



**HAL**  
open science

# Modélisation bayésienne algorithmique de la reconnaissance visuelle de mots et de l'attention visuelle

Thierry Phénix

► **To cite this version:**

Thierry Phénix. Modélisation bayésienne algorithmique de la reconnaissance visuelle de mots et de l'attention visuelle. Sciences cognitives. Ecole doctorale: Ingenierie pour la sante la Cognition et l'Environnement, 2018. Français. NNT: . tel-01735606

**HAL Id: tel-01735606**

**<https://theses.hal.science/tel-01735606>**

Submitted on 16 Mar 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## THÈSE

Pour obtenir le grade de

### **DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES**

Spécialité : **PCN - Sciences cognitives, psychologie et neurocognition**

Arrêté ministériel : 25 mai 2016

Présentée par

**Thierry PHÉNIX**

Thèse dirigée par **Sylviane VALDOIS**, DR CNRS  
et codirigée par **Julien DIARD**, CR CNRS

préparée au sein du **Laboratoire de Psychologie et NeuroCognition**  
et de l' **École doctorale « Ingénierie pour la Santé, la Cognition et l'Environnement »**

## **Modélisation bayésienne algorithmique de la reconnaissance visuelle de mots et de l'attention visuelle**

Thèse soutenue publiquement le **15 Janvier 2018**

Jury composé de :

**Monsieur MARTIN ARGUIN**

PROFESSEUR, UNIVERSITÉ DE MONTRÉAL-CANADA, Rapporteur

**Monsieur PIERRE BESSIÈRE**

DIRECTEUR DE RECHERCHE, CNRS DÉLÉGATION PARIS, Rapporteur

**Monsieur ALAIN CONTENT**

PROFESSEUR, UNIVERSITÉ LIBRE DE BRUXELLES-BELGIQUE, Examineur

**Madame ANNE GUÉRIN-DUGUÉ**

PROFESSEUR, UNIVERSITÉ GRENOBLE ALPES, Présidente

**Madame SYLVIANE VALDOIS**

DIRECTRICE DE RECHERCHE, CNRS DÉLÉGATION ALPES, Directeur de thèse

**Monsieur JULIEN DIARD**

CHARGÉ DE RECHERCHE, CNRS DÉLÉGATION ALPES, Co-Directeur de thèse





# Table des Matières

Table des Matières	i
Liste des Illustrations	iv
Liste des Tableaux	vi
<b>1 Introduction</b>	<b>1</b>
1 Les enjeux des modèles de reconnaissance visuelle de mots . . . . .	2
1.1 Problématique de la reconnaissance visuelle de mots . . . . .	3
1.2 Contribution . . . . .	4
2 Plan de lecture . . . . .	6
<b>2 Revue de littérature : les modèles computationnels de reconnaissance visuelle de mots</b>	<b>9</b>
1 Les principaux modèles de reconnaissance visuelle de mots . . . . .	9
1.1 <i>Logogen Model</i> (Morton, 1969) . . . . .	9
1.2 <i>Interactive Activation Model</i> (IA ; McClelland & Rumelhart, 1981) . . .	11
1.3 <i>Activation-Verification Model</i> (AV ; Paap, Newsome, McDonald, & Schvaneveldt, 1982) . . . . .	13
1.4 <i>Multiple Object Recognition and attention SElection</i> (MORSEL ; Mozer & Behrmann, 1990) . . . . .	16
1.5 <i>Multiple Read-out Model</i> (MROM ; Grainger & Jacobs, 1996) . . . . .	20
1.6 <i>Sequential Encoding Regulated by Inputs to Oscillations within Letter units</i> (SERIOL ; C. Whitney, 2001) . . . . .	22
1.7 <i>Bayesian Reader Model</i> (BR ; Norris, 2006 ; Norris & Kinoshita, 2012) .	25
1.8 <i>Overlap Model</i> (Gomez, Ratcliff, & Perea, 2008) . . . . .	28
1.9 <i>Spatial Coding Model</i> (SCM ; Davis, 2010b) . . . . .	30
1.10 <i>Letters in Time and Retinotopic Space Model</i> (LTRS ; Adelman, 2011) .	33
2 Discussion . . . . .	35
2.1 Quelle structure pour un modèle de reconnaissance visuelle de mots ? . .	35
2.2 L’encodage de la position des lettres doit-il être strict, relatif ou distribué ?	41
2.3 Quelles sont les propriétés du système visuel qui modulent la perception des mots ? . . . . .	43
2.4 Conclusion . . . . .	45
<b>3 BRAID Bayesian word Recognition using Attention, Interference and Dynamics</b>	<b>49</b>
1 Méthodologie : modélisation bayésienne algorithmique . . . . .	49
1.1 Variables du modèle . . . . .	50

1.2	Distribution conjointe et décomposition . . . . .	51
1.3	Formes paramétriques . . . . .	51
1.4	Inférence bayésienne . . . . .	52
2	Présentation des modules de <i>BRAID</i> . . . . .	53
2.1	Sous-modèle sensoriel des lettres . . . . .	54
2.2	Sous-modèle perceptif des lettres . . . . .	58
2.3	Sous-modèle des connaissances lexicales . . . . .	59
2.4	Sous-modèle d'attention visuelle . . . . .	61
2.5	Sous-modèle d'appartenance lexicale . . . . .	63
3	<i>BRAID Bayesian word Recognition using Attention, Interference and Dynamics</i> . . . . .	65
3.1	Le modèle BRAID . . . . .	65
3.2	Modèle de cohérence (sans variable de contrôle) . . . . .	66
3.3	Application du modèle de cohérence . . . . .	66
3.4	Modèle de cohérence contrôlé . . . . .	68
4	Conclusion . . . . .	69
<b>4</b>	<b>Inférences probabilistes et calibrage du modèle <i>BRAID</i></b> . . . . .	<b>73</b>
1	Simulation de tâches cognitives par <i>BRAID</i> . . . . .	73
1.1	Identification d'une lettre isolée sans influence lexicale : $Q_{P_n}^T$ . . . . .	73
1.2	Reconnaissance visuelle de mots : $Q_W^T$ . . . . .	77
1.3	Identification d'une lettre isolée avec influence lexicale : $Q'_{P_n}^T$ . . . . .	81
1.4	Décision lexicale : $Q_D^T$ . . . . .	83
2	Calibrage du modèle . . . . .	88
2.1	Paramètres dépendant de l'expérience à simuler . . . . .	88
2.2	Paramètres sous le contrôle du modélisateur . . . . .	91
2.3	Conclusion sur le calibrage du modèle . . . . .	101
<b>5</b>	<b>Simulations sur la perceptibilité des lettres</b> . . . . .	<b>107</b>
1	Méthodologie . . . . .	108
2	Effet de supériorité des mots et des pseudo-mots . . . . .	109
2.1	Simulation A01 : Effet de supériorité des mots . . . . .	109
2.2	Simulation A02 : Effet de supériorité des pseudo-mots . . . . .	112
2.3	Simulation A03 : Effet du contexte non orthographiquement légal . . . . .	114
2.4	Discussion . . . . .	116
3	Étude du rôle du contexte <i>mot</i> . . . . .	117
3.1	Simulation A04 : Effet de la durée du présentation du contexte . . . . .	117
3.2	Simulation A05 : Effet additif de la durée de présentation de la cible et du contexte . . . . .	118
3.3	Simulation A06 : Effet du nombre de lettres du contexte . . . . .	121
3.4	Discussion . . . . .	123
4	Étude du rôle des contextes <i>pseudo-mot</i> et <i>non-mot</i> . . . . .	123
4.1	Simulation A07 : Effet de la durée de présentation en contexte pseudo-mot . . . . .	124
4.2	Simulation A08 : Effet de la durée de présentation en contexte non-mot . . . . .	125
4.3	Discussion . . . . .	128
5	L'effet de supériorité du mot est-il robuste ? . . . . .	128
5.1	Simulation A09 : Effet de la prédictibilité de la position de la lettre cible . . . . .	128
5.2	Simulation A10 : Effet d'espacement des lettres . . . . .	130
5.3	Discussion . . . . .	132
6	Discussion sur la perceptibilité des lettres . . . . .	133

<b>6</b>	<b>Simulations sur la reconnaissance visuelle de mots et la décision lexicale</b>	<b>141</b>
1	Méthodologie . . . . .	141
2	Effets de fréquence des mots . . . . .	142
2.1	Simulation B01 : Effet de fréquence en reconnaissance de mot . . . . .	143
2.2	Simulation B02 : Effet de fréquence en décision lexicale . . . . .	145
2.3	Discussion . . . . .	146
3	Effets classiques en décision lexicale . . . . .	147
3.1	Simulation B03 : Effet du statut lexical . . . . .	147
3.2	Simulation B04 : Effet de légalité des non-mots . . . . .	149
3.3	Discussion . . . . .	151
4	Effet de la fréquence du voisinage . . . . .	152
4.1	Simulation B05 : Effet de la fréquence du voisinage dans une tâche de décision lexicale . . . . .	152
4.2	Simulation B06 : Effet de la fréquence du voisinage dans une tâche de reconnaissance visuelle de mot . . . . .	156
4.3	Discussion . . . . .	158
5	Effet de la densité du voisinage . . . . .	159
5.1	Simulation B07 : Effet facilitateur de la densité du voisinage dans une tâche de décision lexicale, en condition <i>mot</i> . . . . .	160
5.2	Simulation B08 : Effet inhibiteur de la densité du voisinage dans une tâche de décision lexicale, en condition <i>non-mot</i> . . . . .	163
5.3	Simulation B09 : Effet de la densité du voisinage dans une tâche de re- connaissance visuelle de mot . . . . .	164
5.4	Discussion . . . . .	167
6	Effet d'interaction entre la fréquence des stimuli et leur voisinage . . . . .	168
6.1	Simulation B10 : les effets d'interaction fréquence-voisinage . . . . .	169
7	Discussion sur la reconnaissance de mots et la décision lexicale . . . . .	172
<b>7</b>	<b>Conclusion générale</b>	<b>177</b>
1	Principales contributions . . . . .	177
2	Discussion . . . . .	180
2.1	Quelle structure pour un modèle de reconnaissance visuelle de mots ? . .	181
2.2	L'encodage de la position des lettres doit-il être strict, relatif ou distribué ?	182
2.3	Quelles sont les propriétés du système visuel qui modulent la perception des mots ? . . . . .	184
2.4	Les limites actuelles de <i>BRAID</i> . . . . .	189
3	Perspectives . . . . .	192
3.1	<i>BRAID</i> : un cadre pour la comparaison de modèles . . . . .	192
3.2	Extensions du modèle <i>BRAID</i> . . . . .	193
	<b>Bibliographie</b>	<b>197</b>
	<b>A Matériel utilisé dans les simulations présentées</b>	<b>205</b>
	<b>B Glossaire : définition des termes utilisés</b>	<b>211</b>

# Liste des Illustrations

1.1	Illustration du modèle conceptuel . . . . .	4
2.1	Schéma du modèle des logogènes (Morton, 1969) . . . . .	10
2.2	Schéma du modèle IA de McClelland et Rumelhart (1981) . . . . .	12
2.3	Schéma du modèle AV (Paap et al., 1982) . . . . .	14
2.4	Fonctionnement détaillé du modèle AV (Paap et al., 1982) . . . . .	15
2.5	Schéma du modèle MORSEL (Mozer & Behrmann, 1990) . . . . .	17
2.6	Schéma du modèle de décision de MROM (Jacobs & Grainger, 1992) . . . . .	21
2.7	Schéma du modèle SERIOL (C. Whitney, 2001) . . . . .	23
2.8	Schéma de l'identification des lettres (C. Whitney, 2001) . . . . .	24
2.9	Schéma du modèle BR (Norris, 2006) . . . . .	26
2.10	Schéma de fonctionnement du modèle BR (Norris, 2006) . . . . .	27
2.11	Schéma de l'encodage positionnel dans le modèle Overlap (Gomez et al., 2008) . . . . .	29
2.12	Schéma du modèle SCM (Davis, 2010b) . . . . .	30
2.13	Schéma du <i>spatial coding</i> du modèle SCM (Davis, 2010b) . . . . .	31
2.14	Schéma du <i>superposing matching</i> dans le modèle SCM (Davis, 2010b) . . . . .	32
2.15	Schéma du modèle LTRS (Adelman, 2011) . . . . .	34
2.16	Architecture des modèles, première partie . . . . .	36
2.17	Architecture des modèles, seconde partie . . . . .	37
2.18	Illustration du modèle conceptuel . . . . .	46
3.1	<i>BRAID</i> , sous-modèle sensoriel des lettres . . . . .	54
3.2	<i>BRAID</i> , distributions de probabilité $P(\Delta I_n^t)$ . . . . .	56
3.3	<i>BRAID</i> , sous-modèle perceptif des lettres . . . . .	58
3.4	<i>BRAID</i> , sous-modèle des connaissances lexicales . . . . .	60
3.5	<i>BRAID</i> , sous-modèle d'attention visuelle . . . . .	61
3.6	<i>BRAID</i> , sous-modèle d'appartenance lexicale . . . . .	63
3.7	Schéma de dépendance complet du modèle <i>BRAID</i> . . . . .	71
4.1	Tâche d'identification d'une lettre isolée sans influence lexicale . . . . .	74
4.2	Tâche de reconnaissance visuelle de mot . . . . .	78
4.3	Distribution de l'attention sur un mot de 5 lettres, en fonction de $\sigma_A^t$ . . . . .	90
4.4	Distributions attentionnelles pour les valeurs de $\sigma_A^t$ choisies . . . . .	92
4.5	Performances d'identification de lettre, par position, pour les valeurs de $\sigma_A^t$ choisies . . . . .	93
4.6	Matrices de confusion du modèle <i>BRAID</i> . . . . .	95
4.7	Calibrage des paramètres $Leak_P$ et $Scale_I$ . . . . .	96
4.8	Rôle de $Leak_P$ dans la mise à jour de la distribution d'un percept . . . . .	97
4.9	Décours temporel du taux d'identification d'une lettre isolée . . . . .	97
4.10	Calibrage du paramètre d'acuité visuelle $Slope_G$ . . . . .	98

4.11	Calibrage du paramètre d'interférence latérale $\theta_i$ . . . . .	100
4.12	Influence du paramètre $Leak_W$ en reconnaissance visuelle de mot . . . . .	102
4.13	Influence du paramètre $Leak_W$ en décision lexicale . . . . .	103
4.14	Influence du paramètre $Leak_W$ en fonction de la tâche . . . . .	104
5.1	Simulation A01, Johnston (1978), expérience 2, effet de supériorité du mot . . . . .	111
5.2	Simulation A02, Chase et Tallal (1990), effet de supériorité du pseudo-mot . . . . .	113
5.3	Simulation A03, Rumelhart et McClelland (1982), expérience 10, effet du contexte non orthographiquement légal . . . . .	115
5.4	Simulation A04, Rumelhart et McClelland (1982), expérience 1, effet de la durée du présentation du contexte . . . . .	119
5.5	Simulation A05, Rumelhart et McClelland (1982), expérience 5, effet additif de la durée de présentation de la cible et du contexte . . . . .	120
5.6	Simulation A06, Rumelhart et McClelland (1982), expérience 6, effet du nombre de lettres du contexte . . . . .	122
5.7	Simulation A07, Rumelhart et McClelland (1982), expérience 7, effet de la durée de présentation en contexte pseudo-mot . . . . .	126
5.8	Simulation A08, Rumelhart et McClelland (1982), expérience 8, effet de la durée de présentation en contexte non-mot . . . . .	127
5.9	Simulation A09, Johnston et McClelland (1974), stimuli mots, effet de la prédictibilité de la position de la lettre cible . . . . .	131
5.10	Simulation A10, Carr, Lehmkuhle, Kottas, Astor-Stetson, et Arnold (1976) ; Massaro (1973), stimuli mots, effet d'éloignement des lettres . . . . .	133
5.11	Corrélation entre les résultats comportementaux et les résultats simulés . . . . .	136
5.12	Relation lettres-mot dans la tâche d'identification de lettre . . . . .	138
5.13	Influence de $Leak_W$ sur la tâche 2AFC . . . . .	139
6.1	Simulation B01, Norris (2006), effet de fréquence en reconnaissance de mots . . . . .	144
6.2	Simulation B02, Norris (2006), effet de fréquence en décision lexicale . . . . .	146
6.3	Simulation B03, Rubenstein, Garfield, et Millikan (1970), effet du statut lexical . . . . .	149
6.4	Simulation B04, Rubenstein, Lewis, et Rubenstein (1971), effet de légalité des non-mots . . . . .	151
6.5	Simulation B05, effet de la fréquence du voisinage dans une tâche de décision lexicale . . . . .	155
6.6	Simulation B06, effet de la fréquence du voisinage dans une tâche de reconnaissance visuelle de mot . . . . .	157
6.7	Illustration de l'effet de fréquence du voisinage en décision lexicale . . . . .	158
6.8	Illustration de l'effet de fréquence du voisinage en reconnaissance de mots . . . . .	159
6.9	Simulation B07, effet de la densité du voisinage dans une tâche de décision lexicale, condition <i>mot</i> . . . . .	162
6.10	Simulation B08, effet inhibiteur de la densité du voisinage dans une tâche de décision lexicale, en condition non-mot . . . . .	165
6.11	Simulation B09, effet de la fréquence du voisinage dans une tâche de reconnaissance visuelle de mot . . . . .	166
6.12	Illustration de l'effet de densité du voisinage en reconnaissance de mots . . . . .	169
6.13	Simulation B10, effet d'interaction entre la fréquence des stimuli et leur voisinage en décision lexicale . . . . .	171
6.14	Simulation B10, résultats avec mise à l'échelle affine des données simulées . . . . .	174
6.15	Analyse globale des résultats de reconnaissance de mots et de décision lexicale . . . . .	175



# Liste des Tableaux

4.1	Valeurs par défaut des paramètres du modèle <i>BRAID</i> . . . . .	105
5.1	Effets dont les différents modèles présentés dans ce chapitre ont rendu compte dans la littérature . . . . .	137
6.1	Récapitulatif des caractéristiques des mots utilisés dans les expériences comportementales et les simulations de la section B05 . . . . .	154
6.2	Récapitulatif des caractéristiques des mots utilisés dans les expériences comportementales et les simulations des sections B07 et B09 . . . . .	161
6.3	Récapitulatif des caractéristiques des mots utilisés dans l'expérience comportementale et les simulations de la section B10 . . . . .	170

# Chapitre 1

## Introduction

La lecture est une tâche en apparence facile. Il suffit que l'on soit en présence d'un mot, sur une affiche publicitaire, un panneau d'autoroute ou encore dans un sous-titre de film, pour que ce mot soit immédiatement reconnu et prêt à être prononcé si besoin. Mais cette apparente facilité est trompeuse. Lire est une tâche d'une extraordinaire complexité qui met en jeu une grande diversité de processus cognitifs : des mécanismes visuels pour extraire l'information orthographique ; des mécanismes moteurs pour les déplacements oculaires et la prononciation des mots ; pour ne citer que les principaux. Et malgré cette complexité, il ne faut que 200 ms (voire moins) pour reconnaître un mot dans un texte.

Dès lors, on comprend que l'étude de la lecture ait conduit à un nombre impressionnant de recherches comportementales dans le champ de la psychologie et des neurosciences cognitives. D'ailleurs, c'est une fonction cognitive toujours très étudiée, qu'il s'agisse de comprendre le fonctionnement du système de lecture chez le lecteur expert, l'origine des troubles de lecture en contexte pathologique ou les mécanismes d'apprentissage de la lecture chez l'enfant.

### Quels types de modèles pour modéliser quoi ?

L'enjeu des recherches comportementales est non seulement d'identifier les mécanismes cognitifs impliqués dans la lecture, mais également de comprendre comment ces mécanismes interagissent afin de proposer une architecture des liens qu'ils entretiennent. Ces représentations théoriques du système cognitif de la lecture sont appelées des modèles théoriques ou modèles verbaux (Phénix, Diard, & Valdois, 2016). Ces modèles se sont longtemps limités à des descriptions formalisées sous forme de boîtes représentant les connaissances ou les mécanismes en jeu, et de flèches spécifiant leurs relations (Marshall & Newcombe, 1973 ; Morton, 1969, 1980). Aujourd'hui, les modèles de lecture sont des modèles mathématiques, le plus souvent implémentés sur des ordinateurs. On parle alors de modèles computationnels <sup>1</sup>. On reconnaît en général la supériorité des modèles computationnels sur les modèles purement théoriques ou verbaux, pour deux raisons principales :

- Les modèles computationnels demandent un degré de spécification qui dépasse largement celui des modèles verbaux. Par exemple, un modèle verbal peut simplement postuler que la représentation orthographique des mots en mémoire doit être activée pour qu'un mot présenté visuellement soit reconnu. En revanche, un modèle computationnel doit clairement spécifier comment les mots sont encodés en mémoire, comment ils sont activés lors du traitement et quel est le décours temporel de cette activation.

---

<sup>1</sup>Le terme « computationnel » doit ici s'entendre dans le sens « mathématisé et simulé » et non par référence à la hiérarchie de Marr (1982).

- Les modèles computationnels permettent de simuler le comportement étudié. Il est ainsi possible de tester la validité théorique du modèle selon sa capacité à simuler (c'est-à-dire reproduire fidèlement) le comportement humain. Ainsi, un modèle se doit de simuler les effets classiquement observés en lecture, tel que, par exemple, l'effet de fréquence selon lequel un mot fréquent est plus rapidement identifié qu'un mot peu fréquent. La validité théorique des modèles peut ainsi être mise à l'épreuve des faits, via la comparaison des données simulées et des observations comportementales.

## Un champ d'études morcelé

Cette complexité de la lecture est difficile à appréhender expérimentalement, ce qui a conduit à l'émergence de trois champs de recherches distincts (Norris, 2013 ; Rayner & Reichle, 2010). Le premier est la reconnaissance visuelle de mots isolés qui s'intéresse aux processus cognitifs impliqués dans les traitements orthographiques. Quelles sont les opérations mentales qui permettent d'analyser une séquence de lettres écrites pour qu'elle soit reconnue comme un mot ? Le deuxième champ d'étude porte sur l'identification de mots et met l'emphase sur les opérations permettant de transformer l'information orthographique en information phonologique. Comment transforme-t-on un stimulus visuel en sa prononciation orale ? Ces deux champs d'investigation sont limités au traitement des mots isolés. Le dernier champ d'étude s'intéresse aux processus qui rendent compte des mouvements oculaires lors de la lecture de phrases ou de textes composés de plusieurs mots : l'objectif est alors de modéliser le contrôle oculomoteur en lecture.

Cependant, ces trois domaines ont évolué en relative indépendance, ce qui pose question, notamment en ce qui concerne les processus de reconnaissance de mots. Avant de prononcer un mot, il faut avoir identifié au moins partiellement les lettres qui le composent afin que la trace phonologique soit activée. Ce constat est commun à tous les modèles d'identification de mots (lecture à haute voix). Il en est de même lors de la lecture d'un texte : avant de déplacer son regard sur le mot suivant, il faut que l'information extraite du stimulus courant soit suffisante pour permettre son identification. On comprend donc que le processus de reconnaissance visuelle des mots soit un composant central de la lecture (Norris, 2013).

## 1 Les enjeux des modèles de reconnaissance visuelle de mots

C'est probablement cette position centrale dans la lecture qui explique l'intérêt toujours croissant pour le développement de modèles de reconnaissance visuelle de mots. Durant ces dernières décennies, de nombreux modèles ont été proposés : le modèle *Interactive Activation* (IAM ; McClelland & Rumelhart, 1981 ; Rumelhart & McClelland, 1982), le modèle *Activation Verification* (AVM ; Paap et al., 1982), les modèles *Dual and Multiple Read-Out* (DROM et MROM ; Grainger & Jacobs, 1994, 1996 ; Jacobs, Rey, Ziegler, & Grainger, 1998), le modèle *Multiple Object Recognition and attention SElection* (MORSEL ; Mozer & Behrmann, 1990), le modèle *Spatial Coding* (SOLAR et SCM ; Davis, 2010b), le modèle *Sequential Encoding Regulated by Inputs to Oscillations within Letter units* (SERIOL ; C. Whitney, 2001 ; C. Whitney & Lavidor, 2005), le modèle *Open-Bigrams* (OB ; Grainger & van Heuven, 2003), le modèle *Overlap Open-Bigrams* (OOB ; Grainger, Granier, Farioli, Van Assche, & van Heuven, 2006), le modèle *Bayesian Reader* et ses déclinaisons (BR ; Norris, 2006 ; Norris & Kinoshita, 2012 ; Norris,

Kinoshita, & Casteren, 2010), le modèle *Overlap* (Gomez et al., 2008) et finalement, le modèle *Letter in Time and Retinotopic Space* (LTRS ; Adelman, 2011) <sup>2</sup>.

Ce champ d'études se caractérise par une grande diversité des modèles proposés et l'absence de consensus sur deux questions fondamentales au centre de tous les débats. La première question concerne l'architecture des modèles de reconnaissance visuelle de mots et le type de traitement de la séquence de lettres : *Quels sont les niveaux de représentation nécessaires et comment sont-ils connectés entre eux ? Les lettres du mot sont-elles traitées en parallèle ou les unes après les autres ?*

La deuxième question concerne l'encodage de la position des lettres dans le mot. Doit-on postuler *un encodage strict* (chaque lettre est codée dans la position qu'elle occupe dans le mot, sans incertitude), *un encodage relatif* (la position est codée par rapport aux autres lettres), *ou un encodage distribué* (l'information d'identité d'une lettre est répartie sur plusieurs positions) ?

Une troisième question, qui a moins fait débat jusqu'ici mais oppose les différents cadres théoriques et les différents champs d'étude, est relative aux propriétés du système visuel qui modulent la perception des mots : *Doit-on intégrer dans un modèle de reconnaissance visuelle de mots l'acuité visuelle ? Quel est le rôle de l'attention visuelle dans la reconnaissance de mots ?*

## 1.1 Problématique de la reconnaissance visuelle de mots

Une revue critique des différents modèles de reconnaissance de mots nous amène à extraire deux problématiques principales en lien avec les questions identifiées précédemment.

La première porte sur la « complétude » des modèles existants. En effet, si les questions de structure, encodage positionnel et traitements visuels sont centrales, aucun des modèles actuels ne propose un cadre unifié permettant de traiter l'ensemble de ces trois questions. Certains modèles simplifient les traitements perceptifs pour se focaliser sur la dynamique des interactions entre les représentations internes des lettres et des mots (question 1). C'est notamment le cas pour le modèle IA (McClelland & Rumelhart, 1981) et ses descendants ou le modèle AV (Paap et al., 1982). A contrario, d'autres modèles négligent ces interactions pour se focaliser sur les problématiques liées à l'encodage de la position des lettres (question 2), comme le modèle *Overlap* (Gomez et al., 2008) ou le modèle LTRS (Davis, 2010b). Enfin, quelques modèles intègrent des dimensions de traitement visuel comme l'acuité ou l'attention qui étaient ignorées des précédents. Ainsi, C. Whitney (2001) articule Le modèle SERIOL autour de l'acuité visuelle, tandis que Mozer et Behrmann (1990) fait de l'attention la pierre angulaire du modèle MORSEL.

La seconde concerne l'absence de consensus sur les réponses apportées à nos trois questions par les différents modèles. Concernant l'architecture, les modèles adoptent le plus souvent une structure à trois niveaux et un traitement parallèle des lettres. Mais cette architecture est remise en question par certains modèles récents comme BR (Norris, 2006) ou LTRS (Adelman, 2011). Concernant la question de l'encodage de la position, le débat est vif et une grande diversité de réponses a été proposée. Enfin, la question de l'acuité et du rôle de l'attention visuelle dans la reconnaissance de mots est le plus souvent éludée par les modèles focalisés sur la simulation des performances de lecteurs experts obtenues en condition expérimentale classique. La prise en compte de ces dimensions devient par contre cruciale lorsqu'il s'agit de rendre compte de la pathologie, comme dans le modèle MORSEL (Mozer & Behrmann, 1990) ou de conditions expérimentales particulières, comme dans le modèle SERIOL (C. Whitney, 2001).

---

<sup>2</sup>Ces modèles seront présentés en détail au Chapitre 2.

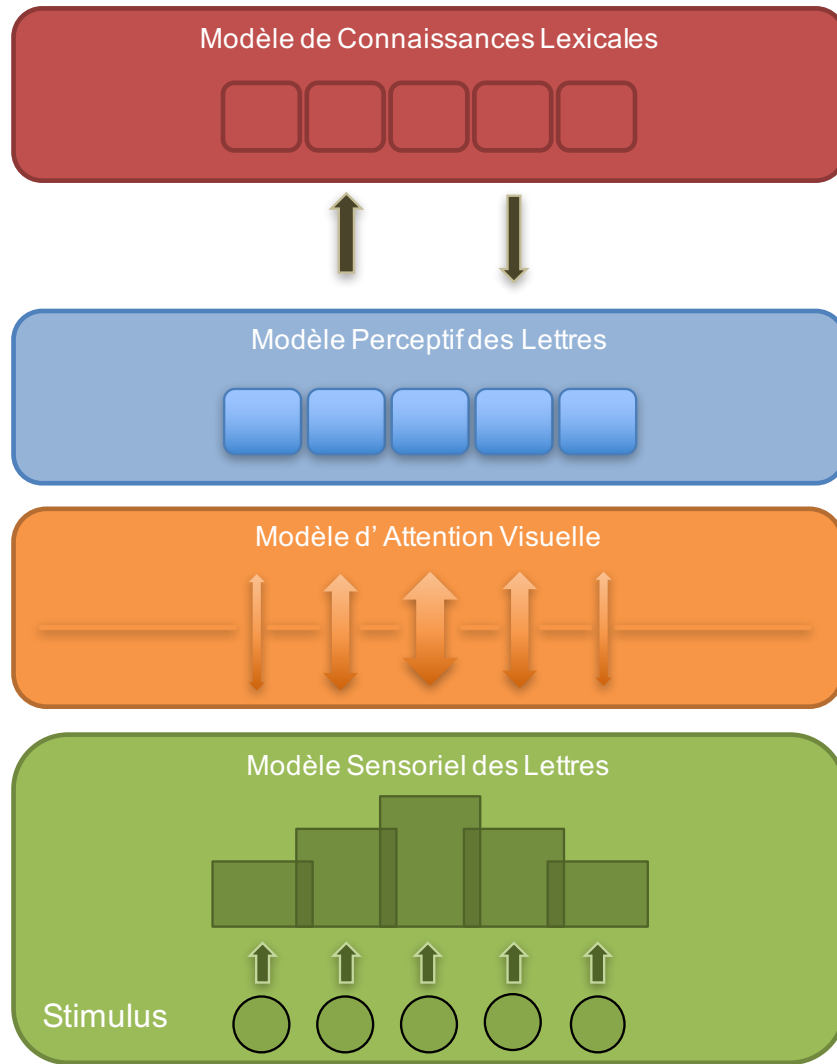


Figure 1.1 – **Illustration du modèle conceptuel.** On retrouve les trois modules classiques dans les modèles de lecture : le sous-modèle sensoriel des lettres (niveau « trait », en vert), le sous-modèle perceptif des lettres (niveau « lettres », en bleu) et le sous-modèle des connaissances lexicales (niveau « mot », en rouge). Le sous-modèle sensoriel intègre l'acuité visuelle matérialisée par la différence de hauteur résultant du traitement sensoriel, et des interférences latérales dues au codage distribué et matérialisées par le chevauchement des rectangles résultant du traitement sensoriel. À cette structure de base s'ajoute le sous-modèle d'attention visuelle qui joue le rôle de filtre attentionnel entre le sous-modèle sensoriel et le sous-modèle perceptif.

## 1.2 Contribution

L'objectif de cette thèse est de proposer un nouveau modèle de la reconnaissance de mots et d'apporter des éléments de validation à l'appui du modèle proposé. Nous nommons *BRAID* ce modèle, pour « *Bayesian word Recognition using Attention, Interference and Dynamics* ». Pour définir ce modèle, nous procédons en deux étapes.

La première étape consiste en une revue critique de la littérature, qui nous permet de définir un cahier des charges théorique pour la conception de ce nouveau modèle. Cette analyse critique vise notamment à évaluer la capacité des modèles existants à répondre aux trois questions fondamentales détaillées précédemment. Pour chaque question, les réponses théoriques proposées par les différents modèles seront confrontées afin de choisir la réponse qui semble la plus per-

tinente en l'état actuel des connaissances. Cette réponse est formulée sous forme d'hypothèse théorique. Ces hypothèses théoriques forment le cahier des charges du modèle, que l'on peut, à cette étape, interpréter comme un modèle conceptuel, « squelette » théorique du modèle *BRAID*. Ce modèle conceptuel est présenté Figure 1.1.

Dans la seconde étape, nous définissons mathématiquement ce modèle conceptuel, nous l'implémentons et évaluons sa capacité à simuler nombre de données expérimentales qui font autorité dans le domaine. Nous avons adopté pour cela un nouveau cadre de modélisation, la *modélisation bayésienne algorithmique*, qui s'avère prometteur pour la modélisation des fonctions cognitives (Diard, 2015 ; Laurent, Barnaud, Schwartz, Bessière, & Diard, 2017). Un seul autre modèle bayésien de reconnaissance de mots, le modèle BR Norris (2006), existe à notre connaissance, mais il utilise le cadre de la modélisation *bayésienne optimale*. La modélisation *bayésienne optimale* repose sur la combinaison de deux distributions : le *prior* qui décrit l'état de connaissance a priori, c'est-à-dire avant la présentation du stimulus, et la *vraisemblance* qui représente la connaissance issue du stimulus. Cet outil de modélisation correspond au niveau *computationnel* de la hiérarchie de Marr (1982). Il se limite à la description formelle de la résolution optimale d'une tâche, sans décrire comment est résolue cette tâche par un système réel, c'est-à-dire sans hypothèse sur les représentations utilisées ou les limites de calcul du système. Avec la modélisation bayésienne algorithmique, nous proposons un formalisme qui permet de décomposer la distribution de vraisemblance en un ensemble complexe de modules. Chaque module est constitué de distributions qui décrivent l'état des connaissances mises en jeu, et de connexions entre ces modules qui sont des relations de dépendance. De ce point de vue, ce nouveau cadre de modélisation se place au niveau *algorithmique* de la hiérarchie de Marr (1982). Le nouveau modèle bayésien algorithmique de reconnaissance de mots que nous proposons est représenté en partie par l'Équation (1.1) :

$$\begin{aligned}
 JD_{BRAID} &= P \left( \begin{array}{cccccccc} W^{0:T} & L_{1:N}^{1:T} & \lambda_{L_{1:N}}^{1:T} & P_{1:N}^{0:T} & A^{1:T} & \mu_A^{1:T} & \sigma_A^{1:T} & C_{A_{1:N}}^{1:T} & \lambda_{P_{1:N}}^{1:T} \\ G^{1:T} & S_{1:N}^{1:T} & \Delta I_{1:N}^{1:T} & I_{1:N}^{1:T} & \lambda_{D_{1:N}}^{1:T} & D^{0:T} & C_{D_{1:N}}^{1:T} & & \end{array} \right) \quad (1.1) \\
 &= \left[ \begin{array}{c} P(W^0) \\ \prod_{n=1}^N P(P_n^0) \\ P(D^0) \end{array} \right] \prod_{t=1}^T \left[ \begin{array}{c} P(W^t | W^{t-1}) \prod_{n=1}^N P(L_n^t | W^t) \\ \prod_{n=1}^N P(\lambda_{L_n}^t | L_n^t P_n^t) \\ \prod_{n=1}^N P(P_n^t | P_n^{t-1}) \\ P(A^t | \mu_A^t \sigma_A^t) P(\mu_A^t) P(\sigma_A^t) \prod_{n=1}^N P(C_{A_n}^t | A^t) \\ \prod_{n=1}^N P(\lambda_{P_n}^t | P_n^t I_n^t C_{A_n}^t) \\ P(G^t) \prod_{n=1}^N [P(S_n^t) P(\Delta I_n^t) P(I_n^t | S_{1:N}^t \Delta I_n^t G^t)] \\ \prod_{n=1}^N P(\lambda_{D_n}^t | \lambda_{L_n}^t C_{D_n}^t) \\ P(D^t | D^{t-1}) P(C_{D_{1:N}}^t | D^t) \end{array} \right].
 \end{aligned}$$

## 2 Plan de lecture

Nous donnons ci-dessous une description détaillée du corps de ce document, chapitre par chapitre.

### **Chapitre 2, Revue de littérature : les modèles computationnels de reconnaissance visuelle des mots**

Dans ce chapitre, nous adoptons un point de vue historique pour présenter une revue de la littérature consacrée aux modèles computationnels de reconnaissance visuelle des mots. Notre objectif n'est pas d'être exhaustif mais plutôt de présenter en détail les modèles qui ont marqué le domaine, soit par leur aspect innovant, soit par la réponse qu'ils apportent à un problème donné. Puis, nous confrontons les solutions proposées par les différents modèles sur les trois questions théoriques qui nous intéressent. À la fin de chaque question, nous formulons notre réponse sous forme d'une hypothèse théorique.

### **Chapitre 3, *BRAID* Bayesian word Recognition using Attention, Interference and Dynamics**

Nous proposons ici la description mathématique complète du modèle *BRAID*, en utilisant un formalisme probabiliste unifié, la modélisation bayésienne algorithmique. Nous reformulons l'ensemble des hypothèses énoncées au Chapitre 2 comme des hypothèses d'indépendance et définissons une distribution conjointe. En nous appuyant sur la structure définie dans ce même chapitre, nous obtenons une décomposition de la distribution conjointe en un ensemble de termes, qui constituent les « briques de connaissance » du modèle. Enfin, à chaque terme de la décomposition nous associons une forme paramétrique que nous décrivons en détail.

### **Chapitre 4, Inférences probabilistes et calibrage du modèle *BRAID***

Nous détaillons dans ce chapitre les quatre tâches que nous simulons avec *BRAID*, en lui posant quatre « questions probabilistes » : identification d'une lettre isolée sans influence lexicale, reconnaissance visuelle de mots, identification d'une lettre isolée avec influence lexicale et décision lexicale. Puis, nous utilisons la simulation de ces tâches pour déterminer les valeurs des paramètres libres des formes paramétriques associées aux différents termes de la décomposition de la distribution conjointe. À l'issue de cette dernière étape, nous disposons d'un modèle entièrement fonctionnel de reconnaissance visuelle des mots.

### **Chapitre 5, Simulations sur la perceptibilité des lettres**

Dans ce chapitre, nous utilisons *BRAID* pour explorer la question de la perceptibilité des lettres. Nous simulons tout d'abord les effets classiques comme l'effet de supériorité du mot ou du pseudo-mot sur le non-mot. Puis nous explorons l'influence du contexte sur l'identification de la lettre cible, en faisant varier les durées de présentation et la qualité du contexte. Enfin, en faisant varier les paramètres attentionnels, nous proposons une explication à la disparition de l'effet de supériorité des mots observée expérimentalement, dans des conditions particulières.

### **Chapitre 6, Simulations sur la reconnaissance visuelle de mots et la décision lexicale**

Ce deuxième chapitre de simulation est dédié à l'identification des mots. Dans les premières simulations, nous nous attachons à reproduire l'effet de fréquence des mots, tant en dénomination qu'en décision lexicale. Puis, nous reproduisons des effets classiques de la décision lexicale, comme l'effet du statut lexical (la décision intervient plus rapidement

lorsque le stimulus est un mot), ou l'effet de légalité des mots (la décision intervient plus rapidement lorsque le stimulus est un non-mot orthographiquement illégal). Enfin, nous explorons l'influence du voisinage du stimulus, en faisant varier sa fréquence et sa densité, sur les performances de reconnaissance en dénomination et en décision lexicale.





# Chapitre 2

## Revue de littérature : les modèles computationnels de reconnaissance visuelle de mots

### 1 Les principaux modèles de reconnaissance visuelle de mots

Dans cette section nous présentons les principaux modèles de reconnaissance visuelle de mots. Nous en avons retenu dix, le choix des modèles reposant sur deux critères : l'innovation, certains modèles sont les premiers à proposer une hypothèse, et l'influence sur le domaine. Pour la présentation nous adoptons un point de vue chronologique. Pour chaque modèle, nous nous attachons à décrire sa structure et son fonctionnement, en mettant en avant ses forces et ses faiblesses. Chaque description d'un modèle se termine par la présentation de ses principaux résultats.

#### 1.1 *Logogen Model* (Morton, 1969)

Le premier modèle auquel nous nous intéressons n'est pas un modèle computationnel mais un modèle mathématique<sup>3</sup>. Le modèle des logogènes est néanmoins très important car il pose les principes de base des futurs modèles computationnels de reconnaissance visuelle de mots. On retrouve ces principes dès les premiers modèles computationnels, dans le modèle IA (McClelland & Rumelhart, 1981) et le modèle AV (Paap et al., 1982), que nous décrirons ensuite.

##### 1.1.1 Description

Le modèle des logogènes postule l'existence d'un lexique mental contenant l'ensemble des représentations lexicales présentes en mémoire à long terme. Ce lexique est composé de logogènes (ou détecteurs). Ce sont des unités lexicales qui peuvent-être activées par des informations sensorielles, auditives ou visuelles, et des informations contextuelles ou sémantiques (voir Figure 2.1). Un logogène définit trois variables associées à chacune des informations lexicales qu'il représente : son niveau d'activation de base, son niveau d'activation courant, et un seuil de réponse.

---

<sup>3</sup>Le modèle n'est pas seulement décrit par des boîtes et des flèches, comme dans un modèle verbal, mais son fonctionnement est décrit par des équations mathématiques qui permettent de faire quelques prédictions, malgré l'absence d'implémentation informatique et de simulations.

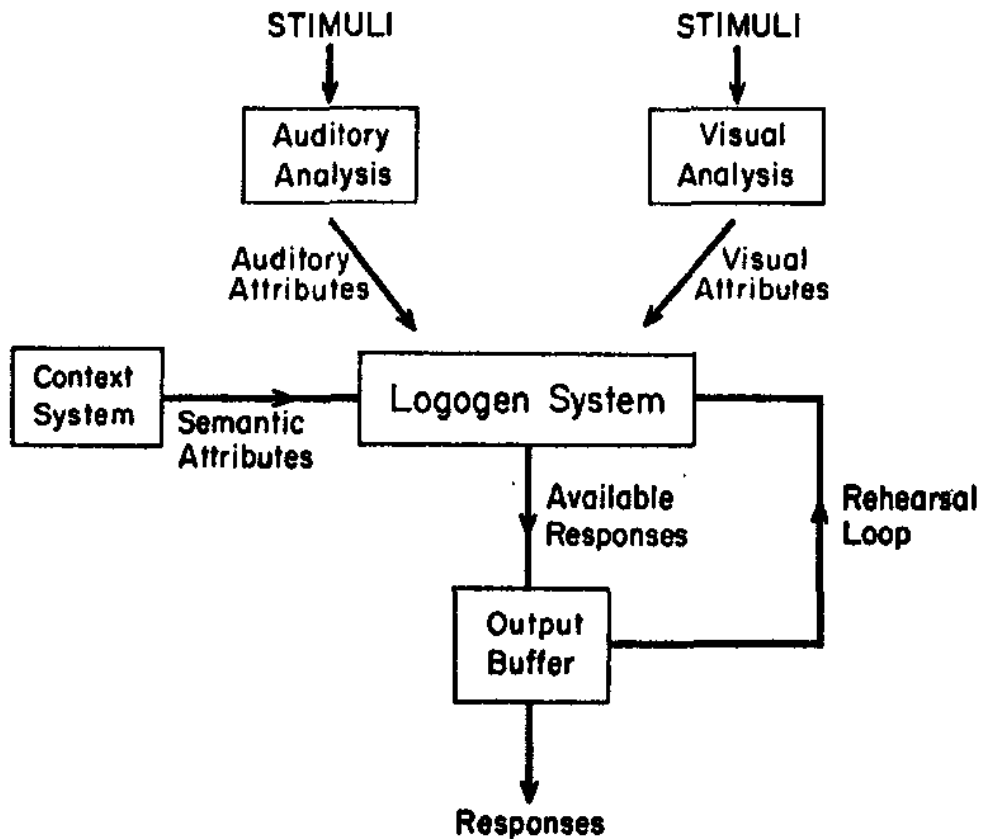


Figure 2.1 – Schéma du modèle des logogènes (Morton, 1969). Représentation schématique de la structure du modèle des logogènes. On observe que les stimuli peuvent-être aussi bien visuels qu’auditifs. Figure tirée de (Morton, 1969, page 166).

Dans une tâche de reconnaissance visuelle de mot, les logogènes reçoivent de l’activation de deux sources différentes : du stimulus et du système cognitif (contexte). Le stimulus fait l’objet d’un traitement visuel qui permet au modèle d’extraire ses attributs. Cette information modifie l’état d’activation de tous les logogènes en maintenant constant le niveau d’activation global du modèle. L’activation des logogènes augmente au cours du temps, proportionnellement au nombre d’attributs qu’ils partagent avec le stimulus. Le mot correspondant au stimulus présenté est rendu disponible dès que l’activation courante d’un logogène dépasse son seuil de réponse. Une fois la réponse produite, le modèle revient rapidement à son état de repos. L’activation produite par un stimulus évolue tout au long du processus de reconnaissance ; ce n’est pas le cas de celle produite par le contexte. En effet, le contexte modifie l’état d’activation de base des logogènes (en l’absence de stimulus). Par exemple, si le contexte de recherche est restreint aux mots de trois lettres, les logogènes représentant les mots de trois lettres voient leur activation de base augmenter, tandis que celle des mots de longueur différente diminue.

Le seuil de réponse des logogènes évolue également, mais plus lentement. Lorsqu’un stimulus est reconnu comme un mot du lexique, le seuil de réponse du logogène correspondant est sensiblement diminué. Cette diminution favorise sa reconnaissance si le mot est présenté à nouveau. Cet avantage est très long à disparaître, ainsi un mot présenté un grand nombre de fois aura un seuil beaucoup plus bas qu’un mot rarement présenté au modèle. On voit ainsi que les seuils des logogènes permettent d’encoder les fréquences d’occurrence des mots et donc de rendre compte de l’effet de fréquence.

Enfin, le modèle identifie le stimulus comme un non-mot lorsqu'aucun logogène ne dépasse son seuil de réponse dans un délai prédéfini.

### 1.1.2 Conclusion

Le modèle des logogènes étant un modèle qualitatif, il ne simule pas de résultats expérimentaux. Cependant, la description détaillée de ce modèle présente les points clés des futurs modèles de reconnaissance visuelle de mots.

- La décomposition en deux modules, l'un dédié au traitement perceptif et l'autre dédié aux connaissances lexicales.
- L'aspect dynamique du modèle : la reconnaissance se fait par accumulation d'activité et le retour vers l'état de repos n'est pas instantané.
- L'activation des unités lexicales se fait sur la base de la similarité entre le stimulus et l'unité lexicale.
- La fréquence d'occurrence des mots qui est codée au niveau lexical fournit un avantage explicite aux mots fréquents.

Pour conclure, notons que ce modèle a connu des évolutions. La première a consisté à dissocier le traitement des sources en créant deux sortes de logogènes, visuels et auditifs, afin de rendre compte des différences de traitement de la fréquence suivant la modalité sensorielle (Morton, 1979). La seconde a permis de prendre en compte l'amorçage par le traitement visuel des mots du système cognitif, avant une tâche verbale, en ajoutant des liens bidirectionnels entre les logogènes visuels et le système cognitif (Jackson & Morton, 1984).

## 1.2 *Interactive Activation Model* (IA ; McClelland & Rumelhart, 1981)

Le premier modèle computationnel de reconnaissance visuelle de mots et sans doute le plus influent est le modèle IA. Il a été développé initialement pour rendre compte de l'effet de supériorité du mot et de l'effet de fréquence des mots. Il revendique sa filiation avec le modèle des Logogènes en se décrivant comme *un modèle des logogènes hiérarchique, non linéaire avec rétroaction entre les niveaux et inhibition entre les logogènes*.

### 1.2.1 Description

Le modèle IA est un modèle connexionniste conçu pour être neurobiologiquement plausible.

**Structure du modèle** Il se présente comme un réseau de neurones artificiels composé de nœuds (codant des unités simples) et de connexions qui les relient (voir Figure 2.2, partie droite). Le codage est localisé : chaque nœud correspond à une unité (trait, lettre ou mot). Le modèle est composé de trois niveaux codant les traits visuels, les lettres et les mots. Le premier niveau correspond à l'acquisition de l'information sensorielle, permettant l'extraction des traits qui caractérisent les lettres. Pour chaque position, il y a 12 unités codant chacune un trait visuel. Le second niveau correspond à l'identification des lettres encodées selon leur position stricte dans la séquence. Le dernier niveau représente le niveau d'identification des mots. Il est composé de 1 179 unités-mots correspondant aux mots monosyllabiques de 4 lettres

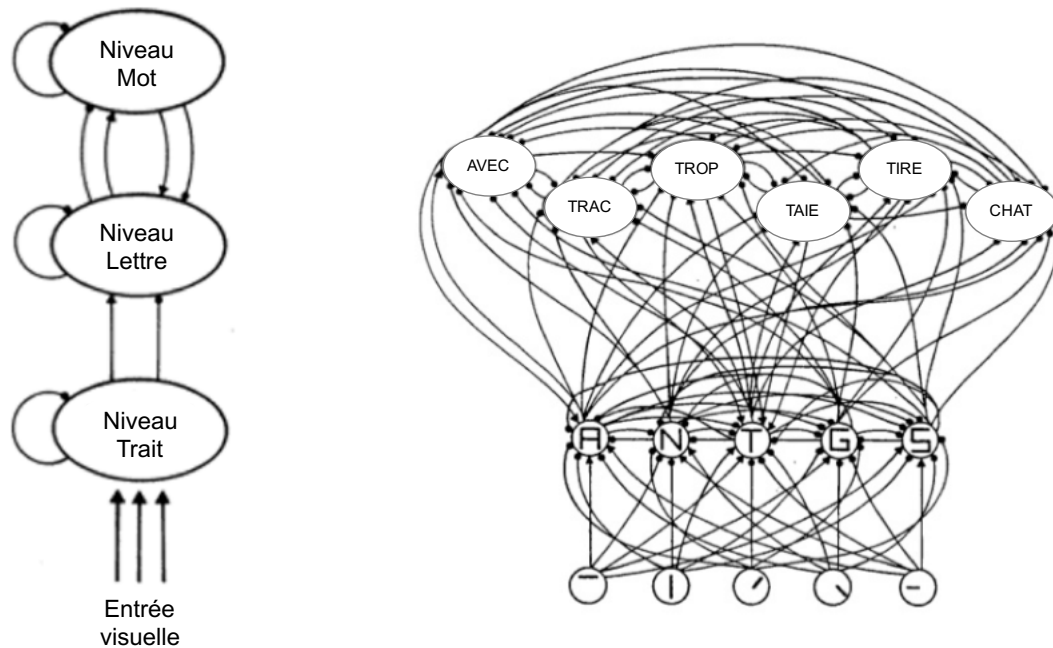


Figure 2.2 – **Schéma du modèle IA de McClelland et Rumelhart (1981)**. À gauche la structure générale du modèle, Figure tirée de (McClelland & Rumelhart, 1981, page 379) ; à droite un exemple de traitement de la lettre *T* comme première lettre du stimulus présenté, Figure tirée de (McClelland & Rumelhart, 1981, page 380). Dans les deux graphiques, les flèches symbolisent des connexions excitatrices, les ronds des connexions inhibitrices.

en langue anglaise. Le modèle postule un traitement parallèle : l'identification des traits se fait en parallèle sur l'ensemble des lettres.

L'intérêt de ce modèle vient de la manière dont sont connectés les nœuds du réseau (voir Figure 2.2, partie gauche). Il existe deux sortes de connexions : les connexions excitatrices et inhibitrices qui, respectivement, augmentent ou diminuent le niveau d'activation des unités auxquelles elles sont connectées. Les connexions excitatrices relient des niveaux différents (connexions inter-niveaux), tandis que les connexions inhibitrices relient des unités inter ou intra-niveau. Alors que les connexions sont unidirectionnelles entre les traits et les lettres, elles sont bidirectionnelles entre les lettres et les mots. Ce réseau de connexions permet de formaliser les deux hypothèses principales du modèle IA.

- La première hypothèse est le renforcement de l'activation des lettres grâce aux connexions excitatrices venant du niveau des mots. L'activation est initialement transmise des traits aux lettres et des lettres aux mots, puis l'activation se propage par rétroaction des mots vers les lettres, ce qui augmente leur activation, permettant ainsi de rendre compte de l'effet de supériorité du mot. On parle de rétroaction de l'activation.
- La seconde hypothèse est celle d'un mécanisme d'inhibition latérale de compétition lexicale. L'activation des unités lettres est transmise aux unités mots, proportionnellement au nombre de lettres en commun, de sorte qu'un grand nombre d'unités mots sont activées. Une fois activée, l'unité mot cible entre en compétition avec les autres unités mots à l'aide des connexions inhibitrices. L'inhibition envoyée par une unité activée est proportionnelle à l'activation reçue, donnant ainsi l'avantage aux mots qui partagent le plus de lettres avec le stimulus.

**Fonctionnement du modèle** Prenons l'exemple de la Figure 2.2 (McClelland & Rumelhart, 1981). Le stimulus visuel contient la lettre T en première position. Lorsque le trait vertical du milieu est détecté au niveau « trait », les connexions excitatrices augmentent l'activation de toutes les lettres qui contiennent ce trait (*B*, *D*, *I* et *T*) et les connexions inhibitrices diminuent l'activation de toutes celles qui ne le contiennent pas. L'activation se propage ensuite au niveau « mot ». L'activation des unités « mots » contenant un *T* en position initiale (*TIME*, *TRIP*, *TAKE*, *TRAP* dans l'exemple) augmente, tandis que l'activation des autres mots (*ABLE*, *CART*) diminue. Les unités mots activées renforcent par rétroaction l'activation du *T* au niveau « lettre ». Elles entrent en compétition au niveau lexical ; le mot le plus activé est celui qui contient le plus de lettres communes avec le stimulus et ce mot est aussi celui dont le pouvoir inhibiteur sur ses compétiteurs est le plus élevé. En pratique, les connexions inhibitrices intra-niveau du niveau lettre ont été supprimées du modèle.

### 1.2.2 Principaux résultats

Le modèle IA est le premier modèle computationnel de reconnaissance visuelle de mots, et il a connu un succès retentissant dans le domaine. Il a notamment permis de simuler fidèlement l'effet de supériorité du mot (Rumelhart & McClelland, 1982) : les simulations montrent que la lettre *E* est plus activée lorsqu'elle est présentée par exemple dans le mot anglais *READ* que hors contexte, comme dans le stimulus *#E##*. Le modèle simule également l'effet de supériorité des pseudo-mots (*A* mieux reconnu dans *TAVE* que dans *#A##*). Les mots qui contiennent des lettres communes avec le pseudo-mot sont activés, et par effet de rétroaction, renforcent l'activation de la lettre. Le modèle rend également compte de l'effet de fréquence et de l'effet de densité du voisinage.

Pourtant, le modèle IA n'est pas exempt de faiblesses. Il ne propose pas de mécanisme permettant de simuler la décision lexicale et propose un codage positionnel strict qui ne permet pas de rendre compte des effets de transposition des lettres en situation d'amorçage (par exemple, l'amorce *CRATON* facilite la reconnaissance de *CARTON*). Les modèles MROM et SCM présentés ci-dessous proposent des réponses à ces limites.

## 1.3 *Activation-Verification Model* (AV ; Paap et al., 1982)

Le modèle AV a été développé peu de temps après le modèle IA. Il revendique une double paternité : le modèle des logogènes pour la partie activation du modèle et le modèle de Vérification (Becker, 1976, 1980) pour la partie vérification. Tout comme le modèle IA, le modèle VA a été conçu avant tout pour simuler l'effet de supériorité du mot. Il est important de noter que le modèle AV est un modèle décrit mathématiquement uniquement, et donc que contrairement aux modèles computationnels, sa capacité à produire des effets par simulations est limitée.

### 1.3.1 Description

Le modèle AV simule la reconnaissance d'un mot en deux étapes ; une étape d'activation et une étape de vérification. L'étape d'activation ou d'encodage reprend les concepts proposés par le modèle des logogènes (Morton, 1969). On retrouve une structure à deux niveaux : l'alphabetum qui correspond au niveau « lettre » et le lexique mental qui correspond au niveau « mot ». L'information sensorielle est encodée progressivement dans l'alphabetum, ce qui produit de l'activation sur l'ensemble du lexique mental. La propagation de l'activation est unidirectionnelle, de l'alphabetum vers le lexique mental. L'étape de vérification suit l'étape d'encodage

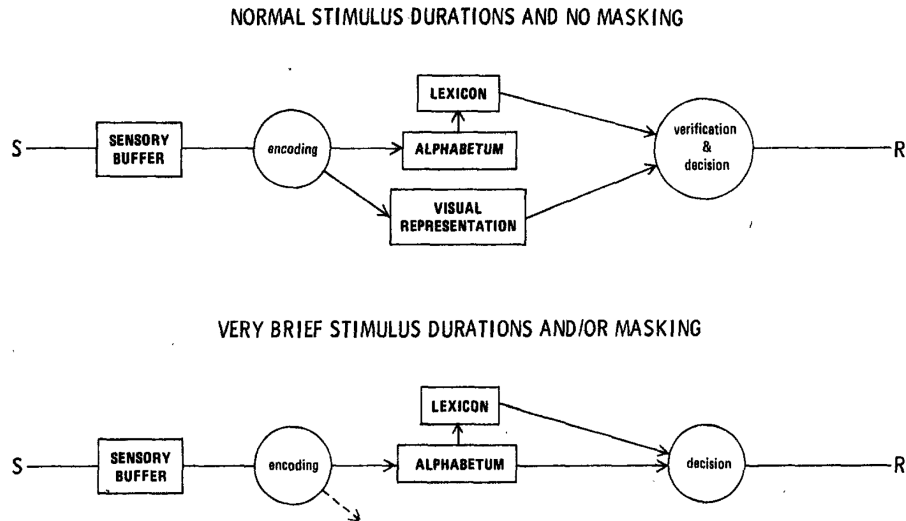


Figure 2.3 – **Schéma du modèle AV (Paap et al., 1982)**. Le modèle a deux modes de fonctionnement ; dans une tâche sans post-masquage du stimulus, le modèle dispose d'une représentation visuelle et peut donc utiliser l'étape de vérification pour prendre une décision quant au mot présenté. Dans une tâche avec post-masquage du stimulus, le modèle ne disposant plus de représentation visuelle du stimulus, la décision est prise directement sur la base des informations de l'alphabetum et du lexique. Figure tirée de (Paap et al., 1982, page 575).

et doit être vue comme une analyse top-down, indépendante de l'encodage, qui conduit à la reconnaissance consciente d'une unique entrée lexicale.

**Structure du modèle** Le modèle a une structure extrêmement simple, composée de deux niveaux, l'alphabetum et le lexique mental (voir Figure 2.3). L'alphabetum contient quatre groupes de 26 unités, une pour chacune des lettres de l'alphabet. Chaque unité contient la valeur de son niveau d'activation courante. Pour chaque groupe, la somme des activations est toujours égale à 1. Ainsi l'activation d'une lettre se fait au détriment des autres lettres du groupe. Le lexique mental est composé uniquement de mots de quatre lettres. Chaque unité du lexique mental est composée, en plus du mot qu'elle représente, de sa fréquence d'occurrence et de son niveau d'activation courant.

**Fonctionnement du modèle** Le traitement perceptif repose sur une matrice de confusion construite expérimentalement. Cette matrice représente la probabilité de reconnaître une lettre  $X$  sachant que la lettre présentée est une lettre  $Y$ , les variables  $X$  et  $Y$  décrivant n'importe quelle lettre de l'alphabet. Cette matrice capture la possibilité de confusion des lettres entre elles. Le modèle prend également en compte la perte d'acuité due à l'excentricité des lettres par rapport à la position du regard sur le stimulus. En pratique, on a donc une matrice de confusion pour chacune des quatre positions dans le stimulus. Enfin, ces matrices évoluent en fonction de la durée de présentation du stimulus. Plus elle est longue, et plus les valeurs sur les diagonales sont proches de 1. Prenons un exemple et considérons que l'on présente au modèle le mot anglais *PORE* (voir Figure 2.4). La probabilité de reconnaître un  $P$  en première position, sachant que la lettre présentée est un  $P$  est de 0,54. Mais on dispose également de toutes les autres valeurs, par exemple la probabilité de reconnaître un  $R$  est de 0,09, un  $S$  est 0,04, et ainsi de suite.

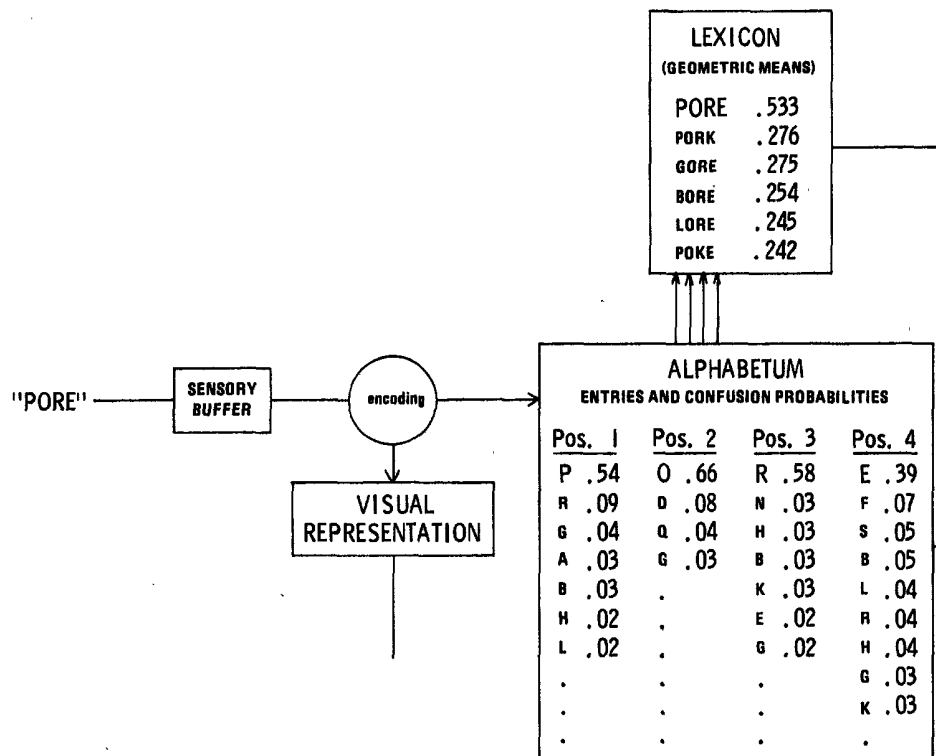


Figure 2.4 – **Fonctionnement détaillé du modèle AV (Paap et al., 1982)**. Le stimulus *PORE* est présenté au modèle. Après encodage des informations sensorielles, l'alphabetum contient quatre distributions de probabilité sur l'espace des lettres, qui créent de l'activation au niveau du lexique mental. Six mots sont suffisamment activés pour accéder à la phase de vérification. Figure tirée de (Paap et al., 1982, page 577).

L'activation ainsi produite au niveau de l'alphabetum se propage au niveau du lexique mental. Pour calculer l'activation des unités « mots », le modèle calcule la moyenne géométrique de l'activation des lettres de l'alphabetum qui composent le mot. Ainsi, dans l'exemple choisi, l'activation du mot *PORE* (0,533) est la moyenne géométrique de 0,54, 0,66, 0,58 et 0,39, tandis que l'activation du mot *PORK* (0,276) est la moyenne géométrique de 0,54, 0,66, 0,58 et 0,03. Ainsi, tous les mots du lexique mental reçoivent une activation proportionnelle au nombre de lettres qu'ils partagent avec le stimulus, et fonction de l'activation de ces lettres au niveau alphabetum.

L'étape de vérification est réalisée une fois la phase d'activation terminée. Elle ne porte pas sur tous les mots du lexique, mais seulement sur les mots dont le niveau d'activation dépasse un certain seuil, fixé à 0,24 (Paap et al., 1982). Cette étape de filtrage est la principale nouveauté de ce modèle. Le processus de vérification est sériel. L'information sémantique, limitée à la fréquence d'occurrence des mots, permet d'ordonner les mots avant l'étape de vérification. Ainsi, la vérification est réalisée en allant du mot sélectionné le plus fréquent au mot sélectionné le moins fréquent. Chaque mot est comparé à la représentation de l'alphabetum. Si la similarité entre les deux est suffisante, le mot est reconnu, sinon le modèle vérifie le mot suivant. Comme cette étape nécessite la présence visuelle du stimulus, elle n'est pas toujours utilisée.

**Tâches simulées** Le modèle AV simule deux tâches expérimentales : une tâche de reconnaissance de mots et une tâche de choix forcé à deux alternatives (2AFC) pour la reconnaissance de lettre isolée.



Pour la tâche de reconnaissance visuelle de mots, le modèle dispose d'une représentation visuelle du stimulus (voir Figure 2.3, schéma du haut). Les mots sélectionnés, classés par ordre décroissant de fréquence d'occurrence, sont comparés à la représentation visuelle. Ainsi, les mots les plus fréquents sont évalués les premiers, ce qui permet de simuler l'effet de fréquence (les mots fréquents sont reconnus plus rapidement) ainsi que les effets de fréquence de voisinage (un mot ayant un voisin orthographique plus fréquent et plus difficile à reconnaître).

Pour la tâche de reconnaissance de lettre isolée, le fait que le stimulus soit masqué supprime la représentation visuelle, et donc rend impossible la comparaison avec les deux lettres proposées (la cible et son alternative). La réponse est donc construite directement, en combinant les informations disponibles à la fois au niveau de l'alphabetum et au niveau du lexique mental (voir Figure 2.3, schéma du bas). Cela est décrit par l'Équation (2.1) :

$$P(C) = P(L)P(C | L) + P(A)P(C | A), \quad (2.1)$$

où  $P(C)$  est la probabilité de répondre correctement,  $P(L)$  est la probabilité que la réponse repose sur le lexique (sa valeur est fonction de la somme des activations des mots sélectionnés),  $P(C | L)$  est la probabilité que la réponse soit correcte sachant qu'elle repose sur le lexique (sa valeur est proportionnelle à la somme des activations des mots sélectionnés qui contiennent la lettre cible à laquelle s'ajoute la demie somme des activations des mots sélectionnés qui ne contiennent ni la lettre cible ni son alternative),  $P(A) = 1 - P(L)$  et enfin,  $P(C | A)$  est la probabilité que la réponse soit correcte sachant qu'elle repose sur l'alphabetum (sa valeur est donnée par l'activation de la lettre cible et la demie somme des activations des lettres autres que les deux lettres présentées).

### 1.3.2 Principaux résultats

Le modèle est utilisé pour simuler l'effet de supériorité du mot et du pseudo-mot (une lettre est mieux perçue lorsqu'elle est placée dans un mot ou un pseudo-mot que lorsqu'elle est placée dans une chaîne de caractères quelconque). Le modèle simule avec succès l'expérience de Johnston (1978) et donne une explication de l'effet de supériorité du mot. Dans l'Équation (2.1), la réponse  $P(C | A)$  est la même que la lettre cible soit présentée dans un mot ou un non-mot, donc l'effet ne peut reposer que sur deux termes :  $P(L)$  et  $P(C | L)$ . Après l'étude des probabilités  $P(C | L)$ , aucune différence n'est observée, que la lettre soit présentée dans un mot, un pseudo-mot ou un anagramme. Il en résulte que l'effet de supériorité du mot repose essentiellement sur  $P(L)$  : l'avantage obtenu lors de l'identification d'une lettre présentée dans un mot est directement lié à la probabilité que la réponse soit basée sur l'information lexicale.

Le modèle rend également compte avec succès de l'effet de fréquence d'occurrence et des effets de voisinage. Cependant, le fait que la fréquence n'intervienne que dans la phase de vérification pose problème. Prenons l'exemple de l'effet de la densité du voisinage orthographique : l'augmentation du nombre de voisins orthographiques plus fréquents que le mot cible conduit le modèle à effectuer plus de vérifications et donc rallonge le temps nécessaire à la décision « OUI » dans une tâche de décision lexicale. Cette prédiction du modèle VA va à l'encontre des données comportementales observées par Grainger et Jacobs (1996).

## 1.4 *Multiple Object Recognition and attention SElection* (MORSEL ; Mozer & Behrmann, 1990)

Le modèle MORSEL a été développé avec deux objectifs : le premier était de concevoir un mécanisme computationnel capable d'analyser simultanément plusieurs objets (des mots d'une

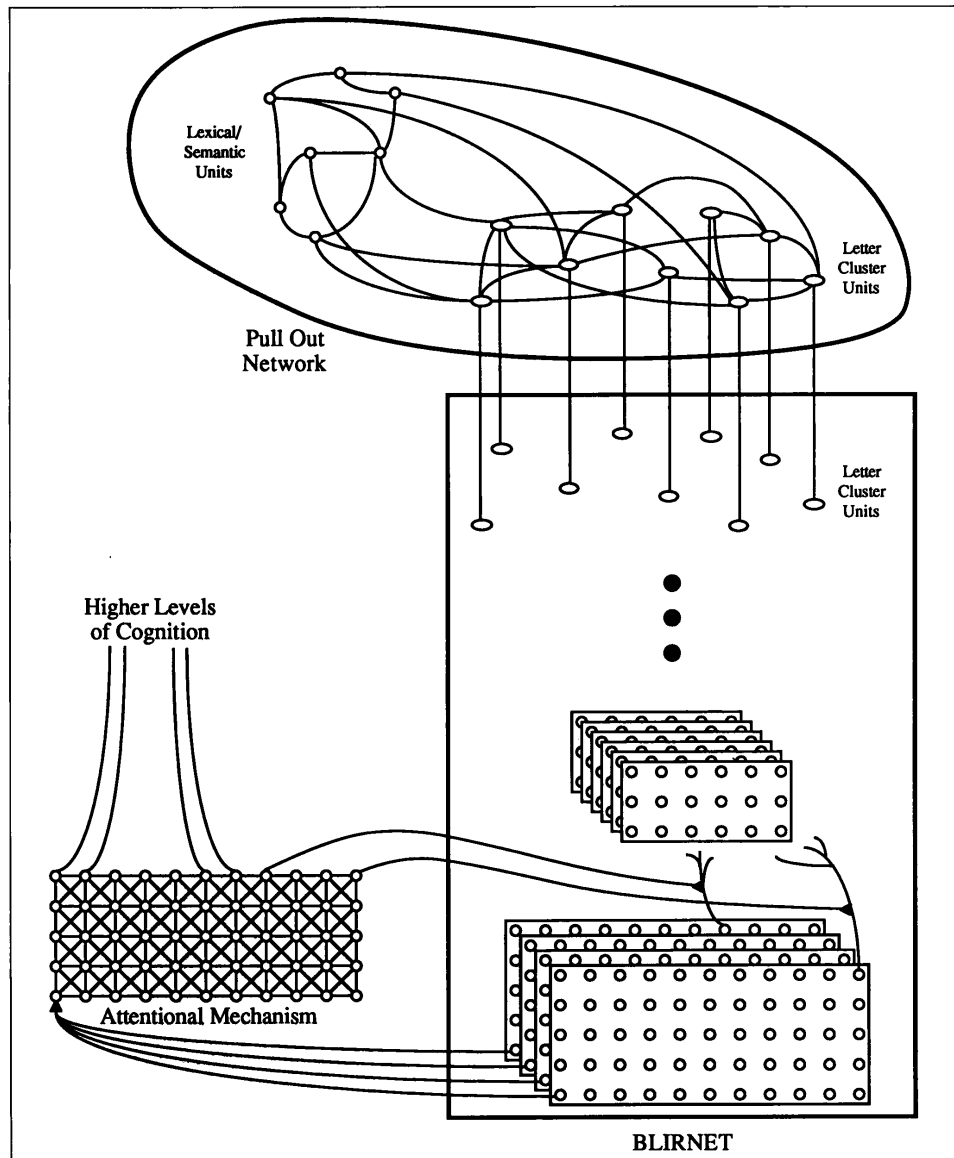


Figure 2.5 – Schéma du modèle MORSEL (Mozer & Behrmann, 1990). À droite le composant central du modèle, Blirnet, dont le rôle est de construire une représentation du stimulus sous forme de tri-grammes. En haut, le composant POnet dont le rôle est d'identifier le mot correspondant le mieux au stimulus, à partir des données bruitées produites par Blirnet. En bas à droite, le module attentionnel qui contrôle l'activité de Blirnet en fonction de contraintes visuelles et sémantiques. Figure tirée de (Mozer & Behrmann, 1990, page 98).

phrase par exemple) et le second était de rendre compte de résultats expérimentaux aussi divers que les erreurs perceptives dues aux multiples objets à analyser, les effets de contexte et de redondance d'informations ou encore de phénomènes attentionnels.

### 1.4.1 Description

Le modèle MORSEL se veut un modèle de reconnaissance d'objets présentés en deux dimensions, adapté à la reconnaissance visuelle de mots, qui intègre un composant attentionnel.

**Structure du modèle** Le modèle se décompose en quatre parties : une couche d'entrée visuelle bidimensionnelle et trois sous-modèles (voir Figure 2.5). La couche d'entrée est une matrice de  $36 \times 6$  éléments sur laquelle sont représentés les mots que le modèle doit identifier. Chaque lettre est représentée par une matrice  $3 \times 3$ , dont les éléments sont des combinaisons de traits de différentes orientations ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ) ou un symbole de fin  $\circ$ . Dans les trois sous-modèles on trouve un composant central en charge de la construction des représentations abstraites des lettres (Blirnet), un composant chargé de construire une réponse à partir des données bruitées fournies par Blirnet (le *Pull-Out Network*, noté POnet), et un composant attentionnel qui guide Blirnet et limite la quantité d'information qu'il traite simultanément (*Attentional Mechanism*, noté AM).

**Blirnet** est un modèle hiérarchique dont la structure est composée de différents niveaux de traitement. Au niveau le plus bas on trouve des détecteurs dont le rôle est d'extraire les attributs visuels de la couche d'entrée en respectant leur position rétinotopique. Au niveau le plus haut, on trouve des détecteurs qui ont perdu leurs caractéristiques rétinotopiques et représentent l'identité abstraite des lettres. Chaque niveau englobe une région rétinotopique plus grande que le précédent et donc un niveau de complexité plus grand.

Les unités qui composent le niveau de sortie de Blirnet sont entraînées à détecter la présence de triplets de lettres, et le codage d'un mot est distribué sur plusieurs triplets. Par exemple, si la couche d'entrée contient le mot *MONEY*, dans la couche de sortie de Blirnet, les triplets suivants sont activés : *\*\*M*, *\*\*\_O*, *\*MO*, *\*\_ON*, *\*M\_N*, *MON*, *M\_NE*, *MO\_E*, *ONE*, *O\_EY*, *ON\_Y*, *NEY*, *NE\_\**, *N\_Y\**, *EY\**, *E\_\*\**, et *Y\*\**. Le symbole « *\_* » signifie que les lettres ne sont pas nécessairement adjacentes, et le symbole « *\** » signifie la présence d'un espace vide. Le point clé de ce modèle est que la ressource dont il dispose pour construire cette représentation est limitée. Ainsi, lorsqu'il y a trop de lettres à traiter (mot long ou plusieurs mots), tous les triplets ne sont pas activés et certains triplets sont activés à tort. Il faut donc un mécanisme de nettoyage, c'est le rôle de POnet.

**POnet** comprend deux types d'unités, les unités triplets de lettres et les unités sémantiques. Les triplets de lettres sont identiques à ceux que l'on trouve en sortie de Blirnet. L'activité résultant du traitement du stimulus peut ainsi être copiée de Blirnet vers POnet. Le réseau POnet connecte toutes ces unités les unes avec les autres, avec des connections excitatrices et des connections inhibitrices. Ainsi, les triplets de lettres compatibles (qui sont présents ensemble dans un mot), comme *MON* et *ONE*, se renforcent mutuellement, tandis que des triplets incompatibles, comme *MON* et *MOV*, s'inhibent les uns les autres. Les unités sémantiques codent le sens des mots et forment un réseau semi-distribué. Ainsi un mot est représenté par plusieurs unités sémantiques, mais une unité sémantique ne peut être connectée qu'à un seul sens d'un mot. Par exemple, l'unité sémantique *MER* pourra être connectée à *LARGE* (comme dans « au large ») et de ce fait ne sera pas connectée à *LARGE* (au sens de vaste). Ces unités sémantiques sont reliées aux unités triplets de lettres qui codent l'orthographe du mot considéré. Des connections inhibitrices connectent les unités sémantiques entre elles pour éviter les traitements simultanés (plusieurs mots ou résidus de l'activité précédente).

**Le mécanisme attentionnel** a été conçu avec deux objectifs : isoler un mot lorsque plusieurs mots sont présents sur la couche d'entrée, et faciliter le traitement des unités qui sont sous le focus attentionnel. L'attention peut être guidée par le stimulus, de manière bottom-up (cas où la couche d'entrée ne contient qu'un mot), ou par un niveau cognitif supérieur (une consigne qui demande d'identifier le mot de droite, par exemple). La structure du composant est la

même que celle de la couche d'entrée du sous-modèle Blirnet. Ceci permet de contrôler le processus de transfert d'information entre les différentes couches du réseau Blirnet (voir Figure 2.5). Plus spécifiquement, l'activité d'une unité Blirnet, pour une position donnée, est transmise au niveau suivant avec une probabilité proportionnelle à l'activité de l'unité correspondante du composant AM. Ce composant possède ainsi trois propriétés essentielles.

- La sélection attentionnelle intervient dès les premières étapes du processus de reconnaissance.
- Lorsque plusieurs mots sont présentés simultanément, l'attention essaie d'isoler chacun d'entre eux en se basant sur la densité des unités de la couche d'entrée, chaque mot formant un groupe dense d'unités activées.
- Toutes les unités de la couche d'entrée sont traitées simultanément, mais le traitement des unités en dehors du focus attentionnel est grandement atténué, sans être supprimé.

**Fonctionnement du modèle** Pour illustrer le fonctionnement du modèle, prenons l'exemple dans lequel les mots *PEA* et *BOY* sont présentés au modèle MORSEL simultanément. La couche d'entrée qui contient le mot sert d'entrée aux sous-modèles Blirnet et AM. Ce dernier focalise l'attention sur la région contenant le mot *PEA*. Blirnet traite alors l'information issue de la couche d'entrée dans son ensemble. Les triplets représentant *PEA* sont activés le plus fortement car ils ont bénéficié de l'action du mécanisme attentionnel. D'autres triplets peuvent émerger, soit du fait d'activations erronées, soit du fait d'interactions avec le second mot *BOY*, même si l'activation reçue par les unités de ce dernier est très faible. POnet effectue ensuite une sélection sur les triplets activés, en se basant sur la plausibilité de trouver deux triplets dans un même mot, et sur les informations en provenance des unités sémantiques.

À ce stade, le modèle connaît l'identité du mot *PEA*, mais pas sa position sur la couche d'entrée. C'est le rôle du composant AM de retrouver cette position. Une fois l'identité et la position du mot connues, elles sont stockées ensemble dans une mémoire à court-terme et rendues disponible aux systèmes cognitifs de plus haut niveau. Le composant AM déplace alors le focus attentionnel sur le second mot *BOY* et le même traitement est réalisé.

#### 1.4.2 Principaux résultats

Bien que ce modèle ait été développé pour simuler les capacités d'un normo-lecteur, il n'a pas été testé sur les tâches classiques comme la tâche de Reicher-Wheeler ou la tâche de décision lexicale. Il est donc difficile de le comparer avec ceux qui l'ont précédé.

En revanche, MORSEL a été utilisé pour simuler des comportements observés chez des patients cérébro-lésés, atteints par exemple de négligence ou de simultagnosie. Pour simuler la négligence, certaines connexions reliant la couche d'entrée au composant AM sont « endommagées ». Les dommages les plus importants sont pratiqués du côté gauche de la couche d'entrée et diminuent progressivement jusqu'au côté droit. Lorsqu'un stimulus est présenté à Blirnet, l'attention a du mal à se focaliser sur les positions endommagées et Blirnet n'arrive pas à détecter les attributs (traits ou terminaison) qui constituent le stimulus sur la couche d'entrée.

Un résultat intéressant montre que malgré des connexions endommagées entre la couche d'entrée et le composant attentionnel, l'effet de supériorité du mot persiste. La reconnaissance des mots semble moins affectée que la reconnaissance des non-mots par les dommages apportés au composant AM. MORSEL explique ce phénomène par l'action de POnet qui est encore capable, même à partir d'un nombre réduit d'éléments, d'identifier le mot correctement. Ce

résultat est intéressant à double titre. Tout d'abord, il montre que la présence d'un effet de supériorité du mot sur un patient dont l'attention est endommagée n'est pas une preuve que la reconnaissance de mot ne requiert pas d'attention. Ensuite, si les interactions entre l'attention et les connaissances lexicales peuvent atténuer l'effet de la négligence sur le traitement des mots, cela peut expliquer que l'attention ne se soit pas distinguée en tant que facteur critique pour la reconnaissance de mots dans un contexte non pathologique.

## 1.5 *Multiple Read-out Model* (MROM ; Grainger & Jacobs, 1996)

Le modèle MROM est le troisième modèle d'une série de quatre modèles (Grainger & Jacobs, 1994, 1996 ; Jacobs & Grainger, 1992 ; Jacobs et al., 1998) ayant pour but de fournir au modèle IA (McClelland & Rumelhart, 1981) les mécanismes nécessaires pour rendre compte des temps de réactions et des distributions des temps de réactions dans des tâches de reconnaissance visuelle de mot et de décision lexicale. Ces modèles sont construits de manière incrémentale, c'est-à-dire que chaque nouveau modèle ajoute des fonctionnalités au modèle qui le précède directement.

### 1.5.1 Description

Pour décrire le modèle MROM, il faut partir du modèle IA (McClelland & Rumelhart, 1981). Ce dernier a été développé initialement pour rendre compte de l'effet de supériorité du mot et n'a pas vraiment été utilisé pour rendre compte de tâches de reconnaissance de mot impliquant la mesure de temps de réaction. Le premier modèle de la série, le modèle SIAM (*semistochastic variant of the IAM* ; Jacobs & Grainger, 1992) introduit de la variabilité dans le modèle IA en remplaçant le seuil de décision (une valeur fixe) par une distribution de probabilité gaussienne. À chaque utilisation du modèle, un tirage est réalisé suivant cette distribution qui fixe la valeur du seuil de décision pour cet essai. Ce modèle est dit semi-stochastique car le processus d'identification est déterministe tandis que le processus de décision est stochastique. Le choix de placer la variabilité sur le processus de décision plutôt que dans le processus d'identification repose sur la capacité de calcul de la machine utilisée. Ce modèle rend compte avec succès des temps de réaction et des distributions des temps de réaction pour des expériences testant l'effet de fréquence et les effets de voisinage (fréquence et voisinage), dans des tâches de reconnaissance de mot.

Avec la version DROM (*Dual Read-Out Model* ; Grainger & Jacobs, 1994), on revient à une tâche de choix forcé à deux alternatives. Le modèle SIAM est enrichi d'un second critère de décision de sorte qu'une réponse positive peut provenir du niveau mot ou du niveau lettre. Dans les deux cas, la décision est prise dès que le mot le plus activé ou la lettre la plus activée passe un certain seuil, variable et défini pour chaque essai.

La version qui nous intéresse ici, le modèle MROM (Grainger & Jacobs, 1996), revient sur les tâches de reconnaissance visuelle de mot et propose un mécanisme de prise de décision basé sur trois critères que nous détaillons au paragraphe suivant. La version la plus aboutie du modèle, MROM-p (*Multiple Read-Out Model of orthographic and phonological processes* ; Jacobs et al., 1998) introduit un composant phonologique au processus décisionnel qui n'entre pas dans le cadre de notre travail.

**Structure du processus de décision** Le modèle MROM fait l'hypothèse que trois processus de décision sont nécessaires pour expliquer les performances observées dans des tâches de décision lexicale (voir Figure 2.6) : deux processus lexicaux pour produire une réponse positive

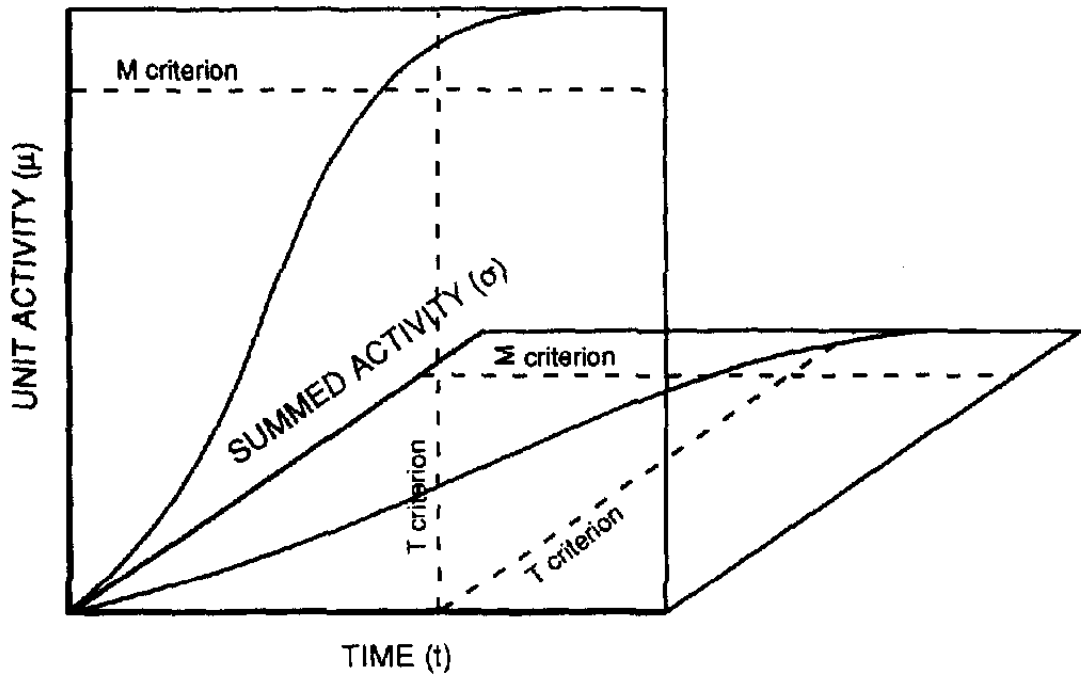


Figure 2.6 – Schéma du modèle de décision de MROM (Jacobs & Grainger, 1992). En abscisse on a le temps. La courbe dans le plan vertical représente le niveau d'activation du mot le plus activé et son seuil de décision  $M$ . La courbe dans le plan incliné représente la somme des activations et son seuil  $\Sigma$ . Dans les deux représentations, le critère de réponse «NON» est matérialisé par un trait en pointillés orthogonal à l'axe des abscisses, noté  $T$ . Figure tirée de (Grainger & Jacobs, 1996, page 522).

et un processus temporel pour produire une réponse négative. Le premier processus lexical repose sur l'activité lexicale produite par chacune des unités « mot ». Cette activation locale est notée  $\mu$  et le seuil associé  $M$ . Ce processus est déjà présent dans le modèle SIAM. Le second processus repose sur la somme des activations générées au niveau lexical par le stimulus. Cette activation globale est notée  $\sigma$  et le seuil de décision associé  $\Sigma$ . Le dernier processus permet de répondre négativement lorsque le stimulus n'est pas un mot, et le critère associé est une limite temporelle notée  $T$ . Les trois critères de décision  $\Sigma$ ,  $M$  et  $T$  sont définis par une distribution de probabilité gaussienne et leur valeur est déterminée par tirage à chaque essai.

**Fonctionnement du modèle** La réponse « OUI » peut intervenir dans deux cas : soit le modèle a identifié un mot en particulier (critère  $M$  dépassé), soit parce que de nombreux mots sont activés (le critère  $\sigma$  est dépassé), ce qui permet au modèle de répondre « OUI » avant qu'un mot ne soit spécifiquement identifié. La réponse « NON » intervient lorsqu'aucun des deux processus lexicaux n'a atteint son seuil de décision dans les  $T$  premières millisecondes. On voit ici que les valeurs de ces différents critères ont une influence massive sur le comportement du modèle. Si la valeur de  $T$  est trop basse ou les valeurs de  $\Sigma$  et  $M$  trop grandes, le modèle produira des faux négatifs (la réponse « NON » est produite alors que le stimulus est un mot). Inversement, si  $T$  est grand ou si l'une des valeurs  $\Sigma$  ou  $M$  est petite, le modèle produira des faux positifs (la réponse « OUI » est produite alors que le stimulus n'est pas un mot).

Dans le modèle, la moyenne de la distribution du critère  $M$  est fixée, mais la moyenne de la distribution des deux autres paramètres est ajustable, car supposée sous le contrôle cognitif du participant. Deux facteurs influencent la valeur de ces paramètres : la nature des stimuli et la

consigne expérimentale qui fixe le compromis à trouver entre vitesse et précision de la réponse. Par exemple, lorsque les stimuli sont des mots et des pseudo-mots,  $\sigma$  augmente rapidement dès les premières itérations du modèle, pour les deux types de stimuli. Il faut donc que  $\Sigma$  et  $T$  soient suffisamment grands pour permettre de différencier les deux types de stimuli. Au contraire, si la consigne met l'accent sur la vitesse de la réponse, alors  $\Sigma$  et  $T$  doivent avoir une valeur basse. La réponse « OUI » est alors le plus souvent produite par  $\sigma$ , ce qui conduit à une augmentation des erreurs. On retrouve ici le compromis entre vitesse et précision.

### 1.5.2 Principaux résultats

La force de ce modèle est sa capacité à rendre compte des temps de réaction moyens, des distributions des temps de réaction et des taux d'erreurs dans des tâches de décision lexicale ou une tâche d'identification perceptive de mot, que le stimulus soit un mot ou un non-mot.

De plus, MROM a démontré sa capacité à simuler les effets de fréquence des mots, de fréquence et de densité du voisinage orthographique et de leurs interactions, ainsi que les effets opposés observés en décision lexicale et en reconnaissance visuelle. Ainsi, l'effet facilitateur de la densité du voisinage en décision lexicale s'explique par l'augmentation de la somme des activations lexicales qui a, par contre, un effet inhibiteur en identification perceptive du fait de l'augmentation des inhibitions latérales.

Enfin, ce modèle met l'accent sur la nécessité de distinguer, dans un même modèle, les processus qui sont spécifiques aux tâches simulées, de ceux qui sont impliqués dans toutes les tâches. Par exemple, le processus décisionnel reposant sur la somme des activités lexicales semble être engendré par la tâche de laboratoire (ici la décision lexicale) tandis que le processus décisionnel reposant sur l'activité lexicale des mots reflète plus directement les mécanismes en jeu en situation écologique de lecture.

## 1.6 *Sequential Encoding Regulated by Inputs to Oscillations within Letter units* (SERIOL ; C. Whitney, 2001)

Le développement du modèle SERIOL a pour but de proposer une méthode d'encodage de la position des lettres d'un stimulus visuel. Pour cela, il introduit deux notions : le traitement séquentiel de gauche à droite des lettres du stimulus (en langue anglaise) qui permet de connaître la position relative des lettres les unes par rapport aux autres, et les bigrammes ouverts qui apportent de la flexibilité à l'encodage de la position. Le modèle SERIOL n'est pas un modèle computationnel et seules certaines portions du modèle sont formalisées mathématiquement, permettant de simuler un nombre très limité d'effets. Cependant, les idées innovantes que propose ce modèle le rendent incontournable.

### 1.6.1 Description

Le modèle SERIOL repose sur une structure à trois niveaux, similaire à celle proposée par le modèle IA (McClelland & Rumelhart, 1981), dont les niveaux « trait » et « lettre » ont été considérablement enrichis. Le modèle de connaissance lexicale, le niveau « mot » du modèle, reste un réseau connexionniste classique, où les unités sont des mots dont l'activation est fonction de la somme des activations transmises par les bigrammes ouverts qui le composent. Enfin, ce modèle ne parle que de la position des lettres et non de leur identification. Ainsi l'activation des différentes unités ne dépend jamais de la nature des lettres, et un *A* est identifié à la même vitesse qu'un *O* du moment qu'ils se trouvent à la même position dans le stimulus.

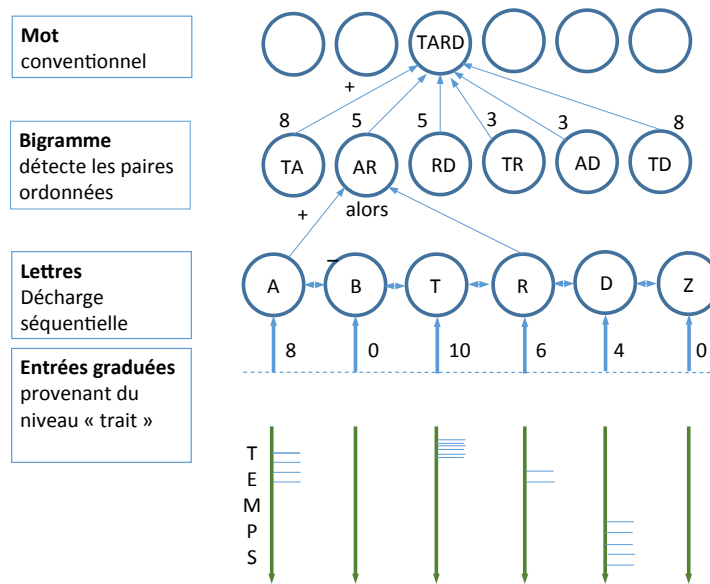


Figure 2.7 – **Schéma du modèle SERIOL (C. Whitney, 2001)**. De l'activation induite par le gradient d'acuité (activation initiale de la lettre *T* puis *A*, *R* et *D*) à l'activation lexicale du mot cible *TARD*. Ce schéma illustre notamment le pattern de décharge temporelle codant la position au niveau « lettre » et la détection de paires ordonnées pour le codage en bigrammes. Figure adaptée de (C. Whitney, 2001, page 224).

**Structure du modèle** Dans le modèle SERIOL, le niveau « trait » est subdivisé en deux, une couche d'entrée et un niveau « trait » proprement dit. Les unités qui composent la couche d'entrée sont les pixels. L'activation produite par le stimulus sur ce niveau n'est pas homogène, mais varie en fonction du *gradient d'acuité* ; elle est maximale au point de fixation et diminue avec l'excentricité (voir Figure 2.8, en bas). Les unités de la couche d'entrée transmettent ensuite l'activation au niveau supérieur, le niveau « trait », en fonction de leur position dans le champ visuel.

Lors de ce transfert d'information, le gradient d'acuité est transformé en *gradient de localisation*. Pour cela, le modèle réalise un traitement différent selon l'hémichamp visuel et l'implication de l'un ou l'autre des hémisphères cérébraux. Les lettres présentes dans le champ visuel gauche sont traitées par l'hémisphère droit. L'activation produite au niveau « trait » par l'hémisphère droit est plus importante que celle de l'hémisphère gauche (voir Figure 2.8, en haut), tout comme l'inhibition latérale qui permet d'inverser l'effet de l'acuité visuelle et d'obtenir ainsi un gradient d'activation décroissant de gauche à droite (voir Figure 2.8, milieu). Selon C. Whitney et Cornelissen (2008), ce mécanisme serait le fruit d'un apprentissage spécifique lié à l'apprentissage de la lecture. Le résultat final de ce premier niveau de traitement est un ensemble de traits caractéristiques des lettres, dont la localisation est rétinotopique et dont l'activation est décroissante de gauche à droite. C'est le *gradient de localisation*. Le deuxième niveau de traitement, le niveau « lettre » est lui aussi subdivisé en deux : un niveau lettre et un niveau bigrammes ouverts. Au niveau « lettre », les unités codant la position des lettres sont connectées à toutes les unités du niveau « trait ». Donc chaque lettre peut représenter n'importe quelle position du stimulus. La lettre qui est connectée aux traits les plus activés reçoit la plus forte activation et devient la première lettre, la deuxième lettre qui reçoit le plus d'activation devient la deuxième lettre et ainsi de suite. Le *gradient de localisation* assure un



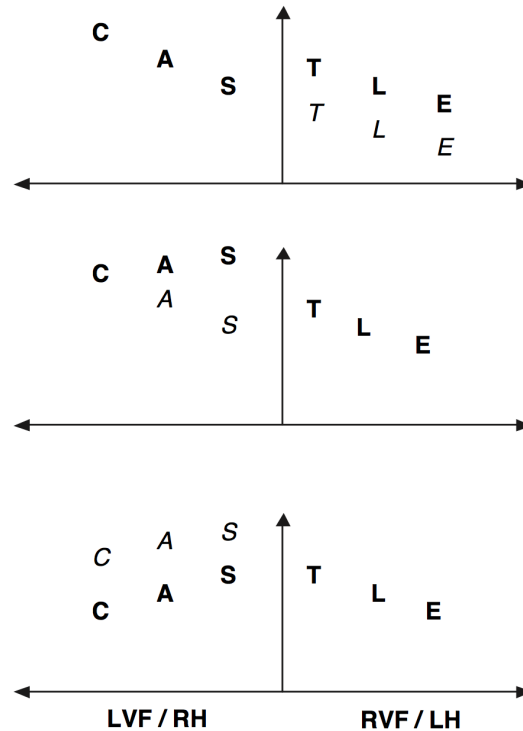


Figure 2.8 – **Schéma de l'identification des lettres** (C. Whitney, 2001). L'axe horizontal représente la position rétinotopique des lettres, l'axe vertical représente le niveau d'activation. Les lettres de l'hémichamp visuel gauche (*left visual field, LVF*) sont traitées par l'hémisphère droite (*right hemisphere, RH*), et les lettres de l'hémichamp visuel droit (*right visual field, RVF*) sont traitées par l'hémisphère gauche (*left hemisphere, LH*). L'image du bas présente l'action du gradient d'acuité, ainsi que le différentiel d'activation entre l'hémisphère droite et l'hémisphère gauche. L'image du milieu présente l'inversion du gradient d'acuité dans l'hémisphère droite et l'image du haut présente l'inhibition trans-hémisphères. Bien que décomposé en trois étapes, ces schémas présentent un unique processus. Figure tirée de (C. Whitney & Cornelissen, 2008, page 147).

parcours de gauche à droite en anglais. C'est donc l'activation enregistrée par le *gradient de localisation* du niveau « trait » qui sert à l'encodage de la position des lettres. Le temps qui sépare l'identification de la position de deux lettres successives est de 25 ms, et l'opération est cyclique avec une fréquence de 5 Hz. Prenons un exemple où le stimulus est le mot *TARD* (voir Figure 2.7). La lettre qui reçoit la première de l'activation est la lettre *T* et c'est également elle qui reçoit le plus d'activation (10). Puis c'est au tour de la lettre *A* qui reçoit un peu moins d'activation (8) et ainsi de suite jusqu'à *D* qui ne reçoit qu'une activation de 4.

Ce décalage obtenu entre les lettres permet de construire les bigrammes ouverts. Les lettres sont appariées dans l'ordre et il n'est pas nécessaire qu'elles soient adjacentes. Ainsi, dans notre exemple (Figure 2.7), les lettres du stimulus *TARD* produisent les bigrammes ouverts suivants : *\*T, TA, AR, RD, TR, AD, TD* et *D\**, où le symbole « *\** » matérialise une extrémité. Le niveau d'activation des bigrammes ouverts ne dépend que de la distance qui sépare les deux lettres (version du modèle de C. Whitney et Cornelissen (2008)). Le niveau total d'activation des bigrammes ouverts est normalisé de sorte que le niveau d'activation des mots courts est supérieur au niveau d'activation des mots plus longs.

### 1.6.2 Principaux résultats

Le modèle offre un cadre explicatif à de nombreux effets mais n'en simule qu'un nombre limité. Son intérêt réside principalement dans l'hypothèse de l'intervention de deux gradients (gradient d'acuité et gradient positionnel) susceptibles d'expliquer certains effets, tels que l'asymétrie de traitement des lettres selon que le mot est présenté dans le champ visuel droit ou gauche, ou des effets plus fins de position optimale du regard, dont la plupart des modèles concurrents ne peuvent rendre compte. Le modèle offre ainsi un cadre théorique potentiel pour l'interprétation des troubles dyslexiques par atteinte des mécanismes visuels ou visuo-attentionnels (voir C. Whitney & Berndt, 1999 ; C. Whitney & Cornelissen, 2005). Il introduit en outre la notion de bigrammes ouverts comme hypothèse explicative de la flexibilité du codage positionnel. Cette idée sera reprise, précisée et implémentée dans d'autres cadres théoriques (Grainger & van Heuven, 2003) pour simuler les effets de transposition de lettres en situation d'amorçage.

### 1.7 *Bayesian Reader Model* (BR ; Norris, 2006 ; Norris & Kinoshita, 2012)

Dans sa version initiale, le modèle BR (Norris, 2006) est un modèle d'un observateur idéal (Geisler, 1989 ; Pelli, Farell, & Moore, 2003) dont les performances sont comparées à des données comportementales. Le concept d'observateur idéal consiste à définir la manière optimale de réaliser une tâche de reconnaissance visuelle de mots. Si les lettres du stimulus sont parfaitement identifiées, alors la tâche consiste simplement à trouver le mot correspondant dans le lexique. En revanche, dès que les lettres perçues ne sont pas parfaitement identifiées (on dit qu'elles sont « bruitées »), la manière optimale de reconnaître un mot, en présence de cette incertitude sur les lettres, consiste à combiner l'information perceptive et les connaissances a priori du lecteur. D'une certaine manière, le lecteur anticipe l'identification du mot à lire sur la base des mots qu'il a l'habitude de lire. La manière optimale de combiner des informations perceptives à des informations a priori est décrite par le théorème de Bayes. L'Équation (2.2) présente la formulation du théorème de Bayes appliqué à la reconnaissance de mot :

$$P(M | S) \propto P(M) f(S | M), \quad (2.2)$$

où  $P(M)$  correspond à la connaissance a priori, calculée à partir de la fréquence des mots et où  $f(S | M)$  est la fonction de vraisemblance du mot  $M$  pour une valeur fixée du stimulus  $S$ .

Le modèle BR repose sur les principes de l'analyse rationnelle Anderson (1990) : pour garantir que les effets observés lors des simulations ne proviennent que des paramètres manipulés (fréquence et voisinage), le modèle repose sur le minimum d'hypothèses possibles. Ayant l'objectif de décrire le processus optimal de reconnaissance visuelle de mot, sans introduire d'hypothèse spécifique, le modèle BR se classe donc dans la couche « computationnelle » de la hiérarchie de modèles de Marr (1982).

Par la suite, ce modèle a connu deux améliorations successives : la version 2010 du modèle (Norris et al., 2010) apporte une incertitude sur la position des lettres selon les principes du modèle Overlap (Gomez et al., 2008), et la version 2012 (Norris & Kinoshita, 2012) ajoute de la variabilité sur la position des lettres dans le processus d'identification, comme le fera plus tard Davis (2010b) dont nous décrivons en détail les travaux plus loin dans ce chapitre, Section 1.9. Ces deux modifications augmentent le nombre d'effets que le BR peut simuler, mais elles n'apportent rien à sa compréhension, c'est pourquoi nous nous limitons ici à la présentation de la version initiale (Norris, 2006).

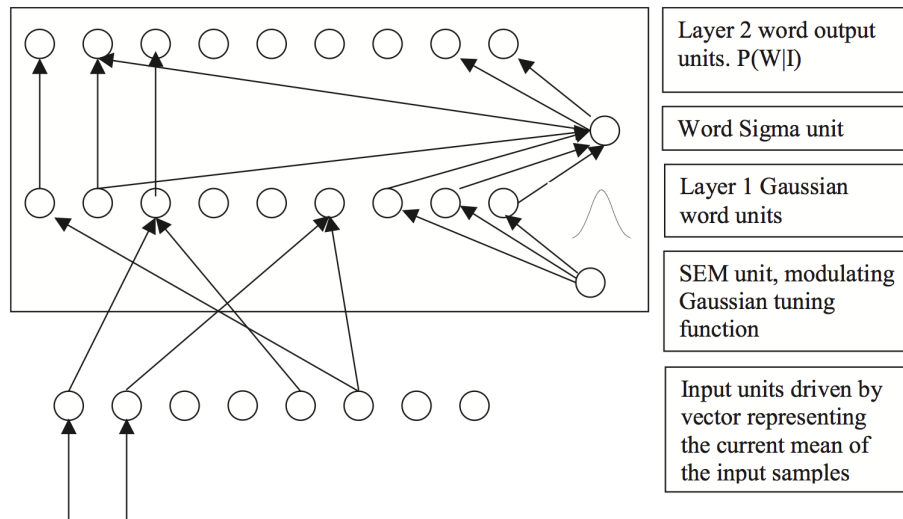


Figure 2.9 – **Schéma du modèle BR (Norris, 2006)**. Ce schéma représente une implémentation du modèle BR avec un réseau de neurones. Le niveau du bas représente le stimulus, le niveau du milieu le résultat de l'échantillonnage sur les activations des mots avant normalisation, à droite la somme des activations pour la normalisation, et le niveau du haut le niveau « mot ». Figure tirée de (Norris, 2006, page 333).

### 1.7.1 Description

Pour définir complètement le modèle, on doit définir la forme paramétrique  $f(S | M)$  de l'Équation (2.2). Le modèle BR calcule une estimation de cette fonction en se basant sur trois hypothèses de fonctionnement :

1. tous les mots peuvent être représentés dans un même espace perceptif multidimensionnel ;
2. les informations perceptives sont accumulées au cours du temps par échantillonnage à taux constant, sur une distribution centrée sur le stimulus, et forment l'entrée du modèle ;
3.  $f(S | M)$  peut être calculée pour chaque mot  $M$  à partir d'une estimation de la variance de la distribution de l'entrée du modèle.

**Structure du modèle** Le modèle ne contient qu'une seule structure de données, la structure des mots. Un mot de  $L$  lettres est représenté par un vecteur de  $26 \times L$  coordonnées. Les mots du lexique sont parfaitement identifiés (il n'y a pas d'incertitude) : ils sont codés par un 1 sur la position correspondant à la lettre présente et par 0 sur les vingt-cinq autres positions. Par exemple, le mot *MAIS* est codé par des 1 sur la 13<sup>ème</sup> position ( $M$ ), sur la 27<sup>ème</sup> ( $26+1$ ) position ( $A$ ), sur la 69<sup>ème</sup> ( $52+9$ ) position ( $I$ ) et sur la 96<sup>ème</sup> ( $78+18$ ) position ( $S$ ). Toutes les autres valeurs du vecteur sont fixées à zéro. Ce système de codage permet de représenter tous les mots dans un même espace perceptif (hypothèse 1).

**Fonctionnement du modèle** Dans ce modèle, il n'y a pas de niveau « trait » ni de niveau « lettre » (voir Figure 2.9). Le processus de traitement perceptif est entièrement résumé par un mécanisme d'échantillonnage. Ainsi, à chaque pas de temps le modèle récupère en entrée un

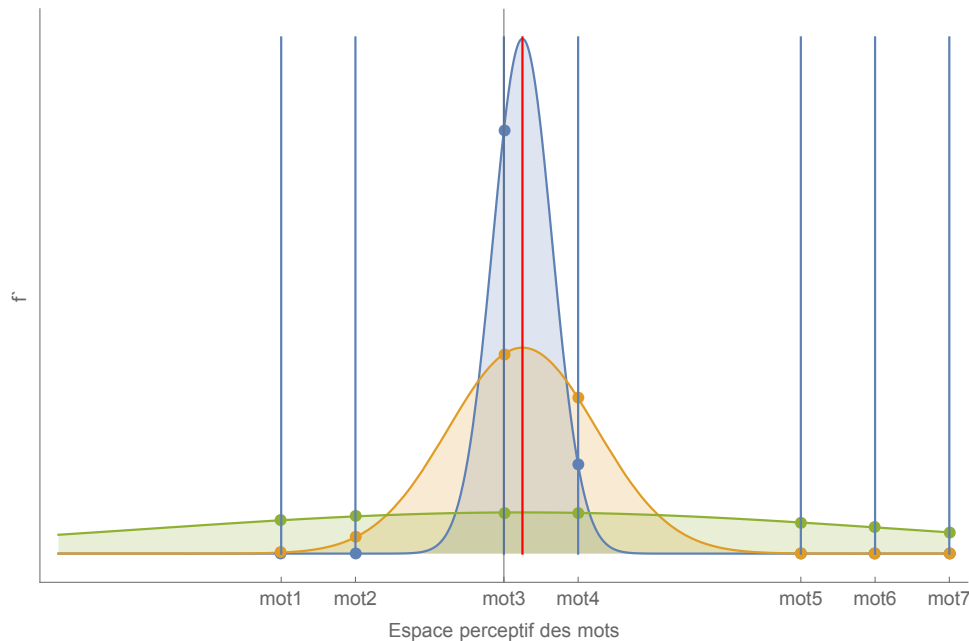


Figure 2.10 – **Schéma de fonctionnement du modèle BR (Norris, 2006)**. Le trait rouge représente l'estimation de la moyenne (cette valeur est réévaluée à chaque nouvel échantillon, mais par souci de clarté, nous avons choisi une unique valeur pour les trois courbes). Les courbes représentent la densité de probabilité décrivant l'erreur autour de la moyenne. La courbe verte correspond à une grande incertitude (peu d'échantillons), la courbe orange à une incertitude moyenne et la courbe bleue correspond à une faible incertitude. Les lignes bleues matérialisent les mots du lexique. Enfin, les ronds matérialisent les valeurs de la vraisemblance perceptive  $f$  pour chacun des mots. On voit que, lorsqu'il y a une grande incertitude, les valeurs sont semblables (points verts). Lorsque l'incertitude diminue, deux valeurs se distinguent et les autres diminuent (points oranges). Enfin, lorsque l'incertitude est petite, un seul candidat se dégage (points bleus).

nouvel échantillon dans l'espace des mots, construit à partir d'une distribution centrée sur le stimulus et de variance constante égale à  $\sigma^2$  (hypothèse 2).

À partir de ces entrées, le modèle calcule deux valeurs, la moyenne  $\mu$  des échantillon dans l'espace des mots et l'erreur standard de la moyenne (ESM) donnée par la formule  $\sigma_M = \sigma' / \sqrt{N}$ , où  $N$  est le nombre d'échantillons et  $\sigma'$  la moyenne des variances de chaque échantillon. L'augmentation du nombre d'échantillons se traduit par la convergence de la moyenne vers un point de l'espace des mots et une réduction de l'incertitude autour de cette valeur. Enfin, la valeur de  $f(S | M)$ , pour le mot  $M$ , est donnée par la valeur de la fonction de densité de la distribution gaussienne de moyenne  $\mu$  et de variance  $\sigma_M$ , évaluée sur le mot courant  $M$  (hypothèse 3). Cela est illustré Figure 2.10.

Ce modèle explique de manière simple l'effet de voisinage orthographique. En effet, si deux mots ne diffèrent que d'une lettre, il faudra que la valeur de SEM soit suffisamment petite, et donc qu'il y ait suffisamment d'échantillons pour que le modèle puisse trancher.

### 1.7.2 Principaux résultats

Le modèle BR rend compte de l'effet de fréquence en reconnaissance de mot et en décision lexicale. Il simule les effets de voisinage observés en décision lexicale et en reconnaissance visuelle de mot : une forte densité du voisinage a un effet facilitateur sur les réponses « OUI »

en décision lexicale et un effet inhibiteur sur les réponses « NON » en décision lexicale et sur les performances en identification perceptive. D'une manière générale, les performances du modèle sont remarquablement similaires à celles des lecteurs humains, suggérant que la reconnaissance de mots pourrait reposer sur des informations bruitées sur les lettres et un processus optimal d'accumulation d'information. Ces résultats fournissent une justification à l'utilisation de la fréquence dans les modèles de lecture sous la forme d'une connaissance a priori.

## 1.8 *Overlap Model* (Gomez et al., 2008)

Le modèle *Overlap* est un autre modèle qui ne s'intéresse qu'au problème de l'encodage de la position des lettres. Il repose sur le postulat selon lequel le problème de l'encodage positionnel serait plus facile à aborder si l'on identifiait la position d'un objet (en l'occurrence une lettre) par une distribution de probabilité sur l'espace des positions possibles, comme le propose Ratcliff (1981), plutôt que par un point de l'espace, comme dans le modèle IA (McClelland & Rumelhart, 1981). L'ambition de ce modèle est de confronter cette idée à différentes situations expérimentales, comme la transposition de lettres par exemple, et de proposer ensuite un cadre pour le développement de futurs modèles de reconnaissance visuelle de mots.

### 1.8.1 Description

Le modèle *Overlap* est minimaliste. Il se focalise sur la question du codage positionnel et ne contient pas de module d'identification des lettres à partir de données sensorielles, ni de module de connaissance lexicale. De plus, il postule une identification parfaite des lettres. La structure du modèle ne porte donc que sur la position.

**Structure du modèle** Le modèle ne traite que les mots de cinq lettres. À chaque lettre du stimulus on associe une distribution de probabilité gaussienne représentant le taux d'encodage de l'identité de la lettre sur les différentes positions. La moyenne de la distribution correspond à la position de la lettre dans la séquence ; la variance varie avec la position, c'est un paramètre libre du modèle, on obtient ainsi une distribution unique pour chacune des positions. Plus la variance est petite, meilleure est l'information de position. Par exemple, si la séquence présentée au modèle est *TRAIL*, comme dans la Figure 2.11 (bas), le point le plus haut est en position 3 pour la lettre *A*, mais la lettre *A* est également associée, à un moindre degré, aux positions 2 et 4 et encore plus faiblement aux positions 1 et 5 (Figure 2.11, deuxième schéma en partant du bas). Dans cet exemple, la variance de la distribution est plus petite pour la lettre *T*, caractérisée par un pic élevé en position 1 et une étendue limitée à la position 2. La lettre *T* est donc plus fortement associée à la position 1 que ne l'est la lettre *A* à la position 3.

**Fonctionnement du modèle** Le fonctionnement du modèle repose sur le calcul du taux de recouvrement entre deux séquences : le stimulus présenté et chacune des deux alternatives proposées (dans une simulation de tâche 2AFC portant sur des mots). Le calcul du taux de recouvrement est obtenu en sommant, sur toutes les positions, les produits des aires sous les courbes. Reprenons l'exemple de la Figure 2.11 et considérons que le mot présenté est *TRAIL* et que l'alternative est *TRIAL*. L'aire sous les courbes est grisée. En position 3, l'aire sous la courbe de la séquence de référence (*TRAIL*) est pleine, ce qui n'est pas le cas pour l'alternative (*TRIAL*). On effectue ce calcul deux fois, pour le mot cible  $o_t$  et pour l'alternative  $o_f$ . La réponse finale est obtenue en calculant :  $p(\text{correct}) = o_t^a / (o_t^a + o_f^a)$  où  $a$  est une valeur supérieure à 1 permettant d'accentuer les petites différences.

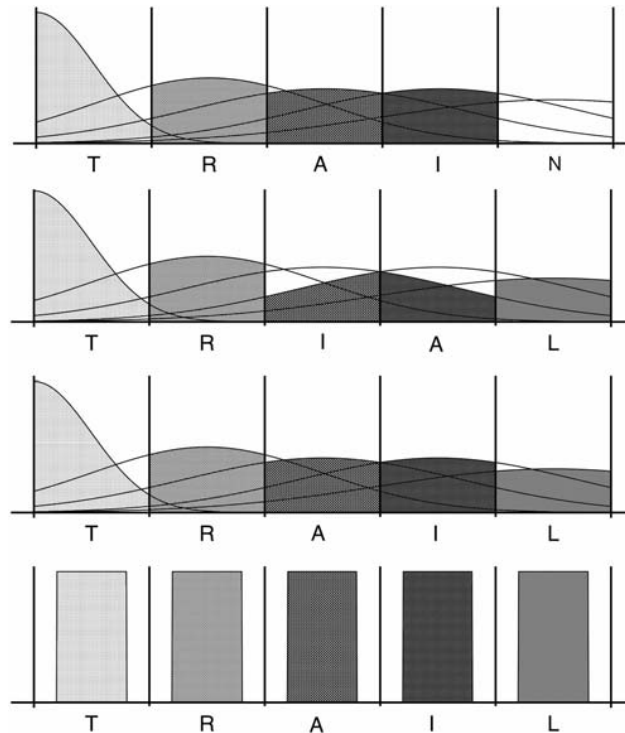


Figure 2.11 – **Schéma de l'encodage positionnel dans le modèle Overlap (Gomez et al., 2008)**. La première ligne (de bas en haut) représente le stimulus *TRAIL* (pas d'incertitude en position) et les surfaces grisées matérialisent le recouvrement avec ce stimulus. La deuxième ligne représente l'encodage positionnel du mot correspondant au stimulus, la troisième un cas d'inversion de lettres (*I* et du *A*) et la dernière ligne (en haut) représente un cas de substitution de lettre (un *N* à la place d'un *L*). Figure tirée de (Gomez et al., 2008, page 256).

### 1.8.2 Principaux résultats

Les paramètres (les variances des distributions) sont fixés indépendamment pour chaque expérience de façon à reproduire les effets observés expérimentalement. Le modèle est testé sur différentes manipulations des séquences de caractères comme la transposition de lettres (*TRAIL-TRIAL*), la répétition de lettres (*MOUSE-MOOSE*), la migration de lettres (*BEARD-BREAD*) et l'insertion/suppression de lettres (*FAULTY-FACULTY*). Les corrélations entre les résultats comportementaux et les simulations sont excellents. De plus, les jeux de paramètres obtenus pour chacune des simulations sont très proches les uns des autres. Les auteurs proposent une fonction (voir l'Equation (2.3)), grâce à laquelle il est possible de générer la valeur de la variance en fonction de la position de la lettre ( $i$ ). Cette modification fait passer le nombre de paramètres libres du modèle de 5 (nombre de position) à 2 (paramètres de l'équation). Ces paramètres sont le taux d'encodage ( $r$ ) et une valeur d'encodage plafond ( $d$ ).

$$s_i = d(1 - \exp(-(i - .5)/r)) \quad (2.3)$$

Le modèle Overlap, doté de ce jeu de nouveaux paramètres, est capable de rendre compte de l'ensemble des effets précédents. La force de ce modèle repose sur la simplicité de la solution trouvée pour résoudre le problème d'encodage de la position. À la suite de ce travail, le principe d'utiliser une distribution pour encoder la position des lettres d'une séquence a été repris par deux modèles : les modèles BR (Norris et al., 2010) et SCM (Davis, 2010b).

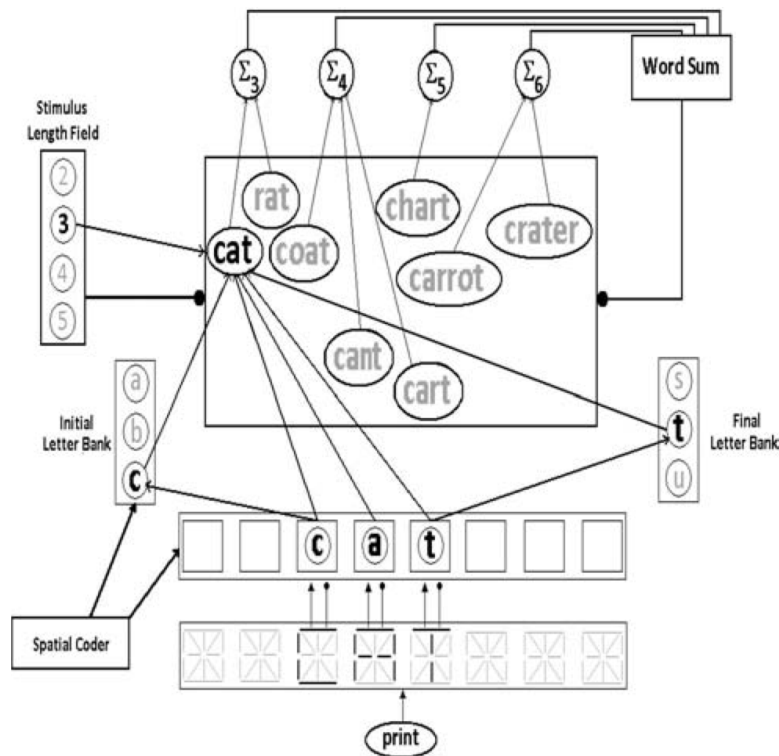


Figure 2.12 – **Schéma du modèle SCM (Davis, 2010b)**. Représentation schématique du traitement du mot *CAT* par le modèle SCM. Les lettres activées à partir de l'extraction des traits sont encodées spatialement (*spatial coder*) ; les lettres initiales et finales sont explicitement encodées comme telles (*initial and final letter bank*) de même que l'information sur la longueur du mot (*stimulus length field*). Plusieurs mots du lexique sont activés en fonction de leur similarité avec le stimulus. La somme de ces activations (*word sum*) est un indicateur de la probabilité que le stimulus soit un mot. Cet indicateur est utilisé en décision lexicale. Figure tirée de (Davis, 2010b, page 723).

## 1.9 *Spatial Coding Model (SCM ; Davis, 2010b)*

Le modèle SCM est un autre modèle qui enrichit le modèle IA (McClelland & Rumelhart, 1981) pour lui permettre de traiter les questions d'amorçage en décision lexicale. Il apporte deux types de modifications. Comme le modèle MROM (Grainger & Jacobs, 1996), il propose un mécanisme de décision lexicale complexe, mais en plus, il enrichit le modèle de base par de nouveaux mécanismes lui permettant de rendre compte de phénomènes comme l'inversion de lettres ou la transposition. Ce modèle est de loin le plus complexe et le plus complet des modèles de reconnaissance visuelle de mots conçus à ce jour.

### 1.9.1 Description

Le modèle reprend la structure à trois niveaux du modèle IA ainsi que le principe des connexions excitatrices et inhibitrices. Il intègre également l'activation progressive au cours du temps des unités qui composent le modèle et il est capable de traiter des stimuli de longueur variable (jusqu'à 12 lettres). Nous nous focalisons ici sur les nouveaux mécanismes postulés par le modèle, en particulier le *spatial coding* et le *superposing matching*.

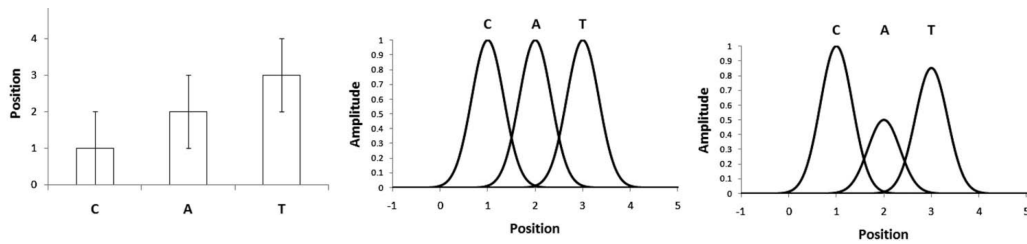


Figure 2.13 – **Schéma du *spatial coding* du modèle SCM (Davis, 2010b)**. Le codage spatial est réalisé en trois phases : 1) décalage temporel pour encoder la position rétinotopique des lettres ; 2) intégration d’incertitude positionnelle (pic en position 2 pour la lettre A, mais A est aussi partiellement associé aux positions 1 et 3) ; 3) ajout d’incertitude sur l’identité des lettres (la lettre initiale est mieux identifiée que la finale et la lettre centrale moins bien. Figure tirée de (Davis, 2010b, page 717).

**Structure du modèle** Le premier mécanisme (*spatial coding*, voir Figure 2.13) représente, dans le modèle, de l’incertitude positionnelle, en encodant la position des lettres par une distribution gaussienne plutôt que par un point (reprenant ainsi l’idée du modèle Overlap (Gomez et al., 2008)). La moyenne de chaque distribution correspond à la position de la lettre dans le stimulus. La variance croît linéairement avec l’excentricité de la lettre par rapport au point de fixation (principe d’acuité visuelle). L’incertitude de position est constante dans le temps et ne dépend donc que de la position de la lettre relativement au point de fixation. Le composant *spatial coding* s’intercale entre le niveau « lettre » et le niveau « mot ». À la sortie de ce composant, les lettres sont représentées suivant deux dimensions : la « hauteur » de la distribution qui encode l’identité de la lettre et le décalage entre les lettres qui encode la position.

Le second mécanisme (*superposing matching*, voir Figure 2.14) permet de calculer l’activation des unités « mot », non sur la base d’une similarité parfaite entre une unité « mot » et la représentation stockée par le *spatial coding*, mais en prenant en compte certaines transformations comme l’inversion de deux lettres (adjacentes ou pas), l’insertion et la suppression de lettres ou encore la substitution de lettres. Le calcul se fait en trois temps.

- la première étape consiste à calculer la différence, pour chaque lettre, entre la fonction représentant le codage spatial du mot et celle du stimulus (dans les faits ce n’est qu’une translation du codage spatial du stimulus car, comme il n’y a pas d’incertitude positionnelle pour les lettres du mot en mémoire, leur codage spatial est un point).
- La deuxième étape consiste à faire la somme de ces fonctions.
- La dernière étape consiste à diviser le maximum de cette fonction somme par le nombre de lettres de l’unité « mot » (voir Figure 2.14).

Enfin, deux composants supplémentaires viennent enrichir ce modèle déjà complexe. Le premier encode la longueur du stimulus. Cette information repose sur le stimulus et contraint l’activation au niveau « mot ». Le second encode les lettres qui se trouvent aux extrémités du stimulus, fournissant un avantage pour l’identification de la première et de la dernière lettre.

**Fonctionnement du modèle** Comme dans IA, les traits identifiés à partir du stimulus induisent de l’activation au niveau des lettres qui les contiennent, en parallèle pour chacune des positions. C’est la phase de traitement au niveau rétinotopique des lettres du stimulus. Puis intervient un processus d’encodage de la position, (*spatial coding*), qui modifie dynamiquement



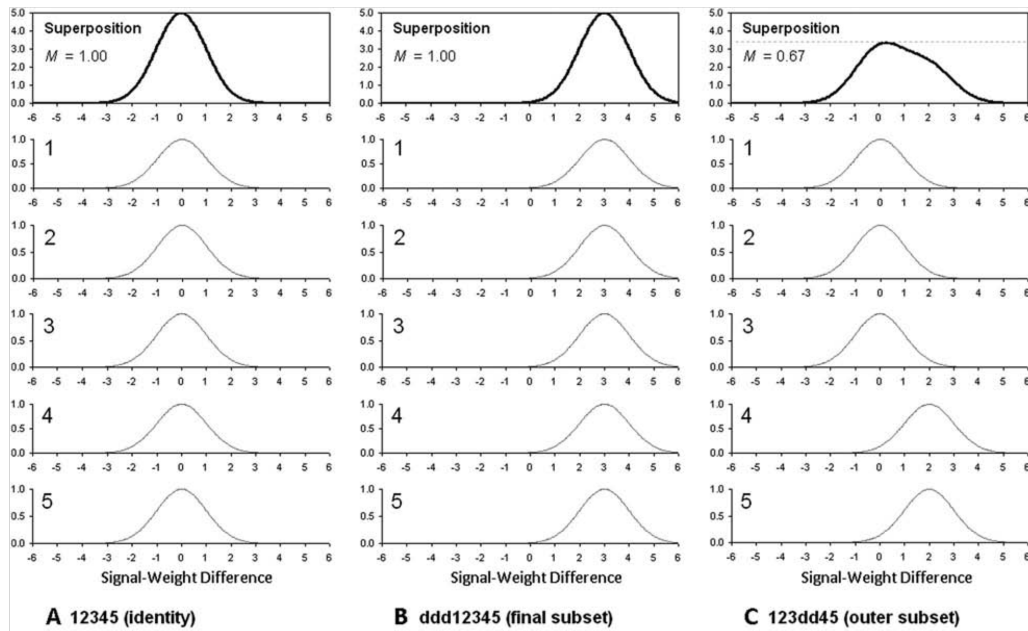


Figure 2.14 – Schéma du *superposing matching* dans le modèle SCM (Davis, 2010b). Chaque ligne représente une fonction de différence signal-poids, où le signal décrit le résultat du traitement bottom-up et le poids décrit la connaissance orthographique. Il y a une ligne par lettre du stimulus. La première colonne correspond au calcul lorsque le mot mémorisé correspond parfaitement à la représentation du stimulus. La deuxième colonne correspond à une translation de 3 positions et la dernière colonne à une insertion. On voit que la translation ne modifie pas la valeur de correspondance ( $M=100$ ). En revanche, l'insertion modifie la valeur maximale de la courbe et donc la valeur de correspondance entre les deux codes. Figure tirée de (Davis, 2010b, page 720).

l'activation des lettres selon leur position. Le profil d'encodage spatial de la séquence est ensuite traité par des unités (*receivers*) qui ont la charge de calculer la meilleure fonction de différence entre la représentation du stimulus et celle des unités mots. La sortie des *receivers* est intégrée par les unités mots, tout en prenant en compte l'information de longueur et celle provenant des lettres des extrémités.

Le modèle SCM est utilisé uniquement dans des tâches de décision lexicale. Le calcul de la décision est le résultat de la compétition de deux processus. Le premier processus calcule la probabilité de répondre « OUI » en combinant la somme de l'information du niveau mot avec l'information du mot qui a la plus grande activation. Ainsi, la réponse favorable intervient quand un mot est reconnu (la valeur d'activation dépasse un certain seuil) ou quand le stimulus génère une importante activité lexicale. Le second processus calcule la probabilité de répondre « NON » sur la base de l'activité créée au niveau lettre. L'idée est que si les lettres sont identifiées et que cela n'a pas créé suffisamment d'activation pour répondre « OUI », alors le stimulus n'est pas un mot. Cependant, dans ce processus, seule l'information sur les lettres est prise en compte.

### 1.9.2 Principaux résultats

Le modèle SCM rend compte d'un nombre important d'effets en décision lexicale avec ou sans amorçage, en simulant les résultats d'une vingtaine d'articles correspondant à une soixantaine de conditions expérimentales. Il est sans nul doute le modèle de reconnaissance de mots actuelle-

ment le plus performant de ce point de vue. En décision lexicale sans amorçage, il rend compte des effets classiques de fréquence (reconnaissance d'autant plus rapide que le mot est plus fréquent), de statut lexical (réponses « OUI » plus rapides que réponses « NON »), de densité du voisinage et de légalité (réponses « NON » plus rapides sur les non-mots illégaux, comme *GZAL*). En décision lexicale avec amorçage, il rend compte des effets lexicaux de l'amorce : facilitation sur les réponses « OUI » quand l'amorce est un non-mot voisin orthographique (*shint* amorce *SHIRT*), inhibition quand l'amorce est un mot voisin (*short-SHIRT*). Sa force est cependant essentiellement de rendre compte des effets fins de transposition des lettres. Par exemple, SCM simule l'effet facilitateur d'une amorce construite par transposition de deux lettres (*caniso-CASINO*) et l'absence d'effet lors de substitutions (*caviro-CASINO*). Il montre l'effet facilitateur d'une amorce composée du squelette consonantique de la cible (*crtn-CARTON*) et ce, même lors de transposition (*ctrn-CARTON*).

## 1.10 *Letters in Time and Retinotopic Space Model (LTRS ; Adelman, 2011)*

La plupart des modèles font l'hypothèse que l'information se propage entre le niveau perceptif (« lettre ») et le niveau lexical (« mot ») graduellement, de sorte qu'une information perceptive incomplète ou bruitée engendre, dès les premières millisecondes de traitement, de l'activation au niveau des mots du lexique. Le modèle LTRS (Adelman, 2011) remet en question cette hypothèse et propose que les informations d'identité et de position sur les lettres du stimulus ne sont disponibles pour le système lexical qu'une fois parfaitement connues. Les effets comportementaux observés ne sont dus, suivant cette hypothèse, qu'aux variations de temps nécessaire à l'identification des lettres et de leur position dans la séquence.

### 1.10.1 Description

Le modèle LTRS propose essentiellement une modélisation du déroulement temporel de l'identification des lettres et de leur position, indépendamment des caractéristiques de ces lettres. Il est alors inutile de postuler une représentation incomplète ou bruitée du stimulus et un système complexe de connexions entre ces niveaux.

**Structure du modèle** La partie lettre du modèle se réduit à deux équations : la probabilité d'identifier une lettre et la probabilité de connaître sa position avec précision. La probabilité d'identifier la lettre en position  $i$  est donnée par  $1 - \exp(1 - \beta_i t)$ . Le paramètre  $\beta_i$  représente la rapidité du traitement. Adelman (2011) propose que certains paramètres ont de l'influence sur sa valeur, comme la position de la lettre, le masquage latéral, l'acuité visuelle et un gradient d'attention, mais ne décrit pas précisément ces mécanismes. La probabilité de connaître la position de la lettre est contrainte par les mêmes propriétés mais a une dynamique différente (paramètre  $\lambda$ ) et est donnée par  $1 - \exp(1 - \lambda \beta_i t)$ .

La représentation du résultat doit suivre certaines contraintes : savoir si une lettre est identifiée ; savoir si deux lettres sont adjacentes et dans quel ordre ; savoir si deux lettres ne sont pas adjacentes sans savoir combien de lettres les séparent ; et enfin connaître l'ordre de deux lettres. Par exemple en notant \* un nombre quelconque de lettre manquante et + l'absence d'au moins une lettre, le mot anglais *CAT* peut-être représenté par les différents états : \*c\*, \*a\*, \*t\*, c\*, \*t\*, \*c\*a\*, \*c\*t\*, c+t, c\*a\*, c\*t\*, \*c\*t, \*a\*t\*, ca+, +at, \*c\*a\*t\*, c\*a\*t\*, \*c\*a\*t, ca\*t\*, c\*a\*t\*, c\*at et cat.

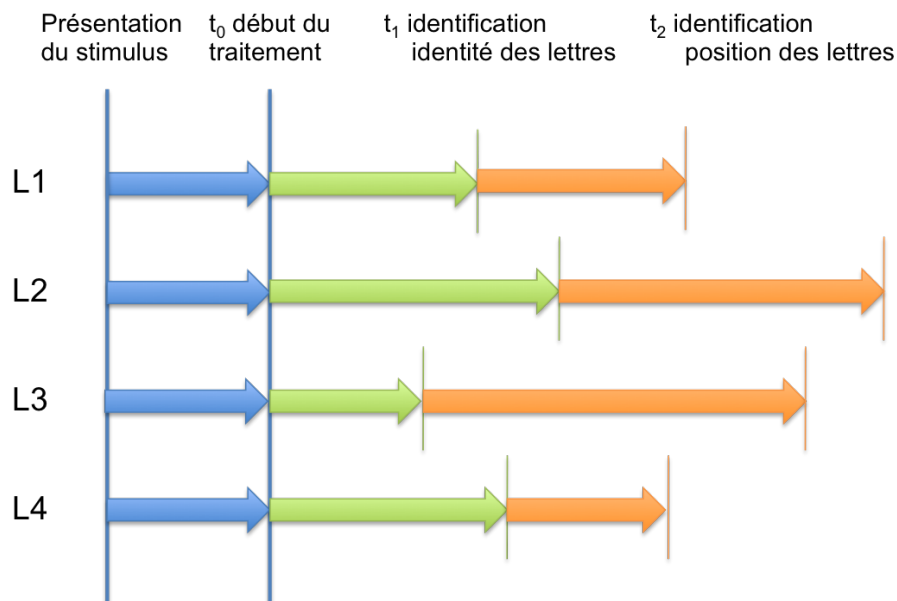


Figure 2.15 – **Schéma du modèle LTRS (Adelman, 2011)**. Déroulement temporel de la reconnaissance d'un mot de quatre lettres, notées L1 à L4, par le modèle LTRS. Le traitement commence en même temps sur les 4 lettres (flèches bleues) mais l'identité précise de chacune des lettres est obtenue après un temps variable (flèches vertes). Une fois l'information d'identité acquise, le traitement positionnel se précise graduellement et l'information de position est obtenue après un temps également variable (flèches orange). Figure tirée de (Phénix, Diard, & Valdois, 2016).

**Fonctionnement du modèle** Le modèle LTRS postule que le traitement commence en parallèle sur l'ensemble des lettres de la séquence (Adelman, Marquis, & Sabatos-DeVito, 2010). Ce traitement est modulé par différents paramètres comme un gradient attentionnel, la visibilité des lettres en fonction de la distance au point de fixation (acuité visuelle) et des effets de masquage latéral (une lettre est moins bien identifiée lorsqu'elle est entourée de deux autres lettres). L'efficacité du traitement conditionne le temps nécessaire à l'identification de chacune des lettres. L'identification d'une lettre ne donne qu'une information approximative sur sa position, de sorte que l'information positionnelle doit être précisée dans un second temps, tel qu'illustré Figure 2.15 pour un mot de 4 lettres. Une lettre n'est reconnue qu'une fois son identité et sa position rétinotopique précisément encodées. À chaque nouvelle lettre identifiée, les mots candidats à l'identification sont soit rejetés (ils ne contiennent pas la nouvelle lettre), soit confirmés (ils contiennent la nouvelle lettre). Ainsi, une fois toutes les lettres identifiées, il ne reste plus aucune ambiguïté sur l'identité du mot s'il est présent dans le lexique. Un mot est identifié dès qu'il n'y a plus qu'un unique candidat dans le lexique ; ceci peut arriver avant que toutes les lettres ne soient identifiées.

### 1.10.2 Principaux résultats

Le modèle LTRS a été utilisé pour simuler les tâches d'identification perceptive et d'amorçage masqué en décision lexicale qui reposent sur la présentation rapide du mot ou de l'amorce. La brièveté des temps de présentation suggère que la réponse est obtenue avant que les informations d'identité et/ou de position n'aient été entièrement traitées. Si l'amorce *JUGDE* est présentée très brièvement, le traitement peut être interrompu alors que seules les lettres *JU\*DE* ont été identifiées avec un codage positionnel approximatif, ce qui permettra d'activer le mot *JUDGE*.

Si l’amorce est *JUNPE* alors le même traitement partiel conduira à identifier *JU\*PE* qui n’amorcera pas le mot *JUDGE*. La force de ce modèle est de montrer que nombre d’effets expérimentaux peuvent être simulés en l’absence de procédure d’appariement lexical. Le modèle ne fait par ailleurs pas d’hypothèse sur la nature du codage positionnel (strict ou bigrammes ouverts). Il montre que des hypothèses simples, limitées au décours temporel de l’extraction des caractéristiques du stimulus (identité et position des lettres), constituent une alternative plausible pour rendre compte d’effets en apparence complexes.

## 2 Discussion

Il arrive que des modèles computationnels reposant sur des hypothèses théoriques différentes traitent une même problématique. Il est alors nécessaire de comparer leurs résultats pour avoir une idée de la validité des hypothèses qui les sous-tendent. Le plus souvent, cette comparaison repose sur la capacité des modèles à simuler des résultats comportementaux, le meilleur modèle étant celui qui prédit le mieux ces observations ou celui qui en prédit le plus. Cette démarche ne peut pas s’appliquer dans le cadre de ce travail de thèse. En effet, il faudrait tout d’abord disposer d’une implémentation fonctionnelle de chacun des neuf modèles considérés, or elles ne sont disponibles que pour les modèles récents (BR, SCM, Overlap). De plus, il faudrait disposer du temps nécessaire pour réaliser les simulations des Chapitres 5 et 6 avec chacun de ces modèles, ce qui semble irréaliste.

Nous avons donc opté pour une autre stratégie. Pour chaque problématique identifiée comme pertinente pour la reconnaissance visuelle de mots, nous analysons les réponses apportées par les modèles, en termes de structure et fonctionnalité. A l’issue de cette analyse, nous concluons sous forme d’une hypothèse théorique qui servira de base au modèle *BRAID*.

### 2.1 Quelle structure pour un modèle de reconnaissance visuelle de mots ?

La question de la structure du modèle englobe en fait deux sous-questions : l’architecture du modèle à proprement parler (quels sont les niveaux de représentation nécessaires et comment sont-ils connectés entre eux ?) et le traitement des lettres du stimulus (le traitement est-il parallèle ou sériel ?).

#### 2.1.1 Une architecture à trois niveaux de traitement

L’architecture proposée par le modèle IA (McClelland & Rumelhart, 1981) est constituée de trois niveaux : « trait », « lettre » et « mot ». Cette architecture sert de support à six des neuf modèles présentés précédemment. Les trois modèles restants, le modèle BR (Norris, 2006), le modèle Overlap (Gomez et al., 2008) et le modèle LTRS (Adelman, 2011) font le choix de n’implémenter qu’une partie de cette architecture. Nous proposons ici une synthèse comparative de ces architectures (voir Figures 2.16 et 2.17).

**Le niveau « trait »** décrit les processus d’extraction des attributs des lettres du stimulus. Pour les modèles IA, MROM et SCM, ces attributs sont des traits verticaux, horizontaux ou obliques (d’où le nom du niveau). Ils sont détectés dans le stimulus lors de la présentation, avec une probabilité dépendant de la qualité visuelle supposée de l’affichage. Le modèle MORSEL reprend l’idée des traits qu’il organise en couches successives au sein d’un composant plus global intégrant également les informations sur les lettres (Blirnet). Dans ce modèle, l’extraction des

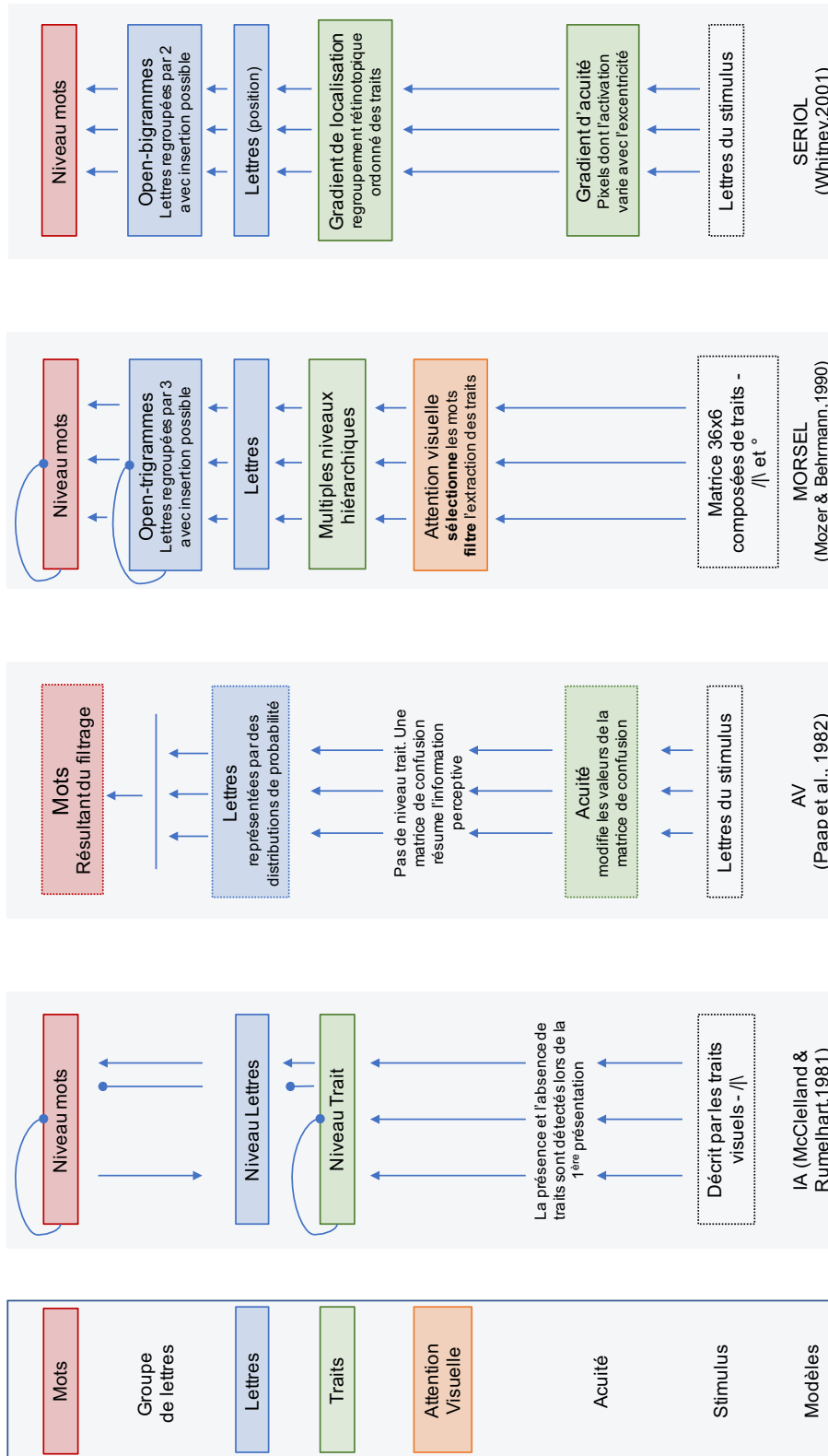


Figure 2.16 – **Architecture des modèles, première partie.** Schéma des niveaux de représentation et de leurs interactions dans les modèles IA, AV, MORSEL et SERIOL. Le modèle MROM, pour ce niveau de détail, a une architecture identique au modèle IA. Les traits finissant par une flèche représentent des connexions excitatrices, les traits finissant par un point représentent des connexions inhibitrices.

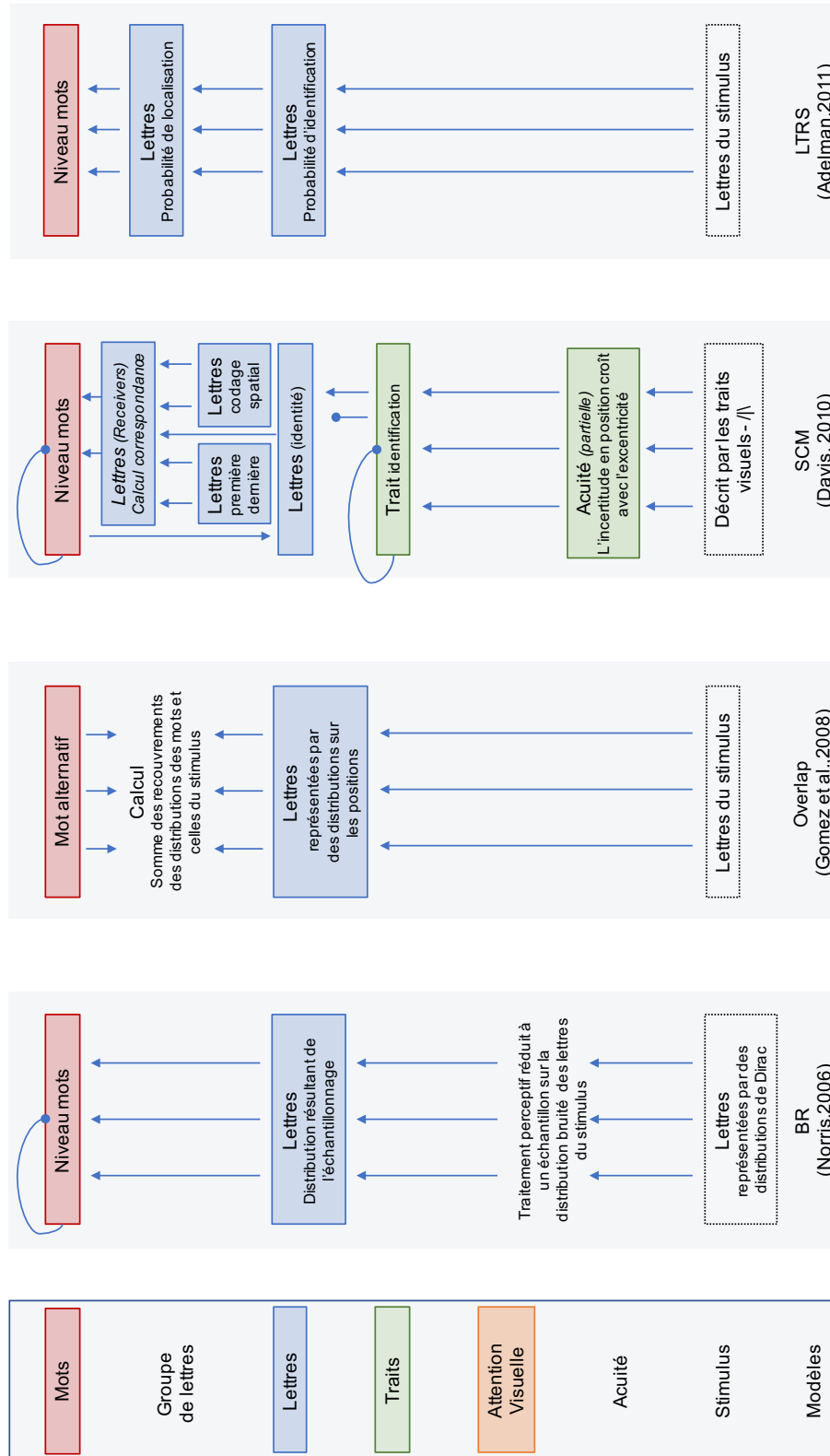


Figure 2.17 – **Architecture des modèles, seconde partie.** Schéma des niveaux de représentation et de leurs interactions dans les modèles BR, Overlap, SCM et LTRS. Les traits finissant par une flèche représentent des connexions excitatrices, les traits finissant par un point représentent des connexions inhibitrices.

attributs se fait avec une probabilité modulée par la charge attentionnelle. Le modèle SERIOL propose un mécanisme d'extraction des traits qui identifie séparément la lettre et sa position relative. Les attributs comprennent les traits mais aussi des angles et des courbes. L'extraction des traits est contrôlée par deux mécanismes : un gradient d'acuité qui favorise l'extraction fovéale et un gradient de localisation qui permet la sérialisation de l'identification des lettres. Ces modèles sont implémentés à l'aide de réseaux de neurones. Chaque trait est représenté par une unité dont on calcule l'activation. Ces unités sont connectées à des unités « lettres » qui agglomèrent ces informations, auxquelles elles transmettent de l'activation inhibitrice, si la lettre ne contient pas le trait, ou excitatrice, si la lettre contient le trait. Dans le modèle VA (Paap et al., 1982), ce mécanisme complexe est résumé par une matrice de confusion. Le résultat du traitement perceptif est donc une distribution de probabilité sur l'espace des lettres. L'intérêt de cette représentation est double : il n'est pas nécessaire de définir les traits pertinents pour l'identification des lettres (ce ne sont pas les mêmes dans IA et SERIOL par exemple) et les probabilités fournissent un mécanisme d'excitation-inhibition naturel par la règle de normalisation.

Trois modèles ne disposent pas d'un niveau « trait » (voir Figure 2.17) : le modèle BR (Norris, 2006), le modèle Overlap (Gomez et al., 2008) et le modèle LTRS (Adelman, 2011). Ces trois modèles ont été conçus pour simuler des effets lexicaux, donc ils sont exclusivement liés à des tâches de reconnaissance visuelle de mot : le modèle BR a pour objectif de rendre compte de l'effet de fréquence tandis que les modèles Overlap et LTRS s'intéressent aux effets liés à l'encodage positionnel, comme la transposition ou l'insertion de lettres. Bien que ces modèles aient simulé avec succès un ensemble d'effets lexicaux, l'absence de niveau « trait » ne leur permet pas, techniquement, de rendre compte d'effets liés à la perceptibilité des lettres. Ainsi, ils ne peuvent pas rendre compte des effets de crowding (une lettre est plus difficile à identifier lorsqu'elle est entourée de deux autres lettres), d'espacement des lettres (l'influence des lettres voisines diminue avec l'augmentation de l'espace qui les sépare) ou d'excentricité (l'influence des lettres voisines augmente avec la distance à la position du regard) (Pelli, Palomares, & Majaj, 2004).

En conclusion, le niveau trait, qu'il soit implémenté par des unités spécifiques ou résumé par une matrice de confusion, est indispensable pour un modèle qui veut rendre compte à la fois des effets lexicaux et des effets sur la perceptibilité des lettres.

**Le niveau « lettre »** est l'élément central de l'architecture à trois niveaux. Son rôle est double : il accumule l'information en provenance du niveau « trait » et transmet le résultat au niveau « mot ». Deux problématiques ont émergé en lien avec ce niveau. La première concerne l'encodage des lettres : doit-il porter à la fois sur l'identité et la position ou peut-on faire l'économie d'une des deux dimensions ? La seconde problématique porte sur les connexions avec le niveau « mot » : doit-on ajouter des connexions « descendantes » du niveau « mot » vers le niveau « lettre » au modèle ?

Seuls trois modèles encodent à la fois l'identité et la position des lettres : les modèles SCM, LTRS et MORSEL. Les autres modèles font le choix de n'encoder qu'une des deux informations et considèrent l'autre comme connue. Par exemple, dans les modèles IA, VA, MROM et BR, l'information sur la position des lettres est connue parfaitement, et l'identité des lettres est incertaine. C'est ce que l'on nomme le *slot coding*. Le fait de considérer la position comme connue, et donc de la fixer, contraint énormément ces modèles. Ils sont notamment incapable de rendre compte d'effets récents comme la transposition, l'insertion ou la suppression de lettre. A contrario, dans les modèles SERIOL et Overlap, c'est l'identité des lettres qui est parfaitement connue et le niveau « lettre » de ces modèles ne représente que l'information de position. Là

encore, ce choix limite énormément l'intérêt de ces modèles, en le restreignant à l'identification des mots.

La seconde question concerne la capacité qu'ont certains modèles de transférer de l'activation du niveau « mot » vers le niveau « lettre ». Présente dans le modèle IA, cette rétro-activation permet d'expliquer l'effet de supériorité du mot (une lettre est reconnue plus efficacement lorsqu'elle est présentée dans un mot que lorsqu'elle est présentée seule). Les modèles VA et MROM proposent une alternative permettant de s'affranchir de ces connexions, avec un processus de décision d'identification de la lettre basée à la fois sur le niveau « lettre » et sur le niveau « mot ». Le modèle MROM, dans sa version précédente (DROM ; Grainger & Jacobs, 1994) montre que si ces deux options permettent d'obtenir des performances similaires pour les mots, seul le modèle avec rétro-activation permet de rendre compte de l'avantage des pseudo-mots.

En conclusion, le codage partiel des lettres (identité ou position) est trop limitant pour un modèle. Il faut donc que le niveau « lettre » propose de coder la position et l'identité des lettres. De même, la capacité du modèle à renforcer l'information portée par le niveau « lettre » à partir de l'information du niveau « mot » est indispensable pour rendre compte de certains effets lexicaux.

**Le niveau « mot »** est l'aboutissement du processus de reconnaissance visuelle. En effet, pour presque tous les modèles présentés, c'est à partir de l'activation des unités qui encodent les mots que se fait l'identification du stimulus présenté ou que se prend la décision sur son appartenance aux mots de la langue.

Les modèles présentés étant implémentés (quand ils le sont) par des réseaux de neurones, les mots sont représentés par des unités qui possèdent une activation de base et une activation courante. Ces unités sont connectées soit directement à des unités « lettres », soit à des unités « groupement de lettres », comme les bigrammes ouverts. Deux modèles ne suivent pas cette architecture : le modèle MORSEL et le modèle LTRS. Dans le modèle MORSEL il n'y a pas d'unité « mot ». Le résultat du traitement du stimulus est représenté par les trigrammes du composant POnet qui sont activés à la fin du processus. Le modèle LTRS, quant à lui, n'a pas de niveau « mot » à proprement parler. Le modèle est dédié à simuler une tâche de choix forcé à deux alternatives et calcule la similarité entre le résultat du niveau « lettre » et chacune des deux alternatives.

Le schéma de fonctionnement de ce niveau est globalement le même pour tous les modèles. L'activation de base des unités « mot » est fonction de leur fréquence d'occurrence dans le lexique utilisé par le modèle. Dans les modèles IA, MROM et SCM, la fréquence est directement utilisée pour fixer l'activation de base des unités. Dans le modèle BR, cette fréquence est normalisée et sert de probabilité a priori  $P(W)$ . Pour finir, dans le modèle VA, la fréquence des mots détermine l'ordre de la vérification qui conduit à la décision. Norris (2006) a montré que favoriser les mots fréquents suffisait pour rendre compte de l'effet de fréquence, qui est un effet fondamental de la reconnaissance visuelle de mots. On peut noter que les modèles MORSEL, SERIOL, Overlap et LTRS n'ont pas implémenté cette fonction. La raison est liée aux tâches qu'ils simulent, qui sont peu sensibles à l'influence de la fréquence ; cependant, de ce fait, ils ne peuvent pas rendre compte de l'effet de fréquence.

L'activation initiale des mots du lexique est modifiée par l'activation au niveau « lettre », produite par la présentation d'un stimulus. Cette modification se fait sur la base d'un calcul de similarité entre le résultat du niveau « lettre » et l'information orthographique des unités mots. Pour le modèle LTRS, la similarité repose uniquement sur la présence des lettres dans le mot. Dans les modèles IA, AV, MROM, SCM et BR, l'activation est proportionnelle à la



présence des lettres et à leur niveau d'activation. Ainsi, une lettre présente mais peu activée aura peu d'effet sur les unités du niveau « mot ». Enfin, pour les modèles MORSEL et SERIOL, la similarité est basée sur la présence des groupements de lettres et leur niveau d'activation.

Enfin, certains modèles ajoutent un mécanisme de compétition interne entre les unités du niveau « mot » afin d'accélérer la convergence du processus d'identification. C'est le mécanisme d'inhibition des modèles IA, MROM, SCM et MORSEL ou le mécanisme de normalisation du modèle BR.

En conclusion, les ingrédients théoriques des modèles de connaissance lexicale semblent faire consensus : un avantage est donné aux mots fréquents sur les mots plus rares ; l'augmentation de l'activation des unités représentant les mots est influencée par la force de la similarité entre la représentation du stimulus construite au niveau « lettre » et les formes orthographiques mémorisées des mots connus ; et enfin, une compétition entre les unités du niveau « mot » permet de garantir la convergence du processus lorsque le stimulus est un mot connu.

**Conclusion et hypothèse** Un modèle unifié de reconnaissance de mots doit être à même de simuler des effets liés à la perceptibilité des lettres. Or, nous avons vu que cela nécessitait la présence d'un niveau « lettre ». Il doit également rendre compte des effets lexicaux majeurs comme l'effet de fréquence. Nous avons vu que cela nécessitait un niveau « mot », qui intègre comme état de base la fréquence d'occurrence des mots. Enfin, nous avons vu que la transmission de l'activation du niveau « trait » au niveau « mot » passait par un niveau « lettre » dont le rôle était l'intégration des informations en provenance du niveau « trait », permettant un calcul de similarité avec les éléments du niveau « mot ». Fort de ces conclusions nous proposons une première hypothèse pour notre modèle.

#### HYPOTHÈSE H1

L'architecture du modèle *BRAID* repose sur une structure à trois niveaux : un niveau « trait », un niveau « lettre » et un niveau « mot ».

- Le niveau « trait » permet d'extraire les informations du stimulus.
- Le niveau « lettre » permet d'encoder l'identité et la position des lettres et ses connexions avec le niveau « mot » doivent être bidirectionnelle.
- Le niveau « mot » donne un avantage aux mots fréquents et calcule l'activation sur la base de la similarité entre le niveau « lettre » et les mots du lexique.

### 2.1.2 Le traitement des lettres est-il parallèle ou séquentiel ?

La question posée ici est celle du traitement sensoriel des lettres du stimulus : les lettres sont-elles traitées toutes en même temps (on parle alors de traitement parallèle), ou bien les unes après les autres (on parle alors de traitement sériel ou séquentiel) ? La réponse choisie par les modèles est, le plus souvent, celle d'un traitement parallèle des lettres. C'est le cas des modèles IA, VA, MORSEL, BR, Overlap et LTRS.

Seuls deux modèles parmi les modèles présentés ont opté pour un traitement sériel des lettres du stimulus : le modèle SERIOL (C. Whitney, 2001) et le modèle SCM (Davis, 2010b). Dans le modèle SERIOL, on introduit un gradient de localisation pour favoriser l'extraction des attributs des lettres du stimulus, dans le sens de la lecture. Ainsi, les lettres qui se trouvent au début du mot dans le sens de la lecture sont activées les premières et les lettres qui terminent le mot sont

activées les dernières. Ce mécanisme permet par la suite de construire des bigrammes ouverts sans connaître la position des lettres dans le stimulus, mais en connaissant la temporalité des décharges d'activation correspondant au traitement des lettres.

Dans le modèle SCM, l'encodage de la position et de l'identité abstraite des lettres intervient simultanément. Pour rendre compte de la position indépendamment de l'identité des lettres, un processus de codage spatial est ajouté au niveau lettre. Ce codage spatial repose sur un parcours attentionnel rapide des détecteurs rétinotopiques dans le sens de la lecture qui conduit à une sérialisation de l'identification des lettres (Davis, 2010a).

Il est important de souligner que, dans ces deux modèles, les attributs visuels du stimulus sont extraits en parallèle. Ce n'est que lors de la construction de la représentation des lettres qu'un mécanisme spécifique vient sérialiser le processus.

On peut se demander ce que la sérialisation de l'identification des lettres du stimulus apporte au modèle. Dans le modèle SERIOL, le processus de sérialisation, appelé le *location gradient*, permet d'ordonner l'activation des lettres dans l'ordre de lecture. Cet ordre est ensuite utilisé pour construire les bigrammes ouverts qui constituent la sortie du niveau « lettre ». À la fin du processus, le niveau d'activation des bigrammes ouverts ne dépend que de la proximité des lettres. Il ne dépend donc ni de l'identité des lettres, ni du moment où elles sont activées, de sorte que l'activation transmise au niveau « mot » est indépendante du processus de sérialisation. Par conséquent, la sérialisation sert à définir les bigrammes ouverts qui représentent le stimulus et ce n'est que de cette manière qu'elle influence les performances de reconnaissance visuelle de mots. Dans le modèle SCM, la sérialisation ne sert qu'à justifier la manière dont le modèle encode la position moyenne des lettres. Cette position est explicite dans les équations et ne dépend pas du temps. Par conséquent, ce mécanisme n'a pas d'influence sur les performances de reconnaissance de mot, et le parcours attentionnel rapide s'apparente ici à un traitement parallèle dans lequel on connaîtrait la position des lettres.

**Conclusion et hypothèse** Le traitement parallèle des lettres du stimulus fait consensus au sein des modèles computationnels de reconnaissance de mots. Bien que deux modèles utilisent un traitement sériel des lettres du stimulus, il ne sert qu'à expliquer des choix d'encodage de la position des lettres et n'a aucune influence directe sur la reconnaissance des mots. Nous formulons donc notre deuxième hypothèse.

#### HYPOTHÈSE H2

Le traitement des lettres du stimulus est réalisé en parallèle.

## 2.2 L'encodage de la position des lettres doit-il être strict, relatif ou distribué ?

Dans les modèles présentés, on trouve trois types d'encodage de la position des lettres : le codage strict ou *slot coding*, qui consiste à encoder chaque lettre du stimulus dans la position qu'elle occupe dans la séquence et à la traiter indépendamment des autres, comme dans le modèle IA (McClelland & Rumelhart, 1981) ; l'encodage relatif qui consiste à encoder la position d'une lettre par rapport aux lettres précédentes, comme les *bigrammes ouverts* (C. Whitney, 2001), où la position d'une lettre est encodée par paires de lettres non nécessairement adjacentes ; et pour finir, l'encodage distribué qui consiste à encoder la position d'un lettre suivant les valeurs d'une distribution de probabilité définie sur l'ensemble des positions, comme dans le modèle Overlap (Gomez et al., 2008).

**L'encodage strict** consiste à encoder l'identité des lettres dans la position exacte où elles apparaissent dans la séquence de lettres qui forme le stimulus. On trouve cette forme d'encodage dans quatre modèles parmi ceux présentés précédemment : les modèles IA, VA, MROM et BR. Cette forme d'encodage implique donc que chaque lettre est identifiée isolément des autres. Par exemple, dans le modèle IA et ses dérivés (Grainger & Jacobs, 1996 ; McClelland & Rumelhart, 1981), les traits associés à une position dans le stimulus sont identifiés et produisent de l'activation uniquement pour la lettre dans la même position. Cette activation se propage ensuite, sous forme d'une activation excitatrice aux mots qui contiennent cette lettre dans la bonne position, et sous forme d'une activation inhibitrice aux autres. Ainsi, lors de la reconnaissance d'un mot, si deux lettres sont transposées dans le stimulus, elles produiront de l'activation inhibitrice qui limitera l'identification du mot. Or on sait depuis les travaux qui ont fait suite à la « lettre de Cambridge »<sup>4</sup> que deux lettres transposées n'empêchent pas la reconnaissance d'un mot.

En conclusion, le codage positionnel strict semble restrictif et nous excluons cette option.

**Le codage relatif** est la première alternative au codage strict. Elle consiste à encoder l'identité d'une nouvelle lettre conjointement avec une ou plusieurs lettres précédentes. Ceci amène à introduire de nouvelles structures, des groupements ordonnés de lettres, intercalés entre les représentations abstraites des lettres et les connaissances lexicales. Ces groupements peuvent être constitués de deux lettres (on parle alors de bigrammes), qui ne sont pas nécessairement adjacentes (on parle alors de bigrammes ouverts). La distance entre deux lettres peut être limitée : par exemple, il n'y a pas plus de deux lettres intercalées. On a présenté une implémentation des bigrammes ouverts dans le modèle SERIOL (C. Whitney, 2001), mais ils sont également présents dans le modèle de Grainger et van Heuven (2003) et le modèle Overlap Open-bigramme (Grainger et al., 2006). On trouve une structure similaire mais avec trois lettres (open-trigrammes) dans le modèle MORSEL (Mozer & Behrmann, 1990) et une structure avec des groupements de bigrammes dans le modèle LCD de Dehaene, Cohen, Sigman, et Vinckier (2005).

Ce mécanisme d'encodage de la position a apporté de la souplesse à l'encodage strict. Il permet notamment aux modèles de reconnaître des mots pour lesquels il manque une lettre (omission) ou pour lesquels une lettre a été ajoutée (insertion). Mais ce modèle se heurte, dans certains cas, au problème de la transposition, que ce soit lettre à lettre ou par blocs de lettres (Kinoshita & Norris, 2013).

En conclusion, le fait d'ajouter une structure spécifique à la reconnaissance visuelle de mots et l'incapacité de rendre compte des effets de transpositions nous incite à rejeter cette forme d'encodage de la position.

**Le codage distribué** est la seconde alternative à l'encodage strict. Il consiste à encoder l'identité de la lettre sur différentes positions. Cette idée est initialement proposée par Ratcliff (1981) et implémentée dans un modèle intégrant la reconnaissance d'objets spatiaux à attention visuelle, le modèle CODE (Logan, 1996). Une première implémentation du codage distribué de la position dans le cadre de la reconnaissance visuelle de mots est proposée par le modèle Overlap (Gomez et al., 2008) et reprise dans des versions ultérieures du modèle BR (Norris & Kinoshita, 2012 ; Norris et al., 2010) et dans le modèle SCM (Davis, 2010b).

---

<sup>4</sup>La lettre de Cambridge fait référence à un courrier électronique qui circula en septembre 2003. Le texte expliquait qu'ont pouvait lire un texte dont les lettres des mots sont mélangées, sans réelle difficulté du moment que la première et la dernière lettre sont les bonnes.

Voir <http://www.mrc-cbu.cam.ac.uk/people/dennis.norris/personal/cambridgeemail/> .

La « force » de l’encodage est définie par une distribution de probabilité gaussienne sur les positions possibles. Ainsi, si la variance de la distribution est petite, l’information sur la position de la lettre est précise (la courbe pique sur une position donnée, comme on peut le voir sur la Figure 2.11, première ligne et première position). Dans le modèle Overlap, la variance dépend de la position de la lettre dans le stimulus.

Cependant, les trois modèles Overlap, SCM et BR diffèrent à la fois par leur approche et par la représentation de l’incertitude positionnelle. Dans le premier modèle, le modèle Overlap, la variance est différente pour chaque position et fixée une fois pour toutes. Cette incertitude sur la position est obtenue expérimentalement et dépend des conditions de présentation du stimulus et des limites sensorielles des participants. De ce fait, les résultats obtenus ne représentent pas seulement un mécanisme d’incertitude positionnelle, mais plutôt la résultante de l’ensemble des traitements sensoriels pour chacune des positions. Dans le modèle SCM, l’incertitude positionnelle interagit avec l’acuité de sorte que la variance augmente avec l’excentricité mais reste constante dans le temps. Enfin, dans le modèle BR, version (Norris & Kinoshita, 2012 ; Norris et al., 2010), l’incertitude sur la position des lettres évolue au cours du temps. Pour chaque nouvel échantillon sensoriel, la variance qui caractérise l’incertitude sur la position se réduit de sorte que l’on peut disposer, à terme, de suffisamment d’échantillons pour que la connaissance de la position devienne certaine.

**Conclusion et hypothèse** Cette dernière forme d’encodage, l’encodage distribué, semble être la plus complète. D’ailleurs, c’est celle qui est adoptée par les modèles les plus récents comme le modèle SCM ou la dernière version du modèle BR (Norris & Kinoshita, 2012). Elle permet de rendre compte des effets d’insertion, d’omission et de transposition de lettres. Nous formulons donc la troisième hypothèse pour notre modèle.

### HYPOTHÈSE H3

L’encodage de la position de chaque lettre est distribué sur l’ensemble des positions du stimulus.

- La distribution de l’encodage ne dépend pas de la position.
- La distribution de l’encodage n’évolue pas au cours du temps (c’est une propriété rétinotopique).

## 2.3 Quelles sont les propriétés du système visuel qui modulent la perception des mots ?

Dans les modèles que l’on a étudiés précédemment, deux mécanismes influencent la reconnaissance visuelle de mots : l’acuité et l’attention. L’acuité se caractérise par un déclin des performances d’identification des lettres croissant avec l’excentricité. L’attention sert à sélectionner le mot à reconnaître dans une phrase, et une fois sélectionné, elle facilite l’extraction des attributs du stimulus dans les zones pertinentes, au détriment des autres.

**L’acuité visuelle** est représentée dans trois modèles parmi ceux étudiés ici : le modèle VA (Paap et al., 1982), le modèle SERIOL (C. Whitney, 2001) et le modèle SCM (Davis, 2010b). Le modèle VA place l’acuité au niveau de l’extraction des attributs qui composent les lettres du stimulus. Pour cela, il propose une matrice de confusion différente pour chaque position dans

le stimulus. Dans ce modèle, l'acuité influence directement les performances du traitement de bas niveau en diminuant la vitesse d'extraction des attributs des lettres en fonction de leur excentricité.

Dans le modèle SERIOL, l'acuité intervient au niveau de la couche d'entrée du modèle et se traduit par une diminution de l'activation des pixels avec l'augmentation de la distance au centre de la fovéa. Ce gradient d'acuité modélise la diminution du nombre de cônes en dehors de la fovéa. À la différence du modèle VA, des mécanismes sont mis en place dans ce modèle pour contrecarrer cet effet : un différentiel d'activation entre les hémisphères (favorisant l'hémisphère droit) et un gradient de localisation. Ainsi, dans ce modèle, la présence d'un modèle de l'acuité n'implique pas que les lettres excentrées sont moins bien identifiées.

Le dernier modèle qui implémente l'acuité visuelle est le modèle SCM. Dans ce modèle, l'acuité n'influence pas le processus d'identification des lettres, mais le processus qui détermine la position des lettres. Cette information de position des lettres est obtenue par un parcours sériel rapide des lettres du stimulus. Le résultat est une distribution gaussienne, centrée sur la position de la lettre, et dont la variance, qui représente l'incertitude sur la position, augmente linéairement avec la distance au point de fixation. Cette fois, l'acuité n'impacte pas directement l'identification de la lettre, mais influence les interactions des lettres entre elles. Ainsi, au point de fixation, l'encodage de l'identité de la lettre dépend essentiellement de la lettre fixée et de ses deux lettres adjacentes, tandis que pour une lettre excentrée, son identité dépendra peut-être aussi des voisins de rang 2 et plus.

**Conclusion et hypothèse** Si le rôle de l'acuité visuelle est indiscutable, la plupart des modèles en font l'économie. La raison en est simple : ces modèles sont étudiés expérimentalement dans des situations où l'acuité n'intervient qu'à la marge, par exemple où les mots présentés n'excèdent pas 5 lettres et ils sont toujours présentés au centre du champ de vision. Cependant, dès que l'on excentre le stimulus, que l'on utilise des stimuli plus longs ou que l'on manipule l'espace entre les lettres, on ne peut plus faire l'économie de modéliser l'acuité.

#### HYPOTHÈSE H4

Le modèle contient un mécanisme permettant de simuler l'effet de l'acuité.

**L'attention visuelle** est pratiquement absente des modèles computationnels de reconnaissance visuelle de mots. En effet, même si elle est suggérée dans le modèle VA par la phase de vérification qui découle des travaux de Becker (1976), elle n'est explicitement présente que dans le modèle MORSEL (Mozer & Behrmann, 1990).

Dans ce modèle, l'attention visuelle est représentée par un composant dédié dont le rôle est à la fois d'isoler les mots à traiter et de faciliter leur reconnaissance. Ce composant a une structure similaire à la couche d'entrée sur laquelle est codée le stimulus présenté au modèle. Le composant d'attention visuelle est connecté à la couche d'entrée, aux couches qui réalisent l'identification des lettres et à un contrôle cognitif supérieur. La connexion à la couche d'entrée autorise un contrôle *bottom-up* : c'est le stimulus qui guide l'attention. La connexion au contrôle cognitif supérieur permet un contrôle *top-down* : c'est la tâche qui contrôle l'attention. Enfin, par ses connexions avec les différentes couches de traitement, l'attention pilote le flux d'information tout au long du processus de reconnaissance.

Le composant attentionnel reçoit des instructions à la fois du contrôle cognitif et de la couche d'entrée. En mixant ces informations, il construit un « spot » attentionnel sur la zone visuelle à identifier. Dans cette zone, l'activité qui se propage dans le réseau est plus importante que dans

les autres zones, de sorte qu'en sortie de Blirnet, les groupements de lettres qui ont bénéficié de l'attention sont plus activés. On peut également noter que les groupements de lettres qui n'ont pas été sous le spot attentionnel sont quand même activés, mais dans une moindre mesure. Ce composant n'agit donc pas comme un filtre tout-ou-rien, mais stimule les zones d'intérêt et inhibe les autres.

Le modèle MORSEL utilise l'attention pour simuler des pathologies. En endommageant les connexions entre le composant attentionnel est la couche d'entrée, le modèle simule avec succès de nombreux effets liés à la dyslexie par négligence.

**Conclusion et hypothèse** L'attention visuelle permet à la fois d'isoler les mots lorsqu'ils sont présentés dans une phrase et de faciliter leur reconnaissance. Ce composant s'avère fondamental pour simuler des pathologies. Un modèle unifié de reconnaissance visuelle de mots se doit de rendre compte à la fois des normo-lecteurs et des pathologies, et à ce titre, il doit intégrer un composant attentionnel.

#### HYPOTHÈSE H5

Le modèle contient un composant attentionnel dont le rôle est de faciliter le traitement des portions d'intérêt du stimulus, au détriment des autres.

## 2.4 Conclusion

Dans ce chapitre nous avons présenté les modèles qui ont contribué au cours de ces dernières années au développement de l'approche computationnelle dans le cadre de la reconnaissance visuelle de mots. L'étude des spécificités de chaque modèle a mis en valeur la richesse et l'ingéniosité des acteurs de ce domaine d'activité.

Nous avons considéré des modèles généralistes, capables de simuler aussi bien les tâches d'identification de lettre que les tâches de reconnaissance de mot, comme les modèles IA (McClelland & Rumelhart, 1981) ou VA (Paap et al., 1982). Nous avons vu des modèles capables de simuler les tâches de décision lexicale comme les modèles MROM (Grainger & Jacobs, 1996) ou SCM (Davis, 2010b). Nous avons également présenté des modèles ayant des objectifs plus « spécifiques », comme le modèle SERIOL (C. Whitney, 2001) qui propose une approche séquentielle du traitement des lettres, le modèle Overlap (Gomez et al., 2008) qui introduit la notion de codage positionnel distribué, le modèle LTRS (Adelman, 2011) qui étudie la possibilité de traiter consécutivement l'identité et la position des lettres d'un stimulus, ou encore le modèle BR (Norris, 2006) qui apporte une interprétation probabiliste à l'effet de fréquence. Nous avons enfin trouvé un unique modèle, le modèle MORSEL (Mozer & Behrmann, 1990), qui suppose que l'attention visuelle est impliquée dès les premières étapes du traitement sensoriel du stimulus.

Puis, nous avons confronté ces différents modèles aux problématiques qui caractérisent la reconnaissance visuelle de mots : les niveaux « trait », « lettre » et « mot » sont-ils indispensables ? Le traitement des lettres est-il parallèle ou séquentiel ? L'encodage de la position des lettres doit-il être strict, relatif ou distribué ? Et enfin, l'acuité et l'attention doivent-elles être intégrées dans un modèle de reconnaissance visuelle de mots ? Ainsi, pour chacune de ces questions, nous avons proposé une réponse sur la base des critères suivants :

- privilégier les mécanismes qui font consensus,
- privilégier les mécanismes génériques aux mécanismes spécifiques (voire, ad-hoc).

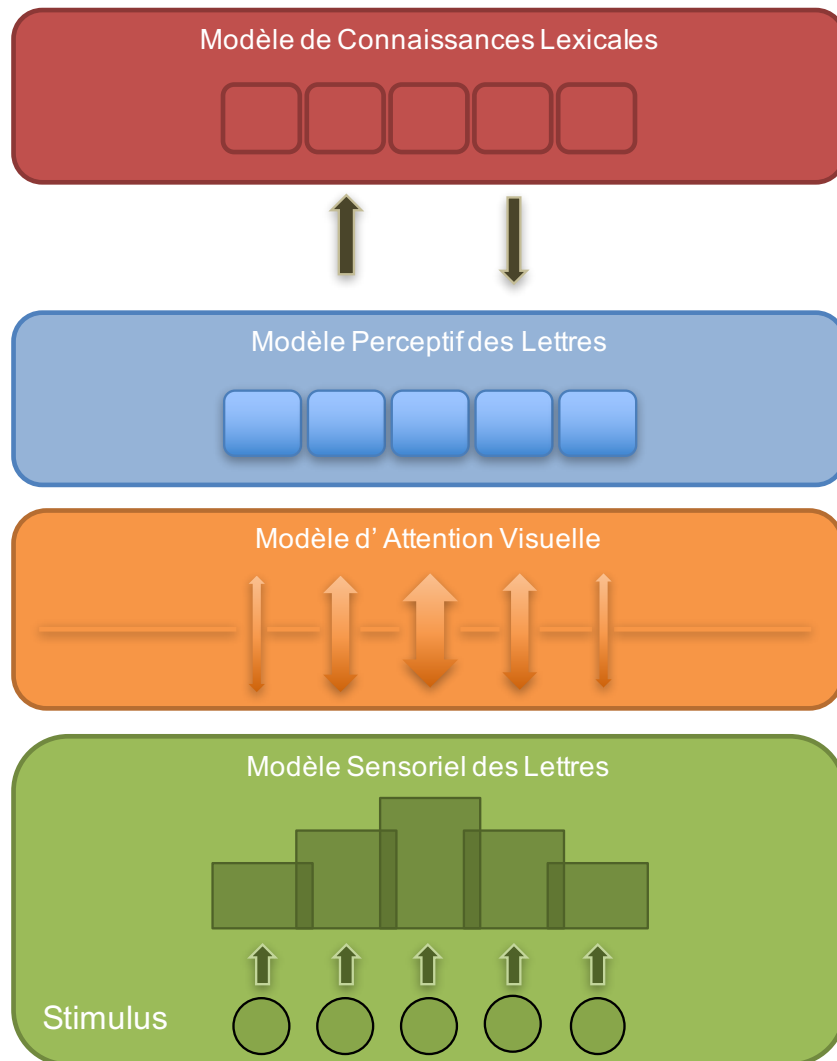


Figure 2.18 – **Illustration du modèle conceptuel.** On retrouve les trois modules classiques dans les modèles de lecture : le sous-modèle sensoriel des lettres (niveau « trait », en vert), le sous-modèle perceptif des lettres (niveau « lettres », en bleu) et le sous-modèle des connaissances lexicales (niveau « mot », en rouge). Le sous-modèle sensoriel intègre l'acuité visuelle matérialisée par la différence de hauteur résultant du traitement sensoriel, et des interférences latérales dues au codage distribué et matérialisées par le chevauchement des rectangles résultant du traitement sensoriel. À cette structure de base s'ajoute le sous-modèle d'attention visuelle qui joue le rôle de filtre attentionnel entre le sous-modèle sensoriel et le sous-modèle perceptif. (Figure répétée de la Figure 1.1, page 4.)

Les réponses apportées à ces questions nous conduisent à proposer un modèle théorique conceptuel de la reconnaissance visuelle de mot présenté Figure 2.18. Ce modèle conceptuel repose sur cinq hypothèses théoriques, qui constituent le « squelette » de son architecture. Ces cinq hypothèses sont rappelées ici.

#### HYPOTHÈSES DU MODÈLE CONCEPTUEL

**HYPOTHÈSE H1 :** L'architecture du modèle *BRAID* repose sur une structure à trois niveaux : un niveau « trait », un niveau « lettre » et un niveau « mot ».

- Le niveau « trait » permet d'extraire les informations du stimulus.
- Le niveau « lettre » permet d'encoder l'identité et la position des lettres et ses connexions avec le niveau « mot » doivent être bidirectionnelle.
- Le niveau « mot » donne un avantage aux mots fréquents et calcule l'activation sur la base de la similarité entre le niveau « lettre » et les mots du lexique.

**HYPOTHÈSE H2 :** Le traitement des lettres du stimulus est réalisé en parallèle.

**HYPOTHÈSE H3 :** L'encodage de la position de chaque lettre est distribué sur l'ensemble des positions du stimulus.

- La distribution de l'encodage ne dépend pas de la position.
- La distribution de l'encodage n'évolue pas au cours du temps (c'est une propriété rétinotopique).

**HYPOTHÈSE H4 :** Le modèle contient un mécanisme permettant de simuler l'effet de l'acuité.

**HYPOTHÈSE H5 :** Le modèle contient un composant attentionnel dont le rôle est de faciliter le traitement des portions d'intérêt du stimulus, au détriment des autres.





# Chapitre 3

## *BRAID* Bayesian word Recognition using Attention, Interference and Dynamics

Nous venons de définir un « modèle conceptuel », un ensemble d'hypothèses qui résultent de notre analyse critique de la littérature dans le domaine de la reconnaissance visuelle de mots, et qui fournissent le squelette de l'architecture de notre modèle. Dans la suite, notre tâche est d'implémenter ce modèle mathématiquement, pour passer d'un modèle conceptuel à un modèle computationnel. Pour cela, nous choisissons un cadre de modélisation, la *modélisation bayésienne algorithmique* (Diard, 2015).

Ce cadre de modélisation hérite de la Programmation Bayésienne des Robots (Bessière, Dedieu, Lebeltel, Mazer, & Mekhnacha, 1998a, 1998b ; Lebeltel, Bessière, Diard, & Mazer, 2004) et de sa généralisation, la Programmation Bayésienne (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013), qui sont toutes les deux des méthodologies de programmation de logiciels ou de robots, appliquées largement dans des domaines variés de l'intelligence artificielle (Bessière, Laugier, & Siegart, 2008). La modélisation bayésienne algorithmique consiste à utiliser cette méthodologie, inchangée, dans le domaine de la définition de modèles d'agent cognitif (Colas, Diard, & Bessière, 2010), en les décrivant au niveau « algorithmique », au sens de la hiérarchie de Marr (1982). Avant de donner la définition mathématique de notre modèle computationnel, nous décrivons succinctement la méthodologie commune à la Programmation Bayésienne des Robots, à la Programmation Bayésienne, et à la modélisation bayésienne algorithmique, de manière à en introduire les concepts et notations nécessaires. Malheureusement, nous ne pourrons pas, ici, décrire de manière extensive le formalisme, son pouvoir d'expression, sa généralité, et les moyens de sa mise en œuvre ; des traitements détaillés sont disponibles ailleurs (Bessière et al., 2013 ; Diard, 2015).

### 1 Méthodologie : modélisation bayésienne algorithmique

L'essence de cette méthodologie consiste à décrire les connaissances d'un agent cognitif par des probabilités (Jaynes, 2003). Nous ne rappelons pas ici les bases mathématiques des probabilités ou du calcul probabiliste, et les supposons connues. Rappelons simplement que les distributions de probabilités sur des variables discrètes et finis suivent deux règles de calcul : la règle de la somme et la règle du produit. La règle de la somme, aussi appelée la règle de normalisation, indique que la somme des probabilités des cas possibles est 1. La règle du produit, elle, parle de la distribution de probabilité sur deux variables,  $P(X Y)$ , et la décompose en un produit de

termes :

$$\begin{aligned} P(X \ Y) &= P(X \mid Y)P(Y) \\ &= P(Y \mid X)P(X) , \end{aligned}$$

où  $P(X \mid Y)$  est la « famille » de distribution de probabilité de  $X$  conditionnelle à  $Y$ , et se lit « probabilité de  $X$  sachant  $Y$  ». Une variante d'écriture de la règle du produit est le théorème de Bayes :

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}, \text{ si } P(Y) \neq 0 .$$

Un modèle bayésien algorithmique est donc un modèle probabiliste, et la méthodologie donne un guide pour la définition de ce modèle. Au lieu de présenter « théoriquement » cette méthodologie, nous l'introduisons en nous appuyant sur un exemple inventé, proche de notre domaine d'intérêt, la reconnaissance visuelle de mots. Nous développerons un modèle bayésien algorithmique de la reconnaissance visuelle d'un mot de trois lettres, en supposant que les lettres qui le composent sont directement observées ; ce qui est évidemment peu réaliste. Nous complétons cet exemple avec quelques définitions de notations, ou de propriétés intéressantes.

## 1.1 Variables du modèle

La première étape de la modélisation bayésienne algorithmique consiste à lister les variables sur lesquelles portent le modèle. Techniquement, chacune des variables du modèle est une *variable probabiliste*, c'est-à-dire l'association entre un symbole (le nom de la variable) et un ensemble de valeurs possibles pour cette variable (un domaine). Dans l'ensemble de notre travail, les domaines des variables probabilistes que nous manipulons sont discrets et finis (par exemple, l'ensemble des lettres de l'alphabet). Cette notion s'oppose à la notion de variable continue pour lesquelles le domaine de définition est représenté par un ensemble infini, ou un intervalle réel, donc non dénombrable (par exemple n'importe quelle valeur réelle entre 0,25 et 0,75).

Dans notre exemple, les mots sont composés de trois lettres. Nous avons ainsi quatre variables probabilistes.

- La variable  $W$  représente l'identité du mot. Son domaine est l'ensemble des mots possibles : dans notre exemple, considérons qu'il n'y a que trois mots possibles, et  $W = \{w_1, w_2, w_3\}$ .
- Trois variables  $L_1$ ,  $L_2$  et  $L_3$  qui, chacune, représente une des lettres du mot. Chacune de ces variables a un domaine qui décrit les lettres possibles de l'alphabet. Par la suite, nous utiliserons un domaine,  $\mathcal{D}_L = \{a, b, \dots, z, \$\}$ , avec une valeur possible pour chaque lettre de 'a' à 'z', et un symbole particulier, '\$', pour représenter une lettre illisible ou manquante.

Remarquons la notation utilisée ici et par la suite. Lorsqu'elles sont indicées, les variables sont « localisées spatialement » : ainsi, dans notre exemple,  $L_1$  est la première lettre du mot,  $L_2$  est la deuxième lettre du mot, et  $L_3$  est la dernière lettre du mot. Plus tard, nous décrirons l'évolution temporelle des variables, et les annoterons par des exposants : ainsi,  $L_2^t$  représentera la lettre en seconde position au temps  $t$ . Introduisons ici quelques raccourcis de notation :

- $X^{1:T}$  est la conjonction des variables  $X^t$  pour les valeurs de temps  $t$  comprises entre 1 et  $T$ , c'est-à-dire  $X^{1:T} = (X^1, X^2, \dots, X^T)$  ;

- ce même raccourci de notation s'applique, pour un pas de temps  $t$  donné, aux positions spatiales. Ainsi  $X_{1:N}^t$  est la conjonction des variables  $X_n^t$ , pour les positions  $n$  allant de 1 à  $N$ , c'est-à-dire  $X_{1:N}^t = (X_1^t, X_2^t, \dots, X_N^t)$  ;
- ce raccourci se combine pour lister simultanément des variables sur plusieurs instants de temps et plusieurs positions, par exemple :

$$X_{1:N}^{1:T} = (X_1^1, X_1^2, \dots, X_1^T, X_2^1, X_2^2, \dots, X_2^T, \dots, X_N^1, X_N^2, \dots, X_N^T) ;$$

- nous notons  $[X_n^t = x_n^t]$  l'assignation de la valeur  $x_n^t$  à la variable probabiliste  $X_n^t$ , et, lorsque celle-ci est implicite, nous l'abrégeons en  $x_n^t$ .

## 1.2 Distribution conjointe et décomposition

La seconde étape de la modélisation bayésienne algorithmique consiste à définir la *distribution de probabilité conjointe du modèle*, c'est-à-dire la distribution de probabilité sur l'espace conjoint formé par toutes les variables listées précédemment. Ainsi, dans notre exemple, c'est la distribution de probabilité conjointe  $P(W L_1 L_2 L_3)$  que nous devons définir. Les distributions de probabilité conjointe portent souvent sur des espaces trop compliqués pour être définies directement. Ainsi, en appliquant la règle du produit, nous décomposons la distribution de probabilité conjointe en un produit de termes. Dans notre exemple, nous choisissons de définir :

$$P(W L_1 L_2 L_3) = P(W)P(L_1 | W)P(L_2 | L_1 W)P(L_3 | L_2 L_1 W) .$$

À cette étape, nous introduisons des *hypothèses d'indépendance conditionnelle* pour simplifier certains des termes de ce produit. On dit que deux variables  $A$  et  $B$  sont indépendantes, conditionnellement à une variable  $C$ , lorsque la probabilité de  $A$  connaissant la valeur de la variable  $C$  ne dépend pas des valeurs de la variable  $B$ , ce qui s'écrit :  $P(A | B C) = P(A | C)$ . Dans notre exemple, nous faisons l'hypothèse que la principale source d'information sur les lettres du mot, est l'identité  $w$  du mot. En d'autres termes, si on suppose le mot  $w$  connu, on peut prédire chacune de ses lettres, indépendamment de la connaissance des autres lettres. Ainsi, le terme  $P(L_2 | L_1 W)$  se simplifie en  $P(L_2 | W)$  : on fait l'hypothèse que  $L_2$  est indépendant de  $L_1$ , conditionnellement à la connaissance de  $W$ . Une hypothèse similaire simplifie  $P(L_3 | L_2 L_1 W)$  en  $P(L_3 | W)$ . Nous obtenons une définition « sous hypothèse » de la distribution de probabilité conjointe :

$$P(W L_1 L_2 L_3) = P(W)P(L_1 | W)P(L_2 | W)P(L_3 | W) . \quad (3.1)$$

On vérifie aisément que les hypothèses d'indépendance conditionnelle permettent de réduire la complexité du modèle : le terme  $P(L_3 | L_2 L_1 W)$  a été approximé par  $P(L_3 | W)$ , qui est une forme mathématique plus simple. En effet, dans  $P(L_3 | L_2 L_1 W)$  il y a autant de distributions de probabilités sur  $L_3$  qu'il y a de combinaisons possibles de valeurs de  $L_2$ ,  $L_1$  et  $W$  (dans notre exemple,  $27*27*3$ ), alors que dans  $P(L_3 | W)$ , il y a seulement trois distributions de probabilités sur  $L_3$ .

## 1.3 Formes paramétriques

Pour rendre opérationnelle la définition du modèle, il faut donner un moyen de calculer chacun des termes qui apparaissent dans la décomposition de la distribution conjointe. Nous associons à chacun de ces termes une *forme paramétrique*, c'est-à-dire une famille paramétrée de distributions de probabilités. Par exemple, nous choisissons que tel terme est une distribution

gaussienne de probabilités (de paramètres libres  $\mu, \sigma$ ), que tel terme est une table de probabilités conditionnelles (avec autant de paramètres que de cases dans les tables), que tel terme est une distribution uniforme de probabilité (sans paramètre libre), etc.

Dans notre exemple, nous devons choisir quatre formes paramétriques, une pour chacun des termes de l'Équation (3.1). Nous choisissons par exemple de définir le terme  $P(W)$  par une distribution uniforme : ainsi,  $P([W = w_1]) = P([W = w_2]) = P([W = w_3]) = 1/3$ . Considérons ensuite le terme  $P(L_1 | W)$  : pour un mot  $w$ , il n'y a qu'une lettre possible en première position, et cette lettre est connue sans aucune incertitude. Ainsi,  $P(L_1 | [W = w])$  est une distribution de probabilité Dirac, c'est-à-dire que la probabilité est 1 sur la bonne lettre, et 0 partout ailleurs. Dans notre exemple, imaginons que le mot  $w_1$  est le mot « bal » : alors  $P([L_1 = 'b'] | [W = w]) = 1$ , et  $\forall l, l \neq 'b', P([L_1 = l] | [W = w]) = 0$ . Définissons que  $w_2$  et  $w_3$  sont les mots « bol » et « lac » : tous les termes  $P(L_1 | W)$ ,  $P(L_2 | W)$  et  $P(L_3 | W)$  sont facilement définis de manière similaire.

Avec ces choix de formes paramétriques pour notre exemple, il ne reste aucun paramètre libre dans la définition de la distribution de probabilité conjointe. Dans le cas général, cependant, il peut rester à cette étape des paramètres libres, et on complète la définition du modèle en indiquant la manière de les calculer à partir, le plus souvent, d'une base de données expérimentale. C'est une étape d'*apprentissage de paramètres* ou de calibrage, dont nous nous passons dans notre exemple.

Notre modèle introductif est donc entièrement spécifié, et sa distribution de probabilité conjointe est calculable en tout point. La phase *déclarative* de la méthodologie, c'est-à-dire la phase dans laquelle on définit sous forme probabiliste l'ensemble des connaissances du modèle, est alors terminée.

## 1.4 Inférence bayésienne

Le modèle étant entièrement spécifié, nous entrons maintenant dans une phase *procédurale*, c'est-à-dire que nous allons maintenant mettre en œuvre le modèle, et les connaissances qu'il contient, pour simuler des tâches cognitives. Pour simuler une tâche, nous définissons une *question probabiliste*, c'est-à-dire la distribution de probabilité d'intérêt qui modélise cette tâche.

Dans notre exemple, nous considérons les tâches de reconnaissance du mot, au vu d'une ou des lettres qui le composent. Par exemple, si la tâche est de reconnaître le mot alors qu'on sait toutes les lettres, la question probabiliste est  $P(W | l_1 l_2 l_3)$ . Si on ne connaît que la première lettre, la question est  $P(W | l_1)$  ; si on connaît la première et la dernière lettre, mais pas la seconde, la question est  $P(W | l_1 l_3)$ , etc.

Remarquons que ces questions sont des distributions de probabilités qui ne sont pas contenues dans la définition de la distribution conjointe du modèle (voir Équation (3.1)). Il faut donc les calculer, par *inférence bayésienne*, c'est-à-dire appliquer les règles du calcul probabiliste pour, à partir de la distribution conjointe, obtenir une expression qui réponde à la question probabiliste (un théorème montre que, la conjointe étant définie, on peut ainsi répondre à n'importe quelle question probabiliste qui porte sur les variables de la conjointe). Cette étape d'inférence ne requiert plus l'intervention du modélisateur, c'est-à-dire que, la question et la conjointe étant posées, la réponse est unique et mathématiquement entièrement déterminée (et trouvable automatiquement). Considérons par exemple la question  $P(W | l_1 l_2 l_3)$ . L'inférence bayésienne

donne :

$$\begin{aligned} P(W \mid l_1 l_2 l_3) &= \frac{P(W l_1 l_2 l_3)}{P(l_1 l_2 l_3)} \\ &= \frac{1}{Z} P(W l_1 l_2 l_3) . \end{aligned}$$

En effet, le terme au dénominateur,  $P(l_1 l_2 l_3)$  est une constante, les valeurs de  $l_1$ ,  $l_2$  et  $l_3$  étant connues. On peut ainsi noter  $Z = P(l_1 l_2 l_3)$ . Il est souvent plus aisé de ne pas calculer cette constante  $Z$  explicitement, et de l'obtenir en fin de calcul par renormalisation de la distribution. On peut donc procéder en calculant « à un facteur près », c'est-à-dire en ne s'intéressant qu'au numérateur : en notant  $\propto$  la relation « est proportionnel à », on réécrit :

$$P(W \mid l_1 l_2 l_3) \propto P(W l_1 l_2 l_3) .$$

Ensuite, on remarque que le calcul n'implique que la distribution conjointe, et on la remplace par sa définition :

$$P(W \mid l_1 l_2 l_3) \propto P(W)P(l_1 \mid W)P(l_2 \mid W)P(l_3 \mid W) .$$

Ici, tous les termes de la distribution conjointe interviennent, mais pour certaines questions, il est possible de simplifier. Par exemple, considérons la question de la reconnaissance du mot ne connaissant que sa première lettre :

$$\begin{aligned} P(W \mid l_1) &= \frac{\sum_{L_2, L_3} P(W l_1 L_2 L_3)}{P(l_1)} \\ &\propto \sum_{L_2, L_3} P(W l_1 L_2 L_3) \\ &\propto \sum_{L_2, L_3} P(W)P(l_1 \mid W)P(L_2 \mid W)P(L_3 \mid W) \\ &\propto P(W)P(l_1 \mid W) \sum_{L_2} P(L_2 \mid W) \sum_{L_3} P(L_3 \mid W) \\ &\propto P(W)P(l_1 \mid W) . \end{aligned}$$

Remarquons que l'on peut interpréter ces calculs comme des processus d'inversion probabiliste de connaissance : le modèle contient des termes qui « prédisent » les lettres pour chaque mot, et, dans ces inférences, la règle de Bayes permet de calculer la relation inverse, dans laquelle, une ou plusieurs lettres étant connues, on « reconnaît » le mot.

Pour conclure notre exemple, imaginons que la première lettre est un 'b', et calculons  $P(W \mid [L_1 = 'b'])$ . Nous obtenons une distribution dans laquelle les mots  $w_1$  (« bal ») et  $w_2$  (« bol ») sont équiprobables, de probabilité 0,5, et le mot  $w_3$  (« lac ») a probabilité 0.

## 2 Présentation des modules de *BRAID*

Au chapitre précédent, nous avons défini un modèle conceptuel (Figure 2.18) contenant un ensemble d'hypothèses sur notre modèle de reconnaissance visuelle de mots. Ce modèle est une architecture, une structure, qui contient des sous-modèles et leurs connexions. Pour traduire mathématiquement ce modèle conceptuel, nous avons présenté un formalisme, la modélisation bayésienne algorithmique, dans lequel un modèle probabiliste est essentiellement défini par

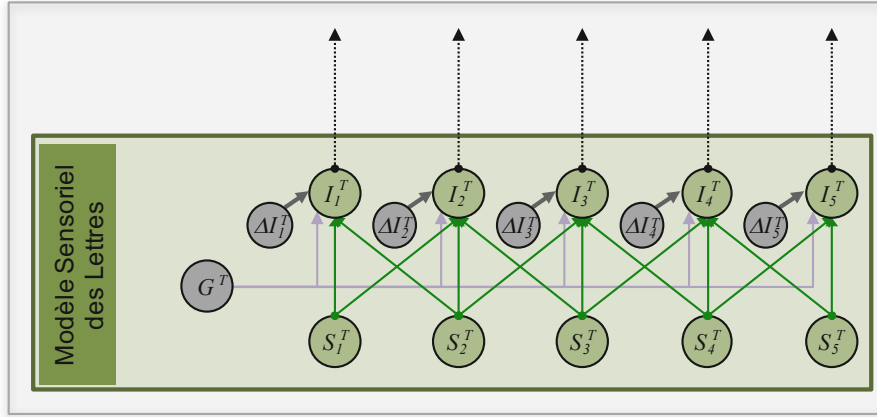


Figure 3.1 – **BRAID**, sous-modèle sensoriel des lettres. Les variables  $S_n^T$  décrivent le stimulus, la variable  $G^T$  décrit la position du regard sur le stimulus et les variables  $\Delta I_n^T$  décrivent les interférences produites par les lettres voisines pour l'identification d'une lettre donnée. Les variables  $I_n^T$  dépendent d'au plus trois lettres du stimulus, de la position du regard et des interférences. Elles décrivent le résultat du traitement sensoriel simulé par le modèle.

sa distribution de probabilité conjointe. Nous allons donc, maintenant, montrer comment nous implémentons notre modèle conceptuel sous la forme d'une distribution de probabilité conjointe. Le modèle obtenu étant structuré, comme l'est le modèle conceptuel, nous nous permettons de définir ce modèle en séquence, présentant tout d'abord chacun des sous-modèles, puis, ensuite, la manière dont ils sont connectés. Pour chacun de ces sous-modèles, nous le présentons comme un modèle bayésien algorithmique, c'est-à-dire en définissant tout d'abord la liste des variables pertinentes, la décomposition de sa distribution conjointe, et enfin, la définition des formes paramétriques des termes qui apparaissent dans cette décomposition.

## 2.1 Sous-modèle sensoriel des lettres

Le premier sous-modèle que nous décrivons est le sous-modèle sensoriel des lettres. Il a pour objectif de rendre compte du processus d'extraction des attributs des lettres du stimulus (**HYPOTHÈSE H1**, niveau « trait »). De tels processus sont décrits par Grainger, Dufau, et Ziegler (2016) ou par le modèle LCD de Dehaene et al. (2005). Parmi les différents ingrédients attendus dans ce sous-modèle, on trouve des éléments permettant l'extraction des traits caractéristiques des lettres, des éléments en charge de pénaliser cette extraction en fonction de l'excentricité de la lettre considérée (**HYPOTHÈSE H4**, acuité) et des éléments en charge d'intégrer de l'information en provenance des lettres adjacentes (**HYPOTHÈSE H3**, interférences latérales). Ce sous-modèle décrit de manière probabiliste ces différents éléments, ainsi que les relations de dépendance entre eux. Le schéma complet de dépendances du sous-modèle sensoriel des lettres est présenté Figure 3.1.

**Variables du modèle** Les variables qui composent le sous-modèle sensoriel des lettres sont les suivantes :

- $S_{1:N}^{1:T}$  : ces variables représentent les images des lettres qui constituent le stimulus. Le processus du traitement visuel des images pour arriver à l'identification des lettres est un sujet à part entière (Grainger, Rey, & Dufau, 2008). Ici, nous ne souhaitons pas modéliser en détail ce processus, mais plutôt nous concentrer sur sa dynamique et les interactions

entre les processus d'identification de lettres voisines. Ainsi, au lieu de représenter le stimulus par des images (tableaux de pixels, par exemple), nous le représentons par l'identité de la lettre contenue dans le stimulus. Les variables  $S_{1:N}^{1:T}$  prennent donc leurs valeurs dans l'espace formé par les lettres de l'alphabet, enrichi du caractère '\$' qui correspond à une lettre illisible ou manquante. Cet ensemble constitue le domaine de définition de ces variables et on le note :  $\mathcal{D}_L = \{a, b, \dots, z, \$\}$ . On note  $|\mathcal{D}_L|$  le cardinal de l'espace  $\mathcal{D}_L$ , c'est-à-dire  $|\mathcal{D}_L| = 27$ .

- $G^{1:T}$  : ces variables représentent la position du regard sur le stimulus et servent à calculer le coefficient d'acuité à appliquer sur chaque position. On fait l'hypothèse que le lecteur pose son regard soit sur une lettre du stimulus, soit à mi-chemin entre deux lettres. Ces variables sont donc discrètes, de domaine de définition  $\mathcal{D}_{Pos} = [1, 1,5, 2, \dots, N - 0,5, N]$ ,  $N$  étant le nombre de lettres du stimulus.
- $\Delta I_{1:N}^{1:T}$  : ces variables encodent le décalage entre la lettre traitée et la lettre à identifier, pour permettre de représenter les interférences latérales entre les processus d'identification de lettres voisines. Dans la version de *BRAID* que nous présentons dans ce document, seules les lettres adjacentes interfèrent avec la lettre à identifier. Nous définissons donc le domaine des variables  $\Delta I_n^t$  par  $\mathcal{D}_{\Delta I} = \{-1, 0, 1\}$ , où  $-1$  correspond à la position immédiatement à gauche de la position de la lettre à identifier, et  $+1$  la position immédiatement à droite.
- $I_{1:N}^{1:T}$  : ces variables représentent le résultat du processus complet d'identification d'une lettre. Comme elles représentent l'identité des lettres, elles sont définies sur le même domaine que les variables  $S_{1:N}^{1:T}$ , à savoir  $\mathcal{D}_L$ .

**Distribution conjointe et décomposition** Ce sous-modèle ne contient qu'une relation de dépendance conditionnelle spatiale, et aucune relation de dépendance temporelle. Nous pouvons donc factoriser les éléments à chaque pas de temps, ce qui conduit à la décomposition de la distribution conjointe suivante :

$$P(S_{1:N}^{1:T} I_{1:N}^{1:T} \Delta I_{1:N}^{1:T} G^{1:T}) = \prod_{t=1}^T \left[ P(G^t) \prod_{n=1}^N [P(S_n^t) P(\Delta I_n^t) P(I_n^t | S_{1:N}^t \Delta I_n^t G^t)] \right] \quad (3.2)$$

Pour un pas de temps donné, nous choisissons que chaque variable  $I_n^t$  en position  $n$  est indépendante des autres variables  $I_{n'}^t$  aux autres positions ( $n' \neq n$ ). Ainsi, il apparaît un produit de termes  $P(I_n^t | \dots)$ , et les variables  $I_n^t$  sont bien séparées en position. Cependant, elles restent dépendantes des variables  $S_{1:N}^t$  sur toutes les positions (en réalité, uniquement les voisines directes grâce aux variables  $\Delta I_n^t$ ), pour représenter les interférences latérales. Cette structure de dépendance implémente donc, dans notre modèle, l'**HYPOTHÈSE H3** de codage distribué.

**Formes paramétriques** Pour finir la description de ce sous-modèle, nous définissons les formes paramétriques des termes qui apparaissent dans la décomposition de la distribution conjointe.

- $\forall t, n, P(S_n^t)$  sont les distributions de probabilité a priori sur les lettres composant le stimulus. Nous faisons l'hypothèse que toutes les lettres ont la même probabilité d'apparaître, ce que l'on modélise par des distributions discrètes uniformes sur le domaine  $\mathcal{D}_L$ . Ce choix n'a pas d'incidence pratique sur les inférences qui sont décrites dans ce manuscrit. En effet, elles sont toutes conditionnées à la présentation d'un stimulus et ainsi, à la connaissance



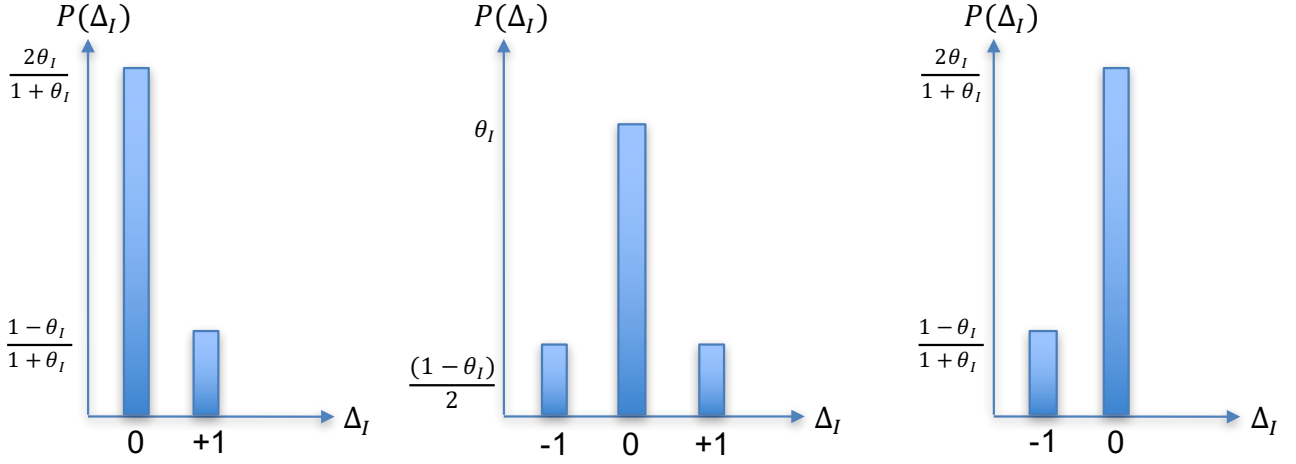


Figure 3.2 – **BRAID**, distributions de probabilité  $P(\Delta I_n^t)$ . A gauche, les distributions de probabilité des variables  $\Delta I_1^{1:T}$  (lettre à identifier en première position) ; à droite, les distributions de probabilité des variables  $\Delta I_N^{1:T}$  (lettre à identifier en dernière position) ; au centre, distribution de probabilité des  $\Delta I_n^{1:T}$  pour les lettres internes du stimulus.  $\theta_I$  est le paramètre qui sert à contrôler l’interférence, plus il est grand et plus les interférences des lettres adjacentes sont importantes (ici  $\theta_I = 0,325$ , voir Chapitre 4, Section 2.2.4).

$s_n^t$  pour chaque variable  $S_n^t$ . Ainsi, la valeur de probabilité  $P([S_n^t = s_n^t])$  peut toujours être intégrée à la constante de normalisation.

- $\forall t, P(G^t)$  sont des distributions de probabilité qui modélisent la position du regard. Nous faisons l’hypothèse générale que le regard peut se poser indifféremment sur toutes les positions du stimulus. Cette hypothèse peut-être remise en question dans certaines expériences (utilisation d’une croix de fixation qui fixerait la position du regard), mais dans ce manuscrit, toutes les inférences seront conditionnées par la connaissance préalable de la position du regard. Ainsi, comme précédemment, ce choix n’aura pas d’incidence en pratique car  $P([G^t = g^t])$  sera intégré à la constante de normalisation.
- $\forall t, n, P(\Delta I_n^t)$  sont les distributions des différents décalages possibles pour l’encodage d’une lettre donnée (voir Figure 3.2). Ce sont donc des distributions de probabilité discrètes sur l’espace des décalages autorisés  $\mathcal{D}_{\Delta I}$ , c’est-à-dire des distributions sur trois positions spatiales (la lettre à identifier, et ses deux voisines directes). Ces distributions représentent par un paramètre  $\theta_I$  la « force » des interférences latérales entre ces positions. Nous faisons le choix de supposer ces interférences symétriques pour les lettres internes du mot (en position  $n$  avec  $n \neq 1$  et  $n \neq N$ ). Nous définissons :

$$P([\Delta I_n^t = i]) = \begin{cases} 1 - \theta_I & \text{si } i = 0 \\ \theta_I/2 & \text{si } i = -1 \text{ ou } i = +1 . \end{cases}$$

Pour les lettres en position 1 et  $N$ , il n’y a qu’un voisin, et « toute l’interférence vient de ce voisin » :

$$P([\Delta I_n^t = i]) = \begin{cases} \frac{2\theta_I}{1+\theta_I} & \text{si } i = 0 \\ \frac{1-\theta_I}{1+\theta_I} & \text{si } i = -1, n = N \text{ ou } i = +1, n = 1 . \end{cases}$$

- $\forall t, n, P(I_n^t | S_{1:N}^t, \Delta I_n^t, G^t)$  sont les distributions des variables représentant le résultat du décodage sensoriel du stimulus. Ce sont des distributions discrètes sur l’espace des lettres

$\mathcal{D}_L$ , fonction du stimulus, des interférences latérales, et de la position du regard. Ces trois effets se mélangent, et la définition mathématique de ces distributions est un peu complexe ; nous construisons cette définition en utilisant des intermédiaires « pédagogiques », introduisant les effets les uns après les autres.

Considérons tout d'abord des distributions de la forme  $P(I_n^t | S_n^t)$ , qui représenteraient l'identification directe des lettres du stimulus, sans interférence latérale ni effet de la position de l'œil. Ces distributions discrètes sont des tables carrées de probabilités conditionnelles, de l'espace des lettres  $\mathcal{D}_L$  vers lui-même. Les paramètres de ces distributions peuvent être identifiés à partir de matrices de confusion expérimentales que l'on trouve dans la littérature (pour une revue, voir Mueller & Weidemann, 2012).

Notons  $\{p_{i,s}\}_{i,s \in \mathcal{D}_L^2}$  la probabilité d'identifier la lettre  $i$  sachant que la lettre  $s$  était présentée. Ces probabilités sont le résultat d'un traitement sensoriel de plusieurs dizaines de millisecondes qui accumule trop d'information sensorielle, par rapport à la granularité temporelle que nous souhaitons. Nous appliquons donc une première transformation sur les paramètres  $\{p_{i,s}\}$ , en les dégradant à l'aide d'un paramètre  $Scale_I$ . Cette transformation s'applique à tous les  $\{p_{i,s}\}$ , ce qui a pour effet de ralentir globalement la dynamique du modèle, sans changer les confusions entre les lettres (techniquement, nous mélangeons la matrice de confusion avec une distribution uniforme, ce qui la « dilue »). Nous définissons ainsi la distribution intermédiaire suivante :

$$P(I_n^t | s_n^t) = \frac{p_{i,s} + Scale_I}{1 + Scale_I}.$$

Introduisons maintenant l'effet de l'acuité visuelle, pour implémenter l'**HYPOTHÈSE H4**, et considérons un terme  $P(I_n^t | S_n^t G^t)$ . Nous appliquons un deuxième paramètre qui dégrade les matrices de confusion. Cette fois, la transformation a pour objectif d'augmenter linéairement la valeur de  $Scale_I$  en fonction de la distance entre la position  $n$  de la lettre et la position du regard  $g^t$ , réduisant ainsi l'information contenue dans  $\{p_{i,s}\}$ . La pente de la transformation linéaire est donnée par le paramètre  $Slope_G$ , ce qui nous donne :

$$P(I_n^t | s_n^t g^t) = \frac{p_{i,s} + Scale_I + |g^t - n|/Slope_G}{1 + Scale_I + |g^t - n|/Slope_G}.$$

Enfin, pour prendre en compte les interférences latérales et obtenir la distribution finale  $P(I_n^t | S_{1:N}^t \Delta I_n^t G^t)$ , nous devons tenir compte des interférences latérales représentées par  $\Delta I_n^t$  : lorsque  $\Delta I_n^t = 0$ , il n'y a pas de décalage et la lettre est bien à la position  $n$  ; lorsque  $\Delta I_n^t = -1$ , la lettre en position  $n - 1$  interfère avec la lettre en position  $n$  ; enfin, lorsque  $\Delta I_n^t = +1$ , la lettre en position  $n + 1$  interfère avec la lettre en position  $n$ . Ainsi, en fonction de la valeur de  $\Delta I_n^t$ , le paramètre  $\{p_{i,s}\}$  adéquat est choisi<sup>5</sup> comme suit :

$$P(I_n^t | S_{1:N}^t \Delta I_n^t G^t) = \begin{cases} P(I_n^t | s_{n-1}^t [\Delta I_n^t = -1] g^t) & \text{cas } \Delta I_n^t = -1 \\ P(I_n^t | s_n^t [\Delta I_n^t = 0] g^t) & \text{cas } \Delta I_n^t = 0 \\ P(I_n^t | s_{n+1}^t [\Delta I_n^t = +1] g^t) & \text{cas } \Delta I_n^t = +1. \end{cases}$$

---

<sup>5</sup>Dans nos simulations, la valeur de  $\Delta I_n^t$  ne sera en pratique jamais connue, et des marginalisations sur cette variable apparaîtront ; elles auront pour conséquence un mélange des reconnaissances à la position considérée et ses positions voisines, pondérées par les probabilités d'interférence  $P(\Delta I_n^t)$ .

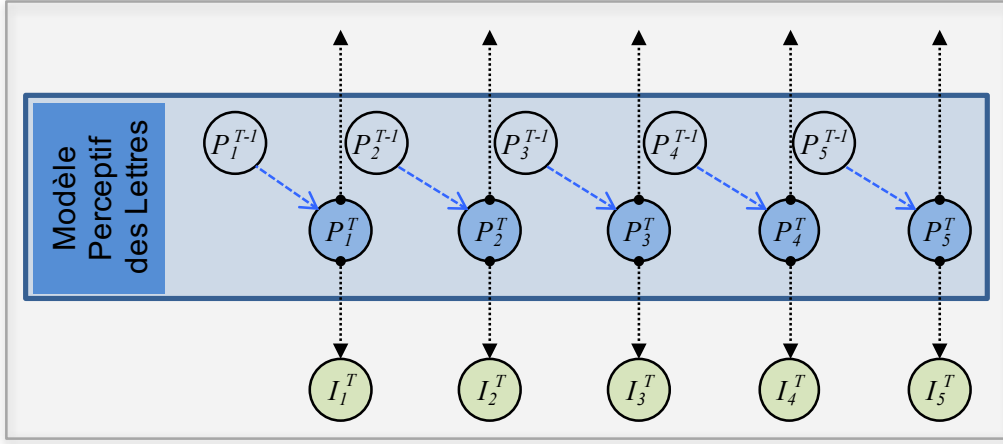


Figure 3.3 – **BRAID**, sous-modèle perceptif des lettres. Les variables  $P_n^t$  décrivent le percept en position  $n$  à l’instant  $t$ . Les flèches en pointillés bleus, entre les variables  $P_n^{T-1}$  et  $P_n^T$  représentent la dépendance temporelle entre ces variables. Les flèches en pointillés noirs reliant les variables  $P_n^t$  aux variables  $I_n^t$  rappellent le lien des variables  $P_n^t$  avec les variables du sous-modèle sensoriel des lettres (voir Section 2.1). Ce lien est implémenté de manière indirecte, par des variables de cohérence (voir Section 3.2).

## 2.2 Sous-modèle perceptif des lettres

Le sous-modèle perceptif des lettres a pour objectif de rendre compte de la génération et du maintien des représentations internes des lettres, les percepts (**HYPOTHÈSE H1**, niveau « lettre »). Pour chaque lettre du stimulus, cette construction repose sur un processus d’accumulation d’information sensorielle qui est réalisée simultanément sur toutes les positions du stimulus (**HYPOTHÈSE H2**). Ce sous-modèle est un modèle temporel, dont l’état initial traduit les connaissances a priori sur les lettres, et qui décrit essentiellement la dynamique temporelle du maintien de l’information. Le schéma complet de dépendances du sous-modèle perceptif des lettres est présenté Figure 3.3.

**Variables du modèle** Les variables qui composent le sous-modèle perceptif des lettres sont les suivantes :

- $P_{1:N}^{0:T}$  : ces variables représentent le résultat de l’accumulation de l’information sensorielle, c’est-à-dire l’identité perçue des lettres. Contrairement aux variables  $I_n^t$  qui sont des représentations « volatiles » (c’est-à-dire que leurs probabilités sont recalculées à chaque itération), ces variables transfèrent et maintiennent leur information au cours du temps. Ces variables représentant des lettres, elles sont définies sur le même domaine que les lettres du stimulus  $S_n^t$  et que les variables encodant le résultat de l’identification sensorielle  $I_n^t$ , c’est-à-dire  $\mathcal{D}_{\mathcal{L}}$ .

**Distribution conjointe et décomposition** Le sous-modèle perceptif des lettres introduit une dépendance temporelle entre les variables encodant l’information perceptive tout en conservant une indépendance spatiale, ce qui conduit à la décomposition de la distribution conjointe suivante :

$$P(P_{1:N}^{0:T}) = \left[ \prod_{n=1}^N P(P_n^0) \right] \prod_{t=1}^T \prod_{n=1}^N P(P_n^t | P_n^{t-1}) \quad (3.3)$$

**Formes paramétriques** Les formes paramétriques des termes qui apparaissent dans la décomposition de la distribution conjointe sont les suivantes :

- $\forall n, P(P_n^0)$  sont les distributions de probabilité qui caractérisent l'état initial des percepts (le prior). Nous faisons l'hypothèse qu'en l'absence prolongée d'information en provenance du stimulus, les percepts ne contiennent aucune information. Ces distributions sont donc des distributions uniformes sur l'espace des lettres  $\mathcal{D}_{\mathcal{L}}$ .
- $\forall t, n, P(P_n^t | P_n^{t-1})$  sont les distributions de probabilité qui constituent le modèle dynamique du sous-modèle, c'est-à-dire qu'elles décrivent l'évolution temporelle des connaissances sur les percepts en absence de stimulus. Ce sont des distributions de probabilité conditionnelles discrètes « quasi Dirac », dont l'essentiel de l'information transmise est portée par la même lettre qu'au pas de temps précédent et dont le reste est distribué uniformément sur les autres lettres. Cette forme traduit mathématiquement un déclin graduel de l'information en absence de stimulus ; ainsi, dans les inférences qui suivront, à chaque itération, la distribution sur  $P_n^t$  déclinera vers son état initial, avec une vitesse de déclin contrôlée par le paramètre  $leak_P$ . Mathématiquement,  $P(P_n^t | P_n^{t-1})$  est un mélange entre une distribution Dirac et une distribution uniforme, telle que :

$$P([P_n^t = l_P^t] | [P_n^{t-1} = l_P^{t-1}]) = \begin{cases} \frac{1+Leak_P}{1+|\mathcal{D}_{\mathcal{L}}| Leak_P} & \text{si } l_P^t = l_P^{t-1} \\ \frac{Leak_P}{1+|\mathcal{D}_{\mathcal{L}}| Leak_P} & \text{sinon .} \end{cases}$$

### 2.3 Sous-modèle des connaissances lexicales

Le sous-modèle de connaissances lexicales a pour objectif de rendre compte des mécanismes mémoriels permettant la reconnaissance de mots (**HYPOTHÈSE H1**, niveau « mot »). Il décrit l'ensemble des mots susceptibles d'être identifiés (désigné sous le terme de lexique dans d'autres cadres théoriques) grâce à la connaissance des lettres qui les composent et la probabilité de les identifier a priori. Il contient également, comme le sous-modèle perceptif des lettres, un modèle dynamique du maintien temporel de l'information sur les mots. Le schéma complet de dépendances du sous-modèle des connaissances lexicales est présenté Figure 3.4.

**Variables du modèle** Les variables qui composent le sous-modèle des connaissances lexicales sont les suivantes :

- $L_{1:N}^{1:T}$  : ces variables représentent les lettres qui composent les mots connus, c'est-à-dire leur orthographe. Elles sont définies sur le même domaine que les variables  $S_n^t$  et  $I_n^t$ , c'est-à-dire  $\mathcal{D}_{\mathcal{L}}$ .
- $W^{1:T}$  : ces variables représentent l'ensemble des mots connus par le modèle. Elles sont définies sur le domaine  $\mathcal{D}_W = \{w_1, w_2, \dots, w_K\}$  où chaque  $w_i$  est un mot particulier et  $K = |\mathcal{D}_W|$  est le nombre total de mots connus.

**Distribution conjointe et décomposition** Ce sous-modèle introduit une dépendance temporelle sur les variables des mots, et fait donc apparaître un terme dynamique. De plus, à chaque instant, le modèle contient une hypothèse d'indépendance entre les lettres, conditionnellement à la connaissance du mot qui contient ces lettres. En d'autres termes, on suppose que, si l'on connaît le mot, on sait donner les lettres dans chacune des positions spatiales, indépendamment

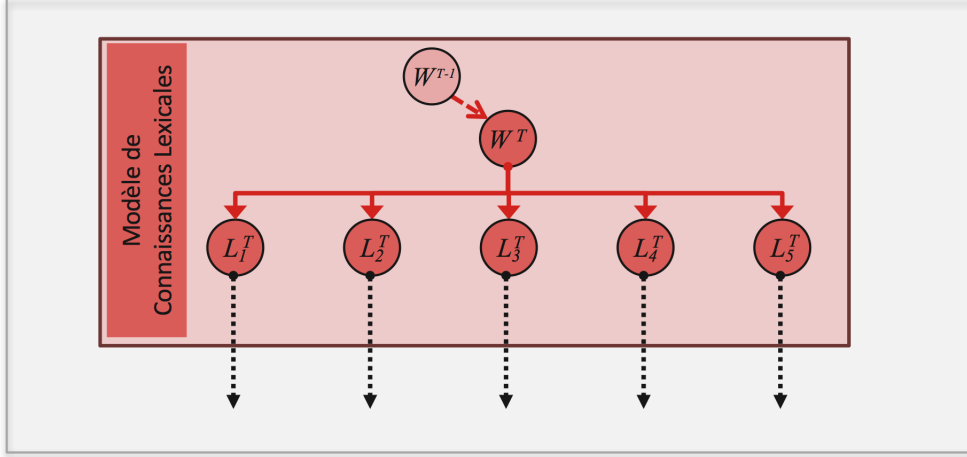


Figure 3.4 – **BRAID**, sous-modèle des connaissances lexicales. Les variables  $W^t$  décrivent la connaissance sur les mots à l’instant  $t$ . Les flèches en pointillés rouges, entre les variables  $W^{T-1}$  et  $W^T$  représentent la dépendance temporelle entre ces variables. Les variables  $L_n^t$  représentent l’information orthographique sur les mots, c’est-à-dire sur les lettres qui les composent.

les unes des autres. Ces deux hypothèses de dépendance conduisent à la décomposition de la distribution conjointe suivante :

$$P(W^{0:T} L_{1:N}^{1:T}) = P(W^0) \prod_{t=1}^T \left[ P(W^t | W^{t-1}) \prod_{n=1}^N P(L_n^t | W^t) \right] \quad (3.4)$$

**Formes paramétriques** Les formes paramétriques des termes qui apparaissent dans la décomposition de la distribution conjointe sont les suivantes.

- $P(W^0)$  est la distribution de probabilité qui caractérise l’état initial, ou de repos, des connaissances sur les mots (le prior). C’est une distribution de probabilité discrète  $P([W^0 = w_i]) = p_{w_i}$  où les paramètres  $p_{w_i}$  sont appris expérimentalement à partir de n’importe quelle base de données proposant des mots et leur fréquence d’occurrence écrite (voir Chapitre 4, Section 2.1.1).
- $\forall t, P(W^t | W^{t-1})$  sont des distributions de probabilité du modèle dynamique sur l’identité des mots. Ces distributions de probabilité conditionnelles discrètes sont, comme dans le modèle perceptif des lettres, des distributions Dirac « un peu diluées », dans lesquelles l’information sur l’identité du mot est quasiment maintenue dans le temps. Contrairement au modèle perceptif des lettres, cette connaissance n’est pas mélangée avec une distribution uniforme mais avec une distribution discrète, identique à  $P(W^0)$ , de sorte qu’en absence prolongée de stimulus, la connaissance sur les mots revienne à son état initial  $P(W^0)$ . La vitesse de ce déclin d’information est pilotée par le paramètre  $Leak_W$ . Nous avons :

$$P([W^t = w^t] | [W^{t-1} = w^{t-1}]) = \begin{cases} \frac{1+p_{w^t} Leak_W}{1+Leak_W} & \text{si } w^t = w^{t-1} \\ \frac{p_{w^t} Leak_W}{1+Leak_W} & \text{sinon .} \end{cases}$$

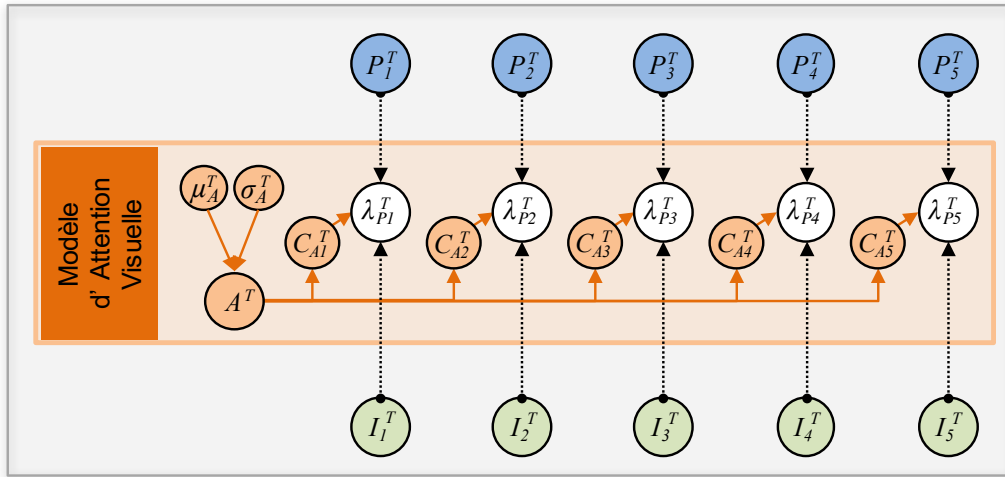


Figure 3.5 – **BRAID**, sous-modèle d’attention visuelle. Les variables  $\mu^T$  et  $\sigma^T$  décrivent les paramètres de la répartition attentionnelle  $A^T$ . Les variables  $C_{A_n}^T$ , dépendantes des  $A^T$ , contrôlent les variables  $\lambda_{P_n}^T$  qui, elles-mêmes, relient les variables  $P_n^T$  aux variables  $I_n^T$ .

- $\forall t, n, P(L_n^t | W^t)$  sont les distributions de probabilité qui caractérisent les connaissances sur les lettres d’un mot. Ce sont des distributions de probabilité conditionnelles « quasi Dirac » : étant donné un mot  $W^t = w^t$  et une position  $n$  dans ce mot, presque toute la probabilité est concentrée sur la lettre correcte  $(1 - (|\mathcal{D}_L| - 1)\epsilon)$ , et le reste est réparti uniformément sur les autres lettres ( $\mathcal{D}_L$  étant le domaine des lettres). Nous avons :

$$P([L_n^t = l] | [W^t = w^t]) = \begin{cases} 1 - (|\mathcal{D}_L| - 1)\epsilon & \text{si } l \text{ est la lettre correcte} \\ & \text{pour le mot } w^t \text{ en position } n \\ \epsilon & \text{sinon .} \end{cases}$$

## 2.4 Sous-modèle d’attention visuelle

Le sous-modèle d’attention visuelle a pour objectif d’implémenter le mécanisme de filtrage attentionnel impliqué dans la reconnaissance visuelle (**HYPOTHÈSE H5**). Il décrit la distribution de l’attention sur les lettres du stimulus, ainsi que les variables de contrôle attentionnel. Le schéma complet de dépendances du sous-modèle des connaissances lexicales est présenté Figure 3.5.

**Variables du modèle** Les variables qui composent le sous-modèle d’attention visuelle sont les suivantes :

- $\mu_A^{1:T}$  : ces variables représentent la position de l’attention sur le stimulus. Elles sont définies sur le même domaine que celui des variables  $G^{1:T}$ , c’est-à-dire sur les positions possibles,  $\mathcal{D}_{Pos}$ . L’attention peut donc se poser sur une lettre ou entre deux lettres. Le choix de modéliser à chaque pas de temps  $t$  la position du regard  $G^t$  et la position de l’attention  $\mu_A^t$  à l’aide de deux variables distinctes nous permet de déplacer l’attention visuelle indépendamment du regard dans le modèle.
- $\sigma_A^{1:T}$  : ces variables représentent la dispersion spatiale de l’attention sur le stimulus, autour de la position  $\mu_A^{1:T}$ . Leur domaine de définition est l’intervalle  $]0, 100]$ . Les deux bornes sont fixées arbitrairement, mais permettent de rendre compte de la totalité des situations envisagées dans ce manuscrit.

- $A^{1:T}$  : ces variables représentent la distribution spatiale de l'attention sur toutes les positions du stimulus. Elles sont définies sur chacune des positions du stimulus.
- $C_{A1:N}^{1:T}$  : ces variables représentent la quantité d'attention allouée à chaque position spatiale du stimulus. Ce sont des variables binaires, dont les valeurs sont notées  $\{0, 1\}$ , avec 0 indiquant qu'il n'y a pas d'attention sur cette position, et 1 indiquant que l'attention porte sur cette position. Techniquement, ces variables sont des variables de « contrôle » dont le rôle est de piloter les variables de cohérence  $\lambda_{P_n}^t$ , contrôlant ainsi la quantité d'information qui est propagée du sous-modèle sensoriel au sous-modèle perceptif des lettres.

**Distribution conjointe et décomposition** L'indépendance conditionnelle spatiale des variables de contrôle nous conduit à la décomposition de la distribution conjointe suivante :

$$P(A^{1:T} \ \mu_A^{1:T} \ \sigma_A^{1:T} \ C_{A1:N}^{1:T}) = \prod_{t=1}^T \left[ P(\mu_A^t) P(\sigma_A^t) P(A^t \mid \mu_A^t \ \sigma_A^t) \prod_{n=1}^N P(C_{A_n}^t \mid A^t) \right] \quad (3.5)$$

**Formes paramétriques** Les formes paramétriques des termes qui apparaissent dans la décomposition de la distribution conjointe sont les suivantes.

- $\forall t, P(\mu_A^t)$  et  $P(\sigma_A^t)$  sont les distributions de probabilité a priori (ou *prior*) sur les paramètres de la distribution d'attention (variable  $A^t$ ). Ce sont toutes les deux des distributions uniformes sur leur domaine respectif. Ce choix de modélisation n'a pas d'incidence pratique sur les inférences qui sont décrites dans ce manuscrit, car elles sont toutes conditionnées par la connaissance de  $\mu_A^t = \mu$ ,  $\sigma_A^t = \sigma$ , de sorte que  $P([\mu_A^t = \mu])$  et  $P([\sigma_A^t = \sigma])$  peuvent toujours être intégrées à la constante de normalisation. Les valeurs  $\mu$  et  $\sigma$  pour  $\mu_A^t$  et  $\sigma_A^t$  dépendent des conditions expérimentales et sont explicitées dans les simulations.
- $\forall t, P(A^t \mid \mu_A^t \ \sigma_A^t)$  sont les distributions de probabilité du modèle de répartition attentionnelle. Ce sont des distributions Gaussiennes de moyenne  $\mu_A^t$  et d'écart type  $\sigma_A^t$ , correctement approximées pour tenir compte de leur support borné et discret (leur domaine étant restreint aux positions possibles des lettres du stimulus (voir Figure 4.4, page 92 pour une illustration)).
- $\forall t, n, P(C_{A_n}^t \mid A^t)$  sont les distributions du modèle d'allocation attentionnelle sur les différentes positions spatiales. Ce sont des distributions de probabilité de Bernoulli pour lesquelles la probabilité que l'attention soit allouée à la position  $n$  (probabilité de succès) vaut  $P(A^t = n)$ . En d'autres termes, elles traduisent sur une variable binaire  $C_{A_n}^t$  par position la distribution attentionnelle connue sur l'ensemble des positions  $P(A^t)$ . Une conséquence directe de ce choix de modélisation est que, puisque l'attention est modélisée par une distribution de probabilité, sa « quantité totale » est 1, par la règle de normalisation. En conséquence, la quantité d'attention totale ne dépend pas de la taille du stimulus, et la distribution (au sens mathématique) de probabilité traduit directement la distribution (au sens spatial) de l'attention. Ainsi, nous avons :

$$\begin{cases} P([C_{A_n}^t = 1] \mid A^t) &= P([A^t = n]) \\ P([C_{A_n}^t = 0] \mid A^t) &= 1 - P([A^t = n]) . \end{cases}$$

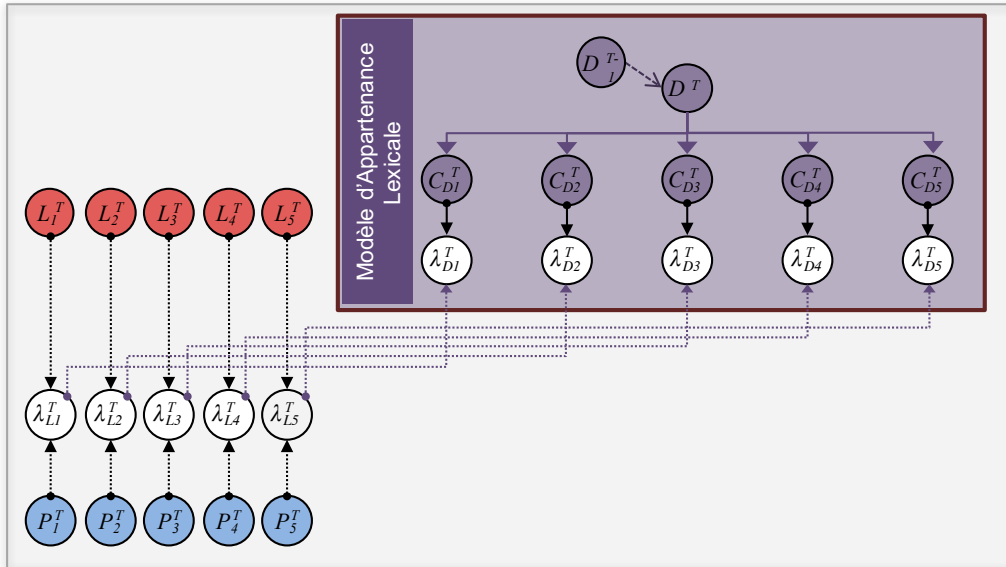


Figure 3.6 – *BRAID*, sous-modèle d'appartenance lexicale. Les variables  $D^t$  décrivent l'état de la décision lexicale à l'instant  $t$ . Les flèches en pointillés violet, entre les variables  $D^{T-1}$  et  $D^T$  représentent la dépendance temporelle entre ces variables. Les variables  $C_{D_1: N}^{1:T}$  décrivent le modèle d'erreur utilisé.

## 2.5 Sous-modèle d'appartenance lexicale

Le sous-modèle d'appartenance lexicale a pour objectif de rendre compte du mécanisme de décision répondant à la question de l'appartenance ou non du stimulus à l'ensemble des mots connus. Il décrit la valeur a priori de la décision (le modèle attend-t-il plutôt un mot ou une chaîne de caractère quelconque ?), un modèle de la dynamique de la décision en l'absence d'informations perceptives, et enfin, un « modèle d'erreur ». Ce modèle décrit la relation attendue entre le modèle de connaissance lexicale et le modèle perceptif des lettres selon que le stimulus est un mot ou pas. En effet, lorsque le stimulus est un mot du lexique, on attend une cohérence entre le sous-modèle perceptif et le sous-modèle lexical, puisqu'un mot du lexique correspond entièrement au stimulus, sans erreur. En revanche, si le stimulus n'est pas un mot du lexique, cette cohérence est en défaut sur au moins une position, et le stimulus, vis-à-vis du lexique, contient au moins une erreur. Le schéma complet de dépendances du sous-modèle d'appartenance lexicale est présenté Figure 3.6.

**Variables du modèle** Les variables qui composent le sous-modèle d'appartenance lexicale sont les suivantes :

- $D^{0:T}$  : ce sont des variables booléennes qui représentent l'appartenance au lexique. Leurs valeurs sont notées  $\{vrai, faux\}$ , *vrai* correspondant à la situation où le stimulus présenté est un mot du lexique.
- $C_{D_1: N}^{1:T}$  : ce sont des variables de « contrôle » qui pilotent l'état des variables de cohérence  $\lambda_{L_n}^t$  reliant les percepts  $P_n^t$  aux lettres des mots du lexique  $L_n^t$ . Ce sont des variables binaires et leurs valeurs sont notées  $\{0, 1\}$ .

**Distribution conjointe et décomposition** Le modèle introduit une dépendance temporelle sur les différents états de la variable d'appartenance lexicale, ce qui conduit à la décomposition



de la distribution conjointe suivante :

$$P(D^{0:T} C_{D_{1:N}}^{1:T}) = P(D_0) \prod_{t=1}^T [P(D^t | D^{t-1}) P(C_{D_{1:N}}^t | D^t)] \quad (3.6)$$

**Formes paramétriques** Les formes paramétriques des termes qui apparaissent dans la décomposition de la distribution conjointe sont les suivantes.

- $P(D^0)$  est la distribution de probabilité a priori de la variable binaire d'appartenance lexicale. Cette distribution de Bernoulli décrit l'attente que l'on a concernant la nature du stimulus qui va être présenté. Cette valeur dépend le plus souvent de l'expérience. Dans ce manuscrit, nous n'utilisons cette variable que dans la simulation de tâches expérimentales, dans lesquelles les mots et les non-mots sont équitablement répartis dans les stimuli. Ceci nous amène à faire l'hypothèse classique que les stimuli seront, dans la moitié des présentations, des mots. Mathématiquement, nous posons donc  $P([D^0 = vrai]) = 1/2$ .
- $\forall t, P(D^t | D^{t-1})$  sont des distributions de probabilité caractérisant l'évolution temporelle du modèle dynamique d'appartenance lexicale. Ce sont des distributions de Bernoulli conditionnelles de paramètre  $Leak_D$ . Ce paramètre contrôle la vitesse de perte d'information sur  $D^t$  en absence de stimulus et intervient comme suit :

$$\begin{cases} P([D^t = vrai] | [D^{t-1} = vrai]) &= 1 - Leak_D \\ P([D^t = faux] | [D^{t-1} = faux]) &= 1 - Leak_D . \end{cases}$$

En d'autres termes, ce terme dynamique est défini par la matrice :

$$\begin{pmatrix} 1 - Leak_D & Leak_D \\ Leak_D & 1 - Leak_D \end{pmatrix} .$$

- $\forall t, n, P(C_{D_{1:N}}^t | D^t)$  sont des distributions de probabilité conjointe des variables d'erreur qui définissent l'appartenance lexicale. Il y a deux cas à considérer. Considérons d'abord que le stimulus est un mot (condition  $D^t = vrai$ ) : les percepts construits à partir des informations sensorielles et les lettres constituant un mot du lexique correspondent exactement. Ainsi, les variables de cohérence les reliant indiquent cette correspondance, et valent 1 en toute position. Les variables  $C_{D_n}^t$  codent ce pattern de réponse et valent 1 partout (aucune erreur) si  $D^t = vrai$ . Considérons maintenant, au contraire, que le stimulus n'est pas un mot (condition  $D^t = faux$ ). Dans ce cas, au moins une des variables de cohérence ne sera pas en correspondance, et donc vaudra 0. Les variables  $C_{D_n}^t$ , qui codent les patterns d'erreurs, sont donc à 0 sur une position, et à 1 partout ailleurs. (Pour limiter les temps de calcul des inférences qui résultent de ce modèle, nous ne considérons que le cas où exactement une variable  $C_{D_n}^t$  est égale à 0, c'est-à-dire les cas à exactement une erreur.) Sans connaissance sur les positions probables des erreurs, nous définissons que les  $N$  patterns possibles d'erreur (un par position de l'erreur) sont équiprobables. Nous avons donc :

$$P([C_{D_{1:N}}^t = \{c_1, \dots, c_N\}] | [D^t = d]) = \begin{cases} 1 & \text{si } d = vrai \text{ et } \forall n, c_n = 1 \\ 1/N & \text{si } d = faux \text{ et } \exists! n, c_n = 0 \\ 0 & \text{sinon .} \end{cases}$$

On rappelle que  $\exists! n, c_n = 0$  se lit « il existe un unique  $n$  tel que  $c_n = 0$  ».

### 3 BRAID Bayesian word Recognition using Attention, Interference and Dynamics

Nous venons de présenter les cinq sous-modèles qui constituent la « machinerie interne » du modèle. En introduisant des connecteurs entre ces différents sous-modèles, nous obtenons le modèle final, *BRAID*. Il ne nous restera plus qu'à décrire ces connecteurs, c'est-à-dire comment ces différents sous-modèles sont agencés les uns aux autres.

#### 3.1 Le modèle BRAID

Chacun des cinq sous-modèles est décrit par une distribution conjointe et sa décomposition. Pour rassembler ces cinq sous-modèles il est nécessaire d'introduire de nouveaux modèles qui servent de connecteurs, que nous appelons des « modèles de cohérence ». Ainsi, pour lier le sous-modèle de connaissance lexicale au sous-modèle perceptif des lettres nous introduisons le modèle de cohérence  $P(\lambda_{L_n}^t | L_n^t P_n^t)$ . De même, pour lier le sous-modèle perceptif des lettres au sous-modèle sensoriel des lettres, nous ajoutons le modèle de cohérence  $P(\lambda_{P_n}^t | P_n^t I_n^t C_{A_n}^t)$ . Enfin, pour lier le sous-modèle d'appartenance lexicale à l'architecture du modèle à trois niveaux, nous ajoutons le modèle de cohérence  $P(\lambda_{D_n}^t | \lambda_{L_n}^t C_{D_n}^t)$ .

En rassemblant ces différentes parties au sein d'un même modèle, nous obtenons la décomposition de la distribution conjointe de l'Équation (3.7), qui constitue le modèle *BRAID*.

$$\begin{aligned}
 JD_{BRAID} &= P \left( \begin{array}{cccccccc} W^{0:T} & L_{1:N}^{1:T} & \lambda_{L_{1:N}}^{1:T} & P_{1:N}^{0:T} & A^{1:T} & \mu_A^{1:T} & \sigma_A^{1:T} & C_{A_{1:N}}^{1:T} & \lambda_{P_{1:N}}^{1:T} \\ G^{1:T} & S_{1:N}^{1:T} & \Delta I_{1:N}^{1:T} & I_{1:N}^{1:T} & \lambda_{D_{1:N}}^{1:T} & D^{0:T} & C_{D_{1:N}}^{1:T} & & \end{array} \right) \quad (3.7) \\
 &= \left[ \begin{array}{c} P(W^0) \\ \prod_{n=1}^N P(P_n^0) \\ P(D^0) \end{array} \right] \prod_{t=1}^T \left[ \begin{array}{c} P(W^t | W^{t-1}) \prod_{n=1}^N P(L_n^t | W^t) \\ \prod_{n=1}^N P(\lambda_{L_n}^t | L_n^t P_n^t) \\ \prod_{n=1}^N P(P_n^t | P_n^{t-1}) \\ P(A^t | \mu_A^t \sigma_A^t) P(\mu_A^t) P(\sigma_A^t) \prod_{n=1}^N P(C_{A_n}^t | A^t) \\ \prod_{n=1}^N P(\lambda_{P_n}^t | P_n^t I_n^t C_{A_n}^t) \\ P(G^t) \prod_{n=1}^N [P(S_n^t) P(\Delta I_n^t) P(I_n^t | S_{1:N}^t \Delta I_n^t G^t)] \\ \prod_{n=1}^N P(\lambda_{D_n}^t | \lambda_{L_n}^t C_{D_n}^t) \\ P(D^t | D^{t-1}) P(C_{D_{1:N}}^t | D^t) \end{array} \right]
 \end{aligned}$$

Pour finir la définition du modèle, nous devons maintenant donner une description des modèles de cohérence. Pour *BRAID*, nous utilisons deux formes distinctes des modèles de cohérence : avec et sans variable de contrôle.

### 3.2 Modèle de cohérence (sans variable de contrôle)

Ce modèle de cohérence est le modèle le plus simple. Du point de vue de l'architecture du modèle, il fait le lien entre deux sous-modèles. Du point de vue des inférences, il permet de sélectionner les sous-modèles qui sont impliqués dans le calcul (Bessière et al., 2013 ; Gilet, Diard, & Bessière, 2011 ; Laurent et al., 2017 ; Moulin-Frier, Diard, Schwartz, & Bessière, 2015 ; Moulin-Frier, Laurent, Bessière, Schwartz, & Diard, 2012 ; Patri, Diard, & Perrier, 2016), mais également de chercher des erreurs, de calculer des similarités ou d'apparier des variables. Pour comprendre intuitivement le modèle de cohérence, nous proposons ici de le définir, de décrire les endroits du modèle *BRAID* où il apparaît, et de montrer synthétiquement l'effet que ce modèle a, ensuite, dans les inférences probabilistes (qui sont détaillées au Chapitre 4).

#### 3.2.1 Définition

Si l'on considère deux variables  $A$  et  $B$ , définies sur un même domaine, le modèle de cohérence introduit une variable supplémentaire, une « variable de cohérence », habituellement notée  $\lambda$ , et telle que  $\lambda$  dépend des variables  $A$  et  $B$ . La distribution conjointe de ces trois variables se décompose alors comme suit :

$$P(\lambda \ A \ B) = P(A)P(B)P(\lambda \mid A \ B) . \quad (3.8)$$

#### 3.2.2 Forme paramétrique

La forme paramétrique  $P(\lambda \mid A \ B)$ , introduite par le modèle de cohérence est définie pour  $\lambda = 1$  :

$$P([\lambda = 1] \mid [A = a] [B = b]) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{sinon .} \end{cases} \quad (3.9)$$

#### 3.2.3 Utilisation dans le modèle *BRAID*

Ce modèle de cohérence est utilisé deux fois dans *BRAID* pour lier deux paires de sous-modèles.

- Pour lier le sous-modèle de connaissance lexicale au sous-modèle perceptif des lettres, les variables de cohérence  $\lambda_{L_{1:N}}^{1:T}$  sont liées par une relation de dépendance probabiliste aux variables  $L_{1:N}^{1:T}$  du sous-modèle de connaissance lexicale, et aux variables  $P_{1:N}^{1:T}$  du sous-modèle perceptif des lettres. Les variables  $L_{1:N}^{1:T}$  et  $P_{1:N}^{1:T}$  sont toutes définies sur le domaine  $\mathcal{D}_L$ .
- Pour lier le sous-modèle d'appartenance lexicale à l'architecture du modèle à trois niveaux, les variables de cohérence  $\lambda_{D_{1:N}}^{1:T}$  sont liées par une relation de dépendance probabiliste aux variables  $C_{D_{1:N}}^{1:T}$  du sous-modèle d'appartenance lexicale, et aux variables  $\lambda_{L_{1:N}}^{1:T}$  qui sont elles-mêmes des variables de cohérence. Les variables  $C_{D_{1:N}}^{1:T}$  et les variables de cohérence  $\lambda_{L_{1:N}}^{1:T}$  sont toutes des variables booléennes.

### 3.3 Application du modèle de cohérence

Du point de vue de la modélisation, le modèle de cohérence ne fait que lier deux sous-modèles. Cependant, suivant la manière dont sont spécifiées ses variables dans les inférences probabilistes, ses propriétés mathématiques diffèrent et il peut donc être utilisé pour réaliser diverses tâches. Nous présentons ici trois cas d'utilisation du modèle de cohérence dans *BRAID*.

### 3.3.1 Inhibition d'une portion du modèle

Le modèle de cohérence peut être utilisé pour « inhiber » certaines portions du modèle, c'est-à-dire contrôler les portions du modèle impliquées dans une inférence, et donc dans lesquelles l'information peut se propager. C'est le cas dans une tâche de reconnaissance de mot pour laquelle le sous-modèle d'appartenance lexicale n'est pas sollicité (voir Chapitre 4, Section 1.2). Pour « inhiber » ce sous-modèle, on ne décrit pas les valeurs des variables  $\lambda_{D_{1:N}}^{1:T}$  en partie droite de la question probabiliste. Dans *BRAID*, on utilise également cette propriété d'inhibition pour simuler une tâche d'identification de lettre avec ou sans l'accès aux connaissances lexicales. Cette fois, ce sont les variables  $\lambda_{L_{1:N}}^{1:T}$  qui sont spécifiées (voir Chapitre 4, Section 1.3) ou pas (voir Chapitre 4, Section 1.1) en partie droite de la question probabiliste.

### 3.3.2 Calcul de similarité entre deux variables

Que se passe-t-il lorsque ni les valeurs de  $A$ , ni de  $B$  ne sont supposées connues dans une question probabiliste ? L'inférence bayésienne, lorsque  $\lambda = 1$ , laisse alors apparaître des marginalisations sur ces deux variables, et donc une double sommation :

$$\sum_{a \in D} \sum_{b \in D} \left[ P([A = a])P([B = b])P([\lambda = 1] \mid [A = a] [B = b]) \right].$$

En utilisant la définition du terme de cohérence de l'Équation (3.9), cette double sommation se simplifie, ce qui donne :

$$\sum_{a \in D} \left[ P([A = a])P([B = a]) \right].$$

Le résultat de ce calcul peut être interprété comme un indice de similarité entre les deux distributions de probabilité  $P(A)$  et  $P(B)$ . On note cet indice  $\langle P(A), P(B) \rangle$ . Sans faire une étude détaillée de l'indice de similarité dans le cas général, nous avons identifié quelques propriétés mathématiques intéressantes dans le cadre des variables définies sur le domaine des lettres  $\mathcal{D}_L$ .

- Deux distributions dont l'essentiel de la masse de probabilité porte sur la même lettre ( $P([A = l_0]) \approx 1$  et ( $P([B = l_0]) \approx 1$ ) ont un indice de similarité fort.
- Deux distributions dont l'essentiel de la masse de probabilité porte sur deux lettres différentes ( $P([A = l_1]) \approx 1$  et ( $P([B = l_2]) \approx 1$ , avec  $l_1 \neq l_2$ ) ont un indice de similarité faible.
- La valeur de l'indice de similarité est également fonction de l'information portée par les variables. Dans le cas où aucune information n'est disponible, les distributions sont uniformes sur  $\mathcal{D}_L$  et l'indice de similarité vaut  $1/|\mathcal{D}_L|$ . A contrario, lorsque l'information est entièrement portée par une même lettre, l'indice de similarité vaut 1.

### 3.3.3 Recherche d'erreur

La dernière utilisation des modèles de cohérence que nous faisons dans *BRAID* consiste à calculer la similarité entre deux variables, sous la contrainte que les deux variables n'ont pas la

même valeur. Pour réaliser cela, on fixe la valeur de  $\lambda$  à 0. Nous calculons :

$$\begin{aligned}
& \sum_{a \in D} \sum_{b \in D} \left[ P([A = a])P([B = b])P([\lambda = 0] \mid [A = a] [B = b]) \right] \\
& \propto \sum_{a \in D} \sum_{b \in D, a \neq b} \left[ P([A = a])P([B = b]) \right] \\
& \propto \sum_{a \in D} \left[ P([A = a]) [1 - P([B = a])] \right] \\
& \propto \sum_{a \in D} \left[ P([A = a]) \right] - \sum_{a \in D} \left[ P([A = a])P([B = a]) \right] \\
& \propto 1 - \langle P(A), P(B) \rangle .
\end{aligned}$$

Dans *BRAID*, la recherche d'erreur est réalisée uniquement entre le modèle de connaissance lexicale et le modèle perceptif des lettres, dans le cadre d'une tâche de décision lexicale (voir Chapitre 4, Section 1.4). Ainsi, en identifiant  $A$  à  $L_n^t$  et  $B$  à  $P_n^t$ , on obtient que la recherche d'erreur consiste à calculer la similarité entre la distribution associée à l'évènement  $[L_n^t = l]$  et la distribution associée à l'évènement  $[P_n^t \neq l]$ . En utilisant le résultat précédent, on obtient que cela consiste à calculer  $1 - \langle P(L_n^t), P(P_n^t) \rangle$ .

### 3.4 Modèle de cohérence contrôlé

Le modèle de cohérence contrôlé enrichit le modèle précédent de sorte que la variable  $\lambda$  dépende des deux variables  $A$  et  $B$ , et d'une troisième variable  $C$ , une « variable de contrôle ». Nous avons vu précédemment que, lorsque une variable de cohérence vaut 1, les deux sous-modèles qu'elle connecte échangent de l'information. A l'inverse, si la valeur de la variable de cohérence n'est pas connue, les deux sous-modèles ne sont plus connectés. La variable de contrôle permet de passer graduellement du mode « connecté » au mode « déconnecté », et d'avoir une connaissance probabiliste sur l'état de la variable de cohérence « connecteur ».

#### 3.4.1 Définition

Si l'on considère deux variables  $A$  et  $B$ , définies sur un même domaine, et une variable de cohérence  $\lambda$  qui les relie, le modèle de cohérence contrôlé introduit une variable supplémentaire  $C$ , de domaine quelconque, appelée « variable de contrôle ». La distribution conjointe de ces quatre variables se décompose alors comme suit :

$$P(\lambda \ A \ B \ C) = P(A)P(B)P(C)P(\lambda \mid A \ B \ C) , \quad (3.10)$$

où la variable  $A$  appartient à un sous-modèle et la variable  $B$  à un autre.

#### 3.4.2 Forme paramétrique

Pour simplifier les notations, supposons binaire la variable  $C$  (de valeurs 0 ou 1). La forme paramétrique  $P(\lambda \mid A \ B \ C)$ , introduite dans le modèle de cohérence contrôlé est définie par :

$$P([\lambda = 1] \mid [A = a] [B = b] [C = c]) = \begin{cases} 1 & \text{si } c = 1 \text{ and } a = b \\ 0 & \text{si } c = 1 \text{ and } a \neq b \\ 1/|\mathcal{D}_A| & \text{si } c = 0. \end{cases}$$

### 3.4.3 Utilisation dans le modèle *BRAID*

Le modèle de cohérence contrôlé est utilisé pour connecter le sous-modèle sensoriel des lettres (variables  $I_n^t$ ), au sous-modèle perceptif des lettres (variables  $P_n^t$ ). La variable de contrôle provient du sous-modèle attentionnel (variables  $C_{A_n}^t$ ). Dans toutes les inférences que nous présentons dans ce manuscrit, la valeur des  $\lambda_{P_n^t}$  est fixée à 1. La valeur de la forme paramétrique  $P(\lambda_{P_n^t} | P_n^t I_n^t C_{A_n}^t)$  résultant de la décomposition de la conjointe dépend de la variable de contrôle : lorsque  $C_{A_n}^t = 1$ , le modèle fonctionne comme un modèle de cohérence sans contrôle et la distribution de la forme paramétrique est une distribution Dirac ; lorsque  $C_{A_n}^t = 0$ , la distribution de la forme paramétrique est une distribution uniforme sur l'espace des lettres  $\mathcal{D}_L$ . Dans le modèle *BRAID*, la forme mathématique du terme de cohérence contrôlé est donc :

$$P([\lambda_{P_n^t} = 1] | [P_n^t = l_P] [I_n^t = l_I] [C_{A_n}^t = c]) = \begin{cases} 1 & \text{si } c = 1 \text{ and } l_P = l_I \\ 0 & \text{si } c = 1 \text{ and } l_P \neq l_I \\ 1/|\mathcal{D}_L| & \text{si } c = 0. \end{cases}$$

La description de la forme paramétrique au cœur du modèle de cohérence contrôlé montre deux modes de fonctionnement pour ce type de modèle.

- Lorsque la variable de contrôle vaut 1, le modèle fonctionne comme le modèle de cohérence décrit Section 3.4. Lors du calcul des inférences, le fait que  $C_{A_n}^t = 1$  s'interprète comme le fait que la lettre en position  $n$  reçoit de l'attention. Les formes paramétriques  $P([\lambda_{P_n^t} = 1] | [P_n^t = l_P] [I_n^t = l_I] [C_{A_n}^t = 1])$  se comportent alors comme des interrupteurs qui laissent passer l'information entre le sous-modèle sensoriel et le sous-modèle perceptif : lorsque l'attention est portée sur une lettre, son traitement sensoriel est propagé dans le modèle perceptif.
- lorsque la variable de contrôle vaut 0, le modèle utilise la valeur constante  $1/|\mathcal{D}_L|$  pour toutes les combinaisons de lettres associées aux variables  $I_n^t$  et  $P_n^t$ . Lors du calcul des inférences, le fait que  $C_{A_n}^t = 0$  s'interprète comme le fait que la lettre en position  $n$  ne reçoit pas d'attention. L'interrupteur est alors « ouvert », ce qui implique qu'aucune information ne transite entre  $I_n^t$  et  $P_n^t$  (la distribution sur  $I_n^t$  est remplacée par une distribution uniforme).

## 4 Conclusion

Au chapitre précédent, en réponse aux problématiques de la reconnaissance visuelle de mots, nous avons défini un modèle conceptuel reposant sur cinq hypothèses théoriques. L'objectif de ce chapitre était de proposer une implémentation de ce modèle conceptuel, en un modèle computationnel. Nous avons donc construit un modèle probabiliste composé de cinq sous-modèles, reliés entre eux par des variables de cohérence.

Le sous-modèle sensoriel des lettres modélise le processus d'extraction de l'information du stimulus. Pour cela, nous avons résumé le processus complexe d'identification d'une lettre isolée en utilisant une matrice de confusion. Puis, nous avons pris en compte la diminution de l'acuité visuelle avec l'excentricité, permettant ainsi au modèle d'identifier les lettres dans différentes positions (**HYPOTHÈSE H4**). Ce résultat nous a permis ensuite de modéliser le cas où plusieurs lettres sont à identifier simultanément, comme dans un mot. Nous avons proposé que ces lettres soient traitées en parallèle (**HYPOTHÈSE H2**), mais que les processus d'extraction des traits

caractéristiques des lettres interfèrent avec les positions adjacentes (**HYPOTHÈSE H3**). Ce sous-modèle est de loin le plus complexe. Il est représenté en vert, dans le schéma de dépendance Figure 3.7 ainsi que dans l'Équation (3.7).

Le sous-modèle perceptif des lettres modélise la dynamique de l'accumulation d'information sur l'identité des lettres, pour chaque position du stimulus. On peut donc l'interpréter comme une « mémoire » qui a la capacité d'accumuler l'information provenant du sous-modèle sensoriel des lettres, qui, lui, est « volatile ». Il est représenté en bleu, dans le schéma de dépendance Figure 3.7 ainsi que dans l'Équation (3.7).

L'information qui transite entre ces deux sous-modèles est filtrée par le sous-modèle d'attention visuelle (**HYPOTHÈSE H5**). Ce sous-modèle remplit trois fonctions : il limite la quantité d'information totale qui transite du sous-modèle sensoriel au sous-modèle perceptif de sorte qu'elle ne dépende pas de la longueur du stimulus ; il facilite le passage de l'information dans les zones d'intérêt, en pratique autour du point de fixation attentionnel ; il ralentit le passage de l'information partout ailleurs. Il est représenté en orange, dans le schéma de dépendance Figure 3.7 ainsi que dans l'Équation (3.7).

Le sous-modèle de connaissance lexicale modélise les mécanismes mémoriels permettant la reconnaissance de mots. Il se compose d'un lexique, c'est-à-dire l'ensemble des mots connus du modèle et d'un prior basé sur la fréquence d'occurrence à l'écrit. Il décrit un mécanisme orthographique permettant de connaître l'orthographe d'un mot mémorisé ainsi qu'un modèle dynamique de maintien temporel de l'information sur les mots. Il est représenté en rouge, dans le schéma de dépendance Figure 3.7 ainsi que dans l'Équation (3.7).

Les trois sous-modèles principaux, c'est-à-dire le sous-modèle sensoriel des lettres, le sous-modèle perceptif des lettres et le sous-modèle de connaissance lexicale, forment une architecture classique à trois niveaux (**HYPOTHÈSE H1**). Outre le sous-modèle d'attention visuelle, nous ajoutons un dernier sous-modèle à cette structure, le sous-modèle d'appartenance lexicale. Il modélise la manière dont un stimulus peut être jugé comme appartenant ou non au lexique. Il est composé d'un modèle dynamique de décision et d'un modèle d'erreur. Il est représenté en violet, dans le schéma de dépendance Figure 3.7 ainsi que dans l'Équation (3.7).

Le modèle de reconnaissance visuelle de mot, le modèle *BRAID*, que nous avons présenté dans ce chapitre, implémente bien les cinq hypothèses du modèle conceptuel construit en réponse aux trois grandes problématiques actuelles du domaine. Nous avons donc rempli l'objectif, et disposons maintenant d'une implémentation mathématique complète de notre modèle conceptuel, sous la forme d'une distribution de probabilité conjointe.

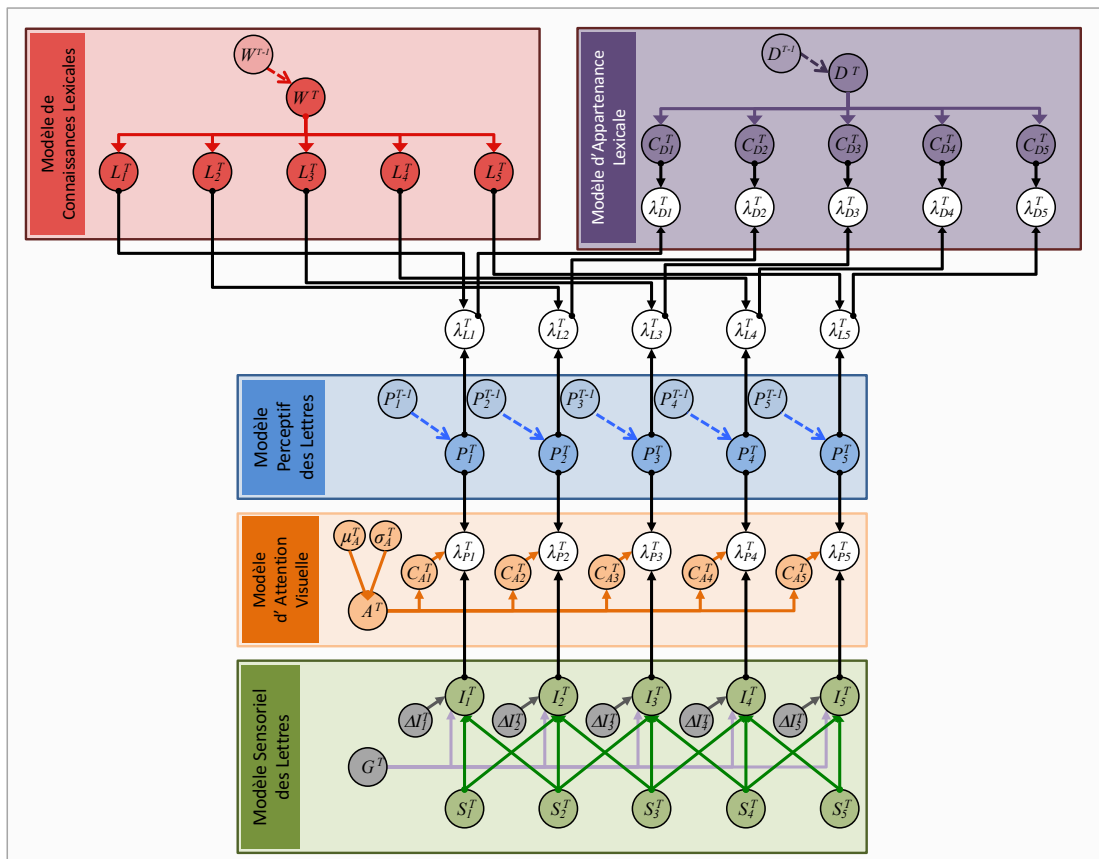


Figure 3.7 – **Schéma de dépendance complet du modèle *BRAID***. Ce schéma représente la structure de dépendance du modèle *BRAID*, au pas de temps  $t$ . En vert, on retrouve le sous-modèle sensoriel des lettres en charge du traitement du stimulus. En bleu, le sous-modèle perceptif des lettres en charge du stockage de l'information sensorielle. En orange, le sous-modèle d'attention visuelle qui régule le transfert d'information du sous-modèle sensoriel vers le sous-modèle perceptif des lettres. En rouge, le modèle de connaissance lexicale. Enfin, en violet, le sous-modèle d'appartenance lexicale.





# Chapitre 4

## Inférences probabilistes et calibrage du modèle *BRAID*

### 1 Simulation de tâches cognitives par *BRAID*

Dans cette section, nous décrivons comment nous utilisons le modèle *BRAID* pour simuler des tâches cognitives liées à la perception des lettres et des mots. Pour chaque tâche, nous procédons en deux étapes. Tout d'abord, nous formulons une question probabiliste, c'est-à-dire un terme probabiliste d'intérêt qui modélise le processus, ou une partie du processus, dont nous pensons qu'il est employé par les sujets pour résoudre cette tâche. Ensuite, nous appliquons les règles du calcul probabiliste de manière systématique et automatique, pour obtenir une expression mathématique qui calcule ce terme à partir de la distribution conjointe du modèle : c'est l'inférence bayésienne.

Afin de faciliter la compréhension de ces calculs, nous représentons la variable d'intérêt, c'est-à-dire la partie gauche de la question probabiliste, en couleur. Nous utilisons ces couleurs également pour la réponse obtenue après inférence. Ces couleurs sont celles utilisées pour décrire le modèle. Pour rappel :

- en *rouge*, les éléments du sous-modèle de connaissances lexicales,
- en *bleu*, les éléments du sous-modèle perceptif des lettres,
- en *vert*, les éléments du sous-modèle sensoriel des lettres,
- en *violet*, les éléments du sous-modèle d'appartenance lexical,
- et en *orange*, les éléments du sous-modèle d'attention visuelle.

#### 1.1 Identification d'une lettre isolée sans influence lexicale : $Q_{P_n^T}$

Nous modélisons l'identification d'une lettre isolée, dans sa forme la plus simple, par la question probabiliste de l'Équation (4.1) : étant donné le stimulus  $S_{1:N}^{1:T}$ , la répartition attentionnelle décrite par les paramètres  $\mu_A^{1:T}, \sigma_A^{1:T}$  la position du regard  $G^{1:T}$ , et que nous autorisons l'information à se propager dans le modèle du stimulus à la représentation abstraite de la lettre en position  $n$  ( $[\lambda_{P_n}^{1:T} = 1]$ ), quelle est la distribution de probabilité sur la lettre  $P_n^T$  en position  $n$  après  $T$  pas de temps ?

$$Q_{P_n^T} = P(P_n^T \mid s_{1:N}^{1:T} [\lambda_{P_n}^{1:T} = 1] \mu_A^{1:T} \sigma_A^{1:T} g^{1:T}). \quad (4.1)$$

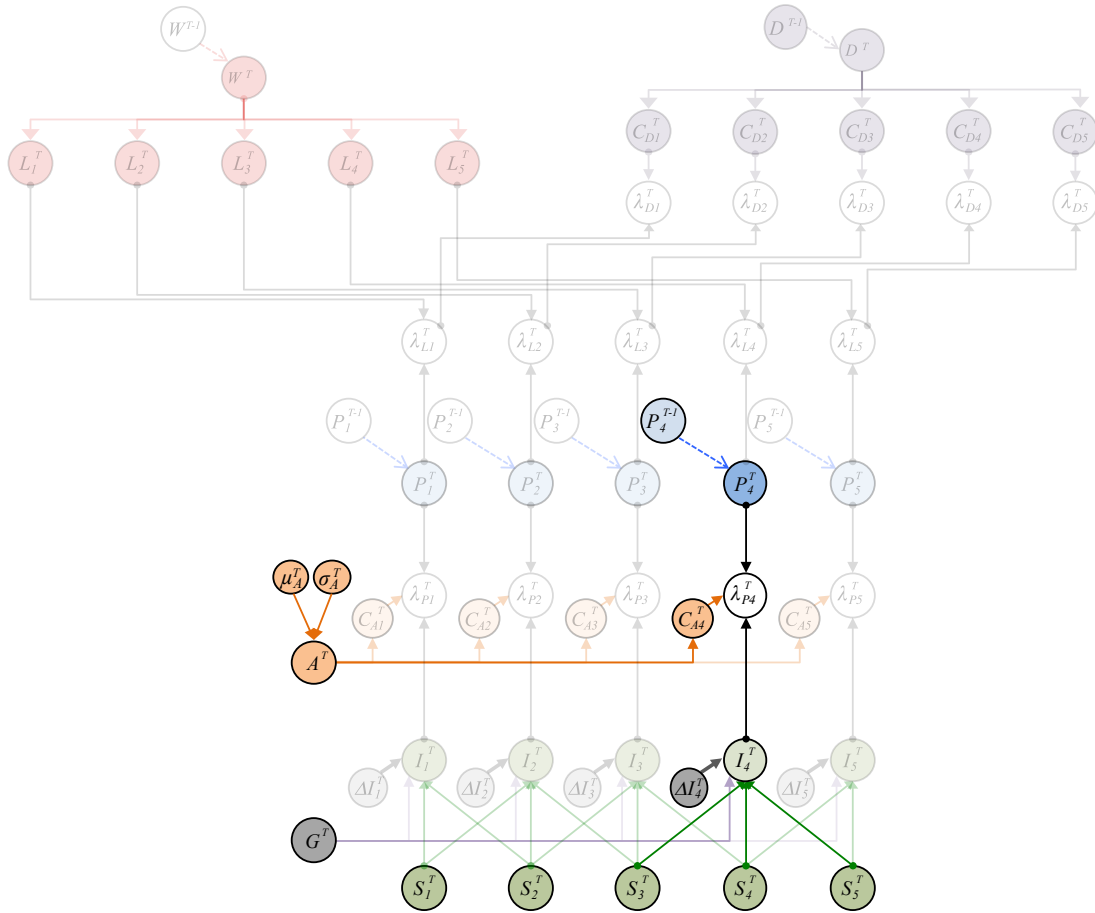


Figure 4.1 – **Tâche d'identification d'une lettre isolée sans influence lexicale.** Schéma de la partie du modèle *BRAID* utilisée pour résoudre la question de l'identification de la quatrième lettre du stimulus. Ce modèle « allégé » exclut la partie du modèle qui se situe au dessus des  $\lambda_{P_m}^t$ , pour  $m \neq 4$ , et également celles qui ne sont pas en position 4, exception faite des variables décrivant le stimulus lui-même.

**Analyse de la question** La réponse attendue à la question  $Q_{P_n}^T$  est une distribution de probabilité sur l'espace des lettres  $\mathcal{D}_{\mathcal{L}}$ . Avant de dériver la réponse en appliquant l'inférence bayésienne à la question  $Q_{P_n}^T$ , nous allons réduire le modèle, et donc la distribution conjointe, aux seules variables mises en jeu dans cette question. En effet, nous remarquons tout d'abord que certaines variables de cohérence du modèle ne sont pas spécifiées dans la question. C'est le cas des variables  $\lambda_L$  et  $\lambda_D$  pour toutes les positions, et des variables  $\lambda_{P_m}^t$  pour les positions  $m \neq n$ . Or, lorsqu'une variable de cohérence n'est pas spécifiée dans la question, la partie du modèle qui se trouve au-delà de ces variables peut être ignorée durant l'inférence (Bessière et al., 2013). On dit que les variables de cohérence fonctionnent comme des interrupteurs (Gilet et al., 2011). Nous pouvons donc réduire le modèle de départ, décrit par la distribution conjointe  $JD_{BRAID}$ , aux variables comprises entre les variables  $S_{1:N}^T$  et celles du percept  $P_n^T$ , sans perte d'informations.

Ensuite, comme nous ne considérons que la position  $n$ , les indépendances conditionnelles du modèle *BRAID* nous permettent d'ignorer les variables  $I_m^t$  et  $\Delta I_m^t$  pour toutes les valeurs de  $m$  différentes de  $n$ . Ces deux remarques nous conduisent, pour répondre à la question  $Q_{P_n}^T$ , à considérer le modèle « allégé » présenté Figure 4.1 et décrit par la distribution de probabilité

conjointe définie par l'équation 4.2 :

$$JD_{Percept_n}^T = P(P_n^{0:T} A^{1:T} \mu_A^{1:T} \sigma_A^{1:T} C_{A_n}^{1:T} \lambda_{P_n}^{1:T} S_{1:N}^{1:T} \Delta I_n^{1:T} I_n^{1:T} G^{1:T}), \quad (4.2)$$

que l'on peut décomposer ainsi :

$$JD_{Percept_n}^T = P(P_n^0) \prod_{t=1}^T \left[ \begin{array}{l} P(P_n^t | P_n^{t-1}) \\ P(A^t | \mu_A^t \sigma_A^t) P(\mu_A^t) P(\sigma_A^t) \\ P(\lambda_{P_n}^t | P_n^t I_n^t C_{A_n}^t) P(C_{A_n}^t | A^t) \\ P(G^t) \prod_{n=1}^N [P(S_n^t)] P(\Delta I_n^t) P(I_n^t | S_{1:N}^t \Delta I_n^t G^t) \end{array} \right] \quad (4.3)$$

**Calcul de la solution** Pour répondre à la question  $Q_{P_n}^T$ , décrite par l'Équation (4.1), nous commençons par marginaliser sur les variables manquantes afin de faire apparaître la distribution conjointe  $JD_{Percept}$ . Ensuite, nous réordonnons les termes afin de faire apparaître une récurrence temporelle. Nous regroupons ensemble les termes d'indice temporel  $T$  pour les placer à la fin de l'équation. Enfin, le terme  $P(P_n^0)$  étant constant, nous pouvons le supprimer, le résultat étant défini à un facteur multiplicatif près (symbole  $\propto$ ).

$$\begin{aligned} Q_{P_n}^T &= P(P_n^T | s_{1:N}^{1:T} [\lambda_{P_n}^{1:T} = 1] \mu_A^{1:T} \sigma_A^{1:T} g^{1:T}) \\ &\propto \sum_{\substack{i_n^{1:T} \\ \Delta i_n^{1:T}}} \sum_{p_n^{1:T-1}} \sum_{\substack{a^{1:T} \\ c_{A_n}^{1:T}}} JD_{Percept_n}^T \\ &\propto \sum_{p_n^{T-1}} \left[ \begin{array}{l} \sum_{\substack{i_n^{1:T-1} \\ \Delta i_n^{1:T-1}}} \sum_{p_n^{1:T-2}} \sum_{\substack{a^{1:T-1} \\ c_{A_n}^{1:T-1}}} \prod_{t=1}^{T-1} \left[ \begin{array}{l} P(p_n^t | p_n^{t-1}) \\ P(a^t | \mu_A^t \sigma_A^t) \\ P(\lambda_{P_n}^t | p_n^t i_n^t c_{A_n}^t) P(c_{A_n}^t | a^t) \\ P(g^t) \prod_{n=1}^N [P(s_n^t)] P(\Delta i_n^t) P(i_n^t | s_{1:N}^t \Delta i_n^t g^t) \end{array} \right] \\ P(P_n^T | p_n^{T-1}) \\ \sum_{\substack{i_n^T \\ \Delta i_n^T}} \sum_{\substack{a^T \\ c_{A_n}^T}} \left[ \begin{array}{l} P(a^T | \mu_A^T \sigma_A^T) \\ P(\lambda_{P_n}^T | P_n^T i_n^T c_{A_n}^T) P(c_{A_n}^T | a^T) \\ P(g^T) \prod_{n=1}^N [P(s_n^T)] P(\Delta i_n^T) P(i_n^T | s_{1:N}^T \Delta i_n^T g^T) \end{array} \right] \end{array} \right] \end{aligned}$$

Le produit temporel de la décomposition laisse apparaître la distribution conjointe au pas de temps précédent. De plus, en prenant en compte les sommations qui précèdent, on peut faire

apparaître le terme de la récurrence temporelle :

$$\begin{aligned}
Q_{P_n^T} &\propto \sum_{p_n^{T-1}} \left[ \sum_{\substack{i_n^{1:T-1} \\ \Delta i_n^{1:T-1}}} \sum_{p_n^{1:T-2}} \sum_{\substack{a^{1:T-1} \\ c_{A_n}^{1:T-1}}} JD Percept_n^{T-1} \right. \\
&\quad \left. P(P_n^T | p_n^{T-1}) \right. \\
&\quad \sum_{\substack{i_n^T \\ \Delta i_n^T}} \sum_{\substack{a^T \\ c_{A_n}^T}} \left[ \begin{array}{l} P(a^T | \mu_A^T \sigma_A^T) \\ P(\lambda_{P_n^T} | P_n^T i_n^T c_{A_n}^T) P(c_{A_n}^T | a^T) \\ P(g^T) \prod_{n=1}^N [P(s_n^T)] P(\Delta i_n^T) P(i_n^T | s_{1:N}^T \Delta i_n^T g^T) \end{array} \right] \\
&\quad \left. \right] \\
&\propto \sum_{p_n^{T-1}} \left[ \begin{array}{l} Q_{P_n^{T-1}} \\ P(P_n^T | p_n^{T-1}) \\ \sum_{\substack{i_n^T \\ \Delta i_n^T}} \sum_{\substack{a^T \\ c_{A_n}^T}} \left[ \begin{array}{l} P(a^T | \mu_A^T \sigma_A^T) \\ P(\lambda_{P_n^T} | P_n^T i_n^T c_{A_n}^T) P(c_{A_n}^T | a^T) \\ P(g^T) \prod_{n=1}^N [P(s_n^T)] P(\Delta i_n^T) P(i_n^T | s_{1:N}^T \Delta i_n^T G^T) \end{array} \right] \end{array} \right]
\end{aligned}$$

Comme dans un modèle de Markov classique (*Hidden Markov Model*, HMM ; Rabiner & Juang, 1993), ou, plus généralement, un filtre bayésien (Bessière et al., 2013), la transition temporelle (première ligne) basée sur le terme de récurrence peut être isolée du reste de l'équation. La seconde partie de l'équation est réorganisée pour faire apparaître la sommation sur les variables de contrôle attentionnel  $C_{A_n}^T$  :

$$\begin{aligned}
Q_{P_n^T} &\propto \left[ \sum_{p_n^{T-1}} [P(P_n^T | p_n^{T-1}) Q_{P_n^{T-1}}] \right. \\
&\quad \left. \sum_{\substack{i_n^T \\ \Delta i_n^T}} \sum_{\substack{a^T \\ c_{A_n}^T}} \left[ \begin{array}{l} P(a^T | \mu_A^T \sigma_A^T) \\ P(\lambda_{P_n^T} | P_n^T i_n^T c_{A_n}^T) P(c_{A_n}^T | a^T) \\ P(g^T) \prod_{n=1}^N [P(s_n^T)] P(\Delta i_n^T) P(i_n^T | s_{1:N}^T \Delta i_n^T g^T) \end{array} \right] \right] \\
&\propto \left[ \sum_{p_n^{T-1}} [P(P_n^T | p_n^{T-1}) Q_{P_n^{T-1}}] \right. \\
&\quad \left. \sum_{\substack{a^T \\ c_{A_n}^T}} \left[ \begin{array}{l} P(c_{A_n}^T | a^T) P(a^T | \mu_A^T \sigma_A^T) \\ P(\lambda_{P_n^T} | P_n^T i_n^T c_{A_n}^T) \\ \sum_{i_n^T} \left[ \begin{array}{l} P(\Delta i_n^T) \\ P(i_n^T | s_{1:N}^T \Delta i_n^T g^T) \end{array} \right] \end{array} \right] \right] \right]
\end{aligned}$$

Il ne reste plus qu'à simplifier la sommation sur les variables attentionnelles. Comme on ne considère ici que la position  $n$ , la sommation sur  $A^t$  se trouve réduite à un unique terme, et la sommation sur la variable  $C_{A_n}^T$  à deux termes ( $C_{A_n}^T = 1$  et  $C_{A_n}^T = 0$ ). On note  $\alpha_n = P([C_{A_n}^T = 1] | [A^T = n])$ . De plus, la sommation sur  $I_n^T$  se réduit à un unique terme, du fait de la présence de la variable de cohérence  $\lambda_{P_n^T}$ , propageant ainsi la valeur courante  $p_n^T$  de la variable  $P_n^T$  à la

variable  $I_n^T$ . On obtient l'écriture suivante :

$$Q_{P_n^T} \propto \left[ \begin{array}{c} \sum_{p_n^{T-1}} [P(P_n^T | p_n^{T-1}) Q_{P_n^{T-1}}(p_n^{T-1})] \\ \left[ \begin{array}{c} \alpha_n \sum_{\Delta i_n^T} \left[ \begin{array}{c} P(\Delta i_n^T) \\ P([i_n^T = p_n^T] | s_{1:N}^T \Delta i_n^T g^T) \end{array} \right] \\ + \frac{(1 - \alpha_n)}{|\mathcal{D}_{\mathcal{L}}|} \sum_{\Delta i_n^T} \left[ \begin{array}{c} P(\Delta i_n^T) \\ \sum_{i_n^T} P(i_n^T | s_{1:N}^T \Delta i_n^T g^T) \end{array} \right] \end{array} \right] \end{array} \right]$$

Finalement, en appliquant la règle de normalisation et en simplifiant les sommes, nous obtenons :

$$Q_{P_n^T} \propto \left[ \begin{array}{c} \sum_{p_n^{T-1}} [P(P_n^T | p_n^{T-1}) Q_{P_n^{T-1}}] \\ \left[ \begin{array}{c} \alpha_n \sum_{\Delta i_n^T} \left[ \begin{array}{c} P(\Delta i_n^T) \\ P([i_n^T = p_n^T] | s_{1:N}^T \Delta i_n^T g^T) \end{array} \right] + \frac{(1 - \alpha_n)}{|\mathcal{D}_{\mathcal{L}}|} \end{array} \right] \end{array} \right] \quad (4.4)$$

**Interprétation de l'équation** La réponse à la question  $Q_{P_n^T}$ , présentée par l'Équation (4.4), peut être facilement interprétée car elle est composée des deux éléments que l'on retrouve classiquement dans une inférence réalisée par un modèle de Markov caché (Rabiner & Juang, 1993). Dans le premier élément, qui est la première ligne de l'Équation (4.4), le terme de récurrence temporelle  $Q_{P_n^{T-1}}$  est tout d'abord multiplié par le terme de transition (ou changement d'état)  $P(P_n^T | P_n^{T-1})$ . Puis, le résultat de cette multiplication est sommé sur le domaine  $\mathcal{D}_{\mathcal{L}}$  (valeurs de  $P_n^{T-1}$ ), ce qui nous donne une prédiction de l'état de la distribution du percept en l'absence d'information. Dans notre cas, ce résultat intermédiaire rend compte d'une perte d'information. En effet, en imaginant que le stimulus soit ôté, l'incertitude de la distribution de probabilité du percept augmenterait graduellement à chaque pas de temps. Cette distribution finirait à son état de repos, c'est-à-dire une distribution uniforme.

Le second terme du produit de l'Équation (4.4) a un effet inverse : il apporte de l'information provenant directement du stimulus. Ce second terme est un modèle d'observation : dans un modèle de Markov caché le modèle d'observation est habituellement un terme unique de la forme  $P(O^t | S^t)$ , avec  $O^t$  les observations et  $S^t$  l'état ; dans le cas de *BRAID*, le modèle d'observation est hiérarchique et structuré. Il repose sur un mécanisme d'extraction des traits du stimulus, sensible à la distance au point de fixation et aux interférences latérales, en vert dans l'Équation (4.4), et limité par la répartition de l'attention sur le stimulus, en orange dans l'Équation (4.4).

## 1.2 Reconnaissance visuelle de mots : $Q_W^T$

La seconde tâche cognitive que nous considérons est la reconnaissance visuelle de mot, modélisée mathématiquement par la question probabiliste présentée par l'Équation (4.5), que l'on peut formuler ainsi : étant donné le stimulus  $S_{1:N}^{1:T}$ , la répartition attentionnelle décrite par les paramètres  $\mu_A^{1:T}$ ,  $\sigma_A^{1:T}$  et la position du regard  $G^{1:T}$ , et sachant que l'information peut se propager dans le modèle, du stimulus jusqu'aux mots ( $[\lambda_{L1:N}^{1:T} = 1]$ ), en passant par la représentation abstraite des lettres ( $[\lambda_{P1:N}^{1:T} = 1]$ ), quelle est la distribution de probabilité sur les mots

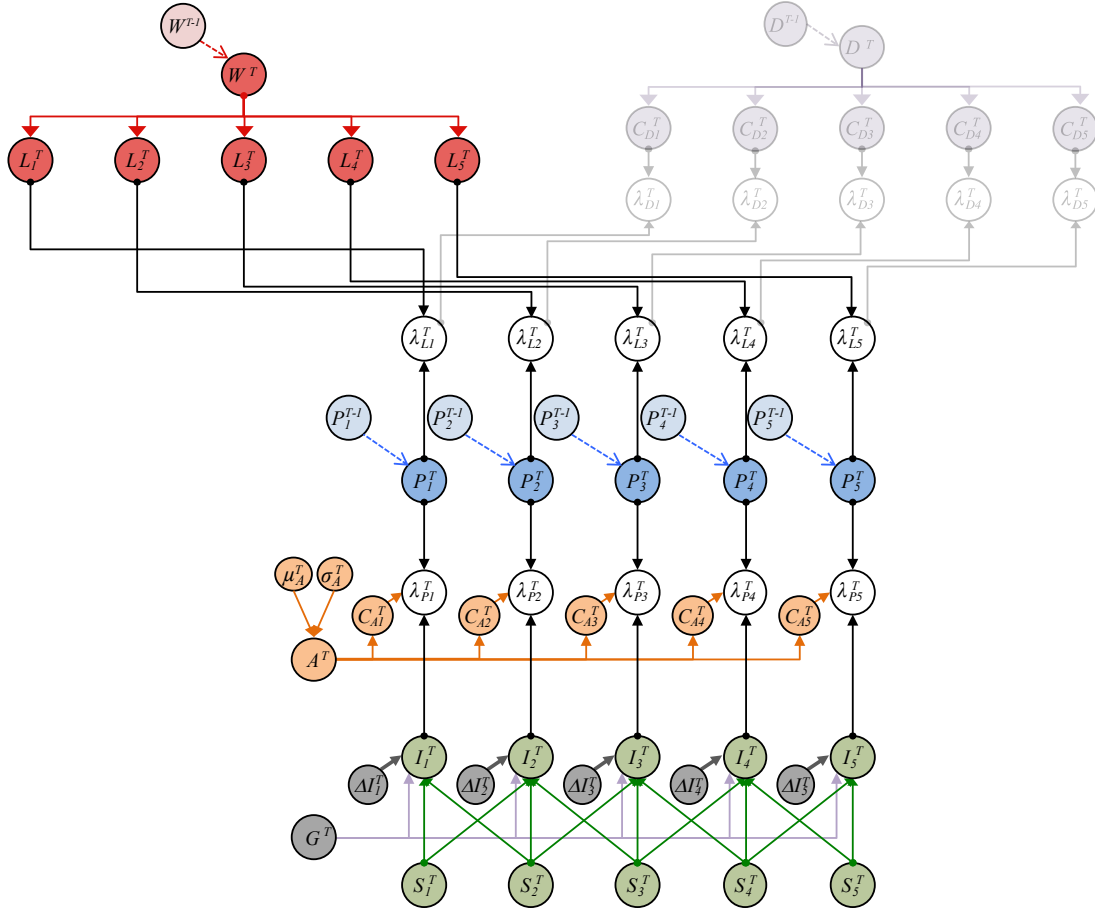


Figure 4.2 – **Tâche de reconnaissance visuelle de mot.** Schéma de la partie du modèle *BRAID* utilisée pour résoudre la question de la reconnaissance visuelle d'un mot de 5 lettres. Ce modèle « allégé » exclut la partie du modèle qui se situe « au dessus » des  $\lambda_{Dm}^t$ .

$W^T$  après  $T$  pas de temps ?

$$Q_{W^T} = P(W^T \mid s_{1:N}^{1:T} [\lambda_{L1:N}^{1:T} = 1] [\lambda_{P1:N}^{1:T} = 1] \mu_A^{1:T} \sigma_A^{1:T} g^{1:T}) \quad (4.5)$$

**Analyse de la question** La réponse attendue à la question  $Q_{W^T}$  est une distribution sur l'espace des mots  $\mathcal{D}_{\mathcal{W}}$ . Comme précédemment, nous observons que les variables de cohérence  $\lambda_D$  ne sont pas spécifiées dans la question. Donc, pour construire la réponse, nous pouvons ignorer la partie du modèle dédiée à la décision lexicale. Ceci nous conduit à considérer la partie du modèle présentée sur la Figure 4.2, c'est-à-dire la distribution conjointe décrite par :

$$JD_{Word}^T = P \left( \begin{array}{c} W^{0:T} \quad L_{1:N}^{1:T} \quad \lambda_{L1:N}^{1:T} \quad P_{1:N}^{0:T} \quad A^{1:T} \quad \mu_A^{1:T} \quad \sigma_A^{1:T} \quad C_{A1:N}^{1:T} \quad \lambda_{P1:N}^{1:T} \\ G^{1:T} \quad S_{1:N}^{1:T} \quad \Delta I_{1:N}^{1:T} \quad I_{1:N}^{1:T} \end{array} \right). \quad (4.6)$$

Le modèle défini par  $JD_{Word}^T$ , contient deux sous-modèles dynamiques parallèles : un sur l'espace des lettres,  $P(P_n^t \mid P_n^{t-1})$ , qui est contenu dans le modèle d'observation du second,  $P(W^t \mid W^{t-1})$ , défini sur l'espace des mots. Cette structure, connue sous le nom de modèle de Markov caché hiérarchique (*Hierarchical Hidden Markov Model*, HHMM ; Murphy, 2002), rend difficile l'inférence, car elle « couple » les évolutions temporelles des distributions sur les percepts et sur les mots.

Différentes techniques permettent de réaliser des inférences sur ce genre de structure tout en garantissant une prise en compte appropriée de ces dépendances temporelles parallèles. Il y a les méthodes exactes et les méthodes approchées. Par exemple, les techniques d'inférence exacte des réseaux bayésiens (*Bayesian Networks*, BN ; Pearl, 1988) ou des réseaux bayésiens dynamiques (*Dynamical Bayesian Networks*, DBN ; Dean & Kanazawa, 1989), équivaldraient à regrouper les variables  $W^t$  et  $P_n^t$  au sein d'une même variable jointe. Ce faisant, nous n'aurions plus qu'une seule variable dynamique, mais sur un espace d'état plus complexe. Ceci serait évidemment beaucoup plus gourmand en mémoire et en capacité de calcul. D'un autre côté, les techniques d'approximation procèdent en propageant l'information dans le modèle de manière itérative jusqu'à la convergence numérique.

La méthode que nous choisissons est inspirée de ces méthodes d'approximation et réalise l'inférence en une passe, mais en deux étapes. À chaque pas de temps, nous commençons par résoudre isolément la question de l'identification des lettres, en utilisant la solution de la question  $Q_{P_n^T}$  décrite par l'Équation (4.4), pour toutes les lettres du stimulus. Puis, pour répondre à la question  $Q_W^T$ , nous considérons le modèle dans lequel toute la partie basse, celle comprise entre le stimulus  $S_{1:N}^{1:T}$  et le percept  $P_{1:N}^{0:T}$ , est remplacée par le résultat de la question de l'identification de la lettre isolée  $Q_{P_n^T}$ . Ceci revient à casser temporairement la dépendance temporelle entre les variables  $P_n^t$  et  $P_n^{t-1}$ , ce qui simplifie massivement le calcul de  $Q_W^T$ .

Dans ce qui suit nous considérons que le calcul de  $Q_{P_n^T}$  est résolu (voir Équation (4.4)) et nous ne considérons que la seconde étape pour résoudre la question  $Q_W^T$ . Ceci nous amène à reformuler la distribution conjointe en adoptant la notation  $K^t = (S_{1:N}^t [\lambda_{P_{1:N}}^t = 1] \mu_A^t \sigma_A^t G^t)$ . Ceci nous permet de proposer une nouvelle définition de la distribution conjointe  $JD_{Word}^T$ , et sa décomposition :

$$\begin{aligned} JD_{Word}^T &= P(W^{0:T} \ L_{1:N}^{1:T} \ \lambda_{L_{1:N}}^{1:T} \ P_{1:N}^{1:T} \ | \ K^{1:T}) \\ &= P(W^0) \prod_{t=1}^T \left[ \begin{array}{c} P(W^t \ | \ W^{t-1}) \\ \prod_{n=1}^N \left[ \begin{array}{c} P(L_n^t \ | \ W^t) \\ P(\lambda_{L_n}^t \ | \ L_n^t \ P_n^t) \\ P(P_n^t \ | \ K^{1:T}) \end{array} \right] \end{array} \right] \end{aligned}$$

La qualité de cette approximation, c'est-à-dire la divergence entre les résultats obtenus avec cette approximation et ceux que l'on aurait obtenus avec une méthode exacte de calcul d'inférence, est une question ouverte. Du point de vue calculatoire, cette approximation concerne essentiellement la vitesse d'accumulation de l'information dans les espaces des lettres et des mots. Nous pensons que les pertes d'informations dues à l'approximation pourraient être contre-balançées par une diminution des valeurs des paramètres de déclin mémoriel  $Leak_P$  et  $Leak_W$ . Comme nous calibrons ces paramètres sur les dynamiques observées expérimentalement à partir de participants humains, il est probable que notre inférence approximative, après le calibrage du modèle, ne modifie pas la dynamique globale des inférences que nous effectuons. Cependant, du point de vue cognitif, cette approximation pose la question de la rétroaction de l'information de l'espace des mots vers l'espace des lettres. Ce point sera discuté en détail dans la conclusion du Chapitre 5.

**Calcul de la solution** Le choix de cette méthode d'approximation nous conduit également à reformuler la question  $Q_W^T$  en une forme simplifiée :  $Q_W^T = P(W^T \ | \ k^{1:T} [\lambda_{L_{1:N}}^{1:T} = 1])$ , avec  $k^t = (s_{1:N}^t [\lambda_{P_{1:N}}^t = 1] \mu_A^t \sigma_A^t g^t)$ . Pour calculer  $Q_W^T$ , nous commençons par faire apparaître la distribution conjointe  $JD_{Word}^T$ , en marginalisant sur les variables inconnues. Puis, nous



réarrangeons les termes de la sommation en faisant apparaître une récursion temporelle :

$$\begin{aligned}
Q_W^T &= P(W^T \mid k^{1:T} [\lambda_{L_{1:N}}^{1:T} = 1]) \\
&\propto \sum_{w^{0:T-1}, l_{1:N}^{1:T}, p_{1:N}^{1:T}} JD_{Word}^T \\
&\propto \left[ \sum_{w^{T-1}} \left[ \sum_{w^{0:T-2}, l_{1:N}^{1:T-1}, p_{1:N}^{1:T-1}} JD_{Word}^{T-1} \right] P(W^T \mid w^{T-1}) \right] \\
&\propto \left[ \prod_{n=1}^N \sum_{l_n^T, p_n^T} \left[ \begin{array}{l} P(l_n^T \mid W^T) \\ P([\lambda_{L_n}^T = 1] \mid l_n^T, p_n^T) \\ P(p_n^T \mid k^{1:T}) \end{array} \right] \right]
\end{aligned}$$

Comme  $\lambda_{L_n}^T = 1$ , nous pouvons réduire la double sommation sur  $l_n^T$  et  $p_n^T$  en une unique sommation sur le domaine  $\mathcal{D}_{\mathcal{L}}$  :

$$Q_W^T \propto \left[ \sum_{w^{T-1}} [Q_W^{T-1}(w^{T-1}) P(W^T \mid w^{T-1})] \right] \left[ \prod_{n=1}^N \sum_{p \in \mathcal{D}_{\mathcal{L}}} \left[ \begin{array}{l} P(L_n^T = p \mid W^T) \\ P(P_n^T = p \mid k^{1:T}) \end{array} \right] \right]$$

Finalement, en notant  $\langle \cdot, \cdot \rangle$  le produit scalaire :

$$\begin{aligned}
Q_W^T &\propto \left[ \sum_{W^{T-1}} [Q_W^{T-1}(w^{T-1}) P(W^T \mid w^{T-1})] \right] \\
&\prod_{n=1}^N \langle P(L_n^T \mid W^T), P(P_n^T \mid k^{1:T}) \rangle \\
&\propto \left[ \sum_{W^{T-1}} [Q_W^{T-1}(w^{T-1}) P(W^T \mid w^{T-1})] \right] \\
&\prod_{n=1}^N \langle P(L_n^T \mid W^T), Q_{P_n^T} \rangle
\end{aligned} \tag{4.7}$$

en remarquant que  $P(P_n^T \mid k^{1:T})$  est la question de reconnaissance des lettres isolées sans influence lexicale,  $Q_{P_n^T}$ .

**Interprétation de l'équation** Comme la réponse à la question de l'identification de la lettre  $Q_{P_n^T}$  présenté par l'Équation (4.4), la réponse à la question de l'identification du mot  $Q_W^T$  présenté par l'Équation (4.7) peut être vue comme le résultat d'une inférence dans un HMM, dont la variable cachée est cette fois la variable  $W^t$ . On retrouve une réponse sous forme du produit de deux termes. Le premier terme, première ligne de l'Équation (4.7), est formé de la somme sur l'espace des mots  $\mathcal{D}_{\mathcal{W}}$  des produits du terme de récurrence temporelle  $Q_W^{T-1}(w^{T-1})$  par le terme de changement d'état  $P(W^T \mid w^{T-1})$ . Le résultat représente la nouvelle prédiction du modèle, en l'absence d'observation provenant du niveau lettre. Ici aussi, ce calcul a pour effet un déclin mémoriel. Toutefois, ce déclin ne conduit pas à une distribution uniforme, représentant une absence de connaissance, mais à l'état au repos de la distribution des mots, représentant la fréquence a priori des mots. Quant au second terme, seconde ligne de l'Équation (4.7), nous pouvons l'interpréter comme le calcul d'une valeur de similarité. En effet,

pour chaque position, le produit scalaire  $\langle P(L_n^T | w^T), Q_{P_n^T} \rangle$  quantifie la « ressemblance » entre la distribution de la lettre perçue et la distribution de la lettre prédite pour un mot donné. Le second terme, analogue d'un modèle d'observation d'un HMM, est le produit des  $N$  valeurs de similarité, pour un stimulus de longueur  $N$ .

Le processus d'identification visuelle d'un mot, basé sur la réponse à la question  $Q_W^T$  est donc un processus itératif. À chaque pas de temps il perd un peu d'information, mais il accumule l'information apportée par la similarité entre les lettres de chaque mot considéré et celles du stimulus présenté. Cette similarité dépend de l'information extraite du stimulus : autrement dit, plus la distribution  $Q_{P_n^T}$  est concentrée sur une lettre, plus la distribution sur les mots accumulera de l'information.

### 1.3 Identification d'une lettre isolée avec influence lexicale : $Q'_{P_n^T}$

Dans la première section, nous avons vu comment la question probabiliste  $Q_{P_n^T}$  de l'identification d'une lettre d'un stimulus, sans influence lexicale, était résolue dans *BRAID* (voir Équation (4.4)). Dans cette question, les variables de cohérences  $\lambda_{L_{1:N}}^{1:T}$  n'étaient pas spécifiées, elles fonctionnaient comme des interrupteurs probabilistes « ouverts », ce qui permettait de « déconnecter » la partie des connaissances lexicales du modèle.

Nous allons maintenant considérer la même tâche cognitive, mais avec les interrupteurs probabilistes « fermés » entre les lettres et les connaissances lexicales. Nous modélisons donc l'identification d'une lettre isolée avec influence lexicale, mathématiquement, en fixant la valeur à 1 pour les variables  $\lambda_{L_{1:N}}^{1:T}$ . Nous posons la question probabiliste de l'Équation (4.8), que l'on peut formuler ainsi : étant donné le stimulus  $S_{1:N}^{1:T}$ , la répartition attentionnelle décrite par les paramètres  $\mu_A^{1:T}, \sigma_A^{1:T}$  et la position du regard  $G^{1:T}$ , et sachant que l'information peut se propager dans le modèle, du stimulus jusqu'aux connaissances lexicales ( $[\lambda_{L_{1:N}}^{1:T} = 1]$ ), en passant par la représentation abstraite des lettres ( $[\lambda_{P_{1:N}}^{1:T} = 1]$ ), quelle est la distribution de probabilité sur la lettre  $P_n^T$  en position  $n$  et au temps  $T$  ?

$$Q'_{P_n^T} = P(P_n^T | s_{1:N}^{1:T} [\lambda_{L_{1:N}}^{1:T} = 1] [\lambda_{P_{1:N}}^{1:T} = 1] \mu_A^{1:T} \sigma_A^{1:T} g^{1:T}) \quad (4.8)$$

**Analyse de la question** La réponse attendue à la question  $Q'_{P_n^T}$  est une distribution sur l'espace des lettres  $\mathcal{D}_{\mathcal{L}}$ . La partie du modèle nécessaire à la résolution de cette question est la même que celle utilisée pour résoudre la question  $Q_W^T$  (voir la Figure 4.2). Nous sommes donc à nouveau confrontés au problème des deux sous-modèles dynamiques qui évoluent en parallèle : un sur l'espace des mots,  $P(W^t | W^{t-1})$ , et un autre sur l'espace des lettres,  $P(P_n^t | P_n^{t-1})$ . La méthode que nous avons retenue consistait à calculer ces deux portions dynamiques du modèle séparément. Pour un pas de temps donné, nous calculions tout d'abord  $P(P_n^T | k^{1:T} [\lambda_{P_n}^{1:T} = 1])$ , en posant  $k^t = (s_{1:N}^t \mu_A^t \sigma_A^t g^t)$ , puis nous injectons ce résultat dans la seconde portion dynamique du modèle, ce qui nous permettait de mettre à jour  $P(W^t)$ . Cette stratégie d'approximation, qui était efficace pour contenir le coût calculatoire, privait les variables  $P_n^t$  de l'information contenue au niveau mot.

Or, pour résoudre la question  $Q'_{P_n^T}$ , nous avons besoin que l'information contenue au niveau mot vienne se combiner à l'information contenue au niveau lettre ; c'est ce qui permettra de différencier la question de l'identité d'une lettre avec influence lexicale ( $Q'_{P_n^T}$ ) et sans influence lexicale ( $Q_{P_n^T}$ ). Nous proposons la solution technique suivante : pour les  $T - 1$  premiers pas de temps, nous plaçons le modèle dans les mêmes conditions que lorsqu'il résout une tâche de reconnaissance de mot, et, au dernier pas de temps, nous injectons l'information contenue au niveau lexical dans la distribution sur les lettres.

**Calcul de la solution** Pour calculer  $Q'_{P_n}$ , nous commençons par faire apparaître la distribution conjointe  $JD_{Word}^T$ , en marginalisant sur les variables inconnues. Puis, nous réarrangeons les termes de la sommation pour faire apparaître les éléments de la récurrence temporelle au pas de temps précédent :

$$\begin{aligned} Q'_{P_n} &= P(P_n^T | k^{1:T} [\lambda_{L_{1:N}}^{1:T} = 1]) \\ &\propto \sum_{\substack{w^{0:T}, \\ l_{1:N}^{1:T}, \\ p_{1:N}^{1:T-1}, p_{m \neq n}^T}} JD_{Word}^T \\ &\propto \sum_{\substack{w^{T-1:T}, \\ l_{1:N}^T, \\ p_{m \neq n}^T}} \left[ \left[ \begin{array}{c} P(W^T | w^{T-1}) \\ \sum_{\substack{w^{0:T-2}, \\ l_{1:N}^{1:T-1}, \\ p_{1:N}^{1:T-1}}} JD_{Word}^{T-1} \end{array} \right] \left[ \begin{array}{c} \prod_{\substack{m=1, \\ m \neq n}}^N P(l_m^T | w^T) \\ P(\lambda_{L_m}^T | l_m^T p_m^T) \\ P(p_m^T | k^{1:T}) \end{array} \right] \left[ \begin{array}{c} P(l_n^T | w^T) \\ P(\lambda_{L_n}^T | l_n^T P_n^T) \\ P(P_n^T | k^{1:T}) \end{array} \right] \right] \end{aligned}$$

Nous pouvons maintenant faire apparaître  $Q_W^{T-1}(w^{T-1})$  qui correspond à la seconde sommation sur  $JD_{Word}^{T-1}$ . Puis nous réorganisons les sommations sur les différentes variables, ce qui permet de faire quelques simplifications :

$$\begin{aligned} Q'_{P_n} &\propto \sum_{w^{T-1:T}} \left[ \begin{array}{c} P(W^T | w^{T-1}) \\ Q_W^{T-1}(w^{T-1}) \end{array} \right] \prod_{\substack{m=1, \\ m \neq n}}^N \left[ \sum_{\substack{l_m^T, \\ p_m^T}} \left[ \begin{array}{c} P(l_m^T | w^T) \\ P(\lambda_{L_m}^T | l_m^T p_m^T) \\ P(p_m^T | k^{1:T}) \end{array} \right] \right] \sum_{l_n^T} \left[ \begin{array}{c} P(l_n^T | w^T) \\ P(\lambda_{L_n}^T | l_n^T P_n^T) \\ P(P_n^T | k^{1:T}) \end{array} \right] \\ &\propto \sum_{w^T} \left[ \begin{array}{c} \sum_{\substack{w^{T-1} \\ N}} [Q_W^{T-1}(w^{T-1}) P(W^T | w^{T-1})] \\ \prod_{\substack{m=1, \\ m \neq n}}^N \langle P(L_m^T | w^T), P(P_m^T | k^{1:T}) \rangle \\ \sum_{l_n^T} P(l_n^T | w^T) P(\lambda_{L_n}^T | l_n^T P_n^T) P(P_n^T | k^{1:T}) \end{array} \right] \end{aligned}$$

Les deux premières lignes correspondent au calcul de  $Q_W^T(w^T)$  mais en omettant de prendre en compte la similarité entre la valeur du percept en position  $n$  et le mot considéré. C'est ici que nous ignorons, temporairement, l'influence des connaissances lexicales sur la reconnaissance de la lettre en position  $n$ . En remarquant que  $\langle P(L_n^T | W^T), P(P_n^T | k^{1:T}) \rangle$  n'est jamais nul, on peut écrire :

$$Q'_{P_n} \propto \sum_{w^T} \left[ \begin{array}{c} Q_W^T(w^T) / \langle P(L_n^T | w^T), P(P_n^T | k^{1:T}) \rangle \\ \sum_{l_n^T} P(l_n^T | w^T) P(\lambda_{L_n}^T | l_n^T P_n^T) P(P_n^T | k^{1:T}) \end{array} \right]$$

Nous avons choisi pour les calculs de remplacer la première ligne par  $Q_W^T$ . Ce choix de multiplier le résultat par  $\langle P(L_n^T | W^T), P(P_n^T | k^{1:T}) \rangle$  revient à finir l'étape d'identification du mot avant d'effectuer la combinaison entre les informations lexicales et les informations extraites du stimulus. Cette modification apportée à la dernière itération facilite l'écriture du résultat final qui devient :

$$Q'_{P_n} \approx \sum_{w^T} \left[ \begin{array}{c} Q_W^T(w^T) \sum_{l_n^T} P(l_n^T | w^T) P(\lambda_{L_n}^T | l_n^T P_n^T) P(P_n^T | k^{1:T}) \end{array} \right] \quad (4.9)$$

**Interprétation de l'équation** La réponse à la question de l'identité de la lettre en position  $n$  avec influence lexicale est une distribution sur l'espace des lettres  $\mathcal{D}_L$ . Cette distribution est calculée en marginalisant sur l'espace des mots, ce qui signifie que tous les mots du lexique considéré contribuent à son calcul. Dans l'Équation (4.9) il apparaît le terme  $P(P_n^T | k^{1:T})$  qui contient l'information sur la  $n^{\text{ème}}$  lettre extraite du stimulus durant les  $T$  premiers pas de temps, et stockée dans le percept correspondant.

Pour comprendre intuitivement comment l'information lexicale se combine avec l'information sur les lettres, il faut considérer la distribution  $P(l_n^T | w^T)$ . Cette distribution est constante au cours du temps. Elle est presque nulle partout, sauf pour une valeur, celle de la lettre en position  $n$  du mot considéré. Par exemple, si on s'intéresse à la contribution du mot *FORK* dans l'identification de la lettre placée en seconde position dans le stimulus, nous avons  $P([L_2^T = 'o'] | [W^T = \text{"fork"}])$  qui est proche de 1 et  $P([L_2^T \neq 'o'] | [W^T = \text{"fork"}])$  qui est proche de 0. Ainsi, pour interpréter ce résultat, on peut résumer la seconde partie de l'équation contenue sous la somme sur  $L_n^T$  par la probabilité du percept, après  $T$  pas de temps, pour la  $n^{\text{ème}}$  lettre du mot considéré  $P(P_n^T | k^{1:T})$ . Dans notre exemple, la contribution des lettres est essentiellement portée par  $P([P_n^T = 'o'] | k^{1:T})$ .

La probabilité que l'identité de la  $n^{\text{ème}}$  lettre du stimulus soit la lettre  $l$  au pas de temps  $T$ , en utilisant l'information lexicale, peut donc être vue comme la somme des probabilités d'identification des mots contenant la lettre  $L$  à la  $n^{\text{ème}}$  position, multiplié par la probabilité que la  $n^{\text{ème}}$  lettre perçue soit la lettre  $l$ , sachant la forme du stimulus, la position du regard et de l'attention visuelle tout au long de ce processus.

## 1.4 Décision lexicale : $Q_D^T$

La quatrième et dernière tâche cognitive que nous considérons est la décision lexicale, c'est-à-dire la tâche qui consiste à décider si un stimulus donné est un mot connu. Cette tâche est modélisée mathématiquement par la question probabiliste présentée par l'Équation (4.10), que l'on peut formuler ainsi : étant donné le stimulus  $S_{1:N}^{1:T}$ , la répartition attentionnelle décrite par les paramètres  $\mu_A^{1:T}$ ,  $\sigma_A^{1:T}$  et la position du regard  $G^{1:T}$ , et sachant que l'information peut se propager partout dans le modèle, quelle est la probabilité que le stimulus soit un mot connu ?

$$Q_D^T = P(D^T | s_{1:N}^{1:T} [\lambda_{D_{1:N}}^{1:T} = 1] [\lambda_{P_{1:N}}^{1:T} = 1] \mu_A^{1:T} \sigma_A^{1:T} g^{1:T}) \quad (4.10)$$

**Analyse de la question** La réponse attendue à la question  $Q_{D_n}^T$  est une distribution sur un espace booléen, c'est-à-dire que les valeurs possibles sont  $\{\text{vrai}, \text{faux}\}$ .

Dans cette question, on observe que les variables de cohérence  $\lambda_{L_{1:N}}^{1:T}$  ne sont pas spécifiées. Dans les cas précédents, cela nous permettait de simplifier le modèle, en écartant les portions du modèle « au-delà » de ces interrupteurs ouverts. Contrairement à ces cas précédents, ici, les variables de cohérence non-spécifiées sont « entre » le stimulus connu et la variable cherchée par la question. Dans ce cas, on ne peut pas simplifier la structure du modèle pour en extraire un sous-modèle suffisant pour répondre à la question.

De ce fait, nous sommes amenés à travailler avec le modèle complet, tel qu'il est présenté Figure 3.7. Nous considérons donc trois sous-modèles dynamiques évoluant en parallèle. Comme pour la reconnaissance visuelle de mots  $Q_W^T$  et l'identification des lettres avec influence lexicale  $Q'_L^T$ , on retrouve le modèle dynamique sur l'espace des mots,  $P(W^T | W^{T-1})$ , celui sur l'espace des lettres,  $P(P_n^T | P_n^{T-1})$ , auxquels il faut ajouter le modèle dynamique de décision lexical,  $P(D^T | D^{T-1})$ .

Pour réaliser l'inférence bayésienne et répondre à la question  $Q_D^T$ , nous utilisons la même stratégie que précédemment. Nous résolvons, de manière isolée du reste du modèle, la récurrence

temporelle sur les variables  $P_n^T$  en répondant à la question  $Q_P^T$  pour toutes les positions  $n$  du stimulus. Puis nous remplaçons, dans le modèle complet, la récurrence temporelle sur la variable  $P_n^T$  par la réponse à la question  $Q_P^T$ . Ceci revient à « casser », dans le modèle complet, la récurrence sur les variables  $P_n^T$ . Dans un souci de clarté, nous reformulons une première fois le modèle pour faire apparaître la décomposition de la distribution conjointe  $JD_{LD}^T$ , en adoptant la notation  $K^t = (S_{1:N}^t [\lambda_{P_{1:N}}^t = 1] \mu_A^t \sigma_A^t G^t)$ , ce qui nous donne la version intermédiaire suivante :

$$\begin{aligned}
 JD_{LD}^T &= P(W^{0:T} L_{1:N}^{1:T} \lambda_{L_{1:N}}^{1:T} P_{1:N}^{0:T} \lambda_{D_{1:N}}^{1:T} C_{D_{1:N}}^{1:T} D^{0:T} | K^{1:T}) \\
 &= P(W^0) \prod_{t=1}^T \left[ \begin{array}{c} P(W^t | W^{t-1}) \\ \prod_{n=1}^N \left[ \begin{array}{c} P(L_n^t | W^t) \\ P(\lambda_{L_n}^t | L_n^t P_n^t) \\ P(P_n^t | K^{1:T}) \end{array} \right] \\ \prod_{n=1}^N P(\lambda_{D_n}^t | \lambda_{L_n}^t C_{D_n}^t) \\ P(D^t | D^{t-1}) P(C_{D_{1:N}}^t | D^t) \end{array} \right]
 \end{aligned}$$

Nous itérons une seconde fois sur le modèle pour simplifier la distribution jointe  $JD_{LD}^T$ . Notre but cette fois est d'isoler la récurrence temporelle sur les variables  $W^T$ , pour pouvoir la remplacer par un terme calculé indépendamment de la récurrence temporelle sur les variables  $D^T$ . Le candidat légitime se trouve dans le résultat de l'inférence sur l'identification des mots  $Q_W^T$ . Rappelons la manière dont est mise à jour cette distribution sur les mots dans cette question :

$$\begin{aligned}
 Q_W^T &= P(W^T | K^{1:T} [\lambda_{L_{1:N}}^{1:T} = 1]) \\
 &\propto \left[ \begin{array}{c} \sum_{W^{T-1}} [Q_W^{T-1} P(W^T | W^{T-1})] \\ \prod_{n=1}^N \langle P(L_n^T | W^T), P(P_n^T | K^{1:T}) \rangle \end{array} \right]
 \end{aligned}$$

Comme nous l'avons vu précédemment, cette équation peut-être décomposée en deux parties. La seconde partie,  $\langle P(L_n^T | W^T), P(P_n^T | K^{1:T}) \rangle$  est classiquement interprétée, dans les HMMs, comme l'étape d'observation. Elle met à jour la distribution sur les mots à partir des informations sensorielles, extraites directement du stimulus. La première partie de l'équation,  $\sum_{W^{T-1}} [Q_W^{T-1} P(W^T | W^{T-1})]$  est interprétée comme une mise à jour dynamique de la distribution sur les mots, provoquée simplement par le changement d'itération. C'est cette partie que nous utilisons pour mettre à jour la distribution sur les mots, indépendamment de la récurrence temporelle sur les variables  $D^T$ . Cela nous conduit à remplacer le terme de récurrence temporelle  $P(W^T | W^{T-1})$  dans le modèle par  $P(W^t | K^{1:t-1} \lambda_{L_{1:N}}^{1:t-1})$ , que nous définissons ainsi :

$$\begin{aligned}
 P(W^t | K^{1:t-1} \lambda_{L_{1:N}}^{1:t-1}) &= \sum_{W^{T-1}} [Q_W^{T-1} P(W^T | W^{T-1})] \\
 &= \sum_{W^{T-1}} [P(W^{T-1} | K^{1:T-1} [\lambda_{L_{1:N}}^{1:T-1} = 1]) P(W^T | W^{T-1})]
 \end{aligned}$$

Nous revenons maintenant à la décision lexicale et présentons la version finale du modèle sous la forme de la distribution conjointe  $JD_{LD}^T$ . Ce modèle ne contient plus qu'un seul terme

de récurrence temporelle  $P(D^t | D^{t-1})$ , le terme de récurrence temporelle sur les lettres ayant été remplacé par  $P(P_n^t | K^{1:t})$  et le terme de récurrence temporelle sur les mots ayant été remplacé par le terme de mise à jour dynamique  $P(W^t | K^{1:t-1} \lambda_{L_{1:N}}^{1:t-1})$ . La distribution conjointe s'écrit alors :

$$JD_{LD}^T = \prod_{t=1}^T \begin{bmatrix} P(W^t | K^{1:t-1} \lambda_{L_{1:N}}^{1:t-1}) \\ \prod_{n=1}^N \begin{bmatrix} P(L_n^t | W^t) \\ P(\lambda_{L_n}^t | L_n^t P_n^t) \\ P(P_n^t | K^{1:t}) \end{bmatrix} \\ \prod_{n=1}^N P(\lambda_{D_n}^t | \lambda_{L_n}^t C_{D_n}^t) \\ P(D^t | D^{t-1}) P(C_{D_{1:N}}^t | D^t) \end{bmatrix} \quad (4.11)$$

**Calcul de la solution** Nous pouvons maintenant calculer l'inférence pour répondre à la question probabiliste  $Q_D^T$ . Comme précédemment, nous commençons par introduire la distribution conjointe en marginalisant sur les variables manquantes de la question :

$$\begin{aligned} Q_D^T &= P(D^T | k^{1:T} [\lambda_{D_{1:N}}^{1:T} = 1] [\lambda_{P_{1:N}}^{1:T} = 1]) \\ &\propto \sum_{\substack{w^{1:T} \\ l_{1:N}^{1:T} \\ p_{1:N}^{1:T}}} \sum_{\substack{d^{0:T-1} \\ c_{D_{1:N}}^{1:T}}} JD_{LD}^T \\ &\propto \sum_{d^{T-1}} \left[ \sum_{\substack{w^{1:T-1} \\ l_{1:N}^{1:T-1} \\ p_{1:N}^{1:T-1}}} \left[ \sum_{d^{0:T-2}} \sum_{c_{D_{1:N}}^{1:T-1}} JD_{LD}^{T-1} \right] P(D^T | d^{T-1}) \right] \\ &\propto \sum_{\lambda_{L_{1:N}}^T} \left[ \sum_{\substack{w^T \\ l_{1:N}^T \\ p_{1:N}^T}} \left[ \prod_{n=1}^N \begin{bmatrix} P(w^T | k^{1:T} \lambda_{L_{1:N}}^{1:T-1}) \\ P(l_n^T | w^T) \\ P(\lambda_{L_n}^T | l_n^T p_n^T) \\ P(p_n^T | k^{1:T}) \end{bmatrix} \right] \right. \\ &\quad \left. \sum_{c_{D_{1:N}}^T} \begin{bmatrix} P(c_{D_{1:N}}^T | D^T) \\ \prod_{n=1}^N P(\lambda_{D_n}^T | \lambda_{L_n}^T c_{D_n}^T) \end{bmatrix} \right] \end{aligned}$$

Puis, nous réorganisons les termes, reconnaissons le terme de récurrence temporelle afin de

faire apparaître  $Q_D^{T-1}$ , et obtenons :

$$Q_D^T \propto \left[ \sum_{d^{T-1}} \left[ Q_D^{T-1} P(D^T | d^{T-1}) \right] \right. \\ \left. \sum_{\lambda_{L_{1:N}}^T} \left[ \sum_{w^T} \left[ \prod_{n=1}^N \sum_{\substack{l_n^T \\ p_n^T}} \left[ \begin{array}{l} P(w^T | k^{1:T-1} \lambda_{L_{1:N}}^{1:T-1}) \\ P(l_n^T | w^T) \\ P(\lambda_{L_n}^T | l_n^T p_n^T) \\ P(p_n^T | k^{1:T}) \end{array} \right] \right] \right] \right. \\ \left. \sum_{c_{D_{1:N}}^T} \left[ \begin{array}{l} P(c_{D_{1:N}}^T | D^T) \\ \prod_{n=1}^N P(\lambda_{D_n}^T | \lambda_{L_n}^T c_{D_n}^T) \end{array} \right] \right] \right]$$

Nous considérons maintenant séparément les calculs des deux réponses possibles à la question  $Q_D^T$ , en commençant par la réponse positive,  $D^T = \text{vrai}$ , et en finissant par la réponse négative,  $D^T = \text{faux}$  <sup>6</sup>. Nous avons vu précédemment que l'hypothèse d'une réponse positive correspondait à « fermer » toutes les variables de cohérence  $\lambda_{L_{1:N}}^T$ , car elles sont contrôlées par les variables  $C_{D_{1:N}}^T$ , qui sont supposées être égales à  $\{1, 1, \dots, 1\}$ . Nous pouvons donc réduire la sommation sur  $C_{D_{1:N}}^T$  à une unique valeur. Ceci se propage dans le modèle, via les variables de cohérence  $\lambda_{D_{1:N}}^T$ , qui jouent ici le rôle d'interrupteur bayésien, aux variables  $\lambda_{L_{1:N}}^T$ , réduisant la sommation sur les  $\lambda_{L_{1:N}}^T$  au terme  $\lambda_{L_{1:N}}^T = \{1, 1, \dots, 1\}$ . On obtient alors :

$$Q_{D=\text{vrai}}^T = P([D^T = \text{vrai}] | k^{1:T} [\lambda_{D_{1:N}}^{1:T} = 1] [\lambda_{P_{1:N}}^{1:T} = 1]) \\ \propto \left[ \sum_{d^{T-1}} \left[ P([D^T = \text{vrai}] | d^{T-1}) Q_{D=d^{T-1}}^{T-1} \right] \right. \\ \left. \sum_{w^T} \left[ \prod_{n=1}^N \langle P(L_n^T | w^T), Q_{P_n^T} \rangle \right] \right] \quad (4.12)$$

Pour le calcul de la réponse  $Q_{D=\text{faux}}^T$ , c'est un peu plus compliqué. On considère cette fois toutes les configurations des variables  $C_{D_{1:N}}^T$  qui contiennent exactement un 0. Cela nous donne  $N$  configurations, une par position. Il en résulte  $P(C_{D_{1:N}}^T | [D^T = \text{faux}]) = \frac{1}{N}$  quand  $C_{D_n}^T = 1 \forall n \neq i$ , et  $C_{D_i}^T = 0$ , et que  $P(C_{D_{1:N}}^T | [D^T = \text{faux}]) = 0$  sinon.

$$Q_{D=\text{faux}}^T = P([D^T = \text{faux}] | k^{1:T} [\lambda_{D_{1:N}}^{1:T} = 1] [\lambda_{P_{1:N}}^{1:T} = 1]) \\ \propto \left[ \sum_{d^{T-1}} \left[ P([D^T = \text{faux}] | d^{T-1}) Q_{D=d^{T-1}}^{T-1} \right] \right. \\ \left. \frac{1}{N} \sum_{i=1}^N \left[ \sum_{w^T} \left[ \prod_{\substack{n=1 \\ n \neq i}}^N \langle P(L_n^T | w^T), Q_{P_n^T} \rangle \right] \right. \\ \left. \langle P(L_i^T | w^T), 1 - Q_{P_i^T} \rangle \right] \right] \quad (4.13)$$

<sup>6</sup>La variable  $D^T$  étant définie sur un espace booléen, on pourrait penser qu'il suffit de calculer la probabilité de répondre  $D^T = \text{vrai}$ , pour ensuite déduire, par la règle de la normalisation, la probabilité de répondre  $D^T = \text{faux}$ . Cependant, ici, nous ne calculons la probabilité de répondre  $D^T = \text{vrai}$  qu'à une constante multiplicative près. Ceci nous impose de calculer les deux probabilités séparément puis de normaliser les résultats obtenus pour  $D^T = \text{vrai}$  et  $D^T = \text{faux}$ .

Une fois  $Q_{D=vrai}^T$  et  $Q_{D=faux}^T$  calculés, nous prenons soin de les normaliser, et obtenons donc une distribution de probabilités sur la variable  $D^T$ . Ainsi, à l'itération suivante, ce sont bien des valeurs de probabilités qui sont utilisées pour l'étape de prédiction temporelle.

**Interprétation de l'équation** Considérons pour commencer le calcul de  $Q_{D=vrai}^T$  décrit par l'Équation (4.12). On y reconnaît la structure habituelle des filtres bayésiens et des HMMs, dans laquelle il y a l'évolution d'une variable temporelle (ici  $D^T$ ) régie par, d'une part, un modèle dynamique, et d'autre part, l'accumulation d'information provenant d'un modèle d'observation. Ici ce modèle d'observation est le second terme de l'Équation (4.12) : c'est l'inférence de la reconnaissance de mots. En effet, ce second terme est très proche de l'Équation (4.7) qui calcule  $Q_W^T$  (excepté qu'on y inclut une étape de prédiction temporelle par la sommation sur  $w^T$ ). Ainsi, le modèle considère tous les mots possibles, et la probabilité que la variable de décision lexicale  $D^T$  soit vraie augmente lorsqu'il existe un mot du lexique qui est probablement reconnu ou si le stimulus a un fort voisinage orthographique dans le lexique.

Comparons maintenant avec le calcul de  $Q_{D=faux}^T$  décrit par l'Équation (4.13). La structure est identique, mais le « modèle d'observation » est différent. En effet, l'hypothèse  $D^T = faux$  contraint les variables  $C_D$ , et donc les  $\lambda_L$ , différemment. Au lieu de fixer toutes les variables de cohérence à 1, ce qui réaliserait la reconnaissance de mots, nous fixons ici une variable de cohérence à 0, et toutes les autres à 1, et cela pour toutes les positions possibles de ce 0. Considérons pour l'instant que ce  $\lambda_L = 0$  est en position  $n$ . La probabilité que la variable de décision lexicale  $D^T$  soit fautive augmente si, une autre lettre que celle présente dans le stimulus en position  $n$  est présente dans un mot du lexique.

En d'autres termes, s'il existe bien un mot du lexique qui correspond en toutes positions, y compris la position  $n$ , alors  $D^T$  est probablement vraie. En revanche, s'il existe un mot du lexique qui correspond au stimulus à condition d'imaginer une erreur en position  $n$  dans le stimulus, alors  $D^T$  est probablement fautive (et on teste cela pour toutes les positions  $n$ , les pondérant par un facteur  $1/N$  qui ne donne aucune préférence positionnelle pour l'erreur). Ainsi, on peut interpréter notre mécanisme de décision lexicale comme un mécanisme de détection d'erreur. S'il n'y a pas d'erreur dans le stimulus, un mot est reconnu, le stimulus est donc un mot et  $D^T = vrai$ . S'il y a une erreur, aucun mot ne correspond au stimulus, parce que le stimulus diffère au moins en une position de tous les mots connus : le stimulus n'est donc pas un mot et  $D^T = faux$ .

Notre modèle de décision lexicale ne considère pas toutes les erreurs possibles. En effet, l'Équation (4.13) contient une sommation de  $N$  cas possibles, et chaque cas fait l'hypothèse d'une erreur en une seule et unique position. En d'autres termes, nous cherchons les « voisins à distance 1 » des mots connus pour les comparer au stimulus. C'est une simplification qui nous permet un gain en temps de calcul, en ne considérant que  $N$  cas parmi les  $2^N - 1$  permutations possibles. L'impact de cette simplification sur la dynamique de notre modèle de décision lexicale est une question ouverte.

Pour finir, observons un parallèle entre nos modèles de reconnaissance de mots et de décision lexicale. Dans la reconnaissance de mots, les connaissances orthographiques servent de connaissances prédictives : si le mot est  $W^T = FORK$ , alors on attend un 'f' en position 1, un 'o' en position 2, etc. Les connaissances orthographiques contiennent un grand nombre d'hypothèses prédictives, une pour chaque mot connu. La reconnaissance de mot procède ensuite en mesurant la similarité entre toutes les hypothèses possibles (les mots) et le stimulus, pour trouver l'hypothèse qui y correspond le mieux. De manière analogue, la décision lexicale se base sur un modèle prédictif, qui contient deux hypothèses :  $D^T$  est soit vraie, soit fautive. Là où la reconnaissance de mot « prédit des  $L_{1:N}$  », la décision lexicale, elle, « prédit des  $\lambda_{L_{1:N}}$  » :



s’il est probable que tous les  $\lambda_{L1:N}$  sont à 1, alors le stimulus est un mot ; si au moins un  $\lambda_{L1:N}$  est probablement 0, alors le stimulus n’est pas un mot. C’est, à notre connaissance, le premier modèle probabiliste pour lequel les variables de cohérence  $\lambda_{L1:N}$  sont observées, et non pas fixées par le calcul. En d’autres termes, notre modèle de décision lexicale est hiérarchique au sens où il « observe » une autre portion du modèle : notre processus de décision lexicale est un observateur de la cohérence du processus de reconnaissance de mots.

## 2 Calibrage du modèle

Cette section est la dernière étape de la définition du modèle *BRAID*. À ce stade nous disposons d’un modèle de connaissances défini par une distribution conjointe. Nous avons décomposé cette conjointe en un produit de termes, et nous avons associé à chacun de ces termes une forme paramétrique. Il nous reste donc à définir les valeurs de ces paramètres.

Notre stratégie pour fixer ces valeurs, et ainsi calibrer le modèle, est d’aligner les performances de notre modèle sur des performances des êtres humains dans certaines tâches comportementales. C’est pourquoi, dans la section précédente, nous avons décrit mathématiquement un ensemble de questions auxquelles *BRAID* est susceptible de répondre. Puis, pour chacune de ces questions, nous avons fourni une réponse en nous appuyant sur les formes paramétriques de la distribution conjointe du modèle.

Nous avons regroupé ces paramètres en deux catégories : les paramètres dont la valeur dépend des conditions de l’expérience comportementale réalisée par le participant que nous simulons, et les paramètres qui caractérisent des processus cognitif indépendamment de la tâche dans laquelle ils sont impliqués.

### 2.1 Paramètres dépendant de l’expérience à simuler

En analysant les questions probabilistes associées aux différentes tâches simulées par *BRAID*, on remarque une certaine constance en partie droite. En effet, si l’on exclut les variables de cohérence qui servent à contrôler les flux d’information dans le modèle, on retrouve les mêmes variables dans toutes les questions :  $(S_{1:N}^t \mu_A^t \sigma_A^t G^t)$ . Le fait que ces variables se trouvent en partie droite des questions probabilistes signifie que leurs valeurs sont connues au moment de l’inférence. Nous allons décrire maintenant comment les valeurs de ces variables sont choisies, et comment les interpréter.

#### 2.1.1 Connaissances lexicales du modèle

La définition de la langue de travail du modèle *BRAID* permet de calibrer les paramètres du sous-modèle de connaissances lexicales : le domaine  $\mathcal{D}_W$  des variables  $W^{0:T}$ , les paramètres  $\theta_W$  du prior  $P(W^0)$  et les paramètres  $\theta_{LW}$  des distributions conditionnelles de la forme  $P(L_n^t | W^t)$ . Ces informations sont fournies par des bases de données lexicales (ou lexiques) que l’on trouve dans la littérature : en effet, ces lexiques fournissent une liste de mots connus, dont la longueur est le cardinal du domaine  $\mathcal{D}_W$ , des fréquences d’occurrence à l’écrit des mots qui identifient les probabilités a priori des mots dans  $P(W^0)$ , et une graphie unique pour chaque mot qui identifie les positions des pics de probabilités dans  $P(L_n^t | W^t)$  (par exemple, pour le mot *FORK*,  $\forall t, P([L_1^t = \text{‘f’}] | [W^t = \text{“fork”}]) \approx 1$ ). Dans la définition mathématique des distributions  $P(L_n^t | W^t)$ , il reste à fixer la valeur du paramètre  $\epsilon$ , qui représente le « niveau de bruit » des représentations lexicales. Cette valeur n’a pas fait l’objet d’une procédure de calibrage systématique, et a été fixée arbitrairement à  $10^{-3}$ .

Toutes les expériences comportementales décrites dans ce manuscrit (Chapitres 5 et 6) utilisent la langue anglaise. Nous disposons de deux lexiques : le *English Lexicon Project* (Balota et al., 2007), qui est composé de 40 481 mots et autant de non-mots et le *British Lexicon Project* (BLP), qui contient 28 730 mots monosyllabiques et dissyllabiques (Keuleers, Lacey, Rastle, & Brysbaert, 2012). Pour les simulations présentées dans cet ouvrage, *BRAID* est configuré avec le lexique *BLP*.

### 2.1.2 Nombre de lettres traitées simultanément

Ce paramètre, noté  $N$ , correspond à la longueur du stimulus. Il est utilisé pour limiter, par commodité, la taille du lexique effectivement utilisé dans les simulations, aux seuls mots de même longueur. Par exemple, si la tâche consiste à identifier le mot *HOME*, le lexique utilisé dans cette tâche est restreint aux mots de 4 lettres du lexique *BLP*. Ce paramètre est également utilisé pour définir le nombre de variables indicées par leur position, c'est-à-dire qu'il fixe la structure spatiale du traitement parallèle dans *BRAID*. Ces choix ont été fait essentiellement pour limiter les temps de calcul, mais il contraint la définition du voisinage orthographique utilisée dans nos expériences.

Ce choix de modélisation affecte la distribution  $P(W^0)$ . La fréquence initiale fournie par le lexique *BLP* est normalisée, non pas sur le lexique entier, mais sur le sous-ensemble des mots de la longueur d'intérêt. Ceci ne modifie pas la fréquence initiale relative des mots les uns par rapport aux autres, à l'intérieur d'une classe de longueur. Mais elle modifie la fréquence relative des mots de longueurs différentes, rendant la comparaison des simulations pour des longueurs différentes impossible. Ainsi, par exemple, l'avantage dans la reconnaissance des mots de trois lettres, qui ont une fréquence moyenne de 297,54, sur les mots de quatre lettres, qui ont une fréquence moyenne de 61,61, disparaît.

### 2.1.3 Position du regard $G^t$

La position du regard est, dans la grande majorité des cas que l'on étudiera par la suite, déterminée par le protocole expérimental de l'expérience comportementale. Par exemple, une consigne sur cette position est indiquée au sujet à l'écran, avant chaque essai, à l'aide d'un point de fixation, d'une croix ou de deux crochets limitant l'espace d'intérêt. Nous faisons donc le choix de simuler systématiquement une position du regard centrée sur le stimulus, c'est-à-dire avec  $\forall t, G^t = (N + 1)/2$ , où  $N$  est la longueur du stimulus (notez que  $G^t$  est un réel et peut avoir une valeur décimale, simulant un oeil posé entre deux lettres du stimulus). Ainsi, pour un mot de 4 lettres, le regard sera placée entre la deuxième et la troisième lettre.

### 2.1.4 Paramètres attentionnels $\mu_A^t$ et $\sigma_A^t$

La répartition de l'attention visuelle, modélisée par le terme  $P(A^t \mid \mu_A^t, \sigma_A^t)$  dans *BRAID*, suit une distribution normale (correctement approximée sur l'intervalle discret des positions de lettres) caractérisée par deux paramètres, sa position moyenne  $\mu_A^t$  et sa variance  $\sigma_A^t$ . Dans le modèle, la position moyenne de l'attention visuelle  $\mu_A^t$  est totalement indépendante de la position du regard  $G^t$ , mais en pratique, nous faisons coïncider ces deux valeurs dans la plupart des simulations. Les seules exceptions sont les simulations dans lesquelles nous manipulons la position de l'attention sans modifier la position du regard. C'est notamment le cas dans les simulations A07 et A08.

La variance de la distribution,  $\sigma_A^t$ , caractérise la répartition de l'attention  $A^t$  sur le stimulus. Lorsque l'attention est interprétée en terme de processus,  $\sigma_A^t$  définit la qualité du filtre

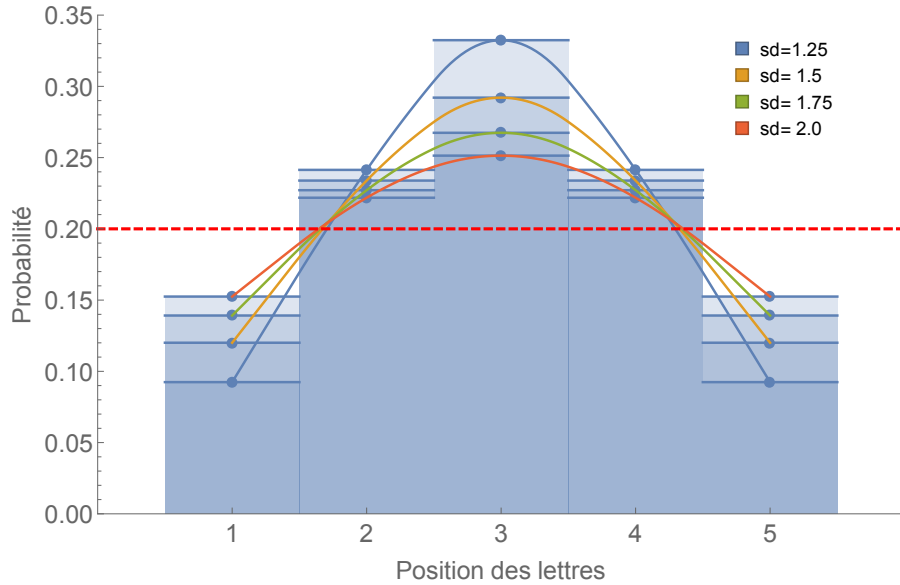


Figure 4.3 – **Distribution de l’attention sur un mot de 5 lettres, en fonction de  $\sigma_A^t$ .** Distributions de l’attention visuelle  $P(A^t | \mu_A^t, \sigma_A^t)$ , sur un mot de cinq lettres (en abscisse), pour  $\mu_A^t = 3$  et pour différentes valeurs de  $\sigma_A^t$ . Le trait en pointillés rouge matérialise une attention répartie uniformément sur les cinq lettres, c’est-à-dire  $\sigma_A^t \rightarrow \infty$ .

attentionnel qui contrôle la création des percepts  $P_n^t$ . Lorsque la valeur de  $\sigma_A^t$  est petite, la répartition de l’attention est concentrée autour de sa valeur moyenne. Dans ce cas, le filtre favorise fortement les lettres centrales au détriment des lettres périphériques. Lorsque la variance augmente, la différence de traitement entre les lettres centrales et les lettres périphériques diminue, jusqu’à devenir nulle lorsque  $\sigma_A^t \rightarrow \infty$ . Dans ce dernier cas, tout se passe comme s’il n’y avait plus de filtre. La Figure 4.3 montre l’effet de différentes valeurs de la variance  $\sigma_A^t$  pour un mot de cinq lettres.

**Objectif de calibrage** Nous faisons l’hypothèse que la valeur de la variance  $\sigma_A^t$  dépend, en conditions expérimentales habituelles chez l’humain, du nombre de lettres du stimulus. Pour calibrer les valeurs de  $\sigma_A^t$ , nous n’avons considéré que les mots de 4 à 6 lettres, car ce sont eux qui sont utilisés dans nos simulations. A notre connaissance, il n’existe pas de donnée comportementale permettant de facilement calibrer  $\sigma_A^t$  en fonction de la longueur du stimulus <sup>7</sup>.

Nous nous sommes donc fixés arbitrairement quatre contraintes : la lettre la plus excentrée ne doit pas recevoir moins de 10% de l’attention totale ; pour les mots de cinq lettres et plus, au moins trois lettres doivent recevoir plus d’attention que la valeur moyenne ; la perte de performance liée à l’excentricité doit être « graduelle » ; enfin, après 1 000 itérations, et quelle que soit la position de la lettre dans le stimulus, la probabilité d’identification ne doit pas être inférieure à 0.1. Les deux premières contraintes concernent directement la distribution de  $A^t$ , tandis que les deux suivantes concernent le résultat de l’identification. De cette manière, nous nous assurons que les stimuli sont identifiés avec un nombre d’itérations relativement homogène.

**Procédure** Pour définir la valeur de la variance  $\sigma_A^t$ , nous procédons en trois temps. Pour vérifier les deux contraintes concernant la forme de la répartition, nous avons étudié les courbes

<sup>7</sup>Dans des travaux en cours dans l’équipe, Emilie Ginestet simule des tâches de décision lexicale pour reproduire une grande base de données et calibrer précisément les paramètres attentionnels (Ginestet, 2016). Ses résultats préliminaires sont consistants avec ceux obtenus par notre méthode.

obtenues en faisant varier la longueur du stimulus entre 3 et 6 lettres, et la valeur de  $\sigma_A^t$  entre 1,25 et 2,5 par pas de 0,25. Ensuite, pour étudier la perte de performance liée à l'excentricité, nous avons construit la surface encadrant la performance d'identification des lettres de l'alphabet (sans influence lexicale, voir Section 1.1) sur les 1 000 premières itérations. Pour cela, à chaque itération, nous avons calculé la performance pour chacune des lettres de l'alphabet, présentées isolément, et nous n'avons retenu que la plus petite et la plus grande performance. Enfin, nous avons répété cette procédure, mais cette fois en utilisant la totalité des mots disponibles, par taille, dans le lexique.

**Résultats** Les valeurs retenues sont les suivantes : pour les mots de longueur 3 et 4,  $\sigma_A^t = 1,5$  ; pour les mots de longueur 5,  $\sigma_A^t = 2$  ; enfin, pour les mots de longueur 6,  $\sigma_A^t = 2,5$ . Les distributions attentionnelles correspondantes, pour les différentes longueurs de mots, sont présentées Figure 4.4. On vérifie bien que la probabilité d'identification est supérieure à 0,1 pour toutes les positions et que pour les mots de longueur 5 et 6, au moins trois positions sont au dessus de la moyenne. On peut également observer la diminution progressive de performance en fonction de l'excentricité sur la Figure 4.5 (colonne de gauche), et la surface encadrant les performances sur tous les mots du lexique, pour chacune des positions considérées, Figure 4.5 (colonne de droite). On vérifie bien que la diminution de performance est progressive avec l'excentricité et que toutes les positions ont une probabilité d'être identifiée après 1 000 itérations d'au moins 0,1. Pour les mots de 4 lettres, en position 3, on observe que la borne inférieure de la surface encadrant les performances semble avoir un comportement anormal. Cela est expliqué par les mots *FIJI* et *JUJU*, qui ont en position 3 la lettre *J* entourée de deux lettres identiques. *BRAID* identifie avec difficulté cette lettre à cause de l'interférence latérale produite par les deux lettres adjacentes identiques. Ces cas atypiques sont rares et sans conséquence pour la suite de notre analyse.

## 2.2 Paramètres sous le contrôle du modélisateur

Pour compléter le calibrage des paramètres du modèle, il reste à définir les valeurs des paramètres qui apparaissent dans les formes paramétriques de la distribution conjointe. Pour cela nous utilisons les tâches définies précédemment, pour aligner les performances de *BRAID* sur les performances humaines. L'objectif de ce calibrage est d'obtenir qu'une itération du modèle corresponde à une milliseconde de temps réel.

### 2.2.1 Matrice de confusion des lettres

Notre modèle de traitement visuel des lettres repose sur un terme  $P(I_n^t | S_{1:N}^t \Delta I_n^t G^t)$ , qui combine l'influence de la position de l'œil et des lettres adjacentes pour l'identification d'une lettre. Cette combinaison, mathématiquement, repose sur une distribution conditionnelle de la forme  $P(I_n^t | s_n^t)$ , qui représente l'identification d'une lettre isolée, et est définie par une table conditionnelle de probabilité. Cette table peut être identifiée à une matrice de confusion entre les lettres, et ses paramètres notés  $\theta_{IAS}$ .

Cette matrice de confusion synthétise un processus visuel complexe qui extrait l'information contenue dans l'image d'une lettre, pour former une représentation interne de la forme de la lettre. Si l'on se réfère au modèle LCD (voir Dehaene et al., 2005), la matrice de confusion résume le processus hiérarchique qui, à partir de l'information contenue au niveau des détecteurs des contrastes locaux (Noyau Géniculé Latéral du thalamus, *LGN*), forme des traits (aire corticale visuelle 1, *V1*), les regroupe en fragments de lettre (aire *V2*) pour finalement construire une

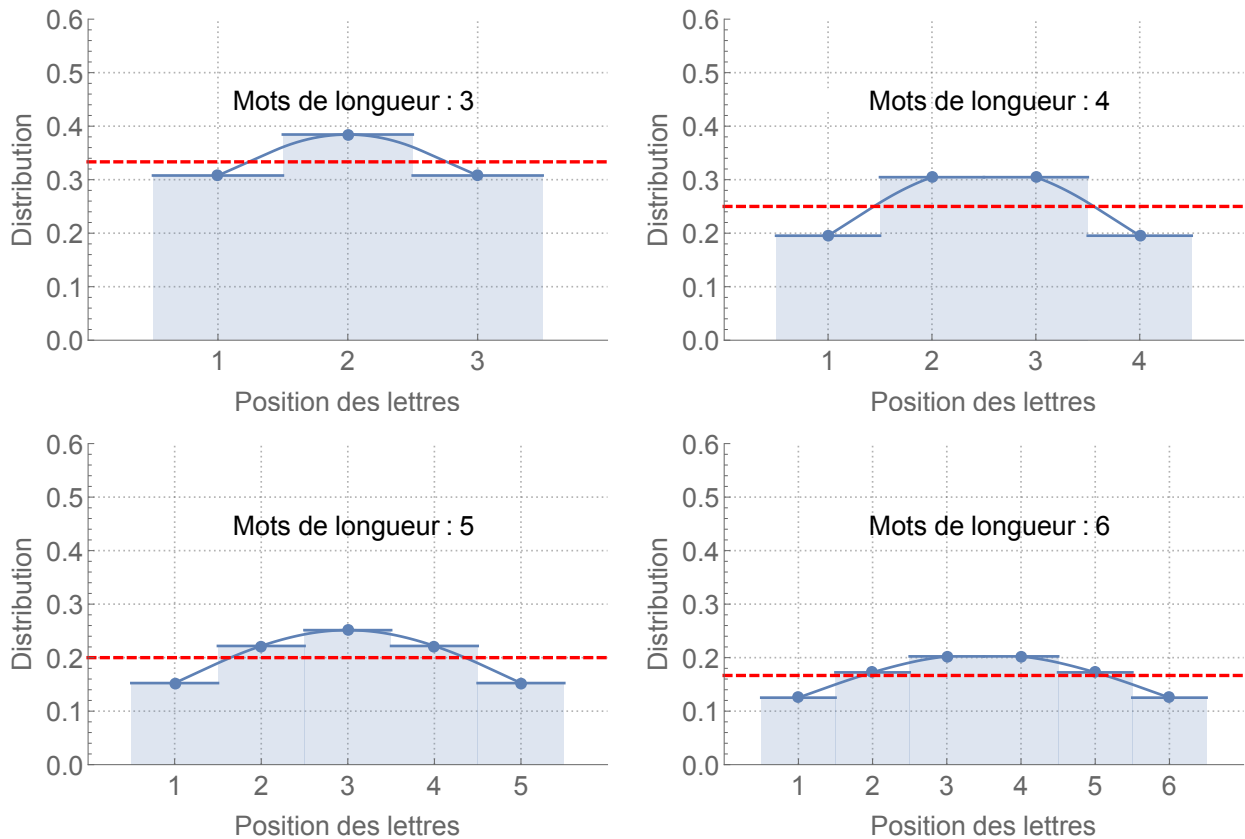


Figure 4.4 – **Distributions attentionnelles pour les valeurs de  $\sigma_A^t$  choisies.** Pour les mots de longueur 3 et 4,  $\sigma_A^t = 1,5$ , pour les mots de longueur 5,  $\sigma_A^t = 2$  et pour les mots de longueur 6,  $\sigma_A^t = 2,5$ . En pointillés rouges la distribution uniforme, qui sert de référence (absence de filtre attentionnel).

forme de lettre abstraite (aire  $V_4$ ). Le résultat de ce processus est dépendant des caractéristiques physiques du stimulus, comme la forme de la lettre présentée (fonte et taille), sa casse (majuscule ou minuscule), le contraste et la couleur d'impression, etc.

**Objectif de calibrage** La table de probabilités conditionnelles, identifiée à une matrice de confusion, doit donc rendre compte des similarités entre les lettres, pour les 26 lettres de l'alphabet.

**Procédure** Pour construire la matrice de confusion utilisée dans *BRAID*, nous avons étudié, dans la littérature, les matrices de similarité qui sont construites sur la base de résultats comportementaux. Mueller et Weidemann (2012) ont recensé 70 matrices complètes, c'est-à-dire qui fournissent les confusions pour toutes les lettres de l'alphabet. Dans cette étude, trois facteurs sont décrits comme influençant la précision de l'identification d'une lettre isolée : la perceptibilité visuelle, la similarité visuelle et les biais de réponse. La perceptibilité visuelle représente la facilité d'identification d'une lettre isolée (le contraste de l'écran ou la durée d'exposition). Nous avons exclu les matrices qui mesureraient ce facteur. Le biais, lui, représente la tendance à répondre a priori, en l'absence de stimulus. Ce facteur ne nous intéresse pas, mais il s'avère difficile à dissocier expérimentalement de la similarité des lettres. Nous avons donc évalué certaines des matrices utilisant ces deux facteurs. Dans leur étude, Mueller et Weidemann (2012) identifient trois matrices qui rendent compte de la similarité et du biais de réponses :  $Matrice_A$

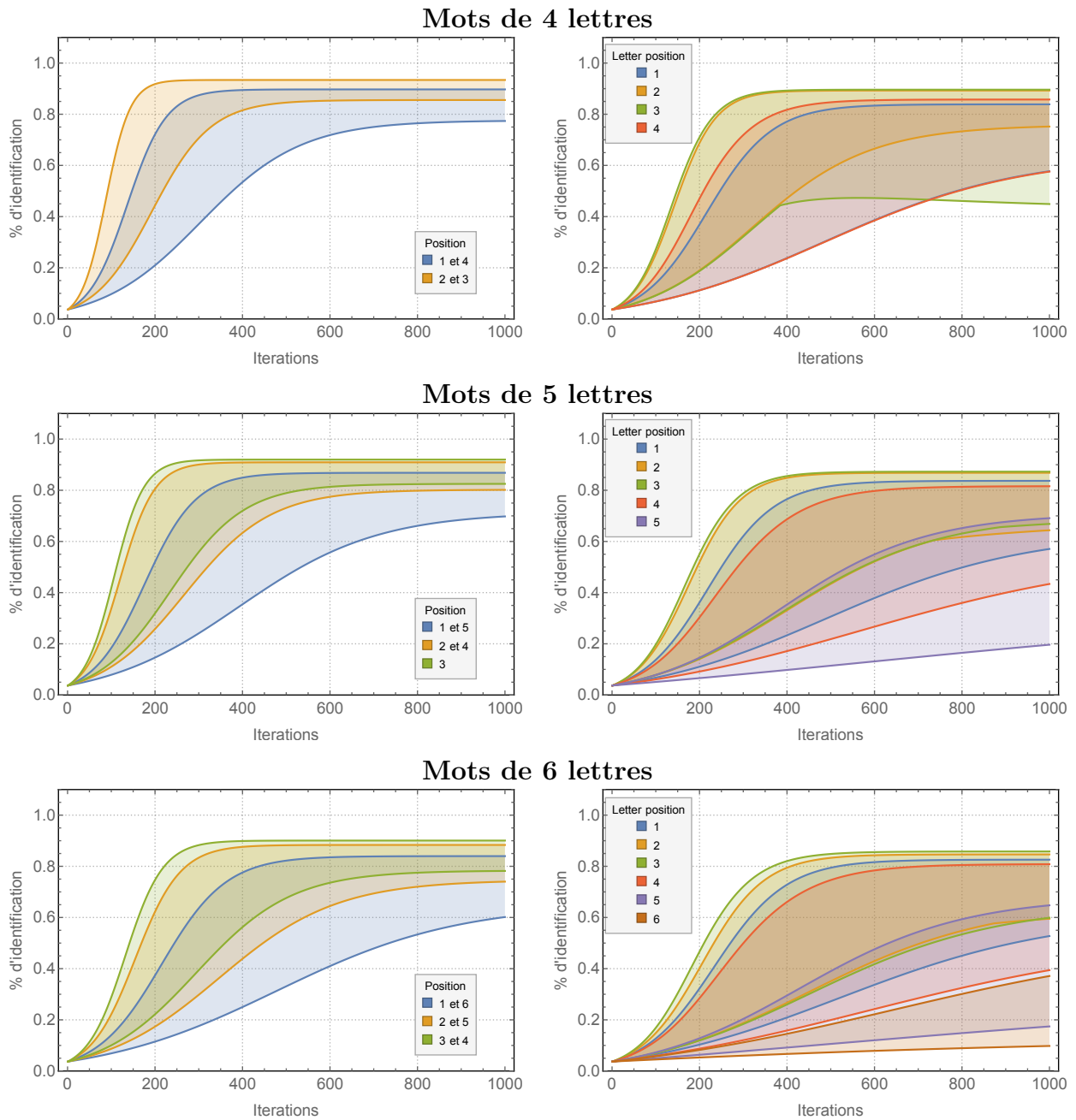


Figure 4.5 – Performances d'identification de lettre, par position, pour les valeurs de  $\sigma_A^t$  choisies. Colonne de gauche : performances d'identification des lettres de l'alphabet, sans influence lexicale, présentées isolément. Colonne de droite : performances d'identification des lettres, présentées en contexte de mot du lexique. Chaque ligne correspond à une taille de mot. Sur chaque graphique, on présente pour chaque itération et chaque position la plus petite et la plus grande probabilité d'identification, ce qui nous donne une surface colorée encadrant les performances d'identification.

(Townsend, 1971b), *Matrice<sub>B</sub>* (Townsend, 1971a) et *Matrice<sub>C</sub>* (Gilmore, Hersh, Caramazza, & Griffin, 1979). Nous avons ajouté la matrice du modèle VA, *Matrice<sub>D</sub>* (Paap et al., 1982) et la matrice « van der Heijden », *Matrice<sub>E</sub>* (van der Heijden, Malhas, & van den Roovart, 1984).

**Résultat** Le choix de la matrice de confusion est essentiellement lié à la dynamique de notre modèle qui impose certaines contraintes de convergence. Nous avons identifié, par expériences, qu’une lettre ne devait pas avoir une probabilité inférieure à 0,3, ni un compétiteur dont la probabilité serait supérieure à la moitié de la probabilité de la lettre elle-même. La matrice la plus consistante avec ces critères est la *Matrice<sub>B</sub>* (Townsend, 1971a). Ce résultat est confirmé par les conclusions de l’étude de (van der Heijden et al., 1984) qui exclut la *Matrice<sub>C</sub>*, et recommande les matrices de Townsend. Cependant, nous avons légèrement modifié la *Matrice<sub>B</sub>*, et augmenté la probabilité de reconnaissance correcte des lettres *G*, *Q* et *T* au détriment de leurs compétiteurs respectifs, *O*, *O* et *I* d’une valeur de 0,006, 0,0029 et 0,006, afin de garantir des temps de convergence similaires aux autres lettres. La matrice *Matrice<sub>B</sub>* (Townsend, 1971a) et sa version modifiée sont présentées Figure 4.6.

### 2.2.2 Paramètres temporels de l’identification des lettres *Leak<sub>P</sub>* et *Scale<sub>I</sub>*

L’information contenue dans la matrice de confusion est le résultat d’une durée d’exposition au stimulus de l’ordre de 50 ms. Utiliser cette matrice telle quelle ferait converger le modèle vers une identification parfaite de lettre en deux ou trois itérations. Ceci a précédemment été remarqué dans le contexte (Pelli et al., 2003). Le rôle du paramètre *Scale<sub>I</sub>*, qui apparaît dans la définition mathématique du terme  $P(I_n^t | S_{1:N}^t \Delta I_n^t G^t)$ , est de limiter l’information qui alimente, à chaque itération, la variable  $I_n^t$ . Ainsi, ce paramètre fait une « mise à l’échelle temporelle » de la matrice de confusion. Un autre paramètre a aussi un rôle sur l’échelle temporelle, c’est le paramètre *Leak<sub>P</sub>*, qui contrôle la vitesse de fuite de l’information contenue dans les variables  $P_n^t$ , dans le modèle dynamique  $P(P_n^t | P_n^{t-1})$ . Nous les calibrons donc conjointement.

**Objectif de calibrage** Pour calibrer ces deux paramètres, il nous faut déterminer un temps de référence correspondant au laps de temps moyen entre la présentation du stimulus et son identification. Dans une étude de magnéto-encéphalographie (MEG), Tarkiainen, Helenius, Hansen, Cornelissen, et Salmelin (1999) identifient le motif de réponse 180 ms après la présentation du stimulus, dans la région occipito-temporale du cortex. Dans la même région corticale, une seconde étude en MEG (Pylkkänen & Marantz, 2003) localise le pic N170, entre 150 ms et 200 ms. Plus récemment, dans une étude d’électro-encéphalographie (EEG), Madec, Rey, Dufau, Klein, et Grainger (2012) trouvent trois pics d’activité sur les électrodes occipitales, à 100 ms, 170 ms et 225 ms. Ils associent le premier pic à l’extraction des traits visuels de bas niveau, le deuxième pic à l’identification des lettres et le troisième pic à l’identification consciente de la lettre présentée. Cela suggère un intervalle « réaliste » pour l’identification d’une lettre isolée allant de 150 ms à 200 ms. Pour calibrer *Leak<sub>P</sub>* et *Scale<sub>I</sub>*, nous choisissons comme temps de référence la valeur basse de l’intervalle, afin de limiter les temps de calcul des simulations, soit 150 ms.

**Procédure** Nous évaluons le modèle sur une tâche d’identification de lettre, en faisant varier le paramètre de déclin mémoriel, *Leak<sub>P</sub>*, sur l’intervalle  $[0, 0,001]$ , et le paramètre de mise à l’échelle de l’information entrante, *Scale<sub>I</sub>* sur l’intervalle  $[0, 10]$ , avec 20 valeurs régulièrement réparties sur chaque dimension. Pour réaliser cette tâche, nous considérons que seule une partie de la capacité attentionnelle est nécessaire aux participants, et nous la fixons arbitrairement à  $P(A^t) = 0,8$ , ce qui correspond à une attention distribuée sur trois positions, dont la moyenne

Matrice de Townsend (1971,b)

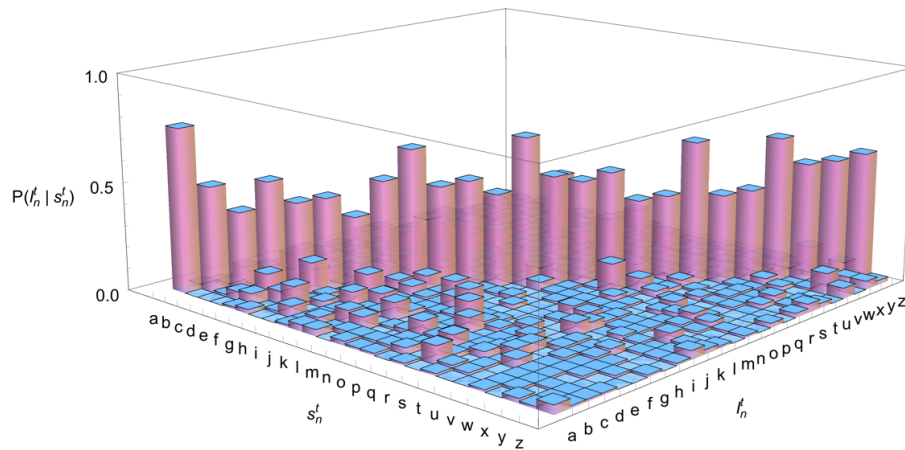
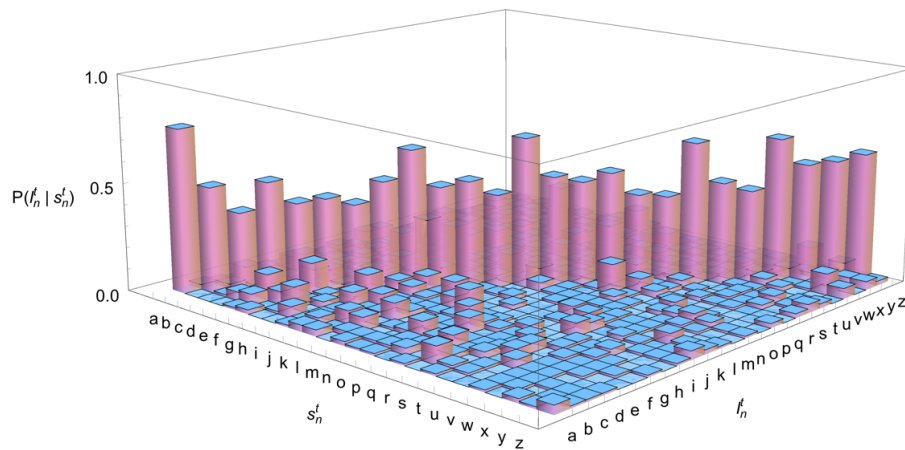
Matrice de *BRAID*

Figure 4.6 – **Matrices de confusion du modèle *BRAID***. en haut, la matrice de confusion de l'étude de Townsend (1971a), en bas, la matrice utilisée par *BRAID*. On peut remarquer que dans cette dernière, les probabilités des lettres *G*, *O* et *T*, sur la diagonale, sont légèrement supérieures.

serait la position centrale, et dont la variance serait 0,5 (soit 10% de l'attention répartie de part et d'autre de la lettre). Pour chaque couple de valeurs  $Leak_P$  et  $Scale_I$ , nous simulons la reconnaissance de lettres sur toutes les lettres de l'alphabet, et obtenons la probabilité moyenne d'identification.

**Résultats** Les résultats de l'identification sont recueillis après 150 itérations (Figure 4.7). Chaque taux de reconnaissance correspond à une région du plan décrit par les paramètres  $Scale_I$  et  $Leak_P$ . Notre objectif est un taux de reconnaissance à 95% après 150 itérations : nous considérons la région correspondante. Pour calibrer les paramètres, il nous faut choisir un point de cette région : il nous faut donc un critère supplémentaire. Pour choisir ce critère, étudions les rôles précis des deux paramètres  $Scale_I$  et  $Leak_P$  dans l'identification d'une lettre isolée.

En effet, ces deux paramètres n'ont pas la même influence sur la mise à jour de la distribution sur  $P_n^t$ . Si le paramètre  $Scale_I$  ne fait que ralentir la convergence de la distribution, le paramètre  $Leak_P$  empêche la distribution  $P(P_n^t)$  de converger, comme le montre la Figure 4.8.



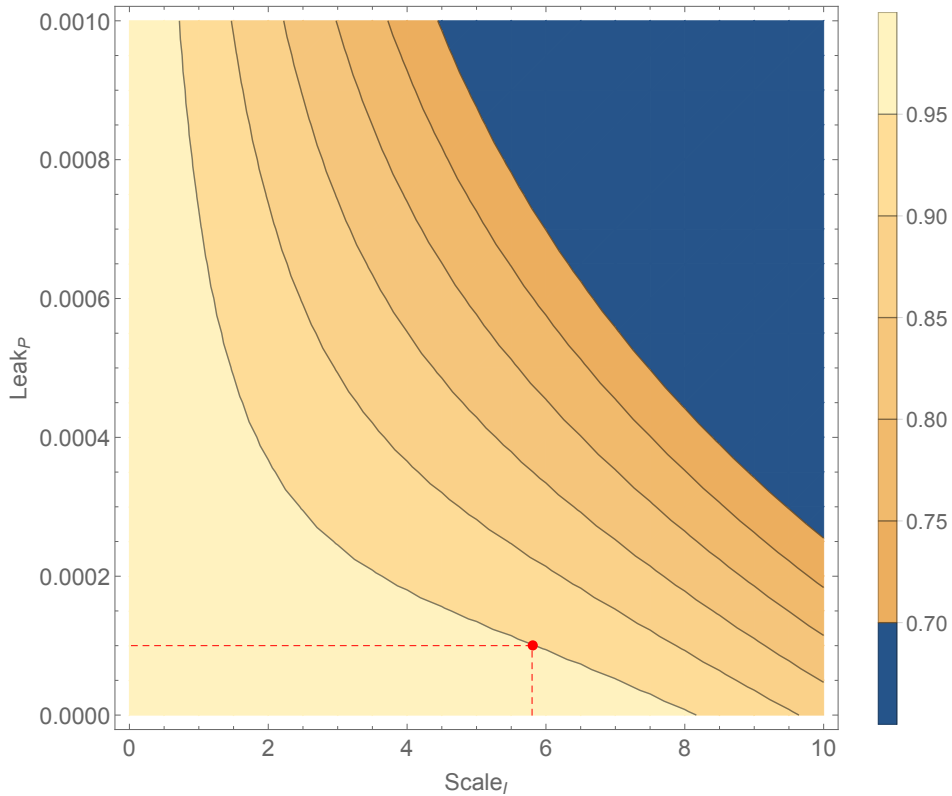


Figure 4.7 – **Calibrage des paramètres  $Leak_P$  et  $Scale_I$ .** Performance moyenne d’identification de lettre en fonction des paramètres de déclin mémoriel,  $Leak_P$  et de mise à l’échelle de l’information entrante,  $Scale_I$ . Les taux de performance fixes décrivent des contours dans cet espace, séparant des régions. Le point rouge correspond aux valeurs de calibrage des paramètres ( $Leak_P = 10^{-4}$  et  $Scale_I = 5,8$ ).

Nous souhaitons limiter le défaut de convergence provoqué par  $Leak_P$  (et donc choisir une petite valeur) sans toutefois éliminer le déclin mémoriel de notre modèle (et donc éviter une valeur nulle). Nous avons donc calibré la valeur de  $Leak_P$  à  $10^{-4}$ , et sélectionné (Figure 4.7) la plus grande valeur de  $Scale_I$  qui garantisse une valeur de probabilité moyenne de 0,95 pour la distribution sur  $P_n^t$  après 150 itérations, soit  $Scale_I = 5,8$ .

Le résultat du déroulement temporel de l’identification de toutes les lettres de l’alphabet est présenté Figure 4.9. On retrouve un taux d’identification moyen de 0,95 après 150 itérations (en vert). Le plus faible taux observé après 150 itérations est à 0,91, mais il arrive à 0,95 après 250 itérations.

### 2.2.3 Pente de l’acuité visuelle $Slope_G$

Maintenant que nous avons paramétré le mécanisme de base permettant l’identification d’une lettre isolée, placée sous le focus attentionnel, nous nous intéressons à l’identification d’une séquence de lettres. Le premier paramètre que nous calibrons est le paramètre qui contrôle la diminution de l’acuité en fonction de l’éloignement de la lettre par rapport à la position du regard,  $Slope_G$ .

**Objectif de calibrage** Pour calibrer ce paramètre, nous nous alignons sur les résultats de l’étude menée par Aghababian et Nazir (2000), qui décrit un modèle linéaire de la diminution

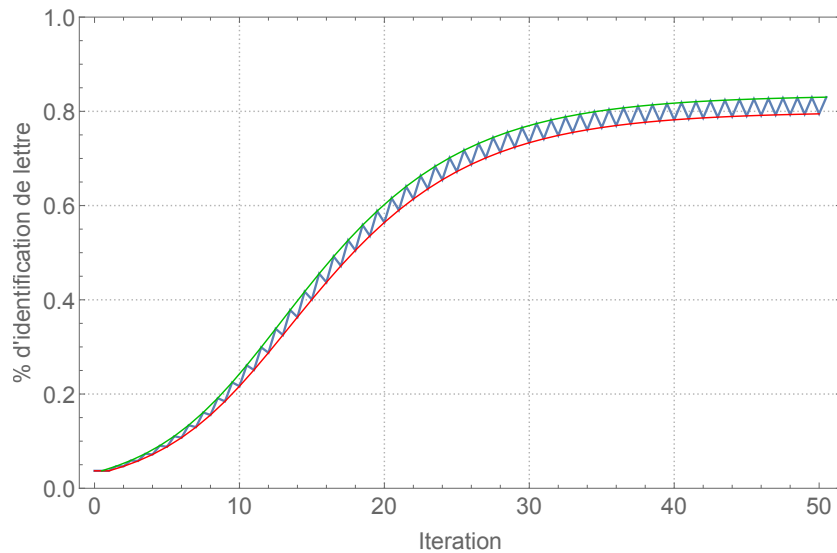


Figure 4.8 – **Rôle de  $Leak_P$  dans la mise à jour de la distribution d'un percept.** Évolution de la probabilité d'identification d'une lettre en fonction des itérations de simulation (courbe rouge). Cette évolution est décomposée en deux étapes pour chaque itération : une étape d'accumulation d'information, et une étape de fuite mémorielle, paramétrée par  $Leak_P$  (courbe bleu en dents de scie). La courbe verte représente la magnitude du déclin à chaque pas de temps.

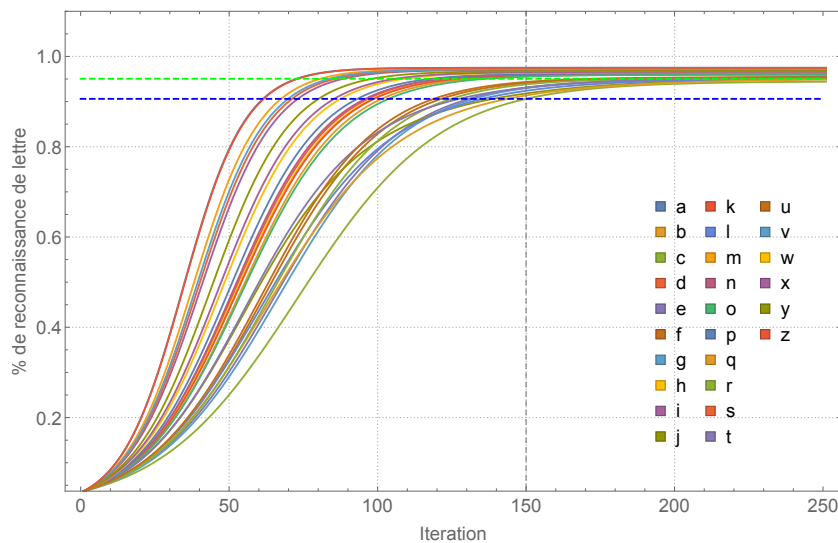


Figure 4.9 – **Décours temporel du taux d'identification d'une lettre isolée.** La ligne verte représente le taux d'identification, moyenné sur toutes les lettres, après 150 itérations. La ligne bleue représente le plus petit taux d'identification (à 0,91), après 150 itérations.

de l'acuité et obtient une valeur de pente comprise entre 3% et 6% après 180 ms de présentation du stimulus.

**Procédure** Dans *BRAID*, lors de l'identification d'une lettre, la diminution de performance en fonction de l'excentricité est due à trois facteurs : la variance de l'attention visuelle, les interférences latérales dues aux lettres voisines et la diminution de l'acuité visuelle. Pour calibrer la pente caractérisant la diminution de l'acuité visuelle, nous minimisons l'influence des

autres paramètres. Pour supprimer l'effet d'interférence, nous utilisons un stimulus composé de cinq lettres identiques, par exemple *AAAAA*. Pour supprimer l'influence de la répartition de l'attention sur le stimulus, nous uniformisons l'attention sur le stimulus en fixant à 0,20 la probabilité  $P(A^T)$  pour toutes les positions. Cette valeur correspond à la quantité d'attention allouée à un stimulus de cinq lettres en position centrale. Enfin, pour réaliser ce calibrage, nous avons placé la position du regard sur la première position et calculé le pourcentage moyen d'identification de toutes les lettres de l'alphabet, pour chacune des cinq positions.

**Résultats** Nous devons montrer deux résultats : d'une part, nous cherchons une valeur du paramètre  $Slope_G$  qui corresponde à une diminution de l'acuité comprise entre 3% et 6%, et d'autre part, nous devons vérifier que cette diminution est linéaire. Pour déterminer  $Slope_G$ , nous avons considéré les cinq valeurs suivantes :  $\{0,5, 0,75, 1,0, 1,25, 1,5\}$  et étudié leur influence, à chaque pas de temps discret compris dans l'intervalle  $[1, 750]$ , sur la pente d'un modèle d'ajustement linéaire calculé sur le taux moyen d'identification des lettres de l'alphabet, pour chacune des cinq positions. Ces résultats sont présentés Figure 4.10, à gauche.

Nous observons que trois valeurs du paramètre (0,5, 0,75 et 1,0) sont dans l'intervalle attendu, c'est-à-dire entre 6% et 8%. Cependant, les performances moyennes d'identification après 150 itérations sont assez faibles, entre 0,41 et 0,18, alors que les performances expérimentales observées sont comprises entre 0,8 et 0,9 (Aghababian & Nazir, 2000). Cette différence s'explique par la probabilité d'attention choisie, qui répond à la contrainte de répartition probabiliste de l'attention dans le modèle, mais qui limite ses performances. Nous avons considéré l'itération 500, pour laquelle le modèle atteint un plafond d'identification pour la première lettre à 0,87. Les valeurs de pente obtenues pour cette itération et les trois valeurs de  $Slope_G$  (0,5, 0,75 et 1,0) sont respectivement 0,019, 0,031, 0,045. Nous avons sélectionné la valeur 1,0, comme valeur du paramètre  $Slope_G$ , car la pente de la diminution de l'acuité qu'il génère est au milieu de l'intervalle de recherche. Le résultat est présenté sur la Figure 4.10, à droite. Ce résultat montre également le caractère linéaire de la diminution de l'acuité, avec une valeur d'ajustement  $\eta^2 = 0,99$ . Donc, que ce soit pour 150 ou 500 itérations, une valeur de  $Slope_G$  à 1,0 remplit les contraintes d'avoir une pente comprise entre 3% et 6%, et que cette diminution soit linéaire.

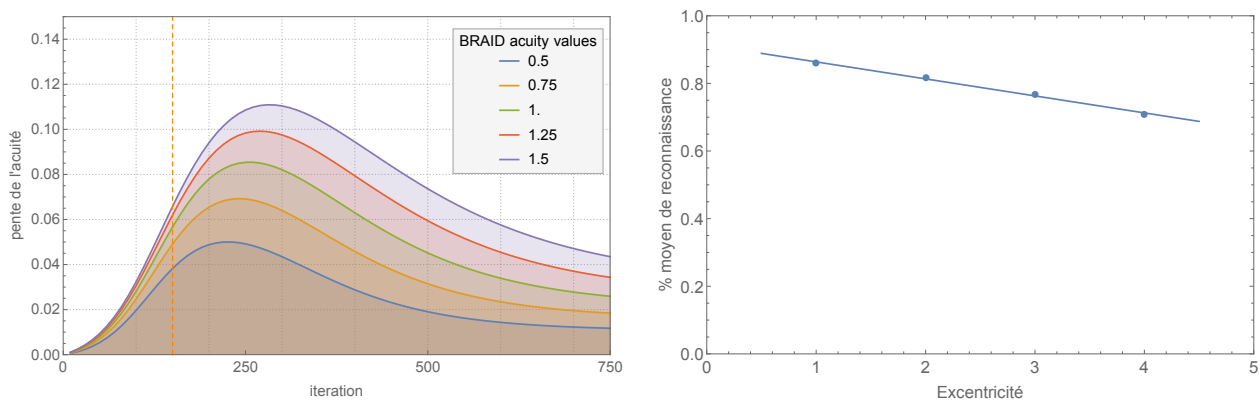


Figure 4.10 – **Calibrage du paramètre d'acuité visuelle  $Slope_G$ .** À gauche : Évolution de la pente du modèle d'ajustement linéaire calculé sur les taux moyens d'identification des lettres pour les cinq positions, calculés à chaque itération et pour différentes valeurs du paramètre  $Slope_G$ . La droite en pointillés oranges matérialise l'itération 150. À droite : Résultat de l'ajustement linéaire après 500 itérations, pour une valeur de paramètre  $Slope_G = 1,0$ .

### 2.2.4 Paramètre des interférences latérales $\theta_i$

Le dernier paramètre qu'il reste à calibrer, avec la tâche d'identification de lettre, est la proportion,  $\theta_i$ , d'interférence latérale produite par les lettres adjacentes.

**Objectif de calibrage** Pour définir la valeur de ce paramètre, nous nous appuyons sur deux études convergentes basées sur une tâche d'identification de lettre, qui comparent deux conditions expérimentales, avec et sans voisins. La première étude montre que le pourcentage d'identification de lettre, après 163 ms, passe de 95% pour une lettre isolée à 83% pour une lettre présentée dans un non-mot de quatre lettres (Massaro, 1979). Dans cette condition, la lettre cible apparaissait aléatoirement à une des quatre positions. La seconde étude repose sur une tâche d'identification de lettres avec amorçage (Goolkasian & King, 1990). Lorsqu'on considère la condition d'amorçage congruente, les performances d'identification observées sont de 96% pour une lettre isolée, et seulement 84% pour une lettre présentée dans un stimulus de trois lettres. Nous ne considérons que les résultats pour lesquels la lettre cible est présentée en position centrale.

**Procédure** Nous évaluons le modèle sur une tâche d'identification de lettre, en faisant varier le paramètre de taux de masquage latéral,  $\theta_i$ , sur l'intervalle  $[0,5, 1]$  par pas de 0,025. Rappelons que lorsque  $\theta_i = 1$ , il n'y a pas d'interférence latérale, et que lorsque  $\theta_i = 0,5$ , chacune des deux lettres adjacentes contribue pour 25% à la reconnaissance d'une lettre. Le matériel utilisé dans ces simulations est composé de dix triplets de lettres, tirés aléatoirement, pour chaque lettre de l'alphabet, la lettre cible étant systématiquement en position centrale. Le paramétrage du modèle utilise les valeurs de  $Leak_P$ ,  $Scale_I$  et  $Slope_G$  définies précédemment. Pour la distribution de l'attention, nous utilisons la répartition utilisée pour calibrer les paramètres précédents  $Leak_P$  et  $Scale_I$ , c'est-à-dire  $\mu_A = 2$  et  $\sigma_A = 0,5$ , ce qui donne une quantité d'attention en position centrale  $P([A^t = 2]) = 0,8$ .

**Résultats** Les résultats des taux d'identification sont recueillis après 150 itérations (voir Figure 4.11). La valeur de paramètre retenue,  $\theta_i = 0,675$ , est celle pour laquelle le taux de performance, 85%, est le plus proche des objectifs.

### 2.2.5 Paramètres lexicaux $Leak_W$ et $Leak_D$

Les deux derniers paramètres qu'il reste à calibrer sont les paramètres  $Leak_W$  et  $Leak_D$  qui caractérisent les modèles dynamiques du sous-modèle de connaissances lexicales,  $P(W^t | W^{t-1})$ , et du sous-modèle d'appartenance lexicale,  $P(D^t | D^{t-1})$ . Le premier modèle dynamique est impliqué dans toutes les tâches simulées par *BRAID* tandis que le second n'intervient que dans la tâche de décision lexicale.

Pour le calibrage de  $Leak_D$ , il est difficile, sinon impossible d'avoir une donnée expérimentale de référence. En effet, dans les études que nous avons considérées, les mesures réalisées avant 400 ms sont interprétées comme le résultat de processus orthographiques, tandis qu'au delà de 400 ms, les mesures sont interprétées comme des processus phonologiques. Il n'est jamais fait mention d'un pic d'activité lié au processus de décision lexicale à proprement parler. Nous nous fixons donc une contrainte relative pour le calibrage de  $Leak_D$  : la décision lexicale doit pouvoir intervenir avant que le mot ne soit entièrement reconnu (Besner & McCann, 1987). Puis, nous fixons la valeur de  $Leak_D$  arbitrairement, et après quelques tests empiriques, à 0,15. Il nous reste à calibrer le paramètre  $Leak_W$  sur des tâches d'identification de mots et de décision lexicale.

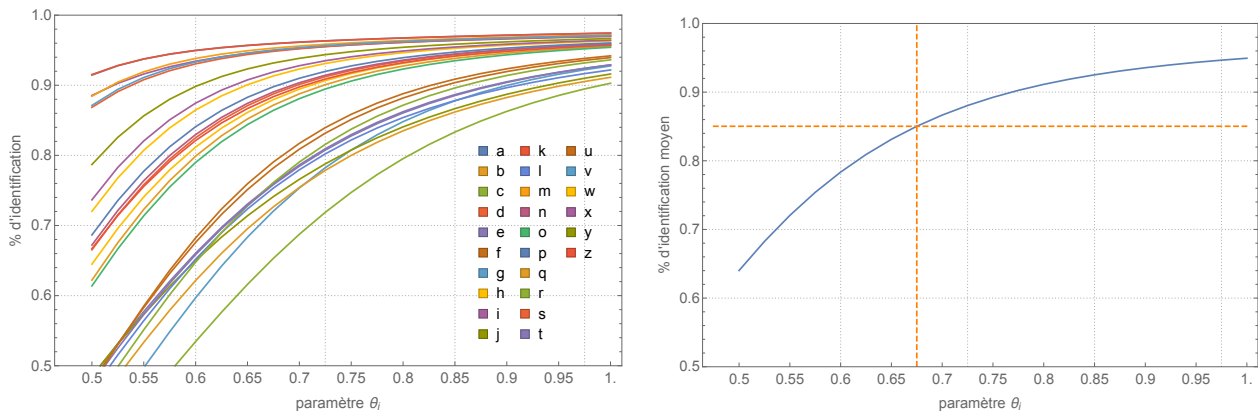


Figure 4.11 – **Calibrage du paramètre d’interférence latérale  $\theta_i$ .** À gauche : taux d’identification pour chaque lettre, après 150 itérations, en fonction de la valeur du paramètre  $\theta_i$ . À droite : taux moyen d’identification, après 150 itérations, en fonction de la valeur du paramètre  $\theta_i$ . La valeur de calibrage du paramètre,  $\theta_i = 0,675$ , est matérialisée par les droites en pointillés oranges.

**Objectif de calibrage** Le paramètre  $Leak_W$  caractérise des processus cognitifs ; il semble donc plus approprié de baser son calibrage sur des données électrophysiologiques obtenues lors de tâches comportementales, que sur des temps de latence qui mettent en jeu d’autres processus, comme des processus de contrôle moteur, par exemple. Pour définir une référence pour le paramètre  $Leak_W$ , nous nous appuyons sur trois études convergentes basées sur une tâche de décision lexicale. La première est une étude MEG qui montre que le pic d’activité M350 est le premier à être sensible à la fréquence des stimuli (Embick, Hackl, Schaeffer, Kelepir, & Marantz, 2001). La seconde étude, également en MEG, montre que ce même pic d’activité n’est pas lié à un processus de décision (Pylkkänen, Stringfellow, & Marantz, 2002). Enfin, la troisième étude est une étude EEG qui montre un pic d’activité P350 reflétant une activité interprétée comme correspondant au niveau mot du modèle IA (Grainger & Holcomb, 2010). Ces trois études placent le maximum d’activité « lexicale » entre 300 ms et 400 ms. Nous proposons donc de calibrer le modèle pour qu’il identifie un mot dans cette fenêtre temporelle.

**Procédure** Pour calibrer  $Leak_W$  nous avons construit des listes de mots en fonction de 3 critères : la fréquence des mots (haute fréquence HF, basse fréquence BF), la densité du voisinage (faible densité FD, haute densité HD), et la fréquence du voisinage (présence d’au moins un voisin plus fréquent HF, absence de voisin plus fréquent BF). Nous avons réalisé des tâches d’identification de mot en considérant 5 valeurs possibles du paramètre : 250, 1 250, 2 500, 5 000 et 10 000. Nous avons également réalisé des tâches de décision lexicale en faisant varier la valeur du paramètre suivant les mêmes 5 modalités.

La contrainte expérimentale choisie est que l’identification des mots intervient entre 300 ms et 400 ms. On définit que *BRAID* a « identifié » un mot lorsque la probabilité d’identification de ce mot a dépassé un seuil à définir. De manière similaire, on définit que *BRAID* a identifié que le stimulus est un mot (réponse « oui » à la question de la décision lexicale), lorsque la probabilité que  $D^t = vrai$  a dépassé un seuil à définir.

Plusieurs remarques sont nécessaires, concernant la définition de ces valeurs de seuil. Tout d’abord, les seuils dépendent évidemment des tâches. En effet, fixer un seuil à 0,75 ne représente pas la même chose en décision lexicale pour laquelle il n’y a qu’une autre alternative, et en reconnaissance de mot pour laquelle il existe plusieurs milliers de mots concurrents. De même,

il est vraisemblable que ces seuils sont sous le contrôle cognitif du participant. Ce dernier ajuste les seuils de décision en fonction de la consigne expérimentale, qui lui demande d’être rapide ou précis ou de trouver un compromis entre précision et rapidité (Norris, 2006). Nous avons donc choisi de ne pas calibrer ces seuils à une valeur unique, mais avons choisi un intervalle,  $[0,70, 0,90]$ , pour les valeurs acceptables de seuil. En effet, au regard des courbes d’identification de mot et des courbes de décision lexicale, cet intervalle semble en contenir l’essentiel de la dynamique.

**Résultats** Les performances de reconnaissance de mots en fonction de la longueur des stimuli, de leurs fréquences et des différentes valeurs du paramètre  $Leak_W$  sont présentées Figure 4.12. Ces résultats montrent que la valeur de  $Leak_W$  a peu d’influence sur la performance de reconnaissance de mots, en condition haute fréquence. En basse fréquence, en revanche, on constate que l’augmentation de la valeur de  $Leak_W$  a deux effets : cela retarde le moment à partir duquel la courbe « décolle », et cela limite la valeur de seuil qu’atteint la courbe après 1 000 itérations. Il n’y a que pour les valeurs de paramètre 250 et 1 250 que la courbe de performance atteint une probabilité moyenne d’identification d’au moins 0,90.

Les résultats en décision lexicale, pour les mots de basse fréquence, et pour les conditions de densité et de fréquence des voisins, sont présentés Figure 4.13. Ces résultats montrent que l’augmentation de la valeur du paramètre  $Leak_W$  influence la décision lexicale pour les mots de basse fréquence dont le voisinage ne contient pas d’élément plus fréquent pour des valeurs de probabilité inférieures à 0,80, et les mots dont le voisinage contient un élément plus fréquent pour des valeurs de probabilité supérieures à 0,80. Il n’y a que pour les valeurs de paramètre 250, 1 250 et 2 500 que la courbe de performance atteint une probabilité moyenne de réponse « oui » d’au moins 0,90.

Pour conclure sur le calibrage du paramètre  $Leak_W$ , nous calculons les performances moyennes, toutes conditions confondues, en reconnaissance de mot et en décision lexicale (voir Figure 4.14). Nous considérons deux fenêtres temporelles : la première, pour la décision lexicale, entre 200 ms et 300 ms et la seconde, pour la reconnaissance de mot, entre 300 ms et 400 ms. Nous considérons le croisement de ces fenêtres temporelles avec notre intervalle acceptable pour les seuils d’identification et de décision (entre 0,7 et 0,9). Nous constatons que la valeur 1 250 pour  $Leak_W$  semble être la mieux adaptée à ces deux contraintes : nous choisissons cette valeur comme valeur par défaut du paramètre.

La valeur 250 permet d’obtenir de meilleures performances, elle sera utilisée pour les tâches d’identification de lettre avec influence lexicale.

## 2.3 Conclusion sur le calibrage du modèle

Avec cette section nous clôturons la phase de calibrage du modèle. Nous avons procédé en deux étapes. Durant la première, nous avons calibré les paramètres contraints par l’expérimentation (voir Tableau 4.1, première partie). On y retrouve notamment la langue du stimulus, le nombre de lettres qui le compose, la position du regard et les paramètres attentionnels. Durant la seconde phase, nous nous sommes appuyés sur des données électrophysiologiques pour calibrer les paramètres qui ont un contrôle sur des processus cognitifs (voir Tableau 4.1, seconde partie). On y trouve les paramètres de traitement perceptif comme la matrice de confusion des lettres, les paramètres de contrôle temporel,  $Leak_P$  et  $Scale_I$ , ainsi que les paramètres d’acuité  $Slope_G$  et d’interférence latérale  $\theta_i$ . On y trouve également les paramètres temporels des connaissances lexicales  $Leak_W$ , et du modèle de décision lexicale,  $Leak_D$ .

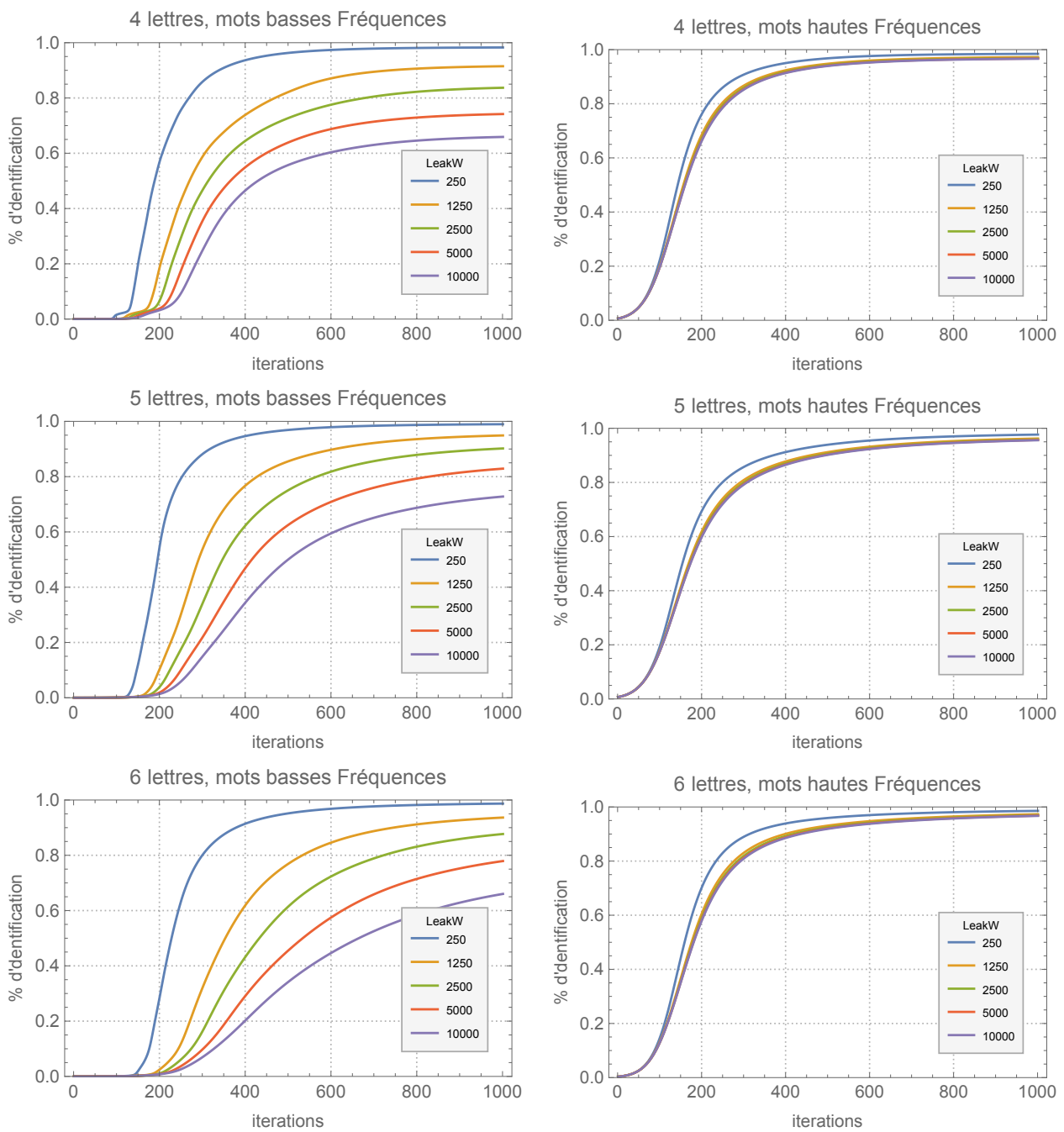


Figure 4.12 – **Influence du paramètre  $LeakW$  en reconnaissance visuelle de mot.** Colonne de gauche : condition de mots de basse fréquence. Colonne de droite : condition de mots de haute fréquence. Lignes de haut en bas : mots de 4, 5 et 6 lettres. Chaque graphique présente les performances moyennes de reconnaissance de mot, en fonction du temps, et pour les différentes valeurs de paramètre  $LeakW$ .

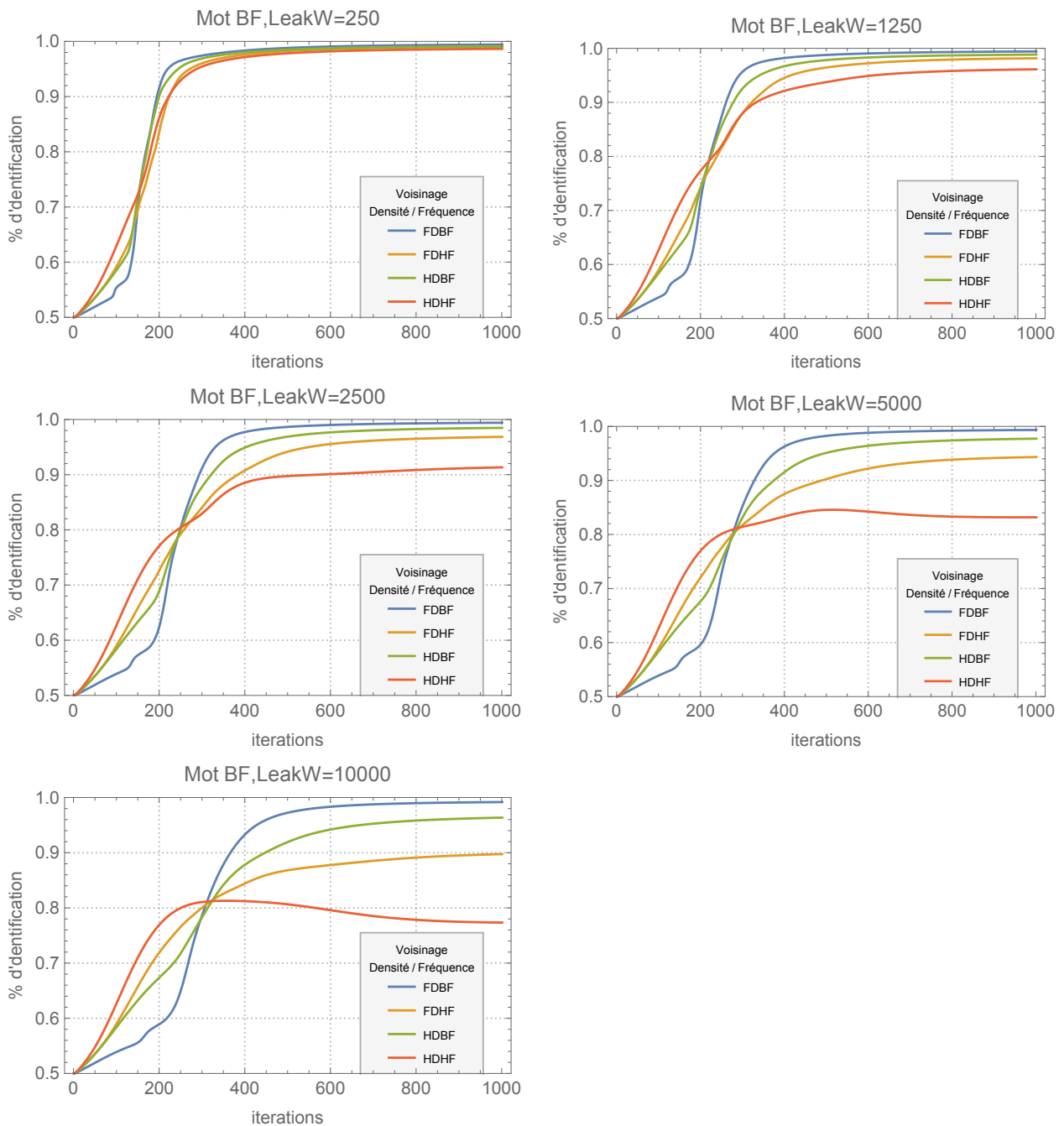


Figure 4.13 – **Influence du paramètre  $Leak_W$  en décision lexicale.** Performances en décision lexicale (probabilité de réponse « oui ») en fonction des itérations, pour une valeur donnée du paramètre  $Leak_W$ , et pour différentes conditions de voisinage.



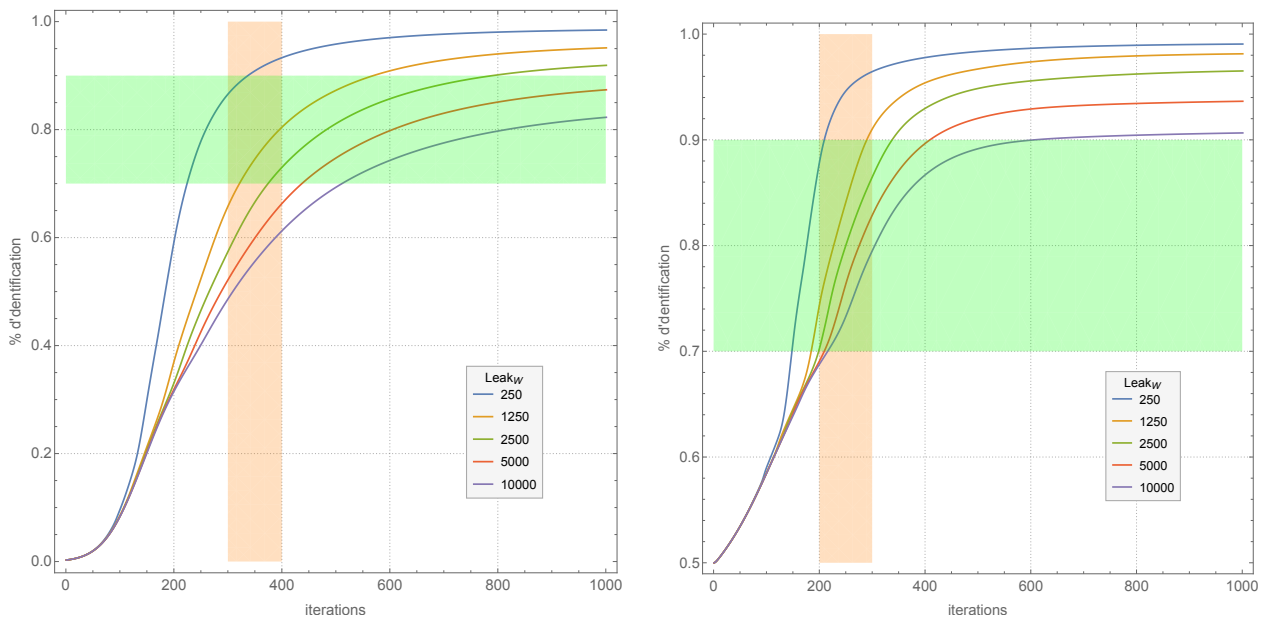


Figure 4.14 – **Influence du paramètre  $Leak_W$  en fonction de la tâche.** À gauche : Taux moyen de reconnaissance de mot en fonction de la valeur du paramètre  $Leak_W$ . À droite : le décours temporel moyen de la décision lexicale (réponse « oui »), en fonction de la valeur du paramètre  $Leak_W$ . La région orange matérialise la zone d'intérêt, comprise entre 300 et 400 itérations pour la reconnaissance de mot, et entre 200 et 300 itérations pour la décision lexicale. La bande horizontale verte matérialise l'intervalle de seuil de décision acceptable.

Tableau 4.1 – Valeurs par défaut des paramètres du modèle *BRAID* : Sauf cas particuliers notés dans le texte, ces valeurs sont les valeurs par défaut utilisées dans les expériences des Chapitres 5 et 6.

<b>Paramètres dépendant de l'expérience à simuler</b>		
<b>Nom</b>	<b>Valeur</b>	<b>Interprétation</b>
$ \mathcal{D}_W $	40 481	Cardinal du domaine des variables $W^{0:T}$ : nombre de mots, fourni par le lexique BLP (anglais)
$\theta_W$	Lexique BLP	Paramètres du prior $P(W^0)$ : fréquence des mots fournie par le lexique BLP (anglais)
$\theta_{LW}$	Lexique BLP	Paramètres des distributions $P(L_n^t   W^t)$ : graphie des mots fournie par le lexique BLP (anglais)
$N$	$\in \{2, \dots, 15\}$	Nombre de lettres du stimulus
$G^t$	$(N + 1)/2$	Position du regard : fixée par défaut au milieu du stimulus
$\mu_A^t$	$(N + 1)/2$	Moyenne des distributions $P(A^t   \mu_A^t \sigma_A^t)$ : fixée par défaut au milieu du stimulus
$\sigma_A^t$	$N \leq 4 : 1,5,$ $N = 5 : 2,0,$ $N \geq 6 : 2,5$	Écart-type des distributions $P(A^t   \mu_A^t \sigma_A^t)$ : fixé par défaut à trois valeurs selon la longueur du stimulus
<b>Paramètres sous le contrôle du modélisateur</b>		
<b>Nom</b>	<b>Valeur</b>	<b>Interprétation</b>
$\theta_{IS}$	<i>Matrice<sub>B</sub></i> modifiée	Paramètres des distributions $P(I_n^t   S_{1:N}^t \Delta I_n^t G^t)$ : identifiés à la matrice de confusion de Townsend (1971a)
$Leak_P$	$10^{-4}$	Paramètre du déclin de l'information dans le modèle dynamique $P(P_n^t   P_n^{t-1})$
$Scale_I$	5,8	Paramètre mettant à l'échelle (temporelle) l'information de la distribution $P(I_n^t   S_{1:N}^t \Delta I_n^t G^t)$
$Slope_G$	1,0	Paramètre mettant à l'échelle (spatiale) l'information de la distribution $P(I_n^t   S_{1:N}^t \Delta I_n^t G^t)$ : diminution de l'acuité visuelle en fonction de l'excentricité
$\theta_I$	0,675	Paramètre contrôlant l'interférence latérale dans la distribution $P(I_n^t   S_{1:N}^t \Delta I_n^t G^t)$ : $\theta_I$ pour la lettre en position $n$ , $(1-\theta_I)$ d'interférence par les lettres adjacentes
$Leak_W$	$\{250, 1\ 250\}$	Paramètre du déclin de l'information dans le modèle dynamique $P(W^t   W^{t-1})$
$Leak_D$	0,15	Paramètre du déclin de l'information dans le modèle dynamique $P(D^t   D^{t-1})$



# Chapitre 5

## Simulations sur la perceptibilité des lettres

Au Chapitre 3 nous avons donné une description détaillée du modèle *BRAID*. Cette description repose sur des variables probabilistes, des relations de dépendances entre ces variables, et elle prend la forme d'une distribution conjointe. Nous avons donné du sens à ce modèle en décrivant chacune des formes paramétriques qui le compose. Au Chapitre 4, nous avons utilisé ce modèle pour répondre, de manière formelle, à un ensemble de questions représentatives des tâches comportementales que l'on retrouve dans les expériences en lien avec la reconnaissance visuelle de mots. A l'aide de ces réponses, nous avons pu calibrer le modèle. Mais à ce stade nous n'avons pas encore confronté *BRAID* à des données comportementales. C'est ce que nous faisons dans ce chapitre (et le suivant) en nous concentrant, tout d'abord, sur la perceptibilité des lettres.

Nous nous intéressons à 10 expériences comportementales portant sur l'effet de supériorité afin d'étudier la capacité de *BRAID* à reproduire les résultats observés. Les trois premières simulations servent à planter le décor. Dans la simulation A01 nous reproduisons l'effet classique de supériorité du contexte *mot* sur le contexte *non-mot* et sur la *lettre isolée* (Johnston, 1978 ; Reicher, 1969 ; Wheeler, 1970). Dans la simulation A02 nous reproduisons l'effet de supériorité du contexte *pseudo-mot* sur le contexte *non-mot* (Baron & Thurston, 1973 ; Spoehr & Smith, 1975). Enfin, dans la simulation A03, nous montrons l'apport du lexique dans la tâche d'identification de lettre.

Les trois simulations suivantes ont pour objectif de mieux cerner le rôle du contexte *mot*. Pour cela, dans la simulation A04, nous faisons varier la durée relative de présentation du contexte par rapport à celle de la lettre cible. Dans la simulation A05, nous étudions les apports relatifs de la lettre et du contexte. Enfin dans la simulation A06, nous étudions l'impact de la qualité du contexte sur les performances d'identification de la lettre cible.

Dans les deux simulations suivantes, nous nous intéressons à l'effet de l'amorçage par des contextes *pseudo-mot* (simulation A07) et *non-mot* (simulation A08) présentées dans l'étude de Rumelhart et McClelland (1982). Enfin, les deux dernières simulations sont consacrées à la disparition de l'effet de supériorité du mot. Dans la simulation A09, nous cherchons à savoir si la connaissance préalable de la position de la lettre cible suffit à faire disparaître l'effet de supériorité du mot, tandis que dans la simulation A10 nous étudions le rôle de l'espacement des lettres sur la disparition de l'effet de supériorité des mots.

Afin d'alléger autant que possible la lecture de ce chapitre, nous allons présenter maintenant, dans une section méthodologie, les règles que nous avons suivies pour réaliser, présenter et analyser ces 10 simulations.

# 1 Méthodologie

Pour garantir une certaine consistance entre ces différentes simulations, nous avons suivi un certain nombre de règles pour présenter les expériences comportementales, pour réaliser nos simulations informatiques, et enfin pour analyser les résultats obtenus. Dans cette section préliminaire, nous présentons ces quelques règles en espérant qu'elle faciliteront la lecture de ce chapitre.

**Présentation des expériences comportementales** Pour décrire les expériences comportementales, nous utilisons systématiquement le même schéma. Après une présentation globale de l'expérience et de ses objectifs, nous décrivons le matériel utilisé, les conditions expérimentales et la procédure. Cette description est cruciale car elle explique les différences entre les expériences, et ces différences peuvent être assez fines. Pour conclure la présentation de l'expérience comportementale, nous présentons les résultats de l'étude sous une forme condensée. Ce qui nous intéresse principalement, ce sont les tendances significatives que les auteurs ont observé dans l'étude, c'est pourquoi nous omettons les détails numériques de ces résultats. Les résultats sont également présentés graphiquement. Pour les simulations A01, A02, A03 et A09 (groupe A1), les observations comportementales sont représentées dans un diagramme barre, en jaune. Pour les simulations A04 à A08 issues de l'étude de Rumelhart et McClelland (1982) (groupe A2), nous présentons le graphique fourni dans l'article avec, à gauche, les résultats comportementaux et à droite, les résultats des simulations avec le modèle IA. Pour ces cinq simulations, les valeurs numériques des moyennes ne sont pas produites dans l'article, nous les avons donc prises directement sur le graphique.

**Méthode de simulations** La méthode utilisée pour présenter les simulations est calquée sur celle utilisée pour présenter les expériences comportementales. Nous commençons par décrire les résultats attendus en nous appuyant en général sur une description intuitive du comportement du modèle. Puis nous décrivons le matériel et la procédure utilisés dans la simulation, pour conclure par les résultats.

Pour le matériel utilisé dans les simulations, nous reprenons autant que possible celui de l'article, en précisant les stimuli qui ont été rejetés et la raison de leur rejet (par exemple, un mot n'est pas présent dans le lexique utilisé dans nos simulations). Quand ce n'est pas possible d'utiliser exactement les stimuli de l'article de référence, nous décrivons en détail la procédure utilisée pour générer nos stimuli. L'ensemble des stimuli est fourni en annexe.

La tâche réalisée par *BRAID* est la même pour toutes les simulations : c'est la tâche d'identification de lettre décrite à la Section 1.3 du Chapitre 4. Cependant, la procédure utilisée varie en fonction des groupes de simulations.

Pour les simulations du groupe A1, nous calculons le décours temporel de la probabilité que la lettre cible soit identifiée, pour chaque stimulus de chaque condition expérimentale (CE). Puis, nous moyennons pour chaque CE et chaque itération les performances du modèle. Le résultat de cette opération est que, pour chaque pas de temps simulé et pour chaque CE, nous disposons de la performance moyenne du modèle. Pour comparer les résultats obtenus aux résultats comportementaux, il nous faut une valeur unique par CE et non le décours temporel complet. Pour obtenir cette valeur, nous fixons une itération de référence pour laquelle la valeur comportementale et le résultat de la simulation sont appariés, pour une condition expérimentale arbitrairement fixée. Le résultat final de la simulation est obtenu en prenant la performance moyenne du modèle à l'itération de référence pour chaque CE.

Pour les simulations du groupe A2, la procédure est différente. Dans ce groupe d'expériences, les simulations correspondent à des expériences avec amorçage, dont la durée de présentation est proportionnelle à celle de l'item considéré. Le calcul du décours temporel de toutes les conditions expérimentales induirait des temps de calcul trop longs. Nous l'avons donc fait uniquement pour la CE fixée et le calcul de l'itération de référence. Pour les autres conditions, nous calculons uniquement la performance du modèle à l'itération de référence. Cette différence de procédure induit de fait une présentation des résultats différente.

**Présentation des résultats des simulations** Pour les simulations du groupe A1, la présentation des résultats se fait graphiquement sur deux figures (voir Figure 5.1). A gauche, nous présentons les courbes de performances moyennes pour chaque pas de temps des différentes CE. Ces courbes nous informent sur l'évolution des performances du modèle au cours du temps. On peut voir ainsi si un effet observé lors d'une expérience comportementale est un effet robuste du modèle ou si, au contraire, il est ponctuel ou s'inverse au cours du temps. Un trait en pointillés orange indique l'itération de référence. A droite, nous présentons un diagramme barre qui compare les résultats comportementaux (en jaune) aux résultats des simulations (en bleu), à l'itération de référence, pour chaque CE. On distingue deux types de résultat : le résultat de la CE qui a servi à choisir l'itération de référence et les résultats des autres CE qui sont prédites par le modèles et serviront pour des analyses complémentaires. Pour les simulations du groupe A2, nous ne disposons que de données ponctuelles pour les simulations. Nous avons donc adopté la forme de présentation utilisée dans l'étude de Rumelhart et McClelland (1982). En complément de ces présentations graphiques nous avons réalisé des analyses statistiques que nous détaillons maintenant.

**Analyse statistiques des résultats** En plus des représentations graphiques des résultats obtenus lors des simulations, nous réalisons une analyse statistique des données simulées. Pour une simulation donnée, une fois définie l'itération de référence, nous disposons de la probabilité d'identification de la lettre cible pour chaque stimulus et chaque CE. Nous réalisons donc une analyse de la variance à mesures répétées, par item. Ces résultats sont présentés à la fin de chaque série de simulations.

## 2 Effet de supériorité des mots et des pseudo-mots

Cette première série de trois simulations a pour objectif de montrer la capacité de *BRAID* à rendre compte des résultats classiques liés à l'effet de supériorité du mot. La simulation A01 s'attache à reproduire l'avantage produit par le contexte *mot* sur les contextes *non-mot* et *lettre isolée*, dans une tâche d'identification de lettre. Avec la simulation A02 nous étudions la capacité du modèle à rendre compte de l'effet de supériorité du pseudo-mot. Enfin, dans la simulation A03, nous étudions l'apport spécifique du niveau lexical du modèle *BRAID*, dans une tâche d'identification de lettre.

### 2.1 Simulation A01 : Effet de supériorité des mots

Le but de cette première simulation est de reproduire l'effet classique de supériorité du mot. Pour cela, nous avons choisi de simuler la seconde expérience de Johnston (1978). Les principes et les résultats de cette étude sont classiquement cités comme représentatifs de l'effet de supériorité du mot (Grainger & Jacobs, 1994 ; Johnston & McClelland, 1980 ; Jordan & Bevan, 1994, 1996 ; Paap et al., 1982).

### 2.1.1 Expérience comportementale : Johnston (1978, expérience 2)

Dans cette expérience, Johnston (1978) compare les performances en identification de lettre dans trois conditions différentes de contexte : la lettre est présentée dans un mot, la lettre est présentée dans un non-mot et la lettre est présentée isolée. L'auteur s'attend à obtenir de meilleures performances d'identification dans la condition *mot* que dans les conditions *non-mot* et *lettre isolée*.

**Matériel et procédure** Les items sont constitués de 4 lettres. Pour former les stimuli *mot*, Johnston (1978) crée 4 groupes de 18 paires de mots, un groupe pour chaque position de la lettre à rappeler (ou lettre cible). Par exemple, les mots *LAST* et *LOST* ont comme contexte commun *L\_ST*, et la lettre cible est en position 2. Lorsque le stimulus est une lettre unique, Johnston (1978) reprend les lettres cibles des mots sélectionnés et garde la même position. Dans notre exemple, *\_A\_\_* et *\_O\_\_*. Enfin les non-mots sont construits à partir des mots en gardant la lettre cible et en mélangeant les lettres du contexte. Par exemple, quand la lettre cible est en position deux, *LAST-SATL* (cas voyelle) et *SHOW-OHSW* (cas consonne). Pour s'assurer que les non-mots ne ressemblent pas à des mots, les contextes sont intervertis. Ce qui donne finalement : *LAST-OASW* et *SHOW-SHTL*.

Les stimuli sont présentés aux participants pour une durée adaptée à chacun (en moyenne 38 ms). Un masquage est réalisé avant et immédiatement après la présentation du stimulus. A l'issue de cette présentation, les participants sont invités à énoncer les lettres formant le stimulus, dans l'ordre, le plus rapidement possible (limite de 4 s). Enfin, ils doivent regarder en dehors du tachistoscope pour répondre à une tâche de choix forcé à 2 alternatives. Le rythme de présentation des stimuli est sous le contrôle des participants.

**Résultats comportementaux** Le résultat majeur de cette expérience est un effet massif du contexte. Les lettres sont mieux reconnues en contexte *mot* qu'en contexte *lettre isolée* et mieux reconnues en contexte *mot* qu'en contexte *non-mot*. L'effet est significatif tant par items que par sujets. Par ailleurs, les lettres sont significativement mieux reconnues lorsqu'elles sont présentées en contexte *lettre isolée* qu'en contexte *non-mot*.

### 2.1.2 BRAID : Simulation A01

Durant la phase de présentation du stimulus, l'activité du niveau lexical du modèle, représentée par la distribution sur  $W^t$ , devrait rapidement se concentrer sur l'item correspondant au mot présenté. Lors de la phase d'identification de la lettre, cette activité lexicale devrait alors contraindre fortement le choix de la lettre cible, même si la distribution sur  $P_n^t$  n'a pas encore piqué<sup>8</sup>.

**Matériel et procédure** Les mots utilisés pour réaliser ces simulations sont ceux proposés par Johnston (1978) à l'exception des mots: *DISK*, *JILL* et *PENT* qui ne figurent pas dans le lexique BLP utilisé dans *BRAID*. Pour la condition mot, nous avons utilisé tous les mots regroupés en fonction de la position de la lettre cible. Pour la condition lettre isolée, nous avons utilisé la lettre cible des mots, en respectant leur position durant la présentation. Enfin pour

---

<sup>8</sup>Même si *BRAID* est un modèle déterministe, la complexité des interactions de ses différents composants rend difficile l'anticipation des résultats produits. C'est pourquoi nous formulons nos attentes sous la forme d'hypothèses. Cependant, cette formulation, que l'on retrouve dans toutes les simulations, ne reflète que notre prudence à l'égard des prédictions que nous formulons sur les résultats retournés par le modèle, et n'est en aucune façon comparable aux hypothèses formulées dans les expériences comportementales.

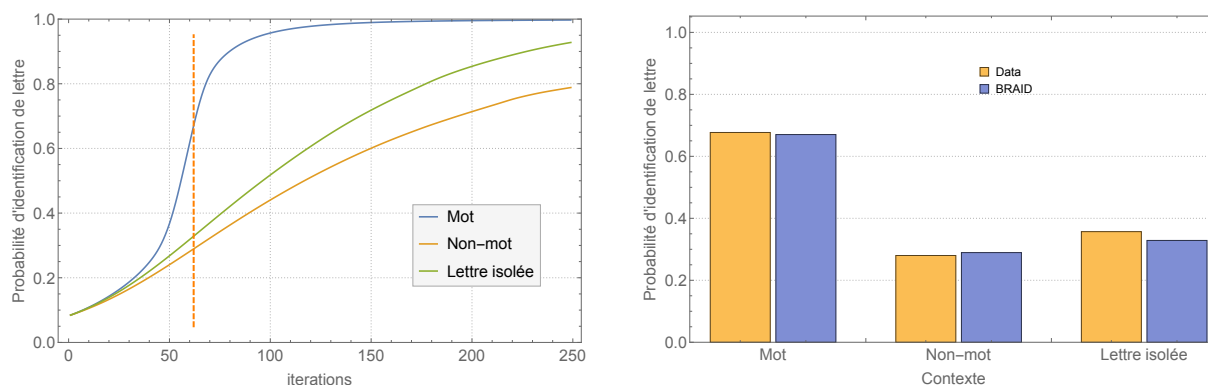


Figure 5.1 – **Simulation A01, Johnston (1978), expérience 2, effet de supériorité du mot.** À gauche, simulation du déroulement temporel de la reconnaissance de lettre pour les trois conditions de contexte. En pointillés rouges, l’itération à laquelle sont extraits les résultats qui sont comparés aux données expérimentales. À droite, présentation des données comportementales (en jaune) et des résultats des simulations (en bleu), pour chacune des trois conditions de contexte (*mot*, *non-mot* et *lettre isolée*).

les non-mots, l’article n’en donnant pas la liste, nous avons utilisé la méthode décrite dans l’article pour construire nos propres non-mots<sup>9</sup>. exemple : *mot* - *LAST*, *lettre isolée* - *\_\_A\_\_* et *non-mot* - *OASW*.

Nous avons simulé une tâche d’identification de lettre, en suivant la méthodologie décrite dans la section 1 de ce chapitre. Les paramètres utilisés sont les paramètres par défaut du modèle.

**Résultats des simulations** Les résultats présentés ont été obtenus après 62 itérations, ce qui correspond à un alignement des performances de *BRAID* et des participants dans la condition *mot*. Les résultats prédits par le modèle sont ceux des conditions *lettre isolée* et *non-mot*. Les résultats des simulations sont présentés dans la Figure 5.1. Les profils de simulations (Figure 5.1, à droite) montrent un effet massif du contexte [ $F(2,542) = 913,7$ ,  $p < 0,001$ ,  $\eta^2 = 0,771$ ]. Une analyse post-hoc par la méthode de Bonferroni montre que les lettres sont mieux reconnues dans le contexte *mot* que dans le contexte *lettre isolée*, [ $t(542) = 28,73$ ,  $p < 0,001$ ]. Elles sont également mieux reconnues dans le contexte *mot* que dans le contexte *non-mot* [ $t(542) = 32,51$ ,  $p < 0,001$ ]. Enfin, elles sont mieux reconnues dans le contexte *lettre isolée* sur la condition *non-mot* [ $t(542) = -12,71$ ,  $p < 0,001$ ]. De plus, les courbes de déroulement temporel (Figure 5.1, à gauche) nous apprennent que par comparaison aux lettres isolées, l’avantage du contexte *mot* était présent dès 43 itérations et continue d’augmenter jusqu’à environ 100 itérations, puis se réduit jusqu’à disparaître à 500 itérations<sup>10</sup>. L’avantage du contexte *mot* par rapport au contexte *non-mot* est par contre observé pour toutes les itérations à partir de la 35ème.

**Conclusion** Les simulations obtenues à l’itération choisie correspondent aux principaux résultats expérimentaux de l’étude de Johnston (1978), en reproduisant l’effet de supériorité du mot par rapport aux deux autres contextes ainsi que l’effet de supériorité des lettres isolées sur les non-mots. Une explication à ce résultat est que les bénéfices produits au niveau lexical par

<sup>9</sup>L’ensemble du matériel utilisé dans les simulations présentées se trouve en annexe. Le matériel de chaque simulation est classé suivant le numéro de la simulation dans laquelle il est utilisé.

<sup>10</sup>Même si nous ne montrons que les 250 premières itérations dans les figures de ce chapitre, l’essentiel de nos simulations a été réalisé sur 500 itérations.



les lettres du contexte dans la condition *non-mot* ne suffisent pas à compenser l'effet délétère des interférences latérales sur la lettre cible, à l'itération retenue.

## 2.2 Simulation A02 : Effet de supériorité des pseudo-mots

Nous avons montré, dans la simulation précédente, qu'un stimulus *mot* produit de l'activité au niveau lexical qui facilite l'identification de la lettre cible. Par ailleurs, nous savons qu'un pseudo-mot produit également une activité au niveau lexical. La question est de savoir si cette activité est suffisante pour faciliter l'identification de la lettre cible.

### 2.2.1 Expérience comportementale : Chase et Tallal (1990)

Dans cette expérience, Chase et Tallal (1990) comparent les performances d'identification de lettres pour trois groupes de sujets : adultes, enfants dyslexiques et enfants témoins. Nous nous sommes focalisés sur les résultats des adultes. Le contexte de présentation des lettres varie suivant trois modalités : *mot*, *pseudo-mot* et *non-mot*. Les auteurs s'attendent à observer de meilleures performances dans les conditions *mot* et *pseudo-mot*.

**Matériel et procédure** Le matériel est constitué de 80 mots de 4 lettres, ayant une fréquence d'au moins 12 pour un million (Kucera & Francis, 1968). Les mots sont regroupés en paires qui ne diffèrent que d'une lettre (la lettre cible) sur chacune des 4 positions. Les non-mots sont construits à partir des mots en réarrangeant les lettres dans l'ordre 3142 (exemple : *THAT-ATTH*). Les pseudo-mots sont construits à partir des mots en changeant la lettre la plus distante de la lettre cible (exemple : *DARK-DARL*). Les voyelles sont remplacées par des voyelles et les consonnes par des consonnes.

La tâche proposée est une tâche de Reicher-Wheeler avec les trois conditions de contexte présentées plus haut. Le protocole est celui des expériences de Rumelhart et McClelland (1982). Chaque essai commence par la présentation d'un point de fixation, suivi, 500 ms après, par la présentation du stimulus. Un masque constitué de la superposition de X et de O apparaît, immédiatement après le stimulus, pendant 100 ms. En même temps que le masque, deux lettres apparaissent au-dessus de la position de la lettre cible. La durée de présentation est ajustée tous les 12 essais de sorte que le participant ait un taux de réussite avoisinant les 75%. Là encore, le participant contrôle le rythme de présentation des stimuli.

**Résultats comportementaux** Le groupe des adultes montre un effet significatif du contexte et de la position ainsi qu'une interaction contexte-position. Une analyse par contrastes polynomiaux montre un effet de supériorité des mots et des pseudo-mots sur les non-mots, mais pas de différence entre mots et pseudo-mots.

### 2.2.2 BRAID : Simulation A02

Lorsque le stimulus présenté est un pseudo-mot, l'activité qu'il produit au niveau lexical est distribuée sur l'ensemble des mots qui sont des voisins orthographiques du stimulus. Le stimulus devrait, comme dans le cas du contexte *mot*, faciliter l'identification de la lettre cible en contraignant sa valeur. Par ailleurs, la présentation d'un non-mot ne produit qu'une activité résiduelle au niveau lexical, c'est-à-dire que la distribution  $P(W^t)$  reste proche de son état initial. Nous attendons donc de meilleures performances en contexte *mot* et en contexte *pseudo-mot* qu'en contexte *non-mot*.

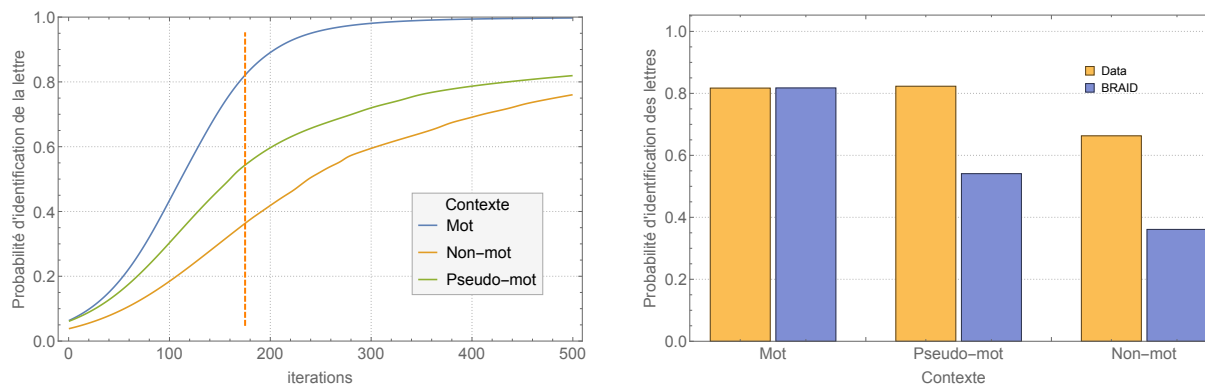


Figure 5.2 – **Simulation A02, Chase et Tallal (1990), effet de supériorité du pseudo-mot.** À gauche, simulation du décours temporel de la reconnaissance de lettre pour les trois conditions de contexte. En pointillés rouges, l’itération à laquelle sont extraits les résultats qui sont comparés aux données expérimentales. À droite, comparaison des données expérimentales (en jaune) avec les résultats des calculs de *BRAID* (en bleu) à l’itération 174, pour chacune des trois conditions de contexte (*mot*, *pseudo-mot* et *non-mot*).

**Matériel et procédure** Le matériel (*mot/pseudo-mot/non-mot*) est fourni dans l’article. Tous les mots sont présents dans le lexique BLP. Tous les pseudo-mots et tous les non-mots sont absents du lexique BLP. Nous n’avons donc apporté aucune modification au matériel fourni pour réaliser ces simulations.

Nous avons simulé une tâche d’identification de lettre, en suivant la méthodologie décrite dans la section 1 de ce chapitre. Les paramètres utilisés sont les paramètres par défaut du modèle.

**Résultats des simulations** Les résultats présentés ont été obtenus après 174 itérations, ce qui correspond à un alignement des performances de *BRAID* et des participants dans la condition *mot*. Les résultats prédits par le modèle sont ceux des conditions *pseudo-mot* et *non-mot*. Les résultats des simulations sont présentés dans la Figure 5.2. Une analyse de la variance à mesures répétées, sur les résultats des simulations montre un effet massif du contexte [ $F(2,158) = 83,65, p < 0,001, \eta^2 = 0,514$ ]. Une analyse post-hoc de Bonferroni montre quant à elle un net avantage du contexte pseudo-mot sur le contexte non-mot [ $t(158) = 4,373, p < 0,001$ ], ce qui reproduit l’effet de supériorité des pseudo-mots décrit dans la littérature. En revanche *BRAID* marque également un net avantage du contexte mot sur le contexte pseudo-mot [ $t(158) = 8,22, p < 0,001$ ] qui n’apparaît pas dans les données comportementales de l’étude. L’avantage des mots sur les pseudo-mots n’est pas observé lors des toutes premières itérations mais apparaît très tôt et se manifeste ensuite de façon stable.

**Conclusion** Les effets expérimentaux principaux, l’effet de supériorité des mots sur les non-mots et l’effet de supériorité des pseudo-mots sur les non-mots, sont correctement reproduits. Les simulations diffèrent des données comportementales de l’expérience de Chase et Tallal (1990) en montrant un effet de supériorité des mots sur les pseudo-mots.

## 2.3 Simulation A03 : Effet du contexte non orthographiquement légal

À la vue de ces premiers résultats, on remarque que les stimuli qui produisent un effet facilitateur sur la perception de la lettre cible sont orthographiquement légaux. Il serait tentant de conclure que les effets observés reposent sur cette caractéristique. Cependant, les résultats précédents suggèrent également que l'avantage obtenu en contexte *mot* et *pseudo-mot* repose uniquement sur l'activité produite par les stimuli au niveau lexical. Selon cette dernière hypothèse, des non-mots illégaux pourraient produire un avantage dans une tâche d'identification de lettre, à condition qu'ils partagent au moins trois lettres avec des mots du lexique (comme c'est le cas pour les pseudo-mots légaux de la simulation A02).

### 2.3.1 Expérience comportementale : Rumelhart et McClelland (1982, expérience 10)

Les auteurs comparent ici les performances d'identification de lettres pour trois conditions de contexte : une condition *pseudo-mot* et deux conditions *non-mot*. Dans la première condition non-mots, notée *non-mot-1*, les stimuli sont construits pour avoir des voisins orthographiques au niveau lexical. Dans la seconde condition non-mot, notée *non-mot-2*, les stimuli sont construits pour ne pas avoir de voisin orthographique au niveau lexical. Les auteurs font l'hypothèse que les performances observées en condition *pseudo-mot* et en condition *non-mot-1* seront similaires entre-elles et supérieures à celles obtenues dans la condition *non-mot-2*.

**Matériel et procédure** Trois groupes de stimuli sont créés à partir de 20 paires de mots (SLOT-SPOT). Les pseudo-mots prononçables sont créés par substitution de la voyelle par une autre voyelle (SLET-SPET). Un premier groupe de non-mots (*non-mot-1*) est créé par remplacement de la voyelle par une consonne (SLCT-SPCT). Chacun de ces non-mots partage son contexte avec au moins trois mots du lexique (exemple pour SPCT : SPAT, SPOT, SPIT). Un second groupe de non-mots (*non-mot-2*) est créé en gardant la lettre cible et en utilisant comme contexte une permutation des trois lettres Q, J, X (exemple : XLQJ-XPQJ) dans le but de minimiser la similarité entre le stimulus et une quelconque trace lexicale.

**Résultats comportementaux** Les lettres sont significativement mieux reconnues dans les non-mots qui partagent leur contexte avec des mots (*non-mot-1*) que dans les *non-mot-2*. Tel que classiquement décrit, on observe également un avantage des pseudo-mots sur les *non-mot-2*, mais un résultat intéressant est que les *non-mot-1*, proches de mots, produisent globalement le même avantage que les pseudo-mots.

### 2.3.2 BRAID : Simulation A03

Par construction, les *pseudo-mots* et les *non-mots-1* devraient produire la même activité au niveau lexical. Nous nous attendons donc à obtenir des performances identiques dans ces deux conditions, et supérieures à la condition *non-mot-2*.

**Matériel et procédure** Le matériel n'était pas fourni dans l'article. Pour construire les stimuli, nous avons utilisé quatre groupes de 18 mots appariés. Les stimuli sont composés de quatre lettres. Les pseudo-mots ont été construits en changeant la lettre la plus éloignée de la lettre cible. Les non-mots partageant le contexte des mots ont été construits en remplaçant la lettre cible par une consonne. La contrainte d'avoir au moins trois mots du lexique partageant

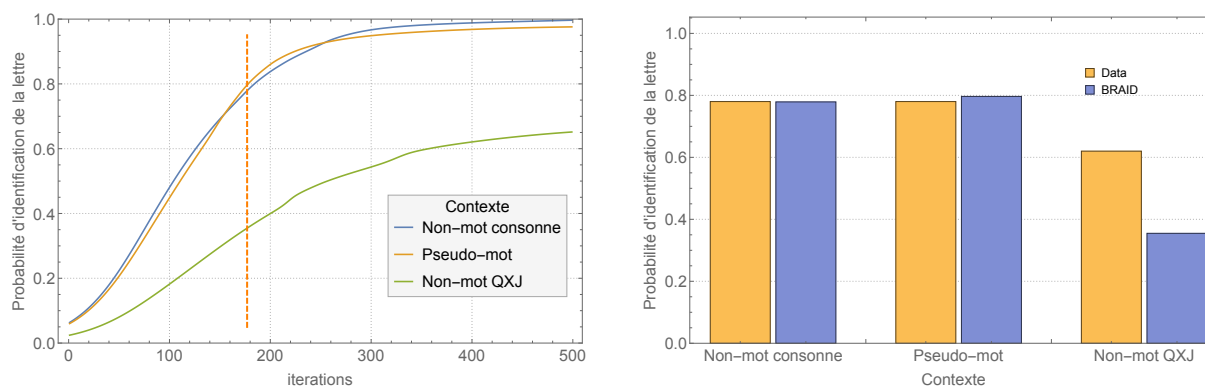


Figure 5.3 – **Simulation A03 Rumelhart et McClelland (1982), expérience 10, effet du contexte non orthographiquement légal.** À gauche, simulation du déroulement temporel de la reconnaissance de lettre pour les trois conditions de contexte. En pointillés rouges, l'itération à laquelle sont extraits les résultats qui sont comparés aux données expérimentales. À droite, comparaison des données expérimentales (en jaune) avec les résultats des calculs de *BRAID* (en bleu), pour chacune des trois conditions de contexte (*non-mot-1*, *pseudo-mot* et *non-mot-2*).

le contexte a été respectée pour 85% des mots, avec un nombre moyen de voisins de 4,8 pour la lettre cible en position 1, de 3,85 pour la position 2, de 8,7 pour la position 3 et de 7,05 pour la position 4. Enfin, les *non-mot-2* sont construits de la même façon que ceux de l'expérience comportementale.

Nous avons simulé une tâche d'identification de lettre, en suivant la méthodologie décrite dans la section 1 de ce chapitre. Les paramètres utilisés sont les paramètres par défaut du modèle.

**Résultats des simulations** Les résultats présentés ont été obtenus après 177 itérations, ce qui correspond à un alignement des performances de *BRAID* et des participants dans la condition *non-mot-1*. Les résultats prédits par le modèle sont ceux des conditions *mot* et *non-mot-2*. Les résultats des simulations sont présentés dans la Figure 5.3. L'analyse de la variance (ANOVA) à mesures répétées, sur les données simulées montre un effet massif du contexte [ $F(2,158) = 71,55, p < ,001, \eta^2 = 0,475$ ]. Une analyse post-hoc de Bonferroni montre que les lettres sont mieux reconnues dans la condition *non-mot-1* que dans la condition *non-mot-2* [ $t(158) = 10,06, p < 0,001$ ]. Comme dans les données expérimentales, l'effet de supériorité des pseudo-mots est le même que l'effet de supériorité des *non-mots-1* [ $t(158) = -0,47, p = 1.$ ]. La courbe de déroulement temporel (Figure 5.3, à gauche) montre qu'un résultat similaire serait obtenu quel que soit le seuil d'itération retenu.

**Conclusion** Les résultats simulés reproduisent les effets expérimentaux de l'expérience 10 de Rumelhart et McClelland (1982). Les non-mots, qui activent partiellement (sur la base de trois lettres) au moins trois mots du lexique, produisent un effet facilitateur sur l'identification de la lettre cible, qu'ils soient orthographiquement légaux ou pas. Ce résultat est indépendant des lettres qui constituent le contexte. C'est un effet robuste qui se manifeste d'emblée et reste stable dans le temps.

## 2.4 Discussion

Dans *BRAID*, l'identification d'une lettre, présentée dans une chaîne de caractère, repose à la fois sur la perception de cette lettre (distribution  $P_n^t$ ) et sur l'activité engendrée au niveau lexical (distribution  $W^t$ ).

Ces trois premières simulations avaient pour objectif de montrer la capacité de *BRAID* à rendre compte des résultats classiques liés à l'effet de supériorité du mot. Dans la première, nous avons simulé l'effet de supériorité du mot sur la lettre isolée et sur une chaîne de caractères quelconque. Dans le modèle *BRAID*, la présentation d'un stimulus mot comme *PARK* entraîne une modification de la distribution sur  $W^t$  (niveau lexical). La probabilité de  $W^t = \text{PARK}$  augmente, contraignant de plus en plus l'identification de la seconde lettre comme étant un *A*. En revanche lorsque le stimulus est une lettre isolée ou un non-mot, la distribution sur  $W^t$  reste proche de son état initial, et n'est pas suffisamment informée pour contraindre efficacement l'identification de la lettre cible.

La seconde simulation portait sur l'effet de supériorité du pseudo-mot. Si nous avons simulé avec succès l'avantage des pseudo-mots sur les non-mots, nous avons également mis en évidence un avantage pour les mots sur les pseudo-mots qui n'est pas présent dans l'expérience de Chase et Tallal (1990). Cependant, une différence de performance entre mots et pseudo-mots a déjà été observée expérimentalement et ce, à de nombreuses reprises (Grainger, Bouttevin, Truc, Bastien, & Ziegler, 2003 ; Paap et al., 1982 ; Rumelhart & McClelland, 1982 ; Simon, Bernard, Largy, Lalonde, & Rebai, 2004). Les travaux de Carr, Davidson, et Hawkins (1978) montrent que les performances sont différentes suivant que les sujets s'attendent à voir des stimuli de même nature (par exemple *mot*) ou des stimuli de nature différentes (par exemple *mot* et *pseudo-mot* mélangés aléatoirement). Ce résultat est confirmé par les travaux de Jordan, Redwood, et Patching (2003). Dans leur étude, les auteurs montrent que la différence de performance observée lors de tâche n'utilisant qu'une sorte de stimulus, comme c'est le cas dans l'étude de Rumelhart et McClelland (1982), devient marginale lorsque les contextes *mot* et *pseudo-mot* sont mélangés comme c'est le cas dans l'étude de Chase et Tallal (1990).

*BRAID* fournit une autre explication possible à cette différence de performance observée entre *mot* et *pseudo-mot* (voir, par exemple, Paap et al., 1982). Lorsque le stimulus présenté est un pseudo-mot, l'activité qu'il produit au niveau lexical est distribuée sur l'ensemble des voisins orthographiques du stimulus. Ainsi deux types de voisins voient leur probabilité augmenter, suivant qu'ils contiennent ou pas la lettre cible. Prenons l'exemple du pseudo-mot *PARL*. Il est construit à partir du mot *PARK* qui facilite l'identification du *A*. Mais il est également voisin du mot *PURL* qui, lui, inhibe l'identification du *A*. Ainsi, pour obtenir des performances identiques en condition *mot* et en condition *pseudo-mot*, il faut que le pseudo-mot n'ait pas de voisin inhibiteur.

La troisième simulation a permis de confirmer que l'avantage obtenu par les *pseudo-mots* n'est dû qu'à l'activité qu'ils génèrent au niveau lexical et non à la régularité orthographique des stimuli. Dans *BRAID*, la présentation de pseudo-mots modifie la probabilité dans  $W^t$  de tous les mots qui partagent la lettre cible et deux autres lettres avec le stimulus. C'est le fait que la distribution de  $W^t$  se concentre sur ces quelques mots et que ces mots contiennent la lettre cible qui donne l'avantage dans la tâche d'identification de lettre. C'est ce que démontre la condition *non-mot-1*. En conclusion, cette troisième simulation a montré que l'effet de supériorité des pseudo-mots ne saurait être interprété comme n'ayant qu'une origine possible, qui serait phonologique.

### 3 Étude du rôle du contexte *mot*

Les trois simulations de cette section portent sur le rôle du contexte *mot*. Pour concevoir ces simulations, nous nous appuyons sur l'étude de Rumelhart et McClelland (1982). Les auteurs y obtiennent une série de résultats comportementaux qu'ils comparent aux résultats de simulations réalisées avec le modèle IA. Dans la simulation A04, nous étudions l'évolution de l'apport du contexte dans le temps en faisant varier la durée de présentation du contexte proportionnellement à celle de la lettre cible. Dans la simulation A05, nous étudions l'apport relatif du niveau lexical ( $W^t$ ) par rapport à l'apport direct du stimulus ( $P_n^t$ ). Enfin dans la simulation A06, nous étudions l'apport du contexte en fonction du nombre de lettres prises en compte au niveau lexical.

#### 3.1 Simulation A04 : Effet de la durée du présentation du contexte

Nous avons vu précédemment qu'une lettre, présentée au sein d'un mot, était plus facilement identifiée que lorsqu'elle est présentée seule ou au sein d'une chaîne de caractères quelconque. Dans cette simulation, il s'agit de déterminer si l'avantage de performance produit par le contexte *mot* est proportionnel à sa durée de présentation.

##### 3.1.1 Expérience comportementale : Rumelhart et McClelland (1982, expérience 1)

Les auteurs dissocient la lettre cible de son contexte, au moment de la présentation du stimulus, afin de manipuler la durée de présentation du contexte indépendamment de celle de la lettre cible. Ils espèrent ainsi observer une augmentation des performances d'identification de la lettre cible avec l'augmentation du temps de présentation du contexte.

**Matériel et procédure** Les stimuli utilisés sont des paires de mots de 4 lettres. La fréquence des mots utilisés est supérieure ou égale à 5 pour un million (Kucera & Francis, 1968). Le ratio de fréquence entre le mot cible et son alternative n'excède pas 2. Toutes les paires possibles de mots dans le lexique ont été utilisées pour chaque position (position 1 : 303 paires, position 2 : 118 paires, positions 3 : 153 paires et position 4 : 176 paires).

Les auteurs utilisent une tâche classique de Reicher-Wheeler dans laquelle ils font varier la durée de présentation du contexte, tout en laissant fixe la durée de présentation de la lettre cible. Cinq rapports de durée sont utilisés (0.6, 0.75, 1.0, 1.33, 1.67), qui représentent la durée de présentation du contexte sur la durée de présentation de la lettre cible. Lorsque le ratio est inférieur à 1, la lettre cible apparaît en premier, puis vient s'ajouter le contexte, tandis que lorsque le ratio est supérieur à 1, le contexte apparaît en premier, puis la lettre cible vient compléter l'affichage. À la fin de la présentation, les quatre lettres du stimulus sont présentes. Elles disparaissent et sont remplacées immédiatement par un masque.

Les simulations effectuées dans l'article avec le modèle IA utilisent 10 paires de mots pour chaque position. 13 cycles de calcul dans le modèle IA ont été nécessaires en moyenne, et le meilleur ajustement a été obtenu 3 cycles après le masquage.

**Résultats comportementaux** Les résultats comportementaux et simulés sont reproduits dans la Figure 5.4 (à droite). Les résultats comportementaux montrent que l'augmentation de la durée de présentation du contexte s'accompagne d'une meilleure reconnaissance des lettres. Pour le ratio le plus bas, les sujets répondent correctement dans 65% des cas, tandis que pour le

ratio le plus haut, ils répondent correctement dans 80%. Chaque résultat est obtenu avec 1 500 observations et un intervalle de confiance de 95% à +/- 2.5%. Les résultats des simulations avec le modèle IA reproduisent assez fidèlement les données expérimentales, produisant 66% de réponses correctes pour la condition de ratio la plus basse, et 81% pour la plus haute.

### 3.1.2 *BRAID* : Simulation A04

Dans *BRAID*, la présentation du contexte (3 lettres) avant le stimulus devrait augmenter la contrainte produite par le niveau lexical et donc entraîner moins d'erreurs. A contrario, la présentation du contexte pour une durée inférieure à celle de la lettre cible devrait diminuer la contrainte lexicale et donc produire plus d'erreurs.

**Matériel et procédure** Les mots utilisés pour les simulations n'étant pas fournis, nous avons utilisé les mots de la simulation A01, construits à partir de ceux de l'article de Johnston (1978). Pour chaque position, nous disposons de 136 paires de mots. Ce matériel est le même pour les trois simulations de cette section qui ont un contexte *mot*.

Nous avons simulé une tâche d'identification de lettre où chaque simulation se déroule en deux temps. Dans un premier temps, le modèle réalise la tâche d'identification à partir d'un stimulus partiel constitué, soit de la lettre cible (ratio < 1), soit du contexte (ratio > 1). Puis, le traitement continue mais en intégrant l'ensemble du stimulus. Le résultat de l'identification est calculé immédiatement à la fin de la présentation. Les paramètres utilisés sont les paramètres par défaut du modèle.

**Résultats des simulations** Les résultats présentés dans la Figure 5.4 (à gauche), ont été obtenus pour une durée de présentation de la lettre cible de 149 itérations, ce qui correspond à un alignement des performances de *BRAID* et des participants sur la première condition de ratio (0.6). Les résultats prédits par le modèle sont ceux des autres conditions (0.75, 1.0, 1.33, 1.67). La probabilité moyenne de bonne réponse obtenue pour le plus petit ratio est de 65%, tandis que celle obtenue pour le plus grand ratio est de 86%. Une analyse de variance à mesures répétées par items met en évidence un effet principal de la durée de présentation du contexte [ $F(4,1084) = 185.3, p < .001, \eta^2 = .406$ ].

**Conclusion** Les simulations réalisées avec *BRAID* reproduisent l'effet de durée du contexte et aboutissent à des pourcentages d'identification proches de ceux obtenus expérimentalement. Les lettres sont d'autant mieux identifiées que le contexte a été présenté plus tôt et par conséquent plus longtemps conformément aux données comportementales.

## 3.2 Simulation A05 : Effet additif de la durée de présentation de la cible et du contexte

Nous venons de voir que l'identification de la lettre cible repose en partie sur le niveau lettre,  $P_n^t$  et en partie sur le niveau lexical,  $W^t$ , lorsque le stimulus est un mot. Avec cette simulation, nous cherchons à évaluer la contribution relative de ces deux sources dans l'identification de la lettre cible.

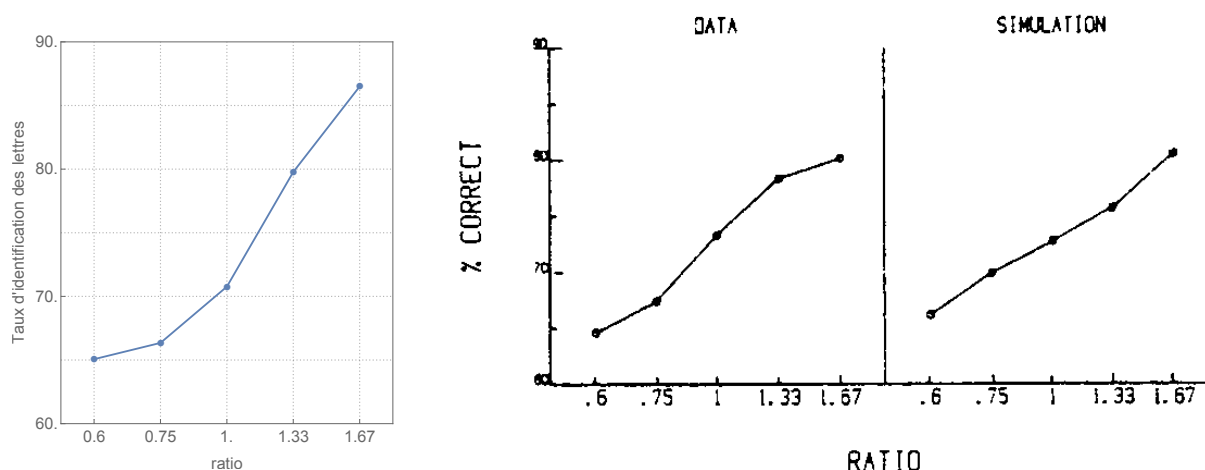


Figure 5.4 – **Simulation A04, Rumelhart et McClelland (1982), expérience 1, effet de la durée du présentation du contexte.** À gauche, la probabilité d'identification de la lettre cible calculée par *BRAID* en fonction du temps de présentation du contexte. À droite, les résultats issus de l'étude de Rumelhart et McClelland (1982), résultats comportementaux et résultats de simulation avec le modèle IA.

### 3.2.1 Expérience comportementale : Rumelhart et McClelland (1982, expérience 5)

Dans cette expérience comportementale, les auteurs font varier le temps de présentation soit de la lettre cible, soit du contexte. Ils cherchent alors à savoir si l'augmentation du temps de présentation de la lettre cible produit les mêmes effets que l'augmentation du temps de présentation du contexte.

**Matériel et procédure** Le matériel est le même que celui utilisé dans l'expérience 1 de Rumelhart et McClelland (1982, voir section précédente).

La tâche est similaire à celle de l'expérience A04. Une durée de présentation de référence  $D$  est calculée pour chaque participant et ajustée de sorte qu'il ait un taux de reconnaissance des lettres de l'ordre de 75%. Les auteurs font varier la durée de présentation suivant deux modalités (durée simple, notée  $D$  ou durée doublée, notée  $2D$ ), et selon l'item considéré (la lettre ou le contexte) pour obtenir quatre conditions de présentation : (lettre- $D$ , contexte- $D$ ), (lettre- $D$ , contexte- $2D$ ), (lettre- $2D$ , contexte- $D$ ) et (lettre- $2D$ , contexte- $2D$ ).

**Résultats comportementaux** Dans l'expérience comportementale (Figure 5.5, à droite), l'augmentation de la durée de présentation de la lettre cible ou du contexte se traduit par une meilleure identification des lettres. Ces deux effets sont significatifs. L'augmentation de durée de présentation de la lettre a un effet plus important que l'augmentation de durée de présentation du contexte et ces deux effets sont cumulatifs. Le modèle IA reproduit correctement ces effets mais les simulations accentuent l'effet de durée de présentation de la cible comparativement à celle du contexte.

### 3.2.2 *BRAID* : Simulation A05

Le doublement du temps de présentation de la lettre cible devrait diminuer l'incertitude dans  $P_n^t$ , tandis qu'un doublement du temps de présentation du contexte devrait diminuer l'incertitude



dans  $W^t$ . Dans ces deux conditions, les performances d'identification devraient être meilleures que pour une durée simple d'exposition. De plus, ces deux effets agissent sur des variables différentes du modèle, ils devraient donc s'additionner, dans la limite d'une valeur plafond d'identification des lettres.

**Matériel et procédure** Le matériel utilisé est le même que pour la simulation A04.

Nous avons simulé une tâche d'identification de lettre en dissociant la présentation de la lettre de la présentation du contexte, comme dans la simulation A04. Pour la condition (lettre-D, contexte-2D), nous avons présenté le contexte seul pour une durée D, puis le stimulus complet. Pour la condition (lettre-2D, contexte-D), nous avons présenté la lettre seule pour une durée D, puis le stimulus complet. Les paramètres utilisés sont les paramètres par défaut du modèle.

**Résultats des simulations** Les résultats présentés dans la Figure 5.5 (à gauche) ont été obtenus pour une durée D de 93 itérations, et un alignement des performances de *BRAID* et des participants sur la condition (lettre-2D, contexte-2D). Les résultats prédits par le modèle sont ceux des trois autres conditions. Les profils de réponse obtenus sont similaires à ceux de l'expérience comportementale. Une analyse de variance à mesures répétées par items des résultats des simulations montre des effets significatifs à la fois de la durée de présentation du contexte [ $F(1,271) = 153.1, p < .001, \eta^2 = .361$ ] et de la durée de présentation de la lettre cible [ $F(1,271) = 1632.51, p < .001, \eta^2 = .858$ ]. Contrairement aux résultats comportementaux, on observe un effet d'interaction [ $F(1,271) = 6.188, p = 0.013, \eta^2 = 0.022$ ] mais il est de faible magnitude. De même, l'effet de la durée de présentation des lettres (+0.31) est supérieur à l'effet de la durée de présentation du contexte (+0.11). Cependant, on peut remarquer que la contribution du contexte dans la simulation (+0.11) est sensiblement identique à celle observée chez les sujets (environ +.09), tandis que la contribution de la lettre dans la simulation (+0.31) est deux fois plus importante que celle des sujets (environ +0.15). Enfin, on observe également l'effet cumulatif des effets lettre et contexte.

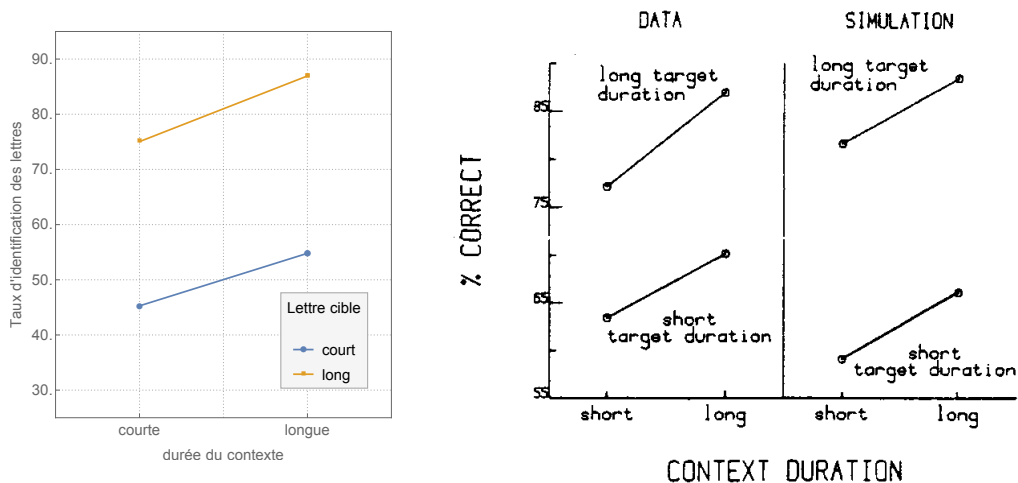


Figure 5.5 – **Simulation A05, Rumelhart et McClelland (1982), expérience 5, effet additif de la durée de présentation de la cible et du contexte.** À gauche, la probabilité d'identification de la lettre cible simulée par *BRAID* en fonction du temps de présentation du contexte. La courbe bleue (resp. jaune) correspond à une durée de présentation de la lettre cible courte (resp. longue). À droite, les résultats issus de l'étude de Rumelhart et McClelland (1982), résultats comportementaux et résultats de simulation.

**Conclusion** Les résultats du modèle *BRAID* reproduisent les effets facilitateurs de l'augmentation de la durée de présentation de la cible et du contexte. On observe notamment que cet effet est plus important lorsque l'on augmente la durée de présentation de la lettre cible que lorsque l'on augmente la durée de présentation au contexte. Cependant, cette différence est plus marquée dans les simulations réalisées par *BRAID* que dans les résultats comportementaux. Enfin, les effets obtenus sont additifs comme dans les données expérimentales.

### 3.3 Simulation A06 : Effet du nombre de lettres du contexte

Après avoir fait varier la durée de présentation, nous nous intéressons à la qualité du contexte. Nous avons déjà vu qu'un stimulus *mot* dont le contexte *utile*<sup>11</sup> est composé de 3 lettres induisait de meilleures performances qu'un stimulus *pseudo-mot* pour lequel le contexte *utile* est au plus composé de 2 lettres. Nous voulons voir si cette observation peut se généraliser. Nous faisons donc l'hypothèse que plus le contexte *utile* est grand, plus son influence sur l'identification de la lettre cible est importante.

#### 3.3.1 Expérience comportementale : Rumelhart et McClelland (1982, expérience 6)

Dans cette expérience, les auteurs présentent une amorce constituée d'une partie des lettres du contexte (0, 1, 2 ou 3 lettres), puis le stimulus complet. Ils font l'hypothèse que l'augmentation du nombre de lettres du contexte présentées en amorce produira une augmentation des performances d'identification de la lettre cible.

**Matériel et procédure** Dans cette expérience, différentes amorces sont créées à partir du contexte initial en faisant varier le nombre et la position des lettres. Huit conditions différentes sont testées : une amorce vide, trois amorces avec 1 lettre, trois amorces avec 2 lettres et une dernière amorce avec les trois lettres. Par exemple, pour le mot *SPOT* et la condition "lettre cible en première position" on obtient :  $\_ \_ \_ \_$ ,  $\_ P \_ \_$ ,  $\_ \_ O \_$ ,  $\_ \_ \_ T$ ,  $\_ PO \_$ ,  $\_ P \_ T$ ,  $\_ \_ OT$ ,  $\_ POT$ .

La tâche à réaliser sont similaires à celles de l'expérience A04. L'amorce est présentée pendant une durée  $D$ , puis elle est remplacée par le stimulus complet pour une durée  $D$  supplémentaire. À l'issue de cette présentation, un masque est appliqué à la place du stimulus.

**Résultats comportementaux** Pour l'expérience comportementale, la reconnaissance de la lettre cible augmente quasi-linéairement avec le nombre de lettres composant l'amorce (Figure 5.6, à droite). Les simulations effectuées avec le modèle IA reproduisent l'augmentation des performances d'identification de la lettre cible en fonction du nombre de lettres du contexte.

#### 3.3.2 *BRAID* : Simulation A06

Dans *BRAID*, la présentation d'une amorce contenant certaines des lettres du stimulus engendre une modification de la distribution de  $W^t$  en faveur des mots qui contiennent ces lettres, dans la bonne position. L'augmentation du nombre de lettres dans l'amorce réduit le nombre de mots activés dans  $W^t$ , augmentant par là-même l'effet de l'amorçage sur le mot présenté comme stimulus. On devrait donc observer une augmentation des performances d'identification de la lettre cible avec l'augmentation du nombre de lettres constituant l'amorce.

<sup>11</sup>Le contexte « utile » est le nombre de lettres du stimulus, autre que la lettre cible, faisant écho à un mot du lexique.

**Matériel et procédure** Le matériel utilisé est le même que précédemment (voir simulation A04).

Nous avons simulé une tâche d'identification de lettre en deux phases comme dans la simulation A04. L'amorce varie en fonction du nombre de lettres du contexte qui la compose (0, 1, 2 ou 3). Les paramètres utilisés sont les paramètres par défaut du modèle.

**Résultats des simulations** Les résultats présentés dans la Figure 5.6 (à gauche), ont été obtenus après 115 itérations (durée de présentation du stimulus), et un alignement des performances de *BRAID* et des participants sur la condition où deux lettres sont utilisées dans l'amorce. Les résultats prédits par le modèle sont ceux des trois autres conditions. La courbe des résultats reproduit l'effet quasi-linéaire observé dans l'expérience comportementale. Une analyse par items met en évidence un effet principal du nombre de lettres du contexte [ $F(3,1916) = 44.92, p < .001, \eta^2 = .066$ ]. La probabilité moyenne de bonne réponse obtenue en l'absence de contexte est de 63,3%, celle obtenue pour le contexte complet est de 84,4%.

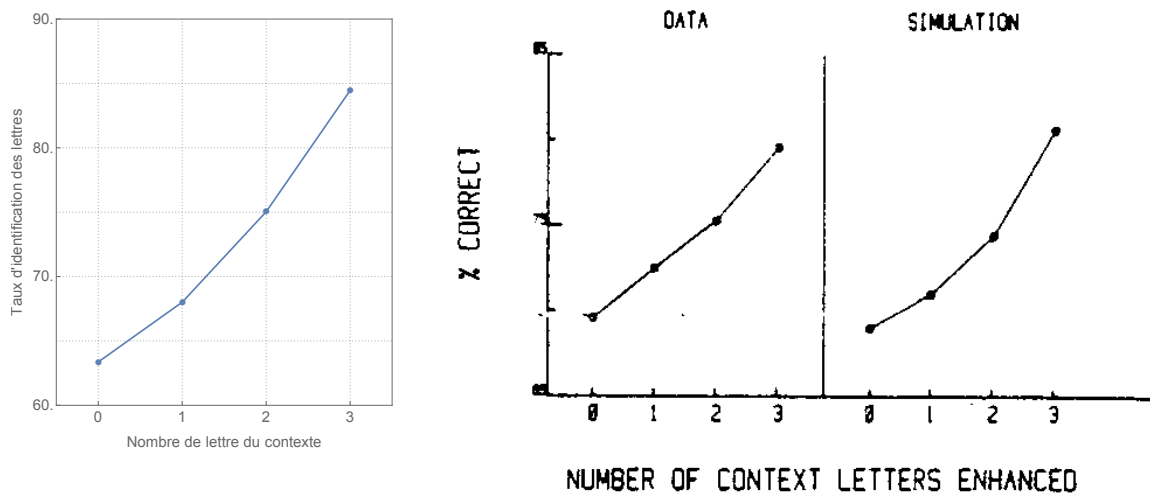


Figure 5.6 – **Simulation A06, Rumelhart et McClelland (1982), expérience 6, effet du nombre de lettres du contexte.** À gauche, la probabilité d'identification de la lettre cible calculée par *BRAID* en fonction du nombre de lettres du contexte. À droite, les résultats issus de l'étude de Rumelhart et McClelland (1982), résultats comportementaux et résultats de simulation.

**Conclusion** Dans *BRAID*, l'augmentation des performances dans la tâche d'identification de lettre est proportionnelle au nombre de lettres du contexte utilisées durant l'amorçage, comme observé expérimentalement par Rumelhart et McClelland (1982). Ce résultat généralise ceux observés dans les premières simulations (simulation A01 et A02), dans lesquelles le contexte *mot*, utilisant les trois lettres du contexte, produit un avantage supérieur au contexte *pseudo-mot* qui, lui, utilise au plus deux lettres du mot. Les trois expériences précédentes permettent donc d'établir que la durée de présentation du contexte, lorsque le contexte est un mot, ainsi que le nombre de lettres du contexte, influencent positivement les performances d'identification de la lettre cible.

### 3.4 Discussion

Les trois simulations précédentes avaient pour objectif d'étudier le rôle du contexte dans la tâche d'identification de lettre en contexte *mot*. Dans la simulation A04, c'est la durée de présentation du contexte relativement à la durée de présentation de la lettre cible qui varie. Pour cette simulation, le décours temporel de l'activité au niveau lettre  $P_n^t$  de *BRAID* ne varie pas suivant les conditions. Les différences de performances observées ne sont dues qu'aux différents niveaux d'activité de  $W^t$ . Or le décours temporel de  $W^t$  pour le stimulus est une fonction sigmoïde. Ceci explique la forme de la courbe de performance de *BRAID*. La même tendance est observée pour les données comportementales.

Cette remarque suggère que l'information accumulée au niveau  $W^t$  se combine à celle du niveau  $P_n^t$ , et améliore l'identification des lettres. C'est ce que montrent les résultats de la simulation A04. En effet, *BRAID* reproduit l'effet facilitateur produit par le doublement de la durée de présentation du contexte d'une part et de la lettre cible d'autre part. Mais il reproduit également l'effet additif de ces deux augmentations de durée. Enfin, le fait que l'avantage produit par le doublement de la durée de présentation de la lettre cible est plus important dans les simulations que dans les résultats comportementaux s'explique par le fait que le décours temporel de l'identification de la lettre cible  $P_n^t$  n'est sans doute pas assez progressif dans *BRAID*.

Finalement, comme pour les données comportementales, la simulation A06 produit une relation linéaire entre le nombre de lettres du contexte et les performances de *BRAID*. Cette série de simulations a donc montré que *BRAID* est capable de rendre compte de l'ensemble des données comportementales liées aux effets du contexte *mot*.

## 4 Étude du rôle des contextes *pseudo-mot* et *non-mot*

Dans les simulations où le stimulus est un mot (voir section 3), nous avons vu que la présentation du contexte, juste avant la présentation du stimulus complet, agissait comme un amorçage et augmentait les performances des participants en identification de lettre.

Dans cette section, nous nous intéressons au cas où le stimulus présenté n'est pas un mot. Les résultats comportementaux rapportés dans l'étude de Rumelhart et McClelland (1982) montrent que l'avantage produit avec un amorçage utilisant le contexte *pseudo-mot* est similaire à celui produit par un contexte *mot*, tandis que le contexte *non-mot* produit un avantage trop faible pour être significatif.

Le résultat obtenu avec le contexte *pseudo-mot* nous semble paradoxal. En effet, bien que les études comportementales montrent une augmentation des performances, le contexte *pseudo-mot* présenté en amorce ne peut produire qu'un effet délétère sur l'identification de la lettre cible. Pour illustrer cette affirmation, prenons l'exemple du mot *HEAP*, de la lettre cible *H* et du pseudo-mot associé *HEAM*. La présentation du contexte *\_EAM* amorce des mots tels que *BEAM*, *REAM*, *SEAM* et *TEAM*, fournissant un avantage aux lettres *B*, *R*, *S* et *T*, mais pas à la lettre cible *H*. D'ailleurs, par construction, la lettre *H* ne peut pas être activée, sinon *HEAM* serait un mot.

Pour expliquer les résultats comportementaux, nous faisons l'hypothèse que les participants ont adopté une stratégie spécifique, induite par ces contextes *pseudo-mot* et *non-mot*. La diffusion lente de l'activité au niveau lexical (aucun mot ne voit sa probabilité augmenter rapidement), associée à l'apparition soudaine de la lettre cible, inciterait le participant à déplacer son attention pour augmenter sa prise d'information. Dans cette hypothèse, le déplacement attentionnel vers la lettre cible se produirait après un temps de latence, durant lequel le participant

traiterait le stimulus complet. Les deux simulations suivantes ont pour objectif de tester cette hypothèse.

## 4.1 Simulation A07 : Effet de la durée de présentation en contexte pseudo-mot

Dans cette simulation, nous cherchons à savoir si la présentation du contexte *pseudo-mot* juste avant la présentation du stimulus complet produit la même augmentation de performance dans la tâche d'identification de lettre que celle produite par le contexte *mot* dans la simulation A05.

### 4.1.1 Expérience comportementale : Rumelhart et McClelland (1982, expérience 7)

Dans cette expérience, les auteurs comparent les performances d'identification de lettre dans deux conditions de contexte : *mot* et *pseudo-mot*, et selon deux conditions de durée de présentation ( $D$  et  $2D$ ).

**Matériel et procédure** Les mots utilisés sont ceux de l'expérience A04. Les pseudo-mots sont construits à partir des mots. Pour chaque mot et chaque position cible, un pseudo-mot est formé en changeant la lettre la plus éloignée de la position cible dans le mot, tout en respectant la nature de la lettre : une voyelle est remplacée par une voyelle et une consonne par une consonne. Par exemple, *WORD* est changé en *WORL* quand la position cible est la position 1 ou 2 et en *MORD* lorsque la position cible est 3 ou 4. Cette procédure de construction des pseudo-mots permet de garantir que les groupes formés par la lettre cible et sa (ses) lettre (s) voisine (s) sont les mêmes pour les mots et les pseudo-mots.

Les conditions expérimentales et la tâche à réaliser sont similaires à celles de l'expérience A05. Les stimuli considérés sont de deux types (*mot* et *pseudo-mot*). Dans la condition  $2D$ , le contexte est présenté sans la lettre cible pendant une durée  $D$ , puis il est remplacé par le stimulus complet pour une durée  $D$  supplémentaire. À l'issue de cette présentation, un masque est appliqué à la place du stimulus.

**Résultats comportementaux** Les auteurs observent un effet significatif de la durée de présentation du contexte (Figure 5.7, à droite). La nature du contexte, *mot* ou *pseudo-mot*, a un effet significatif sur les performances des participants. En revanche, il n'y a pas d'interaction entre la durée de présentation et la nature du contexte. Le contexte pseudo-mot influe de la même manière que le contexte mot sur les performances d'identification de la lettre cible.

Pour les résultats des simulations présentées dans l'article, si le modèle IA réussit à reproduire l'effet de durée de présentation du contexte pour les mots ainsi que l'effet de contexte entre *mot* et *pseudo-mot*, il échoue à reproduire l'effet de durée de présentation du contexte pour les pseudo-mots. Pour obtenir les résultats présentés Figure 5.7 (à droite), les auteurs ont effectué deux ajustements sur les paramètres de leur modèle qui seront discutés plus loin, dans la discussion.

### 4.1.2 BRAID : Simulation A07

Pour rendre compte de l'effet du déplacement attentionnel, nous réalisons deux fois les simulations, une fois avec et une fois sans déplacement attentionnel.

**Matériel et procédure** Le matériel utilisé (mots et pseudo-mots) est le même que pour la simulation A04.

Dans la simulation sans déplacement attentionnel, les paramètres utilisés sont les paramètres par défaut du modèle. La tâche d'identification de lettre est identique à celle de la simulation A05, mais avec les contextes *mot* et *pseudo-mot*. Pour les conditions où la durée de présentation du contexte est  $2D$ , nous avons présenté le contexte seul pour une durée  $D$ , puis le stimulus complet.

La simulation avec déplacement attentionnel ne concerne que la condition dans laquelle le contexte *pseudo-mot* est présenté en amorce (condition *pseudo-mot* et durée  $2D$ ). Dans cette condition, la simulation se déroule en trois étapes. Dans la première étape, le modèle traite le contexte qui lui est présenté seul, sans modification de paramètre, pendant  $D$  itérations (c'est la phase d'amorçage). Dans la deuxième étape, le modèle traite le stimulus complet, toujours sans modification de paramètres, pendant  $3D/4$  itérations (c'est la phase de latence). Enfin, dans la troisième étape, le modèle traite le stimulus complet et le paramètre représentant la moyenne de la distribution d'attention est déplacé sur la lettre cible (c'est la phase avec déplacement attentionnel). La durée de traitement est la durée restante, soit  $D/4$ . Ce ratio  $3/4$ ,  $1/4$  a été choisi empiriquement de manière à conserver une dynamique d'accumulation d'information sur la lettre cible qui ne soit pas trop rapide.

**Résultats des simulations** Les résultats présentés dans la Figure 5.7 (à gauche), ont été obtenus pour  $D = 154$  itérations, et un alignement des performances de *BRAID* et des participants sur la condition contexte *mot* et durée de présentation  $D$ . Les résultats prédits par le modèle sont ceux des trois autres conditions. Les résultats des simulations utilisant les paramètres par défaut sont présentés Figure 5.7 (à gauche). Ils montrent l'incapacité de *BRAID* à rendre compte de l'effet du contexte sur les pseudo-mots, en l'absence d'hypothèse sur le rôle de l'attention. Nous allons donc nous focaliser sur les résultats de la simulation avec modification des paramètres attentionnels (voir Figure 5.7, au centre). Une analyse de la variance à mesures répétées montre que *BRAID* réussit à reproduire l'effet de durée de présentation du contexte [ $F(1,1084) = 337.28$ ,  $p < .001$ ,  $\eta^2 = .23$ ], l'effet de la nature du contexte (*mot* vs *pseudo-mot*) [ $F(1,1084) = 40.06$ ,  $p < .001$ ,  $\eta^2 = .027$ ] et l'absence d'interaction entre les deux conditions [ $F(1,1084) = 2.12$ ,  $p = .146$ ,  $\eta^2 = .001$ ].

**Conclusion** Les résultats des simulations viennent confirmer ceux obtenus lors de la simulation A01. On observe un net avantage de la condition *mot* sur la condition *pseudo-mot*. Les résultats viennent également confirmer les hypothèses formulées en début d'étude. Sans prise en compte de l'attention, l'effet d'amorçage dans la condition *pseudo-mot* a un impact négatif sur les performances d'identification de la lettre cible. En revanche, la prise en compte d'un déplacement attentionnel permet au modèle de simuler les données observées lors de l'étude comportementale.

## 4.2 Simulation A08 : Effet de la durée de présentation en contexte non-mot

Avec cette simulation, nous cherchons à savoir si la présentation du contexte *non-mot* juste avant la présentation du stimulus complet produit une augmentation de performance dans la tâche d'identification de lettre.

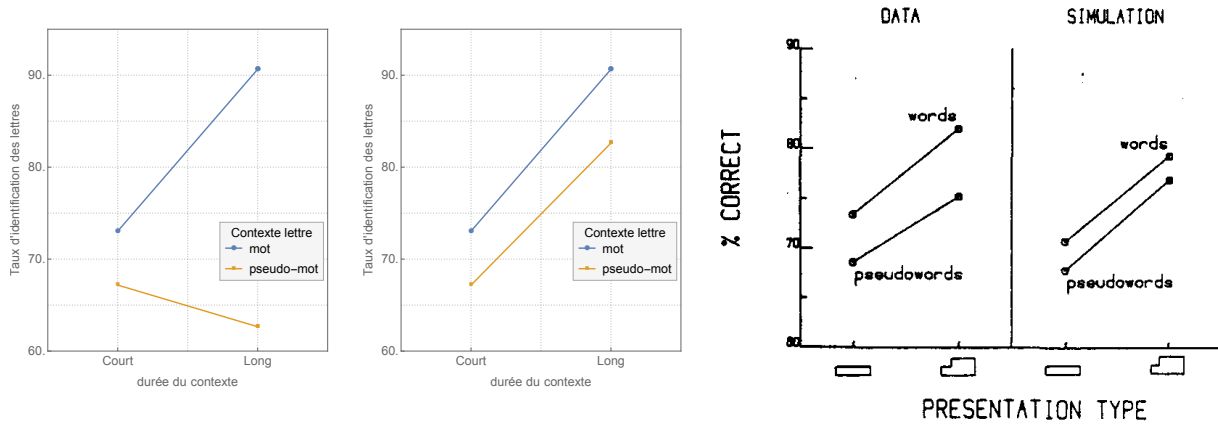


Figure 5.7 – **Simulation A07, Rumelhart et McClelland (1982), expérience 7, effet de la durée de présentation en contexte pseudo-mot.** À gauche, la probabilité d'identification de la lettre cible calculée par *BRAID* en fonction du temps de présentation du contexte, sans déplacement attentionnel (gauche) et avec déplacement attentionnel (droite). La courbe bleue (resp. jaune) correspond au contexte *mot* (resp. *pseudo-mot*). À droite, les résultats issus de l'étude de Rumelhart et McClelland (1982), résultats comportementaux et résultats de simulation.

#### 4.2.1 Expérience comportementale : Rumelhart et McClelland (1982, expérience 7)

Dans cette expérience, les auteurs comparent les performances d'identification de lettre dans deux conditions de contexte : *mot* et *non-mot*, et deux conditions de durée de présentation  $D$  et  $2D$ .

**Matériel et procédure** Les mots utilisés sont ceux de l'expérience A04. Les non-mots sont construits à partir des mots. Pour chaque mot de quatre lettres, on intervertit les deux premières lettres et les deux dernières lettres. Par exemple, le stimulus *mot* *WORD* devient *OWDR*. La durée de présentation des stimuli a été augmentée pour permettre d'obtenir des niveaux de performance pour les non-mots qui ne soient pas trop bas. La conséquence est que le niveau de performance pour les mots est supérieur à celui des expériences précédentes.

Les conditions expérimentales et la tâche réalisée sont similaires à celles de l'expérience A05. Les stimuli considérés sont de deux types (*non-mot* et *mot*). Dans la condition  $2D$ , le contexte de la lettre cible est présenté seul pendant une durée  $D$  (amorçage), puis il est remplacé par le stimulus complet pour une durée  $D$  supplémentaire. À l'issue de cette présentation, un masque est appliqué à la place du stimulus.

**Résultats comportementaux** Les résultats comportementaux montrent un effet significatif de la durée de présentation dans la condition *mot*. En revanche l'effet de durée de présentation dans la condition *non-mot* n'est pas significatif (voir Figure 5.8).

#### 4.2.2 *BRAID* : Simulation A08

Dans cette simulation nous formulons les mêmes hypothèses que dans la simulation A07, en nous intéressant cette fois au contexte *non-mot*.

**Matériel et procédure** Les mots utilisés sont les mêmes que dans la simulation A07. Les non-mots sont construits en suivant les directives de l'article, c'est-à-dire en intervertissant les deux premières lettres et les deux dernières lettres. Ainsi si l'ordre initial des lettres est 1-2-3-4, il devient 2-1-4-3.

La tâche simulée est la même que pour la simulation A07, avec les conditions de contexte *mot* et *non-mot*. Nous réalisons les simulations avec les mêmes paramètres que dans la simulation A07, et notamment nous gardons un temps de latence avant le déplacement attentionnel égal aux trois quarts de la durée de présentation du stimulus complet.

**Résultats des simulations** Nous avons gardé la même durée de calcul pour l'identification de la lettre cible, soit, en condition courte, 154 itérations et nous avons aligné les performances de *BRAID* et des participants sur la condition de contexte *mot*, et la durée de présentation *D*. Les résultats prédits par le modèle sont ceux des trois autres conditions. En l'absence de déplacement attentionnel, la présentation du contexte *non-mot* avant la présentation du stimulus complet (condition longue 2D) a un effet inhibiteur sur les performances de *BRAID* (Figure 5.8, à gauche). Pour les résultats de la simulation avec déplacement attentionnel, une analyse de la variance à mesures répétées montre que *BRAID* réussit à reproduire l'effet de durée de présentation en contexte *mot* [ $F(1,271) = 255,69, p < ,001, \eta^2 = 0,485$ ], l'effet de durée de présentation [ $F(1,271) = 74,91, p < ,001, \eta^2 = 0,217$ ] ainsi que l'effet d'interaction [ $F(1,271) = 83,37, p < ,001, \eta^2 = 0,235$ ].

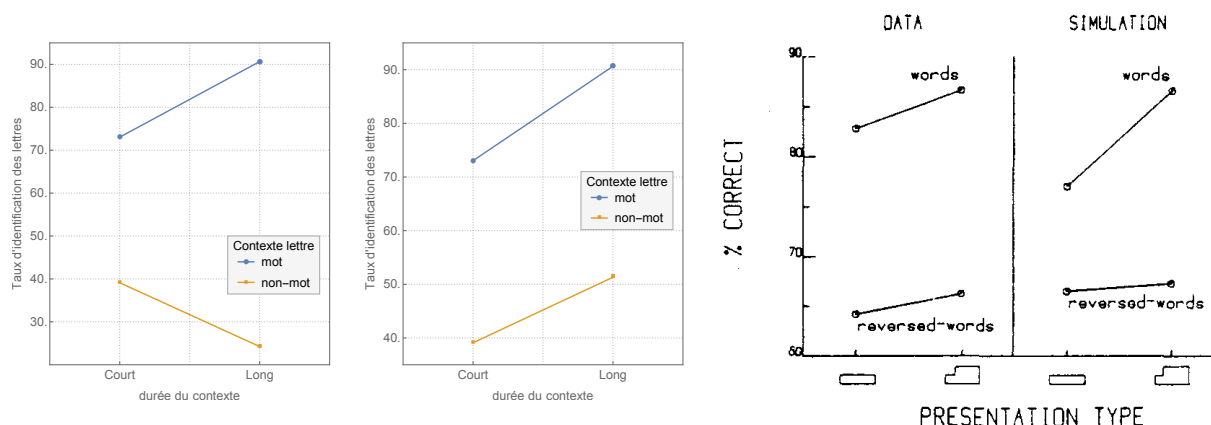


Figure 5.8 – **Simulation A08, Rumelhart et McClelland (1982), expérience 8, effet de la durée de présentation en contexte non-mot.** À gauche, la probabilité d'identification de la lettre cible calculée par *BRAID* en fonction du temps de présentation du contexte, sans déplacement attentionnel (gauche) et avec déplacement attentionnel (droite). La courbe bleue (resp. jaune) correspond au contexte *mot* (resp. *non-mot*). À droite, les résultats issus de l'étude de Rumelhart et McClelland (1982), résultats comportementaux et résultats de simulation.

**Conclusion** Comme dans l'étude comportementale, les résultats des simulations montrent un net avantage de la condition *mot* sur la condition *non-mot*. De même, les résultats confirment les hypothèses formulées en début d'étude. Sans prise en compte de l'attention, l'effet d'amorçage dans la condition *non-mot* a un impact négatif sur les performances d'identification de la lettre cible. En revanche, la prise en compte d'un déplacement attentionnel permet au modèle de simuler les données observées lors de l'étude comportementale.



### 4.3 Discussion

Ces deux simulations proposent des résultats cohérents avec les simulations précédentes. Dans la condition durée de présentation courte, on retrouve l'effet de supériorité du contexte *mot* sur le contexte *pseudo-mot* de la simulation A02, et l'effet de supériorité du contexte *mot* sur le contexte *non-mot* de la simulation A01. On retrouve également l'effet facilitateur produit par l'augmentation de la durée de présentation du contexte en condition *mot*, que l'on peut assimiler à un amorçage, et que l'on a déjà simulé précédemment (simulation A04 et A05).

Ce qui est nouveau dans ces deux simulations, c'est l'amorçage par un contexte *pseudo-mot*, simulation A07, ou par un contexte *non-mot*, simulation A08. L'observation expérimentale selon laquelle les participants obtiennent de meilleures performances dans la condition d'amorçage par un contexte *pseudo-mot*, est difficile à comprendre car cet amorçage ne peut pas favoriser la lettre cible. Rumelhart et McClelland (1982) font le même constat et proposent une solution basée sur la modification des paramètres de leur modèle. L'objectif de ces modifications est de limiter le rôle du contexte tout en sélectionnant les mots du lexique qui ont une influence sur l'identification de la lettre cible. La solution proposée par le modèle IA est difficile à interpréter et n'est soutenue, dans l'article, par aucun argument.

Dans *BRAID*, nous n'avons pas besoin d'imaginer un mécanisme qui contrôlerait quels mots sont mis à contribution pour identifier la lettre cible. C'est au moment de l'identification de la lettre cible que l'information lexicale, dans son ensemble, est mise à contribution par l'intermédiaire d'une marginalisation sur  $W^t$ . Ainsi, l'identification de la lettre cible repose à la fois sur l'information contenue au niveau lettre (distribution  $P_n^t$ ) et au niveau lexical (distribution  $W^t$ ). Or, dans le cas d'un amorçage avec un contexte *pseudo-mot*, si l'information contenue dans  $W^t$  n'est pas favorable à la lettre cible, et que les participants ont de meilleurs résultats, c'est que l'identification repose essentiellement sur la distribution  $P_n^t$ .

## 5 L'effet de supériorité du mot est-il robuste ?

L'effet de supériorité du mot est l'un des effets orthographiques les plus robustes de la littérature (Spoehr & Smith, 1975). Cependant, Massaro (1973) modifie les conditions de l'expérience de Reicher-Wheeler, et réussit à faire disparaître cet effet. Carr et al. (1978) font remarquer que Massaro (1973) avait modifié le protocole classique sur de nombreux points : les mots, les non-mots et les lettres cibles sont en nombre limité et sont connues des participants ; les lettres sont trois fois plus espacées que d'ordinaire ; la position de la lettre cible est toujours la même. Ces modifications rendent difficile l'identification des causes de la disparition de l'effet de supériorité des mots. Carr et al. (1978) rendent responsable l'espacement des lettres du stimulus, mais dans leur étude, ils modifient à la fois l'espacement des lettres et la prédictibilité de la lettre cible.

Pour identifier le rôle de ces deux causes dans la disparition de l'effet de supériorité des mots, nous nous intéressons tout d'abord au rôle de la prédictibilité de la position sur les performances d'identification, indépendamment de l'espacement, en simulant l'expérience de Johnston et McClelland (1974). Puis, nous simulons l'expérience de Carr et al. (1978) et discutons des causes possibles de la disparition de l'effet de supériorité des mots.

### 5.1 Simulation A09 : Effet de la prédictibilité de la position de la lettre cible

Dans cette simulation nous nous intéressons à l'impact de la prédictibilité de la position de la lettre cible, dans une tâche de Reicher-Wheeler. Nous cherchons à savoir si la connaissance préal-

able de la position de la lettre cible a la même influence sur les performances d'identification, en contexte *mot* et en contexte *non-mot*.

### 5.1.1 Expérience comportementale : Johnston et McClelland (1974)

Dans leur expérience, Johnston et McClelland (1974) manipulent l'attention pour modifier l'effet de supériorité des mots sur les non-mots. Ils proposent à des sujets de réaliser une tâche de Reicher-Wheeler dans deux conditions expérimentales. Dans une condition, les participants doivent déployer leur attention sur la séquence entière durant la présentation du stimulus. Ce n'est qu'au moment de l'affichage des deux alternatives qu'ils découvrent la position de la lettre cible. Dans l'autre condition, la position de la lettre cible dans la séquence est indiquée avant la présentation du stimulus. Les participants ont comme instruction de focaliser leur attention sur cette position. Les auteurs s'attendent à observer les mêmes performances en condition *mot* et *non-mot* dans la condition séquence indiquée.

**Matériel et procédure** Les stimuli sont de deux types : des mots de 4 lettres appariés (*COIN-JOIN*) ou des non-mots orthographiquement illégaux et non-prononçables (*CPRD-JPRD*). L'identité et la position des lettres cibles sont les mêmes pour les deux types d'items. Les deux conditions de présentation sont réparties par blocs de 20 stimuli, et le sujet est informé du type d'items (mot ou non-mot) avant chaque série.

Les participants sont soumis à une tâche de 2AFC avec masquage pré et post stimulus. Pour la condition de contexte, deux groupes sont formés, un passe la condition *mot* et l'autre la condition *non-mot*. Chaque participant est soumis aux deux conditions d'attention (*séquence entière* et *séquence indiquée*). Dans la première condition d'attention, les participants fixent le milieu du stimulus et doivent le traiter comme un tout lors de l'identification (condition *séquence entière*). Dans la seconde condition, les participants doivent fixer l'indice de position durant la phase de pré masquage et ensuite essayer de ne traiter que la lettre indiquée (condition *séquence indiquée*).

**Résultats comportementaux** Les performances liées aux contextes ne sont pas comparables, car elles ont été obtenues pour des durées de présentation des stimuli différentes. Les auteurs ne s'intéressent donc qu'aux effets simples de la condition d'attention. Lorsque les stimuli sont des mots, ils observent un effet significatif de l'attention avec un gain de performance de (+6,7%) pour la condition *séquence entière* comparé à la condition *séquence indiquée*. Cette différence est significative tant par sujets que par items. Dans la condition de contexte *non-mot*, l'effet simple de l'attention s'inverse au profit de la condition *séquence indiquée* avec un gain de performance de (+6,5%) par rapport à la condition *séquence entière*. Là encore l'effet est significatif par items et par sujets.

### 5.1.2 BRAID : Simulation A09

Pour simuler dans *BRAID* la prédictibilité de la lettre cible, nous déplaçons dans le modèle la position du regard et de l'attention, de sorte qu'elle coïncide avec la position de la lettre cible. L'accélération de la convergence de  $P_n^t$  associée à la lettre cible devrait rapidement rendre inutile l'apport du niveau lexical  $W^t$ .

**Matériel et procédure** Nous avons repris les mots et les non-mots de la simulation A01. Nous avons simulé une tâche d'identification de lettre, en suivant la méthodologie décrite dans

la section 1 de ce chapitre. Les résultats de la condition *séquence entière* sont ceux déjà calculés lors de la simulation A01, pour les conditions *mot* et *non-mot*. Nous les avons donc repris. Pour simuler la condition *séquence indicée*, nous avons utilisé les paramètres par défaut, et modifié simplement la position du regard et la moyenne de la distribution de l'attention de sorte que ces deux paramètres coïncident avec la position de la lettre cible.

**Résultats des simulations** Nous présentons ici deux séries de résultats, ceux obtenus en condition *mot* présentés Figure 5.9 (en haut) et ceux obtenus en condition *non-mot* présentés Figure 5.9 (en bas). Ces résultats ont tous été obtenus après 68 itérations pour la condition *mot* et 188 itérations dans la condition *non-mot*. Nous avons aligné les performances de *BRAID* et des participants sur la condition *séquence entière*, pour les deux conditions de contexte. Les résultats prédits par le modèle sont ceux de la condition *séquence indicée*.

Une analyse de la variance pour mesures répétées a montré que, dans la condition *mot*, les lettres sont mieux reconnues dans la condition séquence entière que dans la condition indicée [ $t(271) = 14,93, p < .001$ ]. On voit que durant les premières itérations, la tendance est inversée, et de manière significative. Par exemple, à l'itération 40 on note un avantage de la condition lettre [ $t(271) = -11,72, p < .001$ ]. Toutefois, les performances en identification de lettres sont faibles lors des premières itérations et bien en deçà de celles des sujets. Les résultats pour la condition *non-mot* montre que les lettres sont mieux reconnues dans la condition indicée, plutôt que dans la condition *séquence entière* [ $t(271) = -15,99, p < .001$ ].

**Conclusion** Les résultats des simulations reproduisent les effets inversés observés dans l'étude comportementale, entre les mots et les non-mots. L'étude du décours temporel de la tâche permet de visualiser l'apport du niveau lexical présent en condition *mot* uniquement.

## 5.2 Simulation A10 : Effet d'espacement des lettres

L'augmentation de l'espacement entre les lettres du stimulus, dans une tâche de Reicher-Wheeler, pourrait conduire à une disparition de l'effet de supériorité du mot. C'est une des deux pistes qu'explore l'étude de Carr et al. (1976), avec la prédictibilité de la position de la lettre cible. Dans cette simulation, nous évaluons l'hypothèse de l'espacement des lettres dans deux conditions de prédictibilité, pour rendre compte de la disparition complète de l'effet de supériorité du mot observé dans l'étude de Massaro (1973).

### 5.2.1 Expériences comportementales : Carr et al. (1976) ; Massaro (1973)

Dans son expérience, Massaro (1973) propose à des participants une tâche de Reicher-Wheeler dont le protocole est modifié. Tout d'abord, la lettre à identifier est toujours en position centrale. De plus, les lettres qui sont susceptibles d'apparaître sont connues à l'avance par les participants. Enfin l'espacement entre les lettres est de  $3.2^\circ$  d'angle horizontal, contre environ  $1^\circ$  d'angle habituellement. Dans leur étude, Carr et al. (1976) suggèrent que la disparition de l'effet de supériorité du mot sur le non-mot ne serait due qu'à l'espacement augmenté entre les trois lettres du stimulus.

**Matériel et procédure** Les deux expériences utilisent le même paradigme expérimental (Carr et al., 1976 ; Massaro, 1973), qui consiste en une tâche 2AFC dans quatre conditions de contexte : *mot* (*AGE*), *non-mot* (*SGL*), *lettre isolée* (*\_G\_*) et *lettre entourée du symbole %* (*%G%*). Massaro (1973) n'utilise que quatre mots de trois lettres (*ACE, AGE, APE, ARE*)

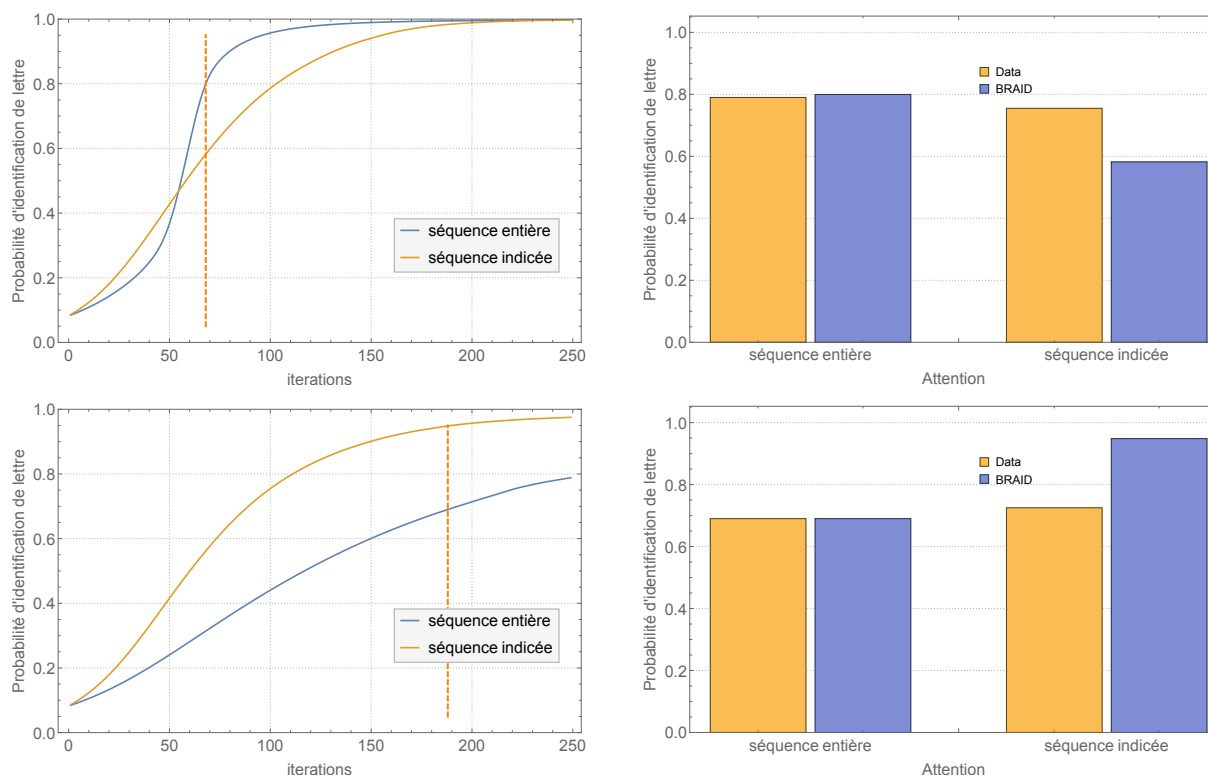


Figure 5.9 – **Simulation A09, Johnston et McClelland (1974), stimuli mots, effet de la prédictibilité de la position de la lettre cible.** Résultats des simulations, en haut la condition *mot*, en bas la condition *non-mot*. Pour chaque ligne, à gauche on trouve le déroulement temporel de la reconnaissance de lettre pour les deux conditions d’attention. En pointillés rouges, l’itération à laquelle sont extraits les résultats qui sont comparés aux données expérimentales. ; à droite, on trouve la comparaison des données expérimentales de Johnston et McClelland (1974) (en jaune) avec les résultats des calculs de *BRAID* (en bleu), pour chacune des deux conditions d’attention.

et quatre lettres cibles (*C*, *G*, *P*, *R*), la lettre cible apparaissant toujours en position centrale. Carr et al. (1976) utilisent les mêmes lettres cibles, mais dans leur expérience, la lettre cible peut apparaître indifféremment dans chacune des trois positions. Ils utilisent donc 12 mots, les 4 mots présents chez Massaro (1973) auxquels ils ajoutent huit autres mots permettant ainsi à la lettre cible d’apparaître dans les deux autres positions (exemple pour la lettre *P* : *POT*, *APE*, *TAP*). Les participants doivent mémoriser la liste des lettres cibles qu’ils devront reconnaître, et sont informés de la nature des stimuli avant chaque bloc expérimental. Enfin l’espacement entre les lettres est ramené à  $1^\circ$  d’angle horizontal, de sorte que les seules différences entre les deux expériences sont la position de la lettre cible et l’espacement entre les lettres du stimulus.

**Résultats comportementaux** Les résultats de Carr et al. (1976) montrent un effet principal du contexte correspondant à l’effet classique de supériorité du mot : les lettres cibles sont identifiées plus précisément en contexte *mot* qu’en contexte *non-mot*, *lettre isolée* ou *lettre entourée du symbole %*. Les résultats de Massaro (1973) en condition d’espacement augmenté et avec un positionnement central de la lettre cible ne montrent aucun effet de supériorité du mot.

### 5.2.2 *BRAID* : Simulation A10

Pour mieux comprendre l'impact des modifications apportées par Carr et al. (1976) sur l'expérience de Massaro (1973), nous réalisons deux simulations. Dans la première, nous étudions le rôle spécifique joué par l'espacement des lettres du stimulus. Dans la seconde, nous ajoutons un déplacement attentionnel pour simuler la prédictibilité de la position de la lettre cible de l'expérience de Massaro (1973).

**Matériel et procédure** Concernant les stimuli, nous constatons que ces deux expériences contiennent très peu de stimuli. Or, pour un même stimulus, *BRAID* a un comportement déterministe, et ne génère donc pas de variabilité intra-essai. Cela ne nous permet pas d'avoir des résultats satisfaisants. Nous avons donc repris le matériel de la simulation A01. Ainsi, nos stimuli sont composés de 4 lettres et non de 3 comme dans les études comportementales. Nous n'avons utilisé que les conditions *mot*, *non-mot* et *lettre isolée* puisque le %, du contexte *lettre entourée du symbole %*, n'est pas un caractère connu du modèle.

Pour simuler l'espacement des lettres du stimulus, nous avons modifié deux paramètres du modèle : la pente de l'acuité ( $Slope_G$ ) et l'interférence latérale ( $\theta_I$ ). La valeur de la pente de l'acuité passe de 1 à 2. Cette augmentation permet de simuler le fait que les lettres excentrées sont perçues de manière plus périphérique et donc avec moins d'acuité. Par ailleurs, l'inhibition latérale passe de 16,25% à 2,5%. Cette diminution de l'interférence latérale ne laisse qu'une influence marginale des deux lettres voisines durant le processus de perception.

**Résultats des simulations** Pour analyser les résultats, nous nous basons essentiellement sur les courbes de décours temporel moyen de l'identification des lettres cibles. Dans la condition *lettres espacées*, on observe un effet robuste de la condition *mot* sur les deux autres conditions, à partir de l'itération 107 ; le modèle est significativement plus performant en *mot* qu'en contexte *lettre isolée* [ $t(271) = 7,6, p < 0,001$ ] ; le modèle est significativement plus performant en *mot* qu'en contexte *non-mot* [ $t(271) = 6.7, p < 0,001$ ].

Si maintenant nous nous intéressons à la simulation qui prend en compte non seulement l'espacement des lettres mais également le déplacement attentionnel induit par la prédictibilité de la position de la lettre cible, alors aucun résultat n'est significatif sur toute la plage d'itérations considérée [ $F < 1$ ].

**Conclusion** Ces deux simulations montrent que les modifications apportées aux paramètres du modèle afin de simuler un espacement des lettres retardent l'apparition de l'effet de supériorité du mot, comparativement aux résultats de la simulation A01. Cependant, cette manipulation ne suffit pas à le faire disparaître. C'est bien l'action combinée de l'espacement et de la prévisibilité de la position de la lettre cible qui permet de supprimer cet effet de supériorité du mot.

## 5.3 Discussion

Les résultats des deux simulations décrites précédemment sont cohérents avec les résultats comportementaux. Dans la Simulation A09, le déplacement du regard et de l'attention sur la lettre cible ne produit pas le même effet suivant la nature du stimulus : il est facilitateur pour les mots et inhibiteur pour les non-mots. Dans la Simulation A10, on montre que l'augmentation de l'espacement des lettres retarde l'apparition de l'effet de supériorité du mot, comparativement

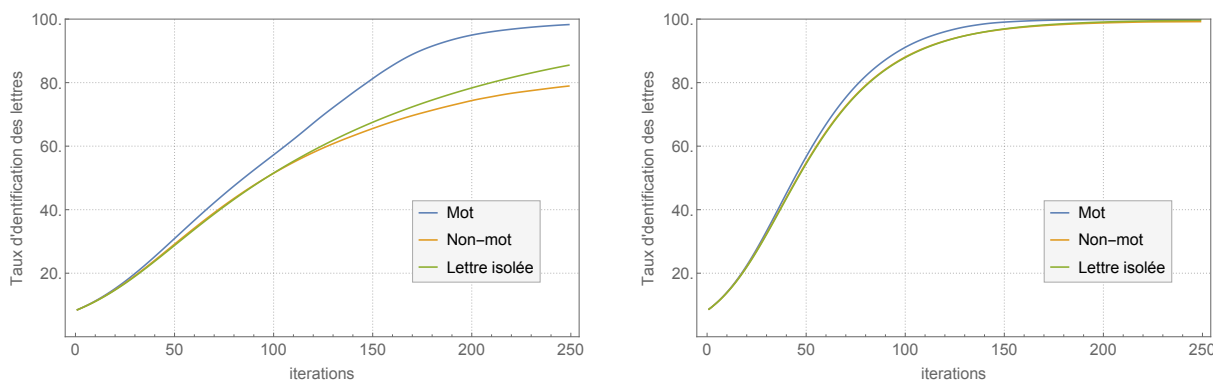


Figure 5.10 – **Simulation A10, Carr et al. (1976) ; Massaro (1973), stimuli mots, effet d'éloignement des lettres.** À gauche, simulation du déroulement temporel de la reconnaissance de lettre en condition de *lettres espacées*, pour les trois conditions de contexte. À droite, simulation du déroulement temporel de la reconnaissance de lettre en condition de *lettres espacées* et de décalage attentionnel, pour les trois conditions de contexte

aux résultats de la simulation A01. Cependant, cette manipulation ne suffit pas à le faire disparaître.

On peut observer l'apport du déplacement du regard et de l'attention, dans la Simulation A09, en comparant les performances obtenues en condition *séquence entière* avec celle obtenues en condition *séquence indicée*. Si on regarde les courbes de déroulement temporel Figure 5.9, en condition de contexte *mot*, on voit que l'avantage de la condition *séquence indicée* apparaît durant les premières itérations. Mais la probabilité d'identification du mot au niveau lexical devient rapidement suffisante pour aider à identifier la lettre cible si bien que l'avantage apporté par l'indigage disparaît. En revanche, lorsque le stimulus est un non-mot, cette probabilité n'est pas suffisante pour aider à identifier la lettre cible, si bien que l'avantage fourni par l'indigage persiste. Cela permet de mieux comprendre l'inversion observée par Johnston et McClelland (1974). Maintenant, si l'on compare les performances simulées obtenues dans la condition *séquence indicée* pour les deux contextes *mot* et *non-mot*, on constate un avantage pour la condition *mot* par rapport à la condition *non-mot*, visible à partir de l'itération 100 et persistant jusqu'à la dernière itération.

Avec la Simulation A10, on observe que l'espacement des lettres à lui seul ralentit l'effet de supériorité du mot induit par le niveau lexical comparativement à la condition sans espacement, Simulation A01. En revanche, si l'on cumule espacement et déplacement du regard et de l'attention, l'apport du niveau lexical n'est plus suffisant pour faire une différence entre les contextes *mot*, *non-mot* et *lettre isolée*. En conclusion, l'effet de supériorité du mot est supprimé par l'action conjointe de l'écartement des lettres et de la prédictibilité de la position de la lettre cible.

## 6 Discussion sur la perceptibilité des lettres

Dans ce chapitre, nous avons présenté dix simulations couvrant les principaux résultats de la littérature sur la perceptibilité des lettres. Les trois premières simulations ont eu pour objectif de reproduire les résultats fondamentaux du domaine. La simulation A01 reproduit les effets classiques de supériorité du mot sur la lettre isolée et sur le non-mot. Elle met également

en évidence la robustesse, dans *BRAID*, de cet effet, grâce aux courbes dynamiques<sup>12</sup>. La simulation A02 reproduit l'effet classique de supériorité du pseudo-mot sur le non-mot, et met en évidence un effet de supériorité du mot sur le pseudo-mot. Enfin avec la simulation A03, nous montrons que ces effets sont dus à l'activité induite par le stimulus au niveau lexical, et non à un mécanisme explicite de vérification de la régularité orthographique du stimulus.

Nous nous sommes intéressés ensuite au rôle que joue le lexique dans les performances d'identification des lettres. Pour cela nous avons fait varier la nature du contexte et son temps de présentation. Les trois simulations suivantes se sont focalisées sur le contexte *mot*. Avec la simulation A04, nous montrons que l'augmentation du temps de présentation du contexte améliore la performance de reconnaissance de la lettre cible. Avec la simulation A05, nous montrons que l'augmentation de la durée de présentation du contexte comme l'augmentation de la durée de présentation de la lettre elle-même améliorent la performance, et que ces effets sont additifs. Enfin, avec la simulation A06, nous montrons que la performance d'identification de la lettre cible augmente linéairement avec le nombre de lettres du contexte.

Les deux simulations qui suivent se placent dans les conditions de la simulation A05 en étendant l'exploration aux contextes *pseudo-mot* (simulation A07) et *non-mot* (simulation A08). Nous montrons que l'effet facilitateur de la durée de présentation du contexte observé sur les mots ne s'étend pas aux deux autres types d'items contrairement aux données comportementales. Néanmoins les résultats comportementaux sont correctement simulés lorsqu'on ajoute un déplacement attentionnel sans déplacement du regard, durant la présentation du stimulus, après l'apparition de la lettre cible.

Enfin, les deux dernières simulations apportent un éclairage nouveau sur deux expériences comportementales qui ont mis en défaut l'effet de supériorité du mot et qui ne sont simulées, à notre connaissance, par aucun autre modèle. Dans la simulation A09, nous montrons que le fait de connaître par avance la position de la lettre cible, donne un avantage en condition *non-mot*, mais pas en condition *mot*. Dans la simulation A10, nous montrons que le fait d'éloigner les lettres les unes des autres diminue l'activité lexicale et de ce fait, l'effet de supériorité des mots.

Globalement *BRAID* s'est avéré capable de simuler non seulement les effets classiques de supériorité du mot et du pseudo-mot, mais également les effets plus subtils de contexte sur la perceptibilité des lettres. Il est également possible de simuler dans *BRAID* les conditions de disparition de l'effet de supériorité du mot et donc de mieux comprendre quels sont les mécanismes à l'origine de cet effet. La plupart des effets décrits dans la littérature ont été retrouvés dans les simulations, mais les analyses ont été menées pour chaque expérience indépendamment. Notre objectif est à présent d'évaluer plus globalement l'adéquation entre données simulées et données comportementales.

**Analyse globale des résultats** Dans toutes les simulations de ce chapitre, nous avons utilisé *BRAID* pour identifier une lettre, pour une position donnée, dans un contexte donné. Le modèle qui simule la dynamique complète du mécanisme calcule la probabilité que la lettre soit identifiée à chaque pas de temps (itération). On obtient ainsi le décours temporel de la probabilité d'identification de la lettre cible. Pour comparer les résultats de *BRAID* aux données comportementales, nous avons dû fixer une itération, et analyser les performances du modèle à cette itération, pour chaque mot et pour chaque condition expérimentale (CE).

Nous avons donc fixé une itération de référence propre à chaque simulation. Cela a du sens dans la mesure où la durée de présentation des stimuli varie d'une expérience comporte-

---

<sup>12</sup>On dit que l'effet est robuste dans le sens où, à partir du moment où il est observé, il reste présent jusqu'à ce que les performances atteignent une valeur plafond, indépendamment de la condition, ce qui fait disparaître les effets statistiques.

mentale à l'autre, et même d'un participant à l'autre. Pour fixer cette itération, nous avons calculé un décours temporel moyen pour chaque condition de la simulation. Puis, nous avons choisi l'itération qui fait coïncider les résultats expérimentaux et les résultats de la simulation dans une condition choisie a priori. Par exemple, dans la simulation A01, c'est la condition de contexte *mot* qui est choisie pour fixer l'itération de référence qui sera ensuite retenue pour toutes les autres conditions. Ainsi, les performances simulées pour la condition de référence ont été calibrées mais les résultats obtenus pour la même itération dans les autres conditions expérimentales sont uniquement prédites par le modèle.

Nous nous sommes basés sur ces résultats prédits pour évaluer, de manière plus globale, le pouvoir prédictif de *BRAID*. Pour cela, nous regroupons les résultats prédits des neuf premières simulations. Nous excluons la dernière expérience (simulation A10) de ce chapitre car nous n'avons pas de données comportementales associées. Nous déterminons la part de variance expliquée par *BRAID* sur l'ensemble des résultats prédits. Nous faisons une régression linéaire avec comme facteur prédictif les résultats des simulations et comme facteur dépendant les données comportementales associées. Les résultats de l'analyse de régression sont représentés Figure 5.11. Le résultat principal est un  $R^2 = 0,58$ , ce qui est plutôt satisfaisant au vu de la diversité des conditions expérimentales regroupées dans cette analyse. Cela montre que l'ensemble de nos simulations a une cohérence d'ensemble, et que *BRAID* est capable de capturer et reproduire une grande variété d'effets liés à la perceptibilité des lettres.

De plus, nous observons deux points qui semblent être atypiques relativement à la tendance générale : ils proviennent de la simulation A01 en condition *non-mot* et *lettre isolée*. Les faibles valeurs observées proviennent du calibrage de l'expérience comportementale qui donne un taux de succès en condition *mot* aux environs de 60% là où les autres expériences sont calibrées pour un taux de succès de l'ordre de 75%. Par comparaison, une autre simulation utilise à la fois des contextes *mot* et *non-mot*, c'est la simulation A08. Pour obtenir des performances satisfaisantes en condition *non-mot*, les auteurs ont calibré l'expérience pour obtenir un taux de succès en condition *mot* de l'ordre de 85%. Donc, bien que ces points proviennent d'une expérience avec un calibrage différent, il n'y a pas de raison de les exclure de l'étude.

**Les modèles concurrents et la perceptibilité des lettres** D'autres modèles computationnels se sont intéressés à la perceptibilité des lettres. C'est le cas du modèle IA (McClelland & Rumelhart, 1981 ; Rumelhart & McClelland, 1982) dont nous avons repris une partie des expériences. Mais on peut citer également le modèle AV (Paap et al., 1982) et plus récemment, le modèle DROM (Grainger & Jacobs, 1994). Ces trois modèles computationnels font des hypothèses différentes pour expliquer l'effet de supériorité du mot.

Pour le modèle IA (McClelland & Rumelhart, 1981), l'avantage observé en contextes *mot* et *pseudo-mot* s'explique par une rétro-activation du niveau *mot* vers le niveau *lettre*, tout au long de la phase de perception. L'avantage produit est proportionnel à l'activation globale du niveau mot. La décision concernant la lettre cible repose uniquement sur l'information contenue au niveau *lettre*. L'activité est transformée en probabilité sur les lettres, puis un tirage est réalisé (c'est la lettre choisie pour le report libre). Si la lettre sélectionnée est l'une des deux lettres proposées dans la simulation, alors le modèle répond directement. Dans le cas contraire, le choix de la lettre se fait au hasard parmi les deux candidats.

Le modèle AV (Paap et al., 1982) fait l'hypothèse d'une sélection lexicale durant la phase de perception, basée sur la similarité du mot avec le stimulus. Ainsi, au moment de la décision, le modèle dispose d'un ensemble de mots susceptibles d'être le stimulus présenté (niveau lexical) et d'une distribution de probabilité sur la lettre cible (niveau alphabetum). La réponse se fait en combinant l'information du niveau lexical avec celle de l'alphabetum comme décrit par



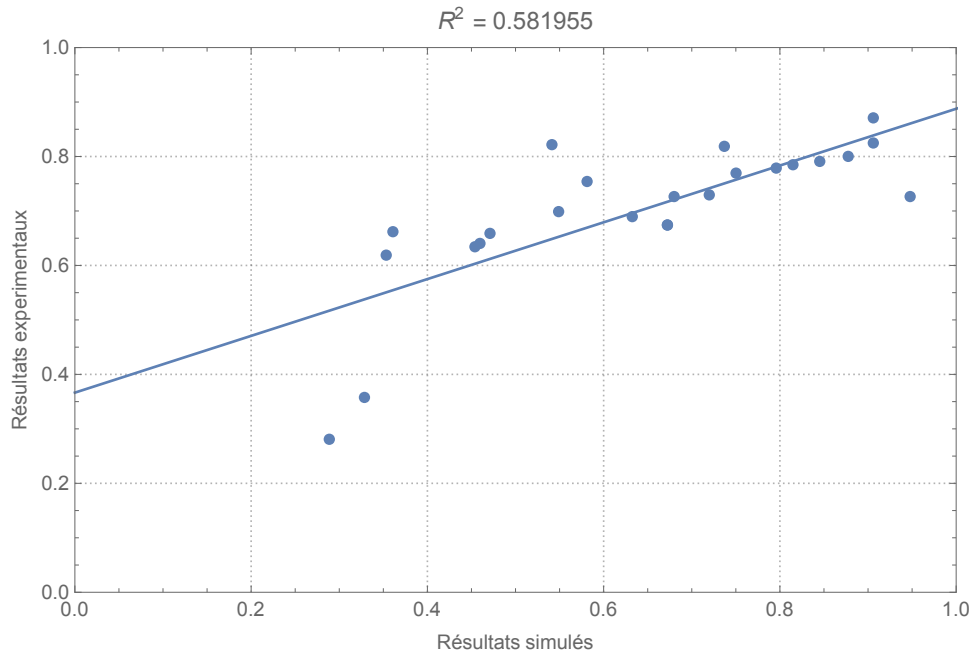


Figure 5.11 – **Corrélation entre les résultats comportementaux et les résultats simulés.** Graphique de corrélation entre les performances de reconnaissance de lettre prédites par *BRAID* (en abscisse) dans les neuf premières simulations, et les résultats comportementaux correspondants (en ordonnée).

l'équation suivante :  $PPC = P(L)P(C|L) + (1 - P(L))P(C|A)$ , où  $PPC$  est la probabilité de répondre correctement,  $P(L)$  est la probabilité de répondre à partir du lexique (proportionnelle à l'activation créée par le stimulus),  $P(C|L)$  la probabilité de répondre correctement à partir du lexique et  $P(C|A)$  est la probabilité de la lettre considérée dans l'alphabetum. Ce processus de décision rend compte de l'avantage du contexte *mot* sur le contexte *pseudo-mot* grâce à  $P(L)$ . Enfin, pour le contexte *non-mot*,  $P(L)$  est petit et la réponse repose essentiellement sur  $P(C|A)$ .

Le modèle DROM (Grainger & Jacobs, 1994) propose une troisième explication. Il reprend la structure du modèle IA avec la rétro-activation du niveau mot vers le niveau lettre, durant la phase de perception. Les auteurs montrent que cette rétro-action n'est pas utilisée dans le modèle et que, pour rendre compte de l'effet de supériorité du mot, il suffit de modifier le processus de décision en ajoutant un critère basé sur l'activation du mot le mieux reconnu au niveau lexical. La réponse à la tâche de choix forcé dépend de la réponse de report libre calculée comme suit :  $P(LR) = P(L) + P(W) - P(L).P(W)$ , où  $P(LR)$  est la probabilité de report libre d'une lettre,  $P(L)$  est la probabilité de la lettre et  $P(W)$  est la probabilité du mot qui a la plus grande activation. La réponse est calculée par l'équation :  $P(FC) = P(LR) + 0.5(1 - P(LR))$  où  $P(FC)$  est la probabilité du choix forcé. Lorsque la lettre est présentée en contexte *non-mot*, les auteurs fixent  $P(W) = 0$ , ce qui fait que le choix de la lettre repose uniquement sur le niveau *lettre*.

En conclusion, ces modèles répondent à la tâche de Reicher-Wheeler de manières différentes. Le modèle IA prend en compte les deux lettres, qui sont proposées comme alternatives, dans sa réponse. Le modèle réalise un tirage suivant la distribution de la lettre considérée. Si la lettre sélectionnée n'est pas une des deux alternatives, un second tirage est réalisé entre les deux alternatives. Le modèle DROM construit sa réponse en se basant uniquement sur la probabilité du report libre de la bonne réponse. Comme dans le modèle IA, il ajoute un second

Tableau 5.1 – **Effets dont les différents modèles présentés dans ce chapitre ont rendu compte dans la littérature.** Le symbole  $\checkmark$  signifie que le modèle rend compte de l’effet considéré. Le symbole  $\sim$  signifie que le modèle rend compte de l’effet considéré après manipulation de ses paramètres.

Simulation	Effet	Modèle			
		IA	AV	DROM	<i>BRAID</i>
A01	Effet de supériorité du mot	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
A02	Effet de supériorité du pseudo-mot	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
A03	Effet du contexte non orthographiquement légal	$\checkmark$			$\checkmark$
A04	Effet de la durée du présentation du contexte	$\checkmark$			$\checkmark$
A05	Effets joints de la cible et du contexte	$\checkmark$			$\checkmark$
A06	Effet du nombre de lettres du contexte	$\checkmark$			$\checkmark$
A07	Effet de la durée de présentation en contexte pseudo-mot	$\sim$			$\checkmark$
A08	Effet de la durée de présentation en contexte non-mot	$\sim$			$\checkmark$
A09	Effet de la prédictibilité de la position de la lettre cible				$\checkmark$
A10	Effet d’éloignement des lettres				$\checkmark$

tirage en cas d’échec. Le fait de ne pas prendre en compte explicitement la probabilité de la réponse alternative a pour effet de surestimer la probabilité de répondre correctement. Enfin, le modèle AV produit une réponse construite en mélangeant l’information en provenance du lexique avec l’information venant directement de l’alphabetum. Un récapitulatif des différentes tâches simulées et reproduites par ces différents modèles est présenté dans le Tableau 5.1.

**La réponse de *BRAID* à une tâche de Reicher-Wheeler** Avec le modèle *BRAID*, nous proposons un mécanisme alternatif à ceux des trois modèles précédents, IA, AV et DROM. En effet, dans notre implémentation, les rétro-propagation des informations lexicales sur l’identité des lettres n’est effectuée qu’en une passe, en fin de processus (voir Chapitre 4, Section 1.3 pour la description mathématique de l’inférence, et voir Chapitre 7, Section 2.4 pour une discussion de l’interprétation de ce choix). La perceptibilité des lettres repose donc sur l’information accumulée dans les distributions du niveau lettre  $P_n^t$ , et du niveau mot  $W^t$ , et sur la dynamique relative des processus d’accumulation d’information dans ces distributions.

***BRAID*, description du processus de perception** Dans *BRAID*, la tâche d’identification de lettres met en œuvre deux processus dynamiques. Le premier processus extrait les caractéristiques de chaque lettre du stimulus  $S_n^t$ , alimentant une représentation volatile temporellement,  $I_n^t$ . Celle-ci alimente une représentation plus stable temporellement,  $P(P_n^t)$ . La vitesse d’évolution de ces distributions dépend des lettres, car certaines sont mieux perçues que d’autres, du fait de leur excentricité et des interférences avec les lettres adjacentes. De plus, l’évolution des distributions sur  $P_n^t$  à partir des informations contenues dans les distributions sur  $I_n^t$  a aussi une vitesse qui est modulée par l’attention. Plus les variables  $P_n^t$  sont proches du point de fixation de l’attention, plus elle accumulent rapidement l’information. On voit, dans la Figure 5.12, le décours temporel des distributions sur  $P_n^t$  pour chaque lettre du stimulus *FORK*.

Le second processus a pour but de faire évoluer la distribution du niveau lexical, c’est-à-dire sur la variable  $W^t$ . Avant la présentation du stimulus, la probabilité des mots du lexique est

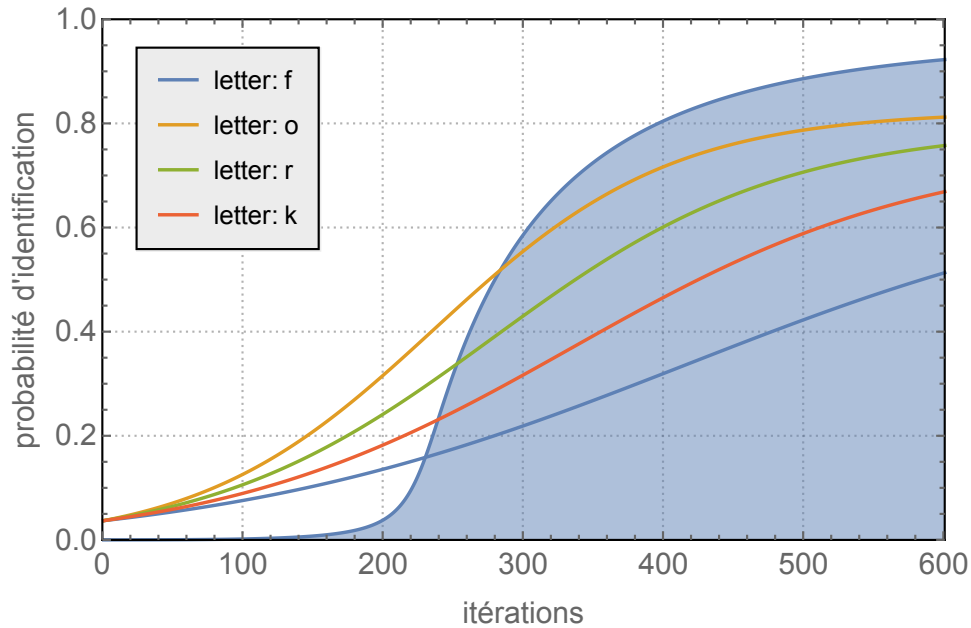


Figure 5.12 – **Relation lettres-mot dans la tâche d'identification de lettre.** La courbe pleine (bleue) présente le décours temporel de l'identification du mot *FORK*. Les quatre autres courbes représentent le décours temporel de l'identification de chacune des lettres du mot. L'attention est portée au milieu du mot, et les lettres centrales accumulent l'information plus rapidement que les lettres extérieures.

proportionnelle à leur fréquence. A chaque itération du processus d'identification de la lettre cible, la modification de la distribution sur  $P_n^t$  induit une modification de la distribution sur  $W^t$ . La probabilité d'un mot de  $W^t$  augmente proportionnellement à sa similarité avec le percept  $P_n^t$  et à la quantité d'information contenue dans la distribution sur  $P_n^t$ . Ainsi, les mots qui partagent des lettres avec le stimulus ont leur probabilité qui augmente. De plus, comme la similarité entre le mot du lexique et le stimulus est calculée comme le produit des similarités des lettres, deux à deux, plus le mot partage de lettres avec le stimulus, plus sa probabilité augmente. De même, si la distribution sur  $P_n^t$  est peu informée (proche de l'uniforme), la similarité entre le mot du lexique et le stimulus sera faible, et la probabilité du mot du lexique sera peu modifiée. On voit, dans la Figure 5.12 (courbe pleine), le décours temporel de la probabilité du mot  $W^t = \text{FORK}$  lorsque le stimulus est *FORK*. On observe bien, pour les premières itérations, une évolution lente de cette probabilité, due au fait que les distributions sur les lettres  $P_n^t$  n'ont pas encore accumulé suffisamment d'information.

On voit sur la Figure 5.13 en orange, le décours temporel de  $P_n^t$  associé à la lettre cible, en bleu le décours temporel de  $W^t$  associé au mot-stimulus. En vert, la réponse produite par *BRAID* construite en marginalisant sur  $W^t$ . Notre réponse n'est pas exactement la réponse à la tâche de Reicher-Wheeler, qui ne devrait pas s'appuyer sur  $P(P_n^t)$  mais seulement sur la probabilité de la lettre cible, la probabilité de son alternative et la normalisation de ces deux probabilités. En omettant le facteur de normalisation, nous sous-estimons la probabilité de répondre correctement. Cependant, la mise à l'échelle compense en moyenne cette sous-estimation. Enfin du point de vue du modélisateur, le fait de travailler directement sur  $P(P_n^t)$  nous en apprend plus sur le fonctionnement de notre modèle.

En conclusion, on voit que l'effet de supériorité du mot en contexte *mot* dépend de la vitesse à laquelle la distribution du percept est construite, mais également de la vitesse à laquelle la distribution des mots converge. Lorsque la lettre est facilement discriminable par rapport aux

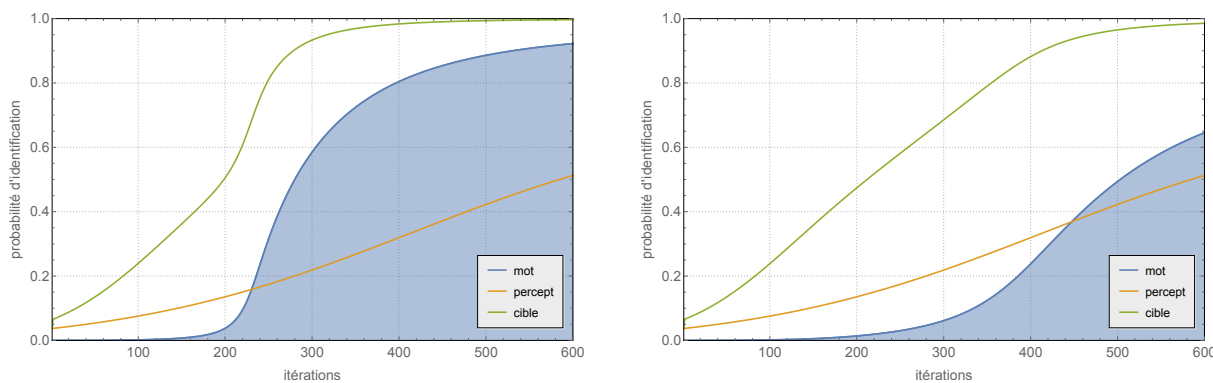


Figure 5.13 – **Influence de  $Leak_W$  sur la tâche 2AFC.** Ces deux figures représentent l'influence du lexique en fonction de la valeur du paramètre de déclin  $Leak_W$  (à gauche,  $Leak_W = 250$ , à droite pour  $Leak_W = 1250$ ). les courbes pleines (bleue) présentent le déroulement temporel de l'identification du mot *FORK*. Les courbes jaunes représente le déroulement temporel de  $P(P_1^t = "F")$  et en vers la probabilité de faire la bonne réponse.

autres lettres de la matrice, comme c'est le cas pour la lettre *A* dans *BRAID*, l'avantage produit par l'identification du mot est assez faible. A contrario, lorsque la lettre est un *F*, qui est difficile à discriminer des lettres *E* ou *P*, l'avantage produit par l'identification du mot est quasi immédiat, comme on le constate dans l'exemple de la Figure 5.13.

Un raisonnement similaire sur les mots devrait amener à prédire un effet de fréquence des mots sur la perceptibilité des lettres, les mots très fréquents devraient produire un avantage supérieur aux mots peu fréquents, la fréquence se traduisant dans *BRAID* par l'état au repos de la distribution  $P(W^t)$ . A noter que Johnston (1978) a observé un effet de fréquence du stimulus sur la perceptibilité des lettres mais en report libre seulement, alors que *BRAID* prédit des effets similaires en report libre et dans la tâche 2AFC. Des expériences nouvelles devront être menées pour valider cette prédiction. De la même façon, *BRAID* prédit qu'un mot-stimulus ayant de nombreux voisins ou ayant un voisin très fréquent ne contenant pas la lettre cible, devrait produire peu d'avantage pour l'identification de la lettre cible.

**L'influence du déclin de  $W^t$  dans l'identification des lettres** D'autres paramètres peuvent moduler l'effet de supériorité du mot. C'est le cas du paramètre de déclin  $Leak_W$  qui intervient dans la construction de la distribution  $P(W^t)$ . Ce paramètre contrôle la vitesse à laquelle la distribution  $P(W^t)$  retrouve son état d'équilibre en l'absence de stimulus et limite la vitesse à laquelle elle se construit. Plus sa valeur est grande et plus la distribution  $P(W^t)$  se construit lentement. Nous pensons que cette valeur est sous le contrôle cognitif du participant et qu'il adapte sa valeur en fonction de la tâche à réaliser. Dans cette thèse, nous n'utilisons que deux valeurs pour  $Leak_W$ . Pour les tâches d'identification de lettre, nous pensons que le participant a intérêt à utiliser au plus vite l'information lexicale, c'est pourquoi nous avons fixé la valeur de  $Leak_W$  à 250. Pour les tâches d'identification de mot, et plus particulièrement pour les tâches de décision lexicale (voir Chapitre 6), le participant a intérêt à avoir une bonne précision sur l'identité des lettres. Nous avons donc fixé la valeur de  $Leak_W$  à 1250 pour retarder l'apport lexical dans ces tâches.

$Leak_W$  pourrait être vu comme un forme d'attention qui pourrait se substituer à notre modèle attentionnel. Mais il n'en est rien. En effet, dans la simulation A08, l'amorçage délétère par un *non-mot* est compensé par le déplacement de l'attention sur la lettre cible. On voit dans cet exemple que l'intervention du lexique n'apporterait rien étant donné que l'amorce est

un *non-mot*.  $Leak_W$  représente sans doute une certaine forme d'attention, mais elle vient en complément du filtre attentionnel présent dans *BRAID*.

En conclusion *BRAID* reproduit avec succès les principales expériences comportementales du domaine. Il nous fournit une compréhension fine des mécanismes mis en jeu dans la tâche de Reicher-Wheeler, notamment grâce au décours temporel des différents niveaux du modèle. Enfin, avec l'attention visuelle, nous proposons une justification théorique à l'effet de l'amorçage en contexte *pseudo-mot* et à la disparition de l'effet de supériorité du mot.

# Chapitre 6

## Simulations sur la reconnaissance visuelle de mots et la décision lexicale

### 1 Méthodologie

La méthodologie de ce chapitre est globalement la même que dans le Chapitre 5 sur la perceptibilité des lettres. Nous avons identifié un ensemble d'effets dans la littérature, liés à la reconnaissance visuelle des mots, dont notre modèle doit rendre compte. À la suite de quoi, nous avons sélectionné des études comportementales sur différents critères : ces études sont des références pour l'effet considéré ; elles rendent compte de la complexité de cet effet notamment en faisant varier les conditions expérimentales ou la tâche à réaliser ; enfin, et c'est un critère important pour les simulations, ces études fournissent les listes des items utilisés.

Cependant, certaines différences avec le chapitre précédent nécessitent d'être soulignées ici : tout d'abord les choix faits pour le paramétrage du modèle, puis les tâches réalisées et leur exploitation et enfin l'analyse des résultats.

**Paramétrage du modèle** Dans ce chapitre, nous n'utilisons qu'un seul jeu de paramètres pour toutes les simulations (les valeurs des paramètres sont présentés dans le Tableau 4.1, Chapitre 4, Section 2.3). Ce jeu de 14 paramètres est presque identique à celui du Chapitre 5, et seule la valeur du déclin au niveau lexical,  $Leak_W$ , change. Sa valeur passe de  $Leak_W = 250$  à  $Leak_W = 1\ 250$ . Cette valeur permet de ralentir l'accumulation d'information dans la distribution sur  $W^t$  et ainsi de donner plus de poids, dans la décision, à l'identification des lettres, conformément à la nature de la tâche de décision lexicale.

Un autre paramètre a sa valeur modifiée d'une simulation à l'autre, en fonction de la longueur des stimuli : c'est le paramètre qui caractérise la dispersion attentionnelle, noté  $\sigma_A$ . Pour les mots de quatre lettres  $\sigma_A = 1,5$ , pour les mots de cinq lettres  $\sigma_A = 2$  et pour les mots de six lettres  $\sigma_A = 2,5$ . Le comportement de ce paramètre n'est pas spécifique à ce chapitre mais est commun à toutes les simulations.

La tâche de décision lexicale met en jeu le sous-modèle d'appartenance lexicale (voir Chapitre 3, Section 2.5), qui est inhibé dans les autres tâches (voir Chapitre 4, Section 1.4). De ce fait, un nouveau paramètre, spécifique à la tâche de décision lexicale, est utilisé ici pour la première fois. C'est le paramètre  $Leak_D$ , qui caractérise le déclin de l'information sur la variable d'appartenance lexicale. Sa valeur est fixée pour toutes les simulations de ce document à  $Leak_D = 0,15$ .

**Calculs réalisés lors des simulations** Dans ce chapitre, le modèle est utilisé pour simuler deux tâches : la décision lexicale et la reconnaissance visuelle de mots. Dans ces deux tâches, *BRAID* calcule le décours temporel du résultat, c'est-à-dire la probabilité conditionnelle que le stimulus soit un mot pour la décision lexicale et la probabilité conditionnelle d'identification du stimulus dans  $W^t$  pour la reconnaissance visuelle de mots. Autrement dit, pour chaque valeur de  $t$  (itération), nous disposons des distributions  $P(D^t \mid \dots)$  ou  $P(W^t \mid \dots)$ , suivant la tâche.

Or, les données expérimentales contiennent essentiellement des temps de réponse, c'est-à-dire des temps observés pour, l'on suppose, que le participant atteigne un seuil de valeur de probabilité sur la réponse qu'il donne. Ainsi, pour comparer les résultats de nos simulations aux données expérimentales, nous avons besoin de l'information inverse, c'est-à-dire qu'il nous faut calculer le nombre d'itérations nécessaires pour atteindre un certain seuil. Donc, pour chaque simulation produite par *BRAID*, on associe une itération de temps à chaque valeur de  $P([D^t = vrai] \mid \dots)$  pour la décision lexicale, et  $P([W^t = w_i] \mid \dots)$  pour la reconnaissance de mots. Pour cela, nous divisons l'espace des probabilités en 101 valeurs allant de 0,0 à 1,0 et pour chaque valeur de probabilité  $s$ , nous sélectionnons le plus petit nombre d'itérations  $t$  tel que  $P([D^t = vrai] \mid \dots) > s$  ou  $P([W^t = w_i] \mid \dots) > s$ . Le résultat est donc une série de 101 itérations correspondant aux 101 valeurs de seuil  $s$  possibles pour prendre la décision. C'est cette information de décours temporel qui est représentée dans les figures de ce chapitre (en partie gauche, voir par exemple Figure 6.1).

**Définition des seuils** Pour analyser les résultats, il reste à définir une valeur de seuil de décision (ou d'identification). Nous faisons l'hypothèse que la valeur de seuil dépend de la consigne expérimentale et se trouve sous le contrôle cognitif du participant. De ce fait, une valeur de seuil « moyenne » différente est utilisée pour chaque expérience comportementale et donc dans chaque simulation. Dans ce chapitre, les valeurs choisies vérifient trois contraintes : premièrement, elles minimisent l'écart entre les données comportementales et les résultats des simulations mis à l'échelle ; deuxièmement, elles permettent de conserver le sens de l'effet quand cela est possible ; et enfin, lorsque les résultats concernent des non-mots, le seuil est défini de sorte que le taux d'erreur simulé corresponde au taux d'erreur observé expérimentalement.

**Analyse statistique des résultats** Les temps de réponse simulés par le modèle sont souvent très inférieurs aux données comportementales. Pour faciliter l'interprétation, nous appliquons un facteur d'échelle sur les résultats des simulations. Ce coefficient multiplicatif est calculé en choisissant une condition expérimentale et en faisant le rapport entre la donnée expérimentale et le résultat du modèle pour cette condition. Les temps de réponse simulés, une fois mis à l'échelle, coïncident avec les données expérimentales pour la condition fixée, et donnent les prédictions du modèle dans les autres conditions expérimentales.

Contrairement au chapitre précédent, nous n'effectuons pas systématiquement de test de significativité par item sur les résultats des simulations. En effet, la majorité des résultats comportementaux rapportés dans ce chapitre étant significatifs par participant mais pas par item, il était impossible de les comparer avec les données des simulations. En revanche, nous réalisons ces tests pour montrer telle ou telle propriété sur les résultats des simulations.

## 2 Effets de fréquence des mots

Cette section est consacrée à l'effet de fréquence des mots. Nous étudions la capacité de *BRAID* à rendre compte de cet effet, en simulant deux tâches comportementales. Dans la simula-

tion A01, nous simulons une tâche de reconnaissance visuelle de mots tandis que, dans la simulation A02, nous simulons une tâche de décision lexicale. Contrairement aux autres sections de ce chapitre, les simulations présentées dans cette section comparent les résultats obtenus avec *BRAID*, non pas à des données comportementales, mais à ceux obtenus par un autre modèle, le *Bayesian Reader* (Norris, 2006) (voir Chapitre 2, Section 1.7).

## 2.1 Simulation B01 : Effet de fréquence en reconnaissance de mot

Dans la simulation B01, nous étudions la capacité de *BRAID* à produire l'effet de fréquence objective dans une tâche d'identification de mot. Pour cela, nous utilisons la même méthodologie que celle proposée par Norris (2006).

### 2.1.1 Simulation du Bayesian Reader : Norris (2006)

Dans son étude, Norris (2006) contrôle la fréquence des mots avant de faire réaliser à son modèle une tâche de reconnaissance visuelle. Son objectif est de montrer que le nombre d'itérations nécessaire au modèle BR, pour reconnaître le stimulus *MOT* présenté, est proportionnel au logarithme de la fréquence du mot dans le lexique.

**Matériel et procédure** Cette simulation est réalisée avec le modèle BR. Son lexique est constitué initialement de 4 106 mots de 5 lettres issus du lexique CELEX. Pour réaliser cette simulation, 130 mots sont sélectionnés aléatoirement et regroupés par paquets de dix. A chaque paquet est attribué une fréquence parmi les treize valeurs suivantes : (1, 2, 5, 10, 20, 50, 100, 200, 500, 1 000, 2 000, 5 000, 10 000). Avant chaque série de simulations, les 10 mots du lexique présents dans un paquet se voient attribuer une nouvelle fréquence de départ. On mesure ensuite le nombre d'itérations nécessaire à BR pour identifier ce mot au seuil d'identification fixé au préalable. La valeur prise en compte pour chaque mot est la moyenne sur 50 essais. Finalement, pour garantir que les autres propriétés du mot étudié n'interfèrent pas dans la mesure, Norris (2006) opte pour un design en carré latin où chacun des 130 mots passe par toutes les conditions de fréquence, ce qui fait un total de treize séries de simulations.

**Résultats du Bayesian Reader** Le modèle BR produit une relation logarithmique entre la fréquence des mots et le nombre d'itérations nécessaire à leur identification. Le résultat présenté est la moyenne de 50 simulations réalisées suivant la procédure décrite précédemment, avec un seuil d'identification fixé à 0,95. La meilleure de ces 50 simulations explique 99% de la variance observée.

### 2.1.2 *BRAID* : Simulation B01

Dans notre modèle, la fréquence d'occurrence des mots correspond à l'état de repos de la distribution de probabilité sur  $W^t$ , c'est-à-dire son état en absence de stimulus, le prior  $P(W^0)$ . Ainsi, un mot plus fréquent « part d'une probabilité plus élevée », et a besoin de moins d'itérations pour atteindre le seuil d'identification. Cependant, s'il paraît évident que l'augmentation de la fréquence initiale des mots réduit les temps de réponse, il est plus difficile d'anticiper une relation logarithmique entre ces deux mesures, tant le modèle *BRAID* est plus complet que le modèle BR.



**Matériel et procédure** Le lexique utilisé dans les simulations réalisées avec le modèle *BRAID* est composé des 4 404 mots de 5 lettres du lexique BLP. Pour les simulations, 130 mots sont sélectionnés aléatoirement (voir Annexe A). Chacun de ces 130 mots est testé dans chacune des 13 conditions de fréquences suivantes : (1, 2, 5, 10, 20, 50, 100, 200, 500, 1 000, 2 000, 5 000, 10 000).

La procédure est la suivante. Après avoir sélectionné un mot  $w_i$ , on lui attribue, comme valeur de repos  $P([W^0 = w_i])$ , une fréquence parmi les 13 fréquences possibles, puis on simule la reconnaissance du mot  $w_i$ . On enregistre les distributions sur l'espace des mots,  $P(W^t | \dots)$ , obtenues pour les 1 000 premières itérations. On recommence cette opération en variant la valeur de fréquence de sorte que chaque mot passe par toutes les conditions de fréquence. Une fois ces étapes terminées, on rétablit la fréquence par défaut du mot. Ainsi, seule la fréquence du mot étudié est modifiée durant la simulation, et les fréquences des autres mots du lexique sont inchangées. On applique cette procédure au 130 mots sélectionnés de sorte que, pour chacun d'eux, on dispose du décours temporel sur 1 000 itérations et 13 conditions de fréquence.

**Résultats des simulations** Nous présentons deux types de résultats : le premier correspond au décours temporel moyen de la tâche de reconnaissance de mots, pour chacune des conditions de fréquence. Ce résultat est obtenu en moyennant les 130 décours temporels, calculés pour chaque condition de fréquence (voir Figure 6.1, à gauche). La régularité des courbes de décours temporel montre que l'effet de fréquence simulé par *BRAID* ne dépend pas de la valeur de seuil choisie et reste valable sur une large plage de valeurs de seuils possibles.

Le second résultat correspond au nombre moyen d'itérations nécessaire à l'identification de mots, pour chaque condition de fréquence. Le seuil d'identification est fixé à 0,85. Cette valeur de seuil a été choisie arbitrairement et est la même pour toutes les simulations de ce chapitre. Un calcul de régression entre le nombre moyen d'itérations et le logarithme de la fréquence nous donne un coefficient de détermination  $R^2 = 0,97$ , caractérisant ainsi une relation linéaire entre ces deux variables. Ce résultat est présenté Figure 6.1 (à droite).

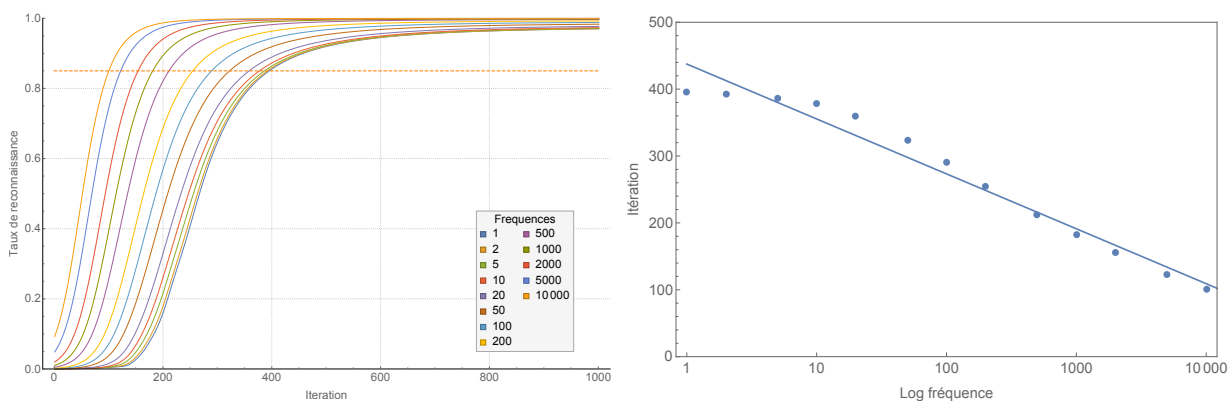


Figure 6.1 – **Simulation B01, Norris (2006), effet de fréquence en reconnaissance de mots.** À gauche, le décours temporel moyen des simulations de la tâche de reconnaissance de mot, calculé sur 1 000 itérations, pour chaque condition de fréquence. La ligne horizontale orange indique le seuil de décision à 0,85. À droite, les performances moyennes du modèle *BRAID*, calculées en fonction du logarithme de la fréquence des mots, au seuil de décision 0,85, et la meilleure droite de régression linéaire entre ces points.

**Conclusion** *BRAID* reproduit les résultats simulés par le modèle BR en reconnaissance de mots. Ces résultats sont stables quelle que soit la valeur de seuil choisie. C'est un résultat robuste du modèle. Contrairement à un pur effet log-linéaire, on observe dans nos points simulés un petit plateau pour les petites fréquences (1, 2 et 5 occurrences par million). Le graphique de décours temporel d'identification des mots montre que la diminution de la fréquence induit un temps de latence croissant sur la courbe d'identification de mot, ainsi qu'une diminution de la pente au point d'inflexion de la courbe.

## 2.2 Simulation B02 : Effet de fréquence en décision lexicale

Dans cette seconde simulation, nous nous intéressons à la tâche de décision lexicale. Dans *BRAID*, cette tâche met en concurrence une tâche de reconnaissance de mot et une tâche de recherche d'erreur présente dans le stimulus. Nous venons de montrer que notre modèle était capable de simuler l'effet de fréquence en reconnaissance visuelle de mot, mais la mise en concurrence de la reconnaissance de mot avec la recherche d'erreur pourrait modifier le rôle de la fréquence objective. C'est ce que nous étudions dans cette section.

### 2.2.1 Expérience comportementale : Norris (2006)

Pour cela, nous considérons la seconde étude de Norris (2006), dans laquelle l'auteur intègre au modèle BR un mécanisme de rejet des *non-mots*, et simule une tâche de décision lexicale.

**Matériel et procédure** Le matériel et la procédure de simulation sont les mêmes que pour la simulation B01. Deux seuils sont cependant nécessaires pour réaliser les simulations dans le modèle BR. Le seuil de réponse OUI pour un *mot* est fixé à 0,95 tandis que le seuil de réponse NON pour un *non-mot* est fixé à 0,5.

**Résultats comportementaux** Le modèle BR produit une relation quasi linéaire entre le logarithme de la fréquence et le nombre d'itérations nécessaire à la prise de décision, avec un  $R^2$  de 0,99, pour un seuil fixé à 0,95.

### 2.2.2 *BRAID* : Simulation B02

Dans la simulation B02, nous utilisons le modèle *BRAID* pour simuler une tâche de décision lexicale. L'évolution de la réponse OUI (celle qui nous intéresse ici) repose essentiellement sur la dynamique de la tâche d'identification de mot, car il n'y a alors pas d'erreur à trouver dans le stimulus. Ceci nous laisse penser que l'on devrait observer ici aussi une relation linéaire entre le nombre d'itérations nécessaire à la réponse OUI et le logarithme de la fréquence objective du mot présenté.

**Matériel et procédure** La procédure expérimentale est la même que pour la simulation B01. Les valeurs de seuil utilisées dans cette simulation sont les mêmes que dans toutes les simulations impliquant le sous-modèle d'appartenance lexicale. Le seuil OUI est fixé arbitrairement à 0,85 et le seuil NON est fixé à 0,5. On obtient 13 courbes représentant le décours temporel moyen des 130 mots pour chaque condition de fréquence (voir Figure 6.2, partie de droite).

**Résultats des simulations** Là encore, nous présentons deux résultats. Le premier correspond au décours temporel moyen de la réponse OUI, pour chacune des conditions de fréquence. Ce résultat est obtenu en moyennant les 130 décours temporels, calculés pour chaque condition de fréquence (voir Figure 6.2, partie de gauche)

Le second résultat présenté Figure 6.2 (à droite) correspond au nombre d'itérations nécessaire par condition de fréquence pour que la réponse OUI atteigne un seuil de 0,85. Le calcul de la régression linéaire du nombre d'itérations nécessaire pour une réponse OUI en fonction du logarithme de la fréquence donne un coefficient de détermination  $R^2 = 0,98$ .

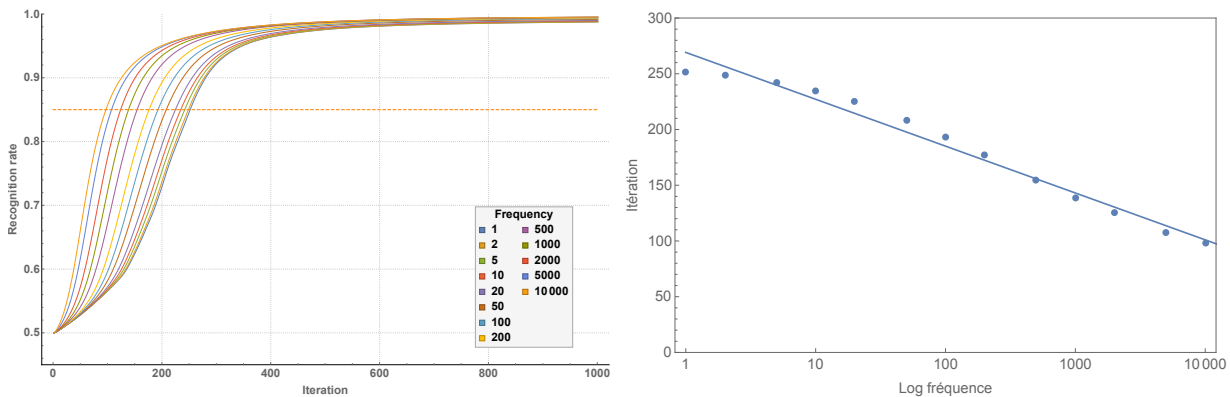


Figure 6.2 – **Simulation B02, Norris (2006), effet de fréquence en décision lexicale.** À gauche, le décours temporel moyen de la réponse OUI, calculé sur 1 000 itérations, pour chaque condition de fréquence. La droite horizontale orange indique le seuil de décision à 0,85. À droite, les performances moyennes du modèle *BRAID*, calculées en fonction du logarithme de la fréquence, pour un seuil de décision de 0,85, et la meilleure droite de régression linéaire entre ces points.

**Conclusion** *BRAID* reproduit les résultats simulés par le modèle BR en décision lexicale. Ces résultats sont stables autour de la valeur de seuil choisie. Le graphique de décours temporel d'identification des mots montre que la diminution de la fréquence induit un ralentissement, dans les premières itérations, sur la courbe de réponse OUI, ainsi qu'une diminution de la pente au point d'inflexion de la courbe.

## 2.3 Discussion

Dans cette section, nous avons répliqué les simulations réalisées par Norris (2006) avec le modèle BR. Nous avons montré que le nombre d'itérations nécessaire pour reconnaître un mot (Simulation B01), ou pour décider si le stimulus présenté est bien un mot (Simulation B02), augmentait linéairement avec le logarithme de la fréquence d'occurrence de ce mot dans le lexique. Mais les résultats que nous obtenons sont beaucoup plus riches, notamment grâce aux courbes de décours temporel (Figures 6.1 et 6.2, partie gauche).

Tout d'abord, l'étude des courbes par fréquence montre qu'elles sont régulièrement espacées, tout au long du décours temporel, pour des valeurs de seuils comprises entre 0,2 à 0,9 pour la tâche de reconnaissance de mot et entre 0,6 à 0,9 pour la tâche de décision lexicale. La régularité de ces espacements matérialise la relation de linéarité entre les performances et le logarithme de la fréquence. Une étude des coefficients de détermination, pour la tâche de reconnaissance de mot et les valeurs de seuil dans l'intervalle 0,2 à 0,9, montre que la plus petite valeur est

$R^2 > 0,96$ . De même, pour la tâche de décision lexicale et les valeurs de seuil dans l'intervalle 0,6 à 0,9, la plus petite valeur est  $R^2 > 0,97$ . On peut ainsi conclure que l'effet de fréquence est un effet robuste du modèle *BRAID*.

Cependant pour les valeurs de fréquence les plus petites (1, 2 et 5 par million de présentations), l'intervalle entre les courbes est plus resserré que pour les autres fréquences et ce, pour la tâche de reconnaissance de mots comme pour la tâche de décision lexicale. Ceci montre que l'influence de la fréquence sur les performances du modèle *BRAID* est sensiblement la même pour ces petites fréquences. Ce résultat diffère de celui observé dans le modèle BR. Cette observation implique que, lorsque les mots sont très rares (moins de 5 occurrences par million), la fréquence n'engendre pas de différence de performance, et dans ce cas, *BRAID* ne se comporte pas comme le modèle BR.

Enfin, les courbes de décours temporel permettent de caractériser le rôle de la fréquence sur les performances : la fréquence donne un avantage temporel aux mots les plus fréquents et facilite leur traitement. En effet, si l'on étudie les courbes sur les 200 premières itérations, on constate que les mots les plus fréquents commencent avec une probabilité de reconnaissance supérieure à l'uniforme. Cette probabilité de départ décroît avec la valeur de la fréquence. Lorsque la fréquence d'occurrence est très petite, la probabilité de départ est inférieure à l'uniforme, ce qui entraîne un retard sur les courbes de décours temporel. Ceci caractérise l'avantage temporel dû à la fréquence. Cependant, si l'on observe les points d'inflexion de ces courbes, on constate que leur pente diminue lorsque la fréquence diminue. Ceci signifie que le gain sur les performances, apporté par l'augmentation du nombre d'itérations, dépend de la fréquence d'occurrence des mots. En conclusion, la fréquence d'occurrence donne un avantage dès le départ, traduit dans le modèle par la distribution prior  $P(W^0)$ , et influence la vitesse d'identification des mots, aussi bien en reconnaissance visuelle qu'en décision lexicale.

### 3 Effets classiques en décision lexicale

Deux effets liés à la tâche de décision lexicale sont aujourd'hui largement admis. Le premier est l'effet de statut lexical (Rubenstein et al., 1970), c'est-à-dire qu'un mot est reconnu comme appartenant au lexique plus rapidement qu'un non-mot est reconnu comme n'en faisant pas partie. Le second est l'effet de légalité des non-mots (Rubenstein et al., 1971), c'est-à-dire qu'un non-mot orthographiquement illégal (comme *BVAGF*) est rejeté plus rapidement qu'un non-mot orthographiquement légal (comme *BRUVE*). Ces deux effets font partie des résultats de référence que Coltheart, Rastle, Perry, Langdon, et Ziegler (2001) identifient comme devant être réalisés par tout modèle de lecture. Dans cette section, nous nous intéressons à la capacité de *BRAID* à simuler ces deux effets. Ces simulations reposent sur deux articles qui fournissent le matériel expérimental (mots, non-mots). Nous avons donc utilisé ce matériel dans nos simulations en prenant soin, quand cela était possible, de comparer les caractéristiques de fréquence et de densité de voisinage avec celles utilisées par *BRAID*.

#### 3.1 Simulation B03 : Effet du statut lexical

Avec cette simulation, nous cherchons à évaluer la capacité du modèle *BRAID* à simuler l'effet de statut lexical pour différentes valeurs de fréquence d'occurrence.

### 3.1.1 Expérience comportementale : Rubenstein et al. (1970)

Dans cette expérience, Rubenstein et al. (1970) s'intéressent au temps nécessaire pour décider si une chaîne de caractères présentée brièvement et isolément est un mot ou un non-mot. Ils font l'hypothèse que le temps de réaction, lorsque le stimulus est un mot, varie en fonction de certaines propriétés : sa fréquence, sa concrétude (le fait que le mot réfère à un concept concret ou abstrait) ou son homographie (le fait qu'une même chaîne de caractère réfère à deux mots se prononçant différemment, par exemple, « les poules du couvent couvent »).

**Matériel et procédure** Les mots comme les non-mots utilisés dans cette expérience ont une longueur de 4 ou 5 lettres et sont proposés en annexe dans l'article. Pour les mots, leur fréquence varie suivant trois modalités : 10, 100, 1 000 apparitions pour 4,5 millions, et sont pris dans des textes de magazine. Les fréquences des mots varient peu au sein de chaque modalité de fréquence. Deux modalités sont utilisées pour la concrétude (« concret » ou « abstrait ») et pour l'homographie (« avec homographe » ou « sans homographe »).

**Résultats comportementaux** Les auteurs trouvent qu'une réponse OUI à la présentation d'un mot est plus rapide qu'une réponse NON à la présentation d'un non-mot. Ils trouvent également un effet significatif de la fréquence et de l'homographie, mais pas de la concrétude, et les mots les plus fréquents sont reconnus plus rapidement que les autres.

### 3.1.2 BRAID : Simulation B03

Le résultat que nous cherchons à reproduire dans ces simulations est l'effet de statut lexical, pour différentes valeurs de fréquence des mots. Nous avons donc regroupé les mots par fréquence, sans tenir compte des conditions d'homographie ou de concrétude. La tâche réalisée par *BRAID* est la tâche de décision lexicale (voir Chapitre 4, Section 1.4). Les paramètres utilisés sont les paramètres de décision lexicale définis par défaut. Les paramètres de la position du regard et de la moyenne de l'attention varient avec la longueur du mot présenté. Dans ces simulations, ces deux paramètres ont la même valeur, et correspondent au centre du stimulus : 2,5 pour les mots de 4 lettres et 3 pour les mots de 5 lettres.

**Matériel** Trois mots parmi ceux utilisés dans l'expérience ne sont pas présents dans le lexique BLP : *TRUE*, *FALSE* et *CRUEL*. Les mots sont regroupés en trois groupes de fréquence (10 ; 100 ; 1 000). Les fréquences moyennes dans nos simulations (4,78 ; 30,63 ; 302,65) sont proches de celles des mots de l'étude initiale de Rubenstein et al. (1970) (4,33 ; 38,89 ; 387,33) en ramenant les valeurs au nombre d'apparitions par million. Les densités de voisinage calculées à partir du BLP sont équivalentes pour chacun des groupes (6,06 ; 6,57 ; 7,27). Pour les non-mots, seule la méthode de construction est fournie par l'article. Nous avons donc créé des non-mots orthographiquement légaux et phonologiquement réguliers, selon les critères précisés par Rubenstein et al. (1970) (voir Annexe).

**Résultats des simulations** Le seuil de décision est fixé à 0,68 pour la réponse OUI comme pour la réponse NON. Ce seuil nous donne un taux de réponse correcte pour les non-mots de 96% identique à celui observé expérimentalement. En revanche, les taux de bonnes réponses obtenus pour les mots sont de 98%, 100% et 100% respectivement pour les mots de fréquence 10, 100, et 1 000, ce qui est supérieur aux taux observés expérimentalement (94%, 97% et 99%). Ces taux sont atteints par *BRAID* en fixant la valeur de seuil à 0,91 pour la réponse OUI.

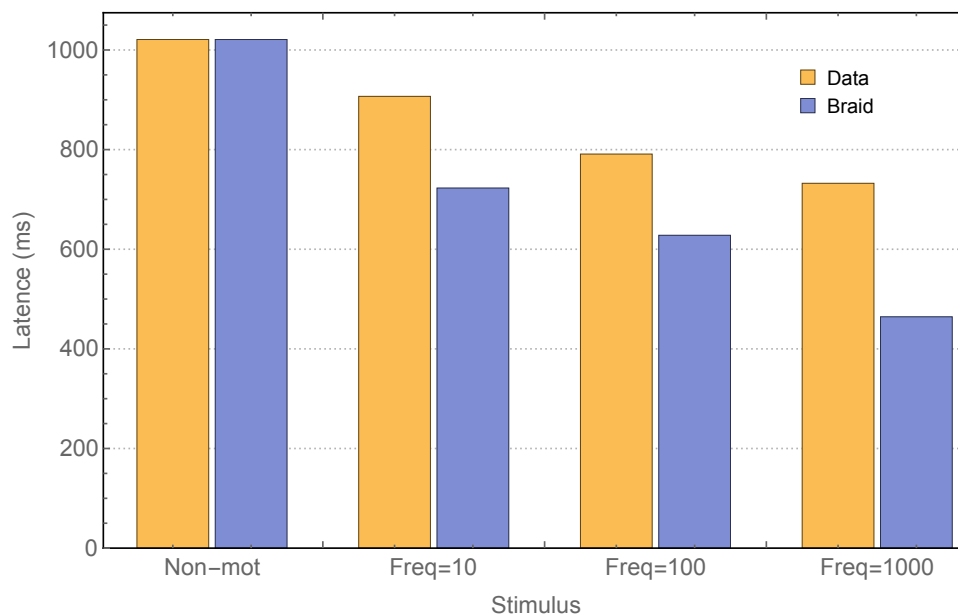


Figure 6.3 – **Simulation B03, Rubenstein et al. (1970), effet du statut lexical.** Comparaison des temps de réaction (en jaune) et des temps de réponse simulés harmonisés (en bleu). La réponse NON pour un non-mot nécessite en moyenne plus de temps que la réponse OUI pour un mot, pour les trois conditions de fréquence étudiées.

Les variances des groupes ne sont pas homogènes, donc pour valider les résultats nous utilisons le test non paramétrique de Kruskal-Wallis par item. De plus, comme la variable dépendante est ordinaire (un non-mot est considéré ici comme un mot dont la fréquence est nulle), nous réalisons les trois comparaisons planifiées suivantes : non-mots contre mots de fréquence 10 :  $[\chi^2(1) = 10,15, p = 0,001]$  ; les mots de fréquence 10 contre les mots de fréquence 100 :  $[\chi^2(1) = 11,69, p < 0,001]$  ; et enfin les mots de fréquence 100 contre les mots de fréquence 1 000 :  $[\chi^2(1) = 47,112, p < 0,001]$ . Comme pour les résultats comportementaux, on obtient un effet de statut lexical ainsi qu'un effet de la fréquence : la rapidité de la décision pour les mots augmente avec la fréquence d'occurrence. Les résultats sont présentés Figure 6.3.

**Conclusion** *BRAID* simule avec succès l'effet du statut lexical, un résultat classique en décision lexicale. De plus, l'effet du statut lexical est d'autant plus marqué que les mots ont une fréquence plus élevée. En revanche, l'écart de performance entre les non-mots et les mots est systématiquement plus important pour le modèle que pour les données comportementales.

## 3.2 Simulation B04 : Effet de légalité des non-mots

Avec cette simulation, nous ne nous intéressons plus qu'aux non-mots. Nous étudions la différence entre les non-mots orthographiquement illégaux et légaux, les premiers devant être rejetés plus rapidement que les seconds.

### 3.2.1 Expérience comportementale : Rubenstein et al. (1971)

**Matériel et procédure** Dans cette expérience, les participants réalisent une tâche de décision lexicale. Les mots et les non-mots utilisés ont une longueur de 4 ou 5 lettres. Nous ne nous intéressons qu'aux résultats obtenus sur les non-mots. Trois ensembles de non-mots sont étudiés.

Le premier ensemble regroupe les non-mots orthographiquement légaux et prononçables (135), qui ont donc toutes les caractéristiques des mots anglais (exemple : « strig »). Les non-mots des deux autres ensembles sont construits en formant un groupe consonantique en fin de chaîne. Dans le deuxième ensemble, ce groupe consonantique est prononçable (exemple : « nraft »), tandis que, dans le troisième ensemble, ce groupe consonantique n'est pas prononçable (exemple : « sbgm »). Les non-mots de ces deux derniers ensembles sont considérés comme illégaux orthographiquement.

**Résultats comportementaux** Après une étude de la variance par sujet, Rubenstein et al. (1971) obtiennent un effet significatif de la légalité orthographique comme de la régularité phonologique, sur les temps de réaction. Ils montrent que les non-mots illégaux sont catégorisés en moyenne 100 ms plus rapidement que les non-mots légaux. Ils obtiennent un fort effet de la légalité et de la régularité sur la précision des réponses. En particulier, les non-mots illégaux et non prononçables sont correctement catégorisés avec une précision de 99%, contre 97% pour les non-mots illégaux prononçables et 94% pour les non-mots légaux.

### 3.2.2 *BRAID* : Simulation B04

Pour simuler l'effet de légalité des non-mots indépendamment des caractéristiques phonologiques, qui ne sont pas représentées dans *BRAID*, nous avons regroupé les non-mots illégaux prononçables et non-prononçables. La tâche réalisée par *BRAID* est la tâche de décision lexicale (voir Chapitre 4, Section 1.4). Les paramètres utilisés sont les paramètres de décision lexicale définis par défaut. Les paramètres de la position du regard et de la moyenne de l'attention sont fixés, comme précédemment, en fonction de la longueur du mot présenté.

**Matériel et procédure** Les non-mots utilisés dans ces simulations sont issus de l'article. Deux non-mots légaux sont néanmoins présents dans le lexique BLP : *TORS* et *FLAN*. Nous les avons donc exclus, et nous n'utilisons donc que 133 non-mots légaux. Le fait de regrouper les non-mots illégaux porte leur nombre à 53. L'étude des voisinages orthographiques laisse apparaître une certaine disparité entre les deux ensembles de stimuli. Le nombre moyen de voisins pour le groupe des non-mots illégaux est de 1,21 tandis que, pour le groupe des non-mots légaux, il est de 4,27.

**Résultats des simulations** Les résultats des simulations présentés Figure 6.4 ont été obtenus avec un seuil de décision à 0,63. Ce seuil permet d'obtenir des taux de réponse correcte simulés identiques à ceux observés, c'est-à-dire 0,99 pour les non-mots illégaux et 0,94 pour les non-mots légaux. Une analyse statistique préliminaire a mis en évidence une différence de variance entre les deux groupes. Nous avons donc réalisé le test non paramétrique de Kruskal-Wallis par item :  $[\chi^2(1) = 6,74, p = 0,009]$ . Les résultats simulent les observations expérimentales : les temps de réaction simulés sont significativement plus longs pour rejeter des non-mots légaux que pour rejeter des non-mots illégaux.

**Conclusion** *BRAID* simule l'effet de légalité des non-mots, en donnant un avantage significatif aux non-mots illégaux. Cependant, la cause de cette différence n'est pas évidente ici. En effet, la densité du voisinage a une influence sur les performances d'identification. Comme nous le montrerons à la Section 5 de ce chapitre, cet effet est inhibiteur. Donc, le fait que les non-mots illégaux aient en moyenne 1,21 voisins, contre 4,27 pour les non-mots légaux, va dans le sens d'une inhibition plus importante pour les non-mots légaux, ce qui devrait donc les pénaliser.

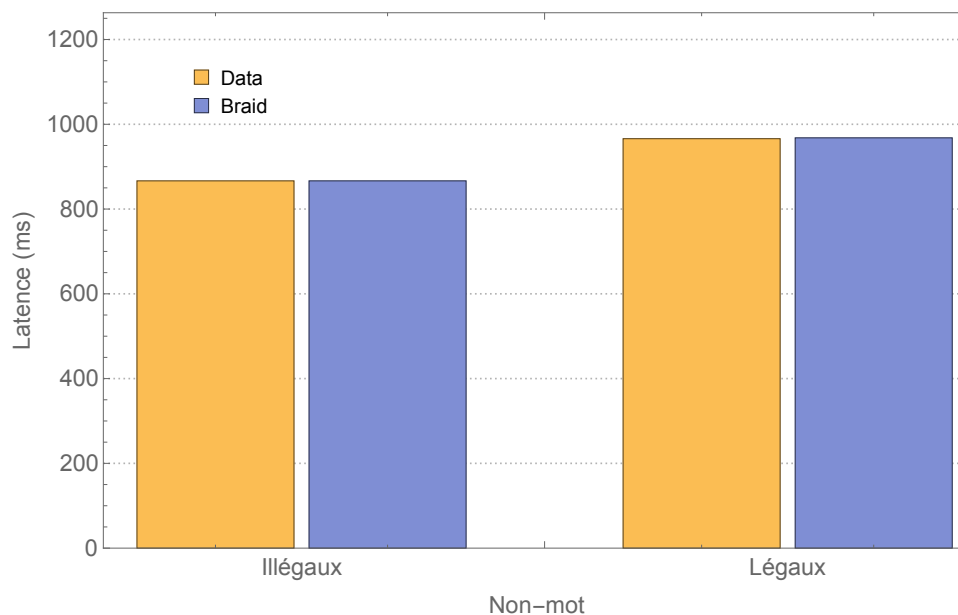


Figure 6.4 – **Simulation B04, Rubenstein et al. (1971), effet de légalité des non-mots.** Comparaison des temps de réactions (en jaune) avec les résultats harmonisés des simulations (en bleu). La réponse correcte NON pour les non-mots légaux nécessite en moyenne plus de temps que la réponse correcte NON pour les non-mots illégaux.

Donc dans cette simulation, comme dans l’expérience comportementale, une partie de l’effet observé pourrait être due à la densité de voisinage.

### 3.3 Discussion

Dans cette section, nous avons reproduit avec succès les deux premiers résultats que Coltheart et al. (2001) identifient comme devant être réalisés par tout modèle de lecture. Tout d’abord, nous avons montré qu’il fallait moins d’itérations à *BRAID* pour accepter un mot que pour rejeter un non-mot (Simulation B03) et que cet effet était plus marqué dans le modèle qu’il ne l’est expérimentalement. Cependant, ce résultat est robuste, car il a été validé pour des mots fréquents mais également pour des mots peu fréquents. Ensuite, avec la Simulation B04, nous avons montré que *BRAID* rejetait plus rapidement les non-mots illégaux que les non-mots légaux. Mais dans ce dernier cas, un doute raisonnable est introduit par la présence d’une différence de densité de voisinage dans les conditions expérimentales. Ces deux simulations nous donnent l’occasion d’apporter des précisions sur le fonctionnement du modèle.

Dans la Simulation B03, le modèle *BRAID* accentue l’effet lexical, pour toutes les valeurs de fréquence, comparativement aux données observées dans les expériences comportementales. En d’autres termes, le modèle est plus lent pour répondre NON lorsque le stimulus est un non-mot <sup>13</sup>, que pour répondre OUI lorsque le stimulus est un mot, comparativement aux données expérimentales. Ce résultat est en parti dû à nos choix de modélisation. Pour rappel, la réponse NON repose sur la détection d’erreur que nous avons limité, pour des raisons de performances calculatoire, à ne considérer qu’une erreur par stimulus. Or, ce choix ne correspond pas aux non-mots utilisés dans ces simulations qui diffèrent le plus souvent des mots du lexique sur plusieurs

<sup>13</sup>On fait la différence ici entre la réponse NON lorsque le stimulus est un non-mot et la réponse NON lorsque que le stimulus est un mot. Dans le premier cas, on étudie la capacité du modèle à rejeter les non-mots tandis que dans le second, la réponse NON est considérée comme une erreur.



lettres. En autorisant le modèle à considérer des erreurs sur plusieurs caractères simultanément, les performances du modèle pour rejeter un non-mot seraient améliorées, réduisant ainsi l'écart entre les temps de réponse NON à un non-mot et les temps de réponses OUI à un mot.

Le modèle SCM (Davis, 2010b) est confronté au même problème. Dans ce modèle, deux méthodes de calcul distinctes sont utilisées pour calculer la réponse OUI à un mot et la réponse NON à un non-mot. Un seuil de réponse commun est fixé à 0,80. À ce mécanisme est ajouté un second seuil à 0,68 qui est utilisé au niveau « mot ». Si un mot du lexique dépasse ce seuil d'activation, le modèle considère que le mot est reconnu et déclenche la réponse OUI. Dans *BRAID*, une possibilité pour réduire la différence serait de fixer également des seuils différents pour ces deux réponses. Par exemple, si nous portons le seuil de réponse OUI à 0,91 en laissant le seuil de réponse NON à 0,68, nous obtenons des résultats très proches des données comportementales.

Dans la Simulation B04, on observe de meilleures performances pour les mots illégaux que pour les mots légaux. Or, on pourrait s'attendre à ce que *BRAID* ne fasse aucune différence, avec une détection d'erreur limitée à une erreur par stimulus. Cette différence est probablement due à la manière dont on calcule la valeur de la réponse NON. Dans l'Équation (4.13) (voir page 86), on voit que la valeur de la réponse NON est calculée en faisant la moyenne, sur tous les patterns d'erreur considérés, de la somme des similarités du stimulus sur lequel on a appliqué le pattern d'erreur avec l'ensemble des mots du lexique. Ce calcul n'est pas facilement interprétable. Néanmoins, tout laisse à penser que le calcul de la similarité en position d'erreur est meilleur en moyenne pour les non-mots illégaux, sans que l'on observe de différence entre les non-mots légaux et les non-mots illégaux pour les autres positions. Ce résultat pourrait être inversé en construisant des non-mots légaux qui seraient des voisins orthographiques de mots du lexiques.

## 4 Effet de la fréquence du voisinage

L'effet de fréquence du voisinage est l'influence que peut avoir un voisin orthographique plus fréquent que le mot présenté, sur le temps nécessaire à l'identification visuelle de ce mot. Cet effet a largement été étudié dans la littérature (voir Andrews (1997) et Ferrand (2007) pour des revues de littérature, bien que les effets qui y sont décrits sont parfois contradictoires et semblent dépendre de la tâche). Dans cette section, nous étudions la manière dont *BRAID* rend compte de l'effet de la fréquence du voisinage dans deux tâches différentes. Dans la Simulation B05, nous nous intéressons à l'effet de la fréquence du voisinage en décision lexicale et présentons une série de trois études comportementales aux résultats contradictoires. Dans la Simulation B06, nous nous intéressons à l'effet de la fréquence du voisinage en reconnaissance visuelle de mot.

### 4.1 Simulation B05 : Effet de la fréquence du voisinage dans une tâche de décision lexicale

Cette première série de simulations a pour objet l'effet de la fréquence du voisinage en décision lexicale. Les résultats obtenus dans la littérature sont variables. Nous considérons ici trois études représentatives des différents types d'observations expérimentales. L'étude A (Sears, Hino, & Lupker, 1995, expérience 1) ne montre pas d'effet de la fréquence du voisinage, l'étude B (Perea & Pollatsek, 1998, expérience 1) met en évidence un effet inhibiteur, tandis que l'étude C (Siakaluk, Sears, & Lupker, 2002, expérience 2) met en évidence un effet facilitateur. Avec ces trois simulations, nous étudions la capacité de notre modèle à rendre compte de ses

observations opposées, et, par là-même, de fournir une piste d'explication aux résultats en apparence contradictoires de la littérature.

#### 4.1.1 Expériences comportementales

Dans les études A et C, les auteurs manipulent deux facteurs indépendants : la densité du voisinage d'une part, qui est de faible densité (FD) ou de haute densité (HD), et, d'autre part, la présence (1HF) ou l'absence (0HF) d'un voisin orthographique plus fréquent. Les deux études diffèrent par la fréquence des stimuli employés. Dans l'étude C, les auteurs n'utilisent que des mots de basse fréquence (BF). Dans l'étude A, les auteurs utilisent des mots de basse fréquence (BF) et de haute fréquence (HF), mais comme ils ne fournissent que les mots de HF, nous ne considérons que ce cas là.

Dans l'étude B, les auteurs manipulent deux facteurs indépendants : la fréquence du voisinage, BF ou HF, et la fréquence des stimuli, BF ou moyenne fréquence (MF).

**Matériel et procédure** Le matériel des trois études est présenté dans le Tableau 6.1. Les fréquences des études comportementales sont calculées selon la norme de la base de Kucera et Francis (1968), tandis que les fréquences de nos simulations sont fournies par le lexique BLP.

Les trois expériences comportementales considérées ici reposent sur des procédures similaires. Chaque essai commence par la présentation d'une croix de fixation pendant une durée allant de 500 ms à 1 s. Puis le mot cible est affiché jusqu'à ce que le participant donne une réponse. Dans l'étude B, un écran blanc est présenté pendant 200 ms entre la croix de fixation et le stimulus. Les participants doivent décider si le stimulus présenté est un mot à l'aide d'un boîtier à deux boutons, l'un pour OUI et l'autre pour NON. La réponse du participant met fin à l'essai. L'intervalle de temps entre deux essais varie entre 1,5 s et 2 s, suivant l'étude. Chaque expérience est précédée par une période d'entraînement durant laquelle les participants sont informés de leurs performances.

Les instructions données aux participants varient suivant l'étude. Dans les études A et B, les participants doivent répondre le plus rapidement et le plus précisément possible. Dans l'étude C, les participants doivent favoriser la précision à la vitesse.

**Résultats comportementaux** Les résultats de l'étude A montrent un effet facilitateur de la taille du voisinage significatif par sujet (mais pas par item), et pas d'effet principal de la fréquence de voisinage. Les résultats de l'étude B montrent un effet principal inhibiteur de la fréquence du voisinage, significatif par participants, et par items seulement pour les mots BF. Enfin, les résultats de l'étude C montrent un effet facilitateur de la fréquence du voisinage, significatif par participants mais pas par items. Les mots avec un voisin plus fréquent sont reconnus 16 ms plus tôt en moyenne pour la condition BF, et 15 ms plus tôt pour la condition HF.

#### 4.1.2 BRAID : Simulation B05

Dans *BRAID*, la décision lexicale met deux processus en concurrence. Le premier accumule l'information au niveau lexical à partir de la représentation interne du stimulus porté par les variables  $P_n^t$ . C'est sur ce premier processus que repose la réponse OUI. Le second accumule de l'information à partir de la représentation interne du stimulus, en faisant l'hypothèse qu'il y a une erreur sur une des lettres. C'est sur ce second processus que repose la réponse NON.

La présence d'au moins un voisin plus fréquent va influencer les deux processus. Pour la réponse OUI, la probabilité du mot stimulus augmente à chaque itération. Parallèlement, le

Tableau 6.1 – **Récapitulatif des caractéristiques des mots utilisés dans les expériences comportementales et les simulations de la section B05.** HF : Haute Fréquence, MF : Moyenne Fréquence et BF : Basse Fréquence. La fréquence du voisinage correspond à la moyenne sur tous les mots du voisin le plus fréquent.

	Fréquence du voisinage								
	Absence de voisin HF				Au moins un voisin HF				
<b>Sears, Hino et Lupker</b> (1995, expériences 1 & 2)	Faible Densité		Haute Densité		Faible Densité		Haute Densité		
	Data	Braid	Data	Braid	Data	Braid	Data	Braid	
	Fréquence	166,7	80,7	178,6	112	139,9	104,5	151,9	63,0
	Taille du voisinage	3,5	3,8	10,6	10,8	3,9	5	10,8	12,1
Fréquence du voisinage	66,1	64,7	101,4	61,4	371,8	358,2	571,8	501,3	
<b>Perea et Pollatsek</b> (1998, expériences 1 & 2)	mots BF		mots MF		mots BF		mots MF		
	Data	Braid	Data	Braid	Data	Braid	Data	Braid	
	Fréquence	3,6	4,4	24,8	27,0	2,8	4,2	26,3	30,3
	Taille du voisinage	2,5	2,1	2,5	2,3	3,0	3,9	4,2	3,9
Fréquence du voisinage		3,6		6,6		90,4		296,3	
<b>Siakaluk, Seras et Lupker</b> (2002, expérience 2)	Faible Densité		Haute Densité		Faible Densité		Haute Densité		
	Data	Braid	Data	Braid	Data	Braid	Data	Braid	
	Fréquence	19,1	17,4	23,8	22,7	20,1	22,3	23,1	26,3
	Taille du voisinage	3,3	3,4	8,7	8,1	3,5	4,2	9,4	9,9
Fréquence du voisinage	8,0	8,5	13,6	18,2	301,0	302,5	290,1	301,3	

voisin plus fréquent, qui a donc une probabilité de départ plus importante, maintient une probabilité haute grâce à sa similarité avec le stimulus. Les probabilités de ces deux points de l'espace  $W$  sont additionnées. En revanche, pour la réponse NON, seul le voisin orthographique plus fréquent a sa probabilité qui augmente. Cela génère un avantage, dans ces conditions, à la réponse OUI. *BRAID* devrait donc produire un effet facilitateur en présence d'un voisin plus fréquent.

**Matériel et procédure** Les stimuli utilisés dans ces trois simulations sont les mêmes que ceux utilisés lors des expériences comportementales, à l'exception du mot *CASINO* de l'étude B qui ne fait pas partie du lexique BLP. Cependant, les caractéristiques des mots diffèrent sensiblement selon le lexique utilisé comme le montre le Tableau 6.1.

La tâche simulée par *BRAID* est la tâche de décision lexicale (voir Chapitre 2, Section 1.4). Les valeurs de seuils sont spécifiques à chaque étude et sont calculés comme décrit dans la section méthodologie du présent chapitre (Section 1). Les résultats présentés ont été obtenus, pour l'étude A avec un seuil à 0,71, pour l'étude B avec un seuil à 0,82 et pour l'étude C avec un seuil à 0,8. Les autres paramètres ont leur valeur par défaut.

**Résultats des simulations** L'analyse des courbes de décours temporel, Figure 6.5 à gauche, montre que la présence d'un voisin orthographique de plus haute fréquence (courbes orange et rouge) produit un avantage dès les premières itérations. Cela est particulièrement visible dans

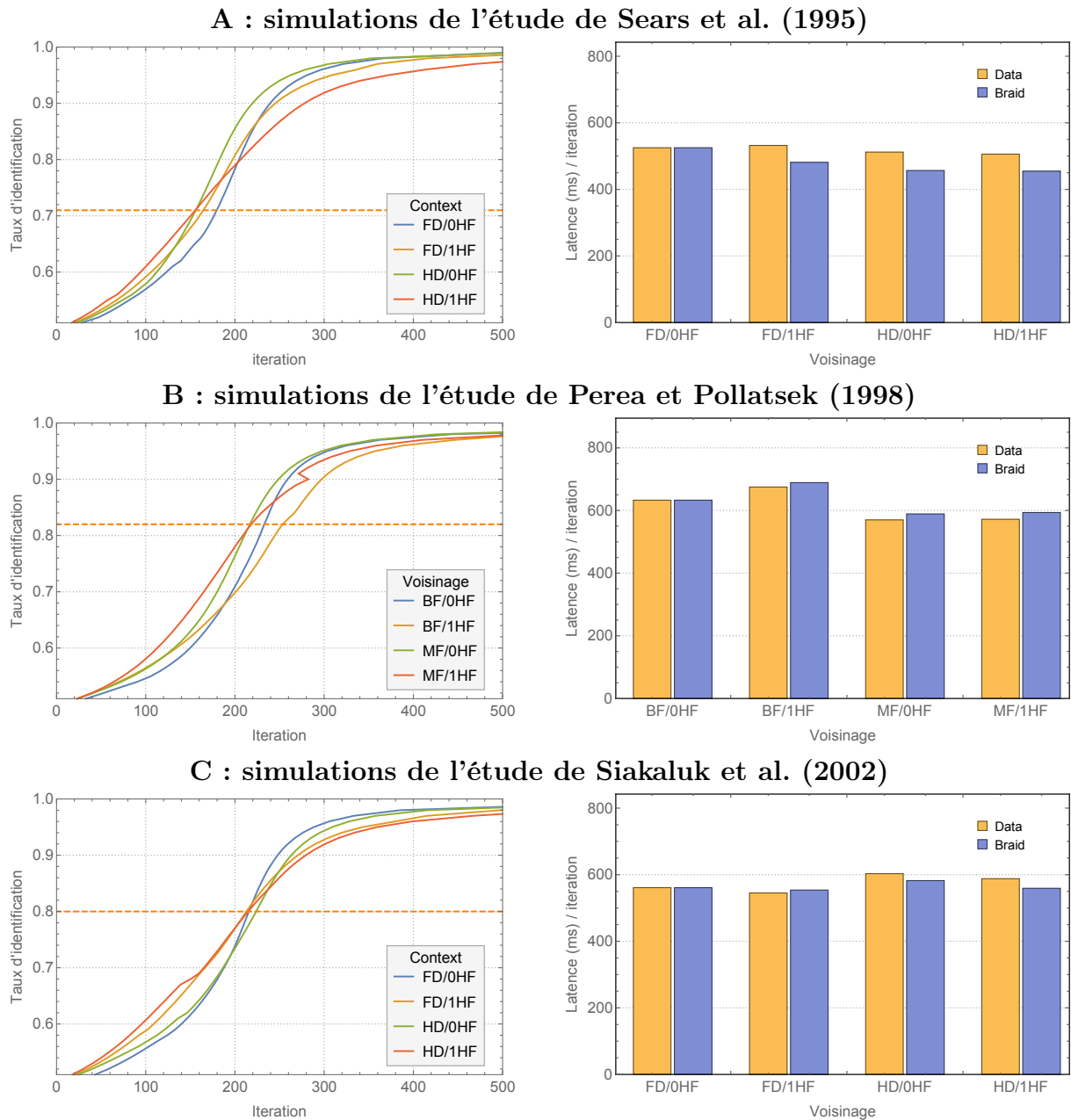


Figure 6.5 – **Simulation B05, effet de la fréquence du voisinage dans une tâche de décision lexicale.** Résultats des simulations des expériences comportementales. La première ligne donne les résultats de Sears et al. (1995), la deuxième ligne donne les résultats de Perea et Pollatsek (1998) et la troisième ligne donne les résultats de Siakaluk et al. (2002). Chaque ligne présente, à gauche, le dérouls temporel de la réponse OUI avec en pointillés, l'itération à laquelle sont extraits les résultats qui sont comparés aux données expérimentales. À droite, on présente les données expérimentales (en jaune) avec les résultats des simulations de *BRAID* (en bleu) pour l'itération choisie.

la simulation de l'étude C. Puis l'avantage s'inverse et les performances deviennent meilleures pour les mots sans voisin orthographique de plus haute fréquence.

Dans l'étude A, l'annulation de l'effet en condition FD se produit pour un seuil à 0,88 et pour la condition HD pour un seuil à 0,71. *BRAID* échoue donc à reproduire une absence d'effet, même si, pour un seuil à 0,71, seule la condition FD/0HF se distingue réellement des deux autres. En revanche, *BRAID* reproduit avec succès les variations des différentes conditions expérimentales des études B et C, pourtant contradictoires.

**Conclusion** Cette série de simulations met en évidence une inversion de l'effet de la fréquence du voisinage en fonction de la valeur de seuil. Pour une réponse rapide (seuil bas), la présence d'un voisin orthographique de haute fréquence a un effet facilitateur. Au contraire, lorsque la réponse produite est plutôt tardive, on observe un effet inhibiteur. Il est intéressant de remarquer que, dans l'étude B, le seuil de réponse choisi est juste avant la première inversion de l'effet produisant un effet facilitateur, et pour l'étude C juste après la seconde inversion de l'effet produisant un effet inhibiteur.

## 4.2 Simulation B06 : Effet de la fréquence du voisinage dans une tâche de reconnaissance visuelle de mot

Cette seconde série de simulations a pour objet l'effet de fréquence dans une tâche de reconnaissance visuelle de mot. La controverse observée en décision lexicale est également présente en reconnaissance visuelle de mot. Nous considérons donc deux études couvrant les deux effets contradictoires relatés dans la littérature. Dans l'étude A (Sears et al., 1995, expérience 2), la présence d'un voisin plus fréquent que le stimulus facilite l'identification de ce dernier pour les mots de basse fréquence et n'a aucun effet pour les mots de haute fréquence. En revanche, dans l'étude B (Perea & Pollatsek, 1998, expérience 2), la présence d'un voisin plus fréquent inhibe l'identification du stimulus. Avec ces simulations, nous cherchons à savoir si notre modèle peut rendre compte d'effets comportementaux opposés, comme c'est le cas pour la décision lexicale.

### 4.2.1 Expériences comportementales

Les études A et B reprennent à l'identique les conditions expérimentales, le matériel et la procédure des études A et B des simulations B05. Seule change la tâche à réaliser. Ici les participants doivent prononcer le mot le plus rapidement et le plus précisément possible.

**Résultats comportementaux** Dans l'étude A, les auteurs trouvent un effet principal de la fréquence du voisinage, sous la forme d'un avantage significatif de 4 ms pour les mots ayant un voisin plus fréquent. Ce résultat est obtenu en incluant les mots de basses fréquences non présentés ici. Si l'on considère uniquement les mots de basse fréquence, la présence d'un voisin plus fréquent est facilitateur pour les hautes densités de voisinage et inhibiteur pour les faibles densités de voisinage. Ces interactions sont significatives par participant et par item. Si l'on considère uniquement les mots de haute fréquence, il n'y a pas d'effet lié à la présence d'un voisin plus fréquent. Dans l'étude B, les auteurs trouvent un effet principal de la fréquence du voisinage. Cette fois, l'avantage observé de 26 ms est en faveur des mots n'ayant pas de voisin orthographique plus fréquent. Cet avantage est significatif par participant mais pas par item.

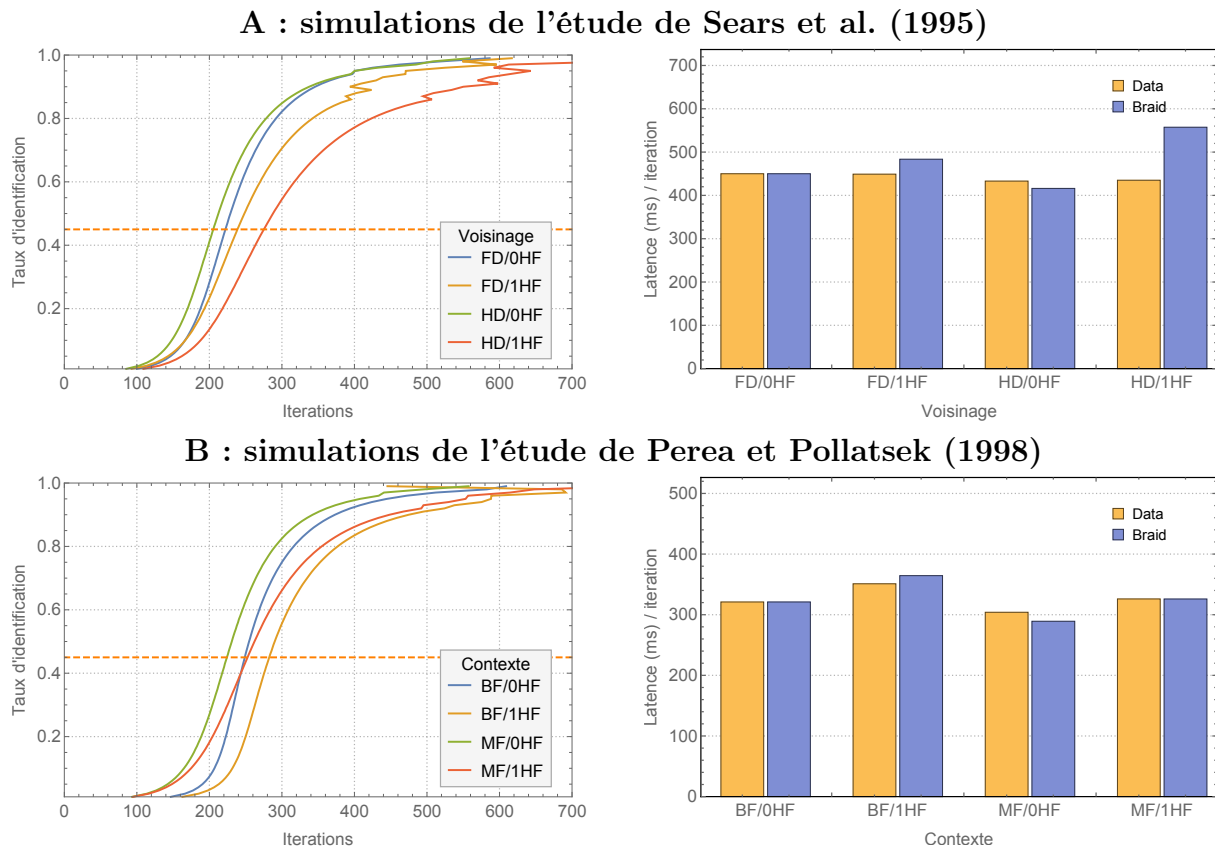


Figure 6.6 – **Simulation B06, effet de la fréquence du voisinage dans une tâche de reconnaissance visuelle de mot.** Résultats des simulations des expériences comportementales, en haut pour l'étude de Sears et al. (1995), et en bas pour l'étude de Perea et Pollatsek (1998). Les graphiques de gauche présentent les décours temporels de la réponse OUI, avec, en pointillés, l'itération de référence. Les graphiques de droite présentent conjointement les résultats des simulations de *BRAID* (en bleu) et les données expérimentales (en jaune).

#### 4.2.2 *BRAID* : Simulation B06

Si le stimulus  $w_i$  a un voisin  $w_v$  orthographique plus fréquent, ce dernier a sa probabilité  $P([W^t = w_v] | \dots)$  qui augmente dans les premières itérations, du fait de sa similarité avec le stimulus. Ainsi, il va ralentir l'augmentation de la probabilité du mot stimulus  $P([W^t = w_i] | \dots)$ . On peut donc s'attendre dans tous les cas à un effet inhibiteur de la présence d'un voisin plus fréquent.

**Matériel et procédure** Les stimuli utilisés sont les mêmes que ceux utilisés dans les simulations B05. La tâche réalisée par *BRAID* est la tâche de reconnaissance visuelle de mot (voir Chapitre 4, Section 1.2). Les résultats présentés dans la Figure 6.5 (à droite), ont été obtenus, pour l'étude A avec un seuil à 0,5, et pour l'étude B avec un seuil à 0,43. Les autres paramètres ont leur valeur par défaut.

**Résultats des simulations** L'analyse des courbes de décours temporel, Figure 6.6 à gauche, montre que la présence d'un voisin orthographique de plus haute fréquence (courbes orange et rouge) produit systématiquement une baisse des performances, quelles que soient les valeurs de seuil considérées. De ce fait, dans l'étude A, *BRAID* échoue à reproduire l'absence d'effet

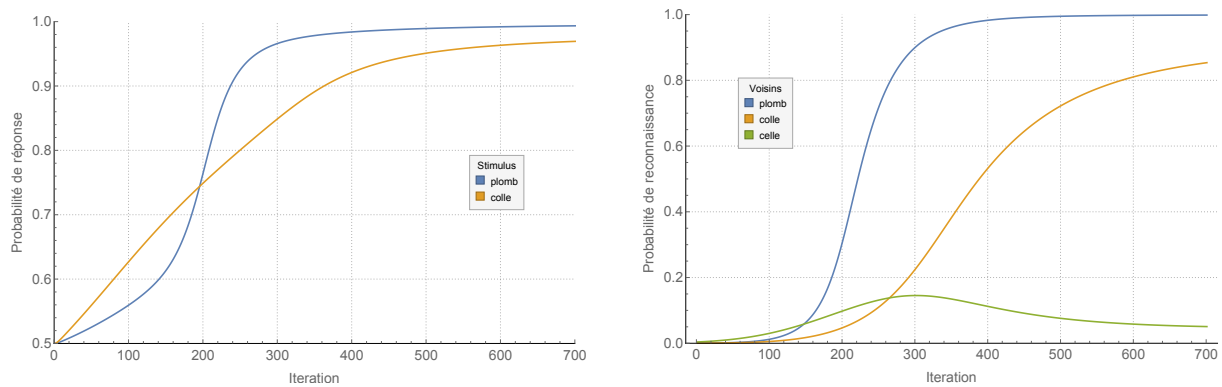


Figure 6.7 – **Illustration de l'effet de fréquence du voisinage en décision lexicale.** À gauche, simulation du déroulement temporel de la décision lexicale du mot *PLOMB* de fréquence 20,27, sans voisin orthographique (condition 0HF), et du mot *COLLE* de fréquence 20,2 qui a pour voisin *CELLE* de fréquence 299,46 (condition 1HF). À droite, simulation du déroulement temporel de la reconnaissance de ces deux mots, et de la reconnaissance du voisin orthographique *CELLE* durant la reconnaissance du stimulus *COLLE*.

observé dans l'expérience comportementale. En revanche, *BRAID* reproduit avec succès l'effet inhibiteur observé dans les différentes conditions de densité de voisinage de l'étude B.

**Conclusion** *BRAID* échoue à reproduire l'absence d'effet observé dans l'expérience comportementale de Sears et al. (1995), étude A, pour les mots de haute fréquence. En revanche, *BRAID* reproduit avec succès les variations observées dans l'expérience comportementale de Perea et Pollatsek (1998), étude B. La principale différence entre ces deux études est la fréquence des mots utilisés. Dans l'étude A, les mots utilisés ont de hautes fréquences d'occurrence, ce qui semble écraser l'effet de fréquence du voisinage dans l'expérience comportementale. Dans l'étude B, la fréquence d'occurrence des mots utilisés est soit faible, soit moyenne.

### 4.3 Discussion

Dans cette section nous avons étudié l'effet de la fréquence du voisinage orthographique sur les deux tâches de décision lexicale et de reconnaissance visuelle de mot. Les résultats obtenus reproduisent en partie les observations expérimentales.

Pour la tâche de décision lexicale (Simulations B05), *BRAID* simule, avec succès, deux résultats pourtant contradictoires : l'effet inhibiteur lié à la présence d'un voisin orthographique de plus grande fréquence, mis en évidence dans l'expérience B, et l'effet facilitateur trouvé dans l'expérience C. En revanche, *BRAID* échoue à rendre compte de l'absence d'effet, obtenu en décision lexicale, dans l'expérience A (Simulations B05). Ce résultat peut s'expliquer, en partie, par la fréquence des stimuli qui est haute, et qui « écrase » les différences liées à d'autres effets. Mais l'on peut malgré tout essayer de comprendre comment un voisin orthographique plus fréquent influence la décision lexicale. Pour cela, prenons l'exemple de la Figure 6.7. On y présente le déroulement temporel de la décision lexicale des mots *PLOMB*, condition 0HF, et *COLLE*, condition 1HF. En parallèle (Figure 6.7, partie droite), on visualise le déroulement temporel de la reconnaissance visuelle des mots *PLOMB*, *COLLE* et son voisin orthographique de plus haute fréquence *CELLE* lorsque le stimulus est *COLLE*. Durant les 200 premières itérations, on observe que la décision lexicale est à l'avantage du stimulus avec un voisin de haute fréquence. Cet avantage est essentiellement dû à la reconnaissance de son voisin orthographique *CELLE*

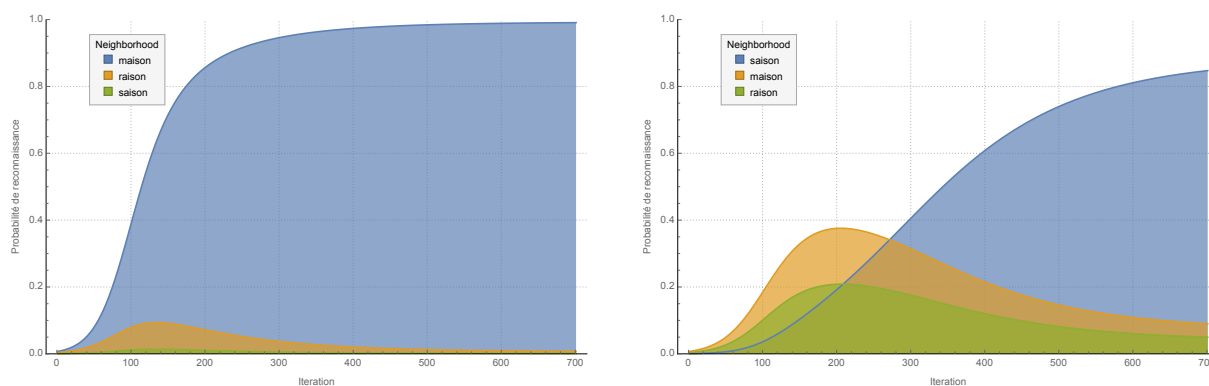


Figure 6.8 – **Illustration de l'effet de fréquence du voisinage en reconnaissance de mots.** À gauche, simulation du déroulement temporel de la reconnaissance du mot *MAISON* de fréquence 461,55, sans voisin orthographique plus fréquent. À droite, simulation du déroulement temporel de la reconnaissance du mot *SAISON* de fréquence 35,54, avec celles de ses voisins orthographiques plus fréquents, le mot *RAISON* de fréquence 247,5, et le mot *MAISON* de fréquence 461,55.

et pas du stimulus lui-même. Donc dans cet exemple, durant les 200 premières itérations, la présence d'un voisin orthographique plus fréquent a un effet facilitateur. Ensuite, la différence entre la probabilité d'identification du stimulus de la condition 0HF et les probabilités cumulées du stimulus de la condition 1HF et de son voisin est suffisamment en faveur du premier pour inverser l'avantage. On observe alors un effet inhibiteur du voisin de haute fréquence.

De même, pour la tâche de reconnaissance visuelle de mots (Simulations B06), le modèle simule l'effet inhibiteur observé dans l'étude B, mais échoue à rendre compte de l'absence d'effet observé dans l'étude A. Dans cette simulation, on constate que *BRAID* simule systématiquement un effet inhibiteur lié à la présence d'un voisin de plus haute fréquence. Ce résultat est observé, que la densité de voisinage soit haute ou faible. Pour comprendre ce qui conduit à ce résultat, prenons l'exemple de la Figure 6.8. Le déroulement temporel de la reconnaissance du mot *MAISON* (à gauche) montre que la présence de voisins orthographiques de basse fréquence a peu d'influence. En revanche (partie droite), la reconnaissance du mot *SAISON* qui a une fréquence de 35,54 est grandement ralentie par la présence de voisins orthographiques plus fréquents, *RAISON* de fréquence 247,5 et *MAISON* de fréquence 461,55. On constate donc que les voisins fréquents ont leur probabilité d'être reconnue qui augmente dans les premières itérations, lorsque l'identité des lettres n'est pas connue avec certitude. Mais dès que l'identité des lettres se précise, l'identité des voisins ne coïncide plus avec le stimulus et c'est le mot stimulus moins fréquent qui prend le dessus, jusqu'à l'identification. Dans le cas présenté, on constate que les deux voisins gardent, même après 700 itérations, une influence sur le résultat final.

## 5 Effet de la densité du voisinage

L'effet de la densité du voisinage est l'influence qu'a le nombre de voisins orthographiques sur le temps nécessaire à l'identification visuelle d'un mot. Cette influence n'est pas la même suivant la tâche demandée aux participants. Pour une tâche de décision lexicale, et lorsque le stimulus est un mot, l'effet de la densité du voisinage est massivement facilitateur dans la littérature (voir Andrews (1997) pour une revue). Nous étudions cet effet facilitateur dans nos



simulations B07. Cet effet s'inverse en décision lexicale lorsque le stimulus est un non-mot et que l'on mesure le temps nécessaire pour produire une réponse NON, ce que nous étudions dans les simulations B08. Enfin, l'influence du nombre de voisins est plus difficile à établir pour la tâche de reconnaissance visuelle de mot (simulations B09).

## 5.1 Simulation B07 : Effet facilitateur de la densité du voisinage dans une tâche de décision lexicale, en condition *mot*

Avec cette série de simulations, nous mettons en évidence l'effet facilitateur du nombre de voisins dans une tâche de décision lexicale. Pour cela nous simulons avec *BRAID* trois expériences comportementales. Deux expériences comportementales fondatrices réalisées par Andrews (1989, 1992), respectivement études A et B, auxquelles nous ajoutons les travaux de Pollatsek, Perea, et Binder (1999), étude C. Nous réutiliserons ces mêmes études dans la série de simulations B08 pour l'effet de la densité du voisinage en reconnaissance visuelle de mot.

### 5.1.1 Expérience comportementale

Dans les études A et B, Andrews (1989, 1992) utilise deux facteurs indépendants, la taille du voisinage, qui peut être de faible densité (FD) ou de haute densité (HD), et la fréquence du stimulus, qui peut être de basse fréquence (BF) ou de haute fréquence (HF). Dans la seconde étude, l'auteur contrôle en plus la fréquence des bigrammes, mais nous ne considérerons pas ce point dans ce travail. Dans l'étude B, les auteurs ne font varier que la densité de voisinage suivant deux modalités, FD ou HD.

**Matériel et procédure** Le matériel des trois études est présenté dans le Tableau 6.2. Les fréquences des études sont calculées avec les normes de la base de Kucera et Francis (1968) pour les études B et C, et avec les normes de la base de Carroll, Davies, et Richman (1971) pour l'étude A, tandis que les fréquences utilisées dans les simulations sont fournies par le lexique BLP. Enfin, les mots de l'étude C ont une fréquence inférieure à 34 par million, et ils ont tous un voisin plus fréquent.

La procédure est la même que celle décrite pour la série de simulations B05. Les mots sont présentés en lettres minuscules dans les trois études. Les participants sont informés avant l'expérience qu'ils doivent répondre le plus vite possible, en maintenant une bonne précision, (études A et B). Pour l'étude C, ils doivent porter une attention égale à la rapidité et à la précision de leurs réponses.

**Résultats comportementaux** Les résultats de l'étude A montrent un effet facilitateur de la taille du voisinage et une interaction entre la taille du voisinage et la fréquence des mots. Cependant, ces deux résultats sont significatifs par participant, mais pas par item. L'étude B reproduit les résultats de l'étude A, et les résultats y sont significatifs par participant et par item. Enfin, le design de l'étude C a conduit les auteurs à réaliser une analyse par item ; les résultats montrent un effet facilitateur de la taille du voisinage par participant et par item.

### 5.1.2 *BRAID* : Simulation B07

Comme pour l'effet de fréquence du voisinage (voir simulations B05), la densité du voisinage devrait faciliter la réponse OUI lorsque le stimulus est un mot, et que les lettres ne sont pas encore suffisamment identifiées pour que le mot correspondant au stimulus ait une forte probabilité dans la distribution sur  $W^t$ .

Tableau 6.2 – **Récapitulatif des caractéristiques des mots utilisés dans les expériences comportementales et les simulations des sections B07 et B09, faisant varier à la fois la fréquence du stimulus et la densité de son voisinage.** HF : Haute Fréquence et BF : Haute Fréquence.

	Densité du voisinage							
	Faible Densité				Haute Densité			
	Stimulus BF		Stimulus HF		Stimulus BF		Stimulus HF	
<b>Andrews</b> <b>(1989, expériences 1 &amp; 3)</b>	Data	Braid	Data	Braid	Data	Braid	Data	Braid
Fréquence	15,4	3,7	1472,6	304,8	16,1	5,9	1444,6	171,9
Taille du voisinage	2,2	4,7	2,5	4,3	12,5	14,8	13,5	15,1
Fréquence du voisinage	66,9		144,4		744,2		967,9	
<b>Andrews</b> <b>(1992, expérience 1)</b>	Data	Braid	Data	Braid	Data	Braid	Data	Braid
Fréquence	2,1	1,8	424,3	413,7	1,4	2,8	339,3	368,1
Taille du voisinage	3,6	5,9	3,4	5,1	12	14,3	11,3	14,3
Fréquence du voisinage	120,4		288,2		722,4		630,8	
<b>Pollatsek &amp; Perea</b> <b>(1999, expériences 1 &amp; 2)</b>	Data	Braid			Data	Braid		
Fréquence	7,4	11,6			8,6	12		
Taille du voisinage	2,2	2,9			8,5	9,7		
Fréquence du voisinage	409	392,9			345	348,2		
<b>Pollatsek &amp; Perea</b> <b>(1999, expérience 3)</b>	Data	Braid			Data	Braid		
Fréquence	20	18,1			19,2	16,6		
Taille du voisinage	0,6	1,8			4,8	6,		
Fréquence du voisinage	111	110,8			75	65,9		

**Matériel et procédure** Les stimuli utilisés dans ces trois simulations sont les mêmes que ceux utilisés lors des expériences comportementales. Les caractéristiques des mots utilisés par le modèle sont présentées dans le Tableau 6.2.

La tâche réalisée par *BRAID* est la tâche de décision lexicale (voir Chapitre 4, Section 1.4). Les résultats présentés dans la Figure 6.9 (à droite), ont été obtenus, pour l'étude A avec un seuil à 0,77, pour l'étude B avec un seuil à 0,85 et pour l'étude C avec un seuil à 0,76. Les autres paramètres ont leur valeur par défaut.

**Résultats des simulations** L'analyse des courbes de décours temporel (Figure 6.9, à gauche) montre qu'une haute densité de voisinage (courbe orange et rouge) produit un avantage dans les premières itérations. Cela est particulièrement visible dans la condition de basse fréquence des stimuli des simulations des études A et B. Puis l'avantage s'inverse et les performances deviennent meilleures pour les mots ayant une faible densité de voisinage (courbe bleue et verte).

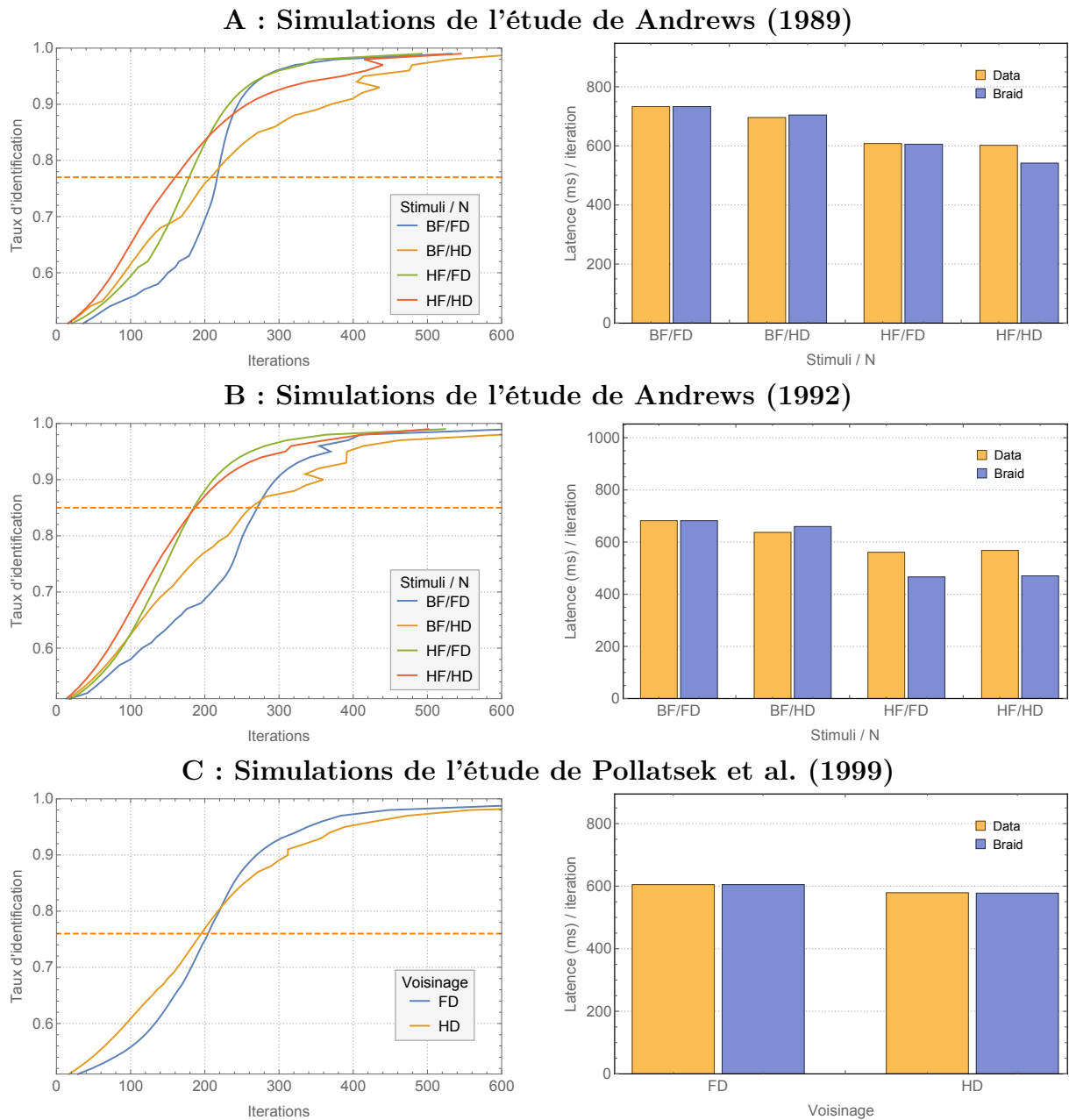


Figure 6.9 – **Simulation B07, effet de la densité du voisinage dans une tâche de décision lexicale, condition *mot***. Résultats des simulations des expériences comportementales. La première ligne donne les résultats de l'étude A, Andrews (1989), la deuxième ligne donne les résultats de l'étude B, Andrews (1992) et la troisième ceux de l'étude C, (Pollatsek et al., 1999). Chaque ligne présente, à gauche, le déroulement temporel de la réponse OUI avec en pointillés la valeur du seuil de décision. À droite sont présentés conjointement les résultats des simulations de *BRAID* (en bleu) et les données expérimentales (en jaune).

**Conclusion** Comme on l'avait déjà observé pour l'effet de fréquence, cette série de simulations met en évidence une inversion de l'effet de la densité du voisinage en fonction de la valeur de seuil de décision choisi. Pour les trois expériences comportementales que nous simulons, le fait de placer le seuil de décision assez bas nous permet de reproduire fidèlement les résultats observés. Ces résultats laissent entrevoir qu'il serait possible d'observer des résultats inverses, pour lesquels la densité du voisinage aurait un effet inhibiteur.

## 5.2 Simulation B08 : Effet inhibiteur de la densité du voisinage dans une tâche de décision lexicale, en condition *non-mot*

Nous nous intéressons maintenant au cas où le stimulus est un *non-mot* et la réponse attendue est NON. La littérature tend à montrer que la densité du voisinage a un effet inhibiteur sur les performances. Nous prenons comme référence une étude de Forster et Shen (1996).

### 5.2.1 Expérience comportementale : Forster et Shen (1996, expérience 2)

Dans cette étude, les auteurs utilisent trois facteurs indépendants : la taille du voisinage qui varie suivant quatre modalités (0, 1, 2, 3/4), la fréquence du voisinage (BF et HF) et enfin la nature du stimulus (mot et non-mot).

**Matériel et procédure** Le matériel est composé de 240 stimuli : 120 mots et 120 non-mots, avec systématiquement 30 mots par conditions de voisinage. Les mots utilisés sont tous de basse fréquence, comprise entre six et sept occurrences par million suivant la norme de la base de Kucera et Francis (1968).

La tâche demandée aux participants est une tâche de décision lexicale. C'est le participant qui démarre l'essai en appuyant sur une pédale. Il n'y a pas de point de fixation et le stimulus apparaît immédiatement après, systématiquement au centre de l'écran. La durée de présentation du stimulus est fixée à 500 ms pour tous les essais. Les participants reçoivent comme instruction de prêter autant d'attention à la précision de leur réponse qu'à la rapidité.

**Résultats comportementaux** Les auteurs réalisent des études statistiques distinctes pour les *mot* et pour les *non-mot*. Ils observent un effet facilitateur de la densité du voisinage en condition *mot*, significatif par participant mais pas par item. De même, ils observent un effet inhibiteur de la densité du voisinage en condition *non-mot*, cette fois significatif par participant et par item. Enfin, les auteurs n'observent pas d'effet de la fréquence du voisinage.

### 5.2.2 BRAID : Simulation B08

Si, dans les premières itérations, la décision en faveur du OUI est prise essentiellement sur la base de l'information en provenance du voisinage du stimulus, on peut penser que cette décision sera plus ou moins la même que le stimulus soit un *mot* ou un *non-mot*. Ceci aura pour conséquence de retarder la réponse NON à un stimulus *non-mot*.

**Matériel et procédure** Les stimuli utilisés dans ces trois simulations sont les mêmes que ceux utilisés lors des expériences comportementales. Cependant, par souci de clarté, nous avons regroupé les items suivant leur nature (*mot* et *non-mot*) et suivant la densité du voisinage, sans tenir compte de la fréquence.

La tâche réalisée par *BRAID* est la tâche de décision lexicale (voir Chapitre 4, Section 1.4). Les résultats présentés (Figure 6.10, à droite) ont été obtenus, pour la condition *mot* avec un seuil à 0,66, et pour la condition *non-mot* avec un seuil à 0,53. Nous avons utilisé un seuil volontairement bas pour la condition *non-mot* afin d'aligner le nombre d'erreurs que commet *BRAID* en condition *non-mot* sur les observations expérimentales.

**Résultats des simulations** L'analyse des courbes de décours temporel en condition *mot* (Figure 6.10, en haut à gauche) montre que la présence de voisins orthographiques (1 ou 2) produit un léger avantage par rapport à la condition sans voisinage (courbe bleue) pour des valeurs de seuils basses. Cet avantage présent également pour la condition 3/4 voisins (courbe rouge) est beaucoup plus prononcé que pour les deux autres conditions, ce qui ne semble pas être le cas dans les observations comportementales. Enfin, on observe que cet avantage s'inverse après environ 220 itérations (la courbe bleue passe au dessus), et que la différence observée entre les trois conditions de voisinage s'inverse également (la courbe rouge passe au dessous).

L'analyse des courbes de décours temporel en condition *non-mot* (Figure 6.10, en bas à gauche) montre le schéma inverse. Les non-mots sans voisins orthographique sont reconnus plus rapidement que les autres. Mais cette tendance n'est valable que pour des seuils de décision très bas (compris entre 0,5 et 0,62). Pour les valeurs de seuil supérieures, la tendance s'inverse. Les dents de scie que l'on observe sur ces courbes révèlent qu'au moins un des non-mots pris en compte dans le calcul de la moyenne n'a pas atteint le seuil et est donc considéré comme non reconnu.

**Conclusion** *BRAID* reproduit avec succès la diminution des temps de réaction consécutive à l'augmentation de la densité du voisinage, observée dans la condition *mot* de l'expérience comportementale de Forster et Shen (1996). *BRAID* reproduit également avec succès l'augmentation globale des temps de réaction liée à l'augmentation de la densité du voisinage, observée dans la condition *non-mot* de cette même étude. On observe cependant une diminution des temps de réaction entre les condition 1 et 2 voisins, ce qui n'est pas le cas dans l'expérience comportementale.

### 5.3 Simulation B09 : Effet de la densité du voisinage dans une tâche de reconnaissance visuelle de mot

Dans cette troisième série de simulations, nous nous intéressons à l'effet de la densité du voisinage dans une tâche de reconnaissance visuelle de mot. On retrouve ici une controverse, similaire à celle concernant l'effet de la fréquence du voisinage. Nous considérons donc trois études pour avoir un éventail complet des effets contradictoires, qui font référence dans la littérature. Dans l'étude A (Andrews, 1989, expérience 3) l'effet de la densité du voisinage est inhibiteur. Dans l'étude B (Pollatsek et al., 1999, expérience 2) les auteurs trouvent un effet facilitateur qu'ils annulent dans l'étude C (Pollatsek et al., 1999, expérience 3). Avec ces trois simulations, nous cherchons à savoir si notre modèle peut rendre compte de ces effets en apparence contradictoires.

#### 5.3.1 Expériences comportementales

Dans l'étude A (Andrews, 1989), les auteurs reprennent les conditions expérimentales, le matériel et la procédure de l'étude A de la série de simulations B07. Seule change la tâche à réaliser. Ici les participants doivent prononcer le mot le plus rapidement et le plus précisément possible.

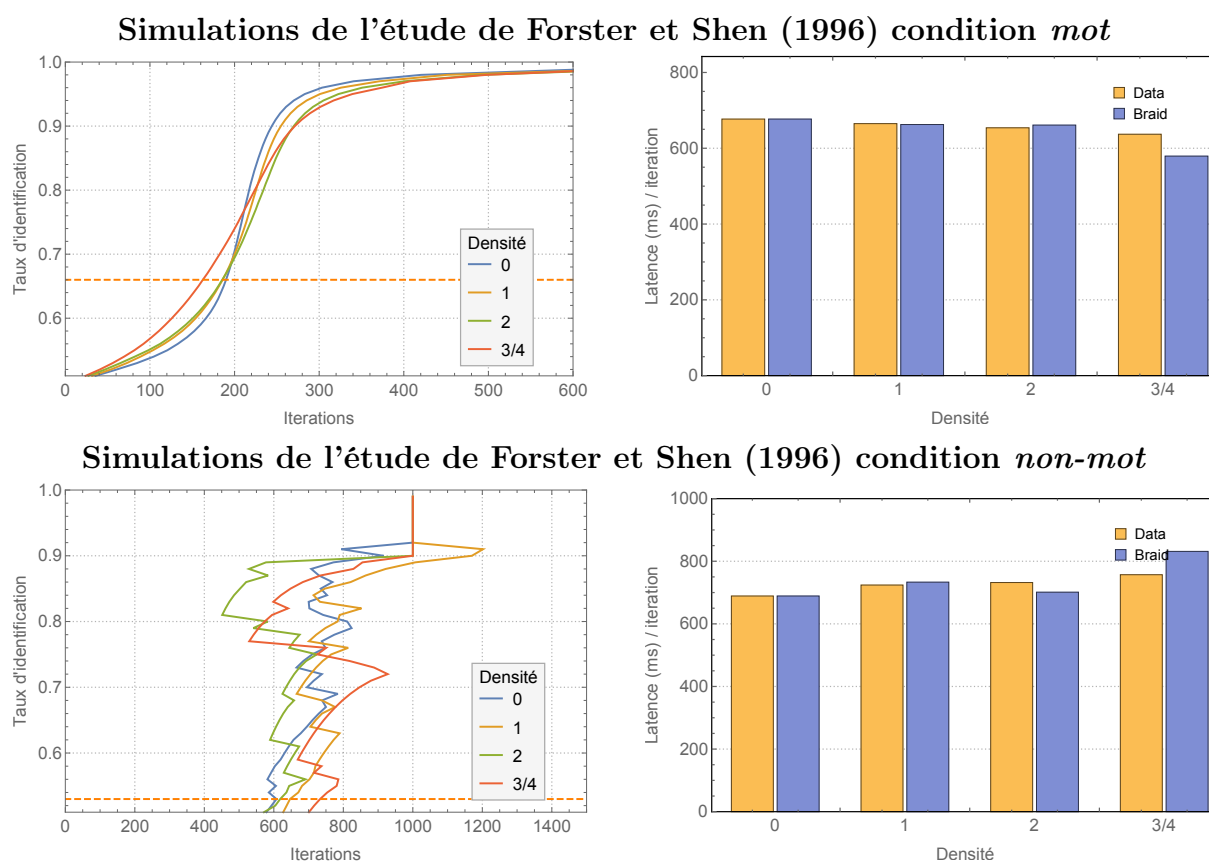


Figure 6.10 – **Simulation B08, effet inhibiteur de la densité du voisinage dans une tâche de décision lexicale, en condition *non-mot*.** La première ligne présente les résultats en condition *mot* et la seconde ligne les résultats en condition *non-mot*. Chaque ligne présente, à gauche, le déroulement temporel de la réponse OUI avec en pointillés l'itération de référence. À droite sont présentés conjointement les résultats des simulations de *BRAID* (en bleu) et les données expérimentales (en jaune).

Pour les études B et C, la procédure est totalement différente. Les auteurs placent les mots dans des phrases et prennent différentes mesures durant la lecture à voix haute, à l'aide d'un oculomètre. Dans l'étude B, les auteurs font varier la densité du voisinage des mots présentés, sans contrôler la fréquence du voisinage, ce qui leur fait suspecter que l'effet facilitateur observé est dû à la fréquence plutôt qu'à la densité. Dans l'étude C, les auteurs contrôlent la fréquence du voisinage et le nombre de voisins plus fréquents (0,9 en moyenne dans les deux conditions).

**Matériel et procédure** Les caractéristiques des stimuli de ces trois études sont présentées dans le Tableau 6.2. La tâche réalisée dans l'étude A est une tâche de dénomination classique. Pour les études B et C, les auteurs mesurent, entre autres la durée de la première fixation, la durée totale du temps de fixation sur la cible et la probabilité de sauter le mot cible. Nous avons choisi comme mesure de référence la durée totale de fixation de la cible.

**Résultats comportementaux** Dans l'étude A, les auteurs trouvent un effet facilitateur de la densité du voisinage significatif par participant et par item. De plus, l'effet simple de la densité du voisinage est significatif en condition BF (49 ms) mais pas en condition HF (10 ms). Dans l'étude B, les auteurs trouvent un effet inhibiteur de 24 ms de la densité du voisinage

significatif par participant et par item. Enfin, dans l'étude C, les auteurs ne trouvent pas que l'avantage observé en condition FD sur la condition HD (5 ms) soit significatif.

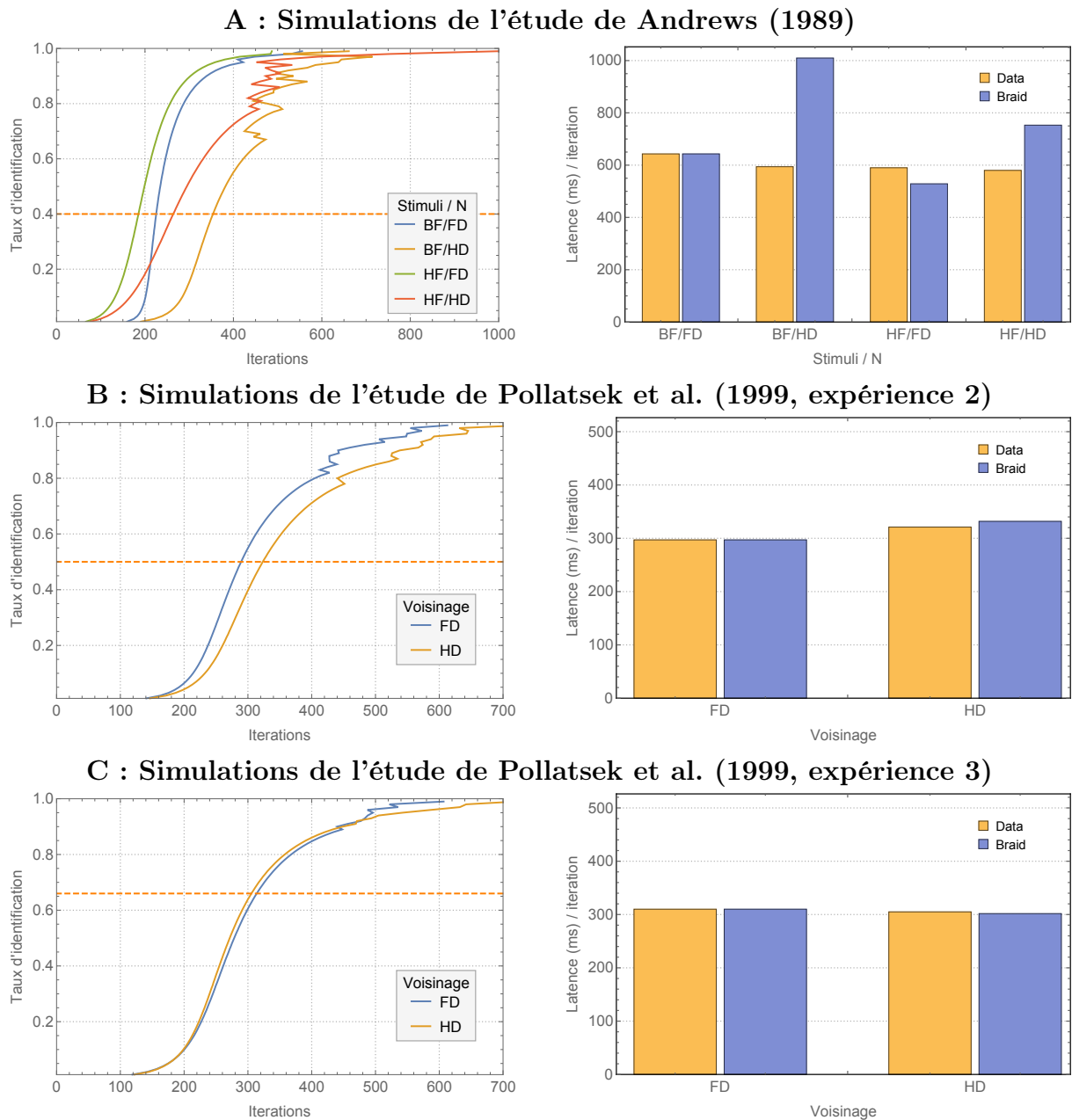


Figure 6.11 – **Simulation B09, effet de la fréquence du voisinage dans une tâche de reconnaissance visuelle de mot.** Résultats des simulations des expériences comportementales. La première ligne donne les résultats de Andrews (1989), la seconde ligne donne les résultats de Pollatsek et al. (1999, expérience 2), la troisième ligne donne les résultats de Pollatsek et al. (1999, expérience 3). Chaque ligne présente, à gauche, le décours temporel de la réponse OUI avec en pointillés l'itération de référence. À droite sont présentés conjointement les résultats des simulations de *BRAID* (en bleu) et les données expérimentales (en jaune).

### 5.3.2 *BRAID* : Simulation B09

Si le stimulus a une forte densité de voisins, durant les premières itérations, où le modèle dispose de peu d'information sur les lettres, sa probabilité ainsi que celles de ses voisins augmentent lentement, créant de fait un effet inhibiteur. Puis, avec l'augmentation de l'information sur l'identité des lettres, la probabilité des voisins arrêter d'augmenter puis décroît, de sorte que l'on devrait observer un effet inhibiteur pour toutes les itérations.

**Matériel et procédure** Les stimuli utilisés sont les mêmes que ceux utilisés dans les simulations B07. La tâche réalisée par *BRAID* est la tâche de reconnaissance visuelle de mot (voir Chapitre 4, Section 1.2). Les résultats présentés (Figure 6.11, à droite), ont été obtenus, pour l'étude A avec un seuil à 0,4, pour l'étude B avec un seuil à 0,5, et pour l'étude C avec un seuil de 0,66. Pour les études B et C, les seuils choisis sont ceux qui minimisent l'écart entre les données expérimentales et les résultats des simulations brutes, sans mise à l'échelle<sup>14</sup>. Les autres paramètres ont leur valeur par défaut.

**Résultats des simulations** Dans les études A et B, les courbes de décours temporel (Figure 6.11, à gauche) montrent que les conditions avec une haute densité de voisinage (courbes orange et rouge) produisent en moyenne de moins bonnes performances que celles avec une faible densité (courbes bleue et verte), quelles que soient les valeurs de seuil considérées. Dans l'étude C, le résultat est inversé mais la différence entre les deux courbes de performance est petite, et le résultat s'inverse pour un seuil de 0,90.

**Conclusion** *BRAID* échoue à reproduire l'effet facilitateur observé dans l'expérience comportementale de Andrews (1989), étude A. En revanche, *BRAID* reproduit avec succès l'avantage d'avoir peu de voisins orthographiques, observé dans l'expérience comportementale de Pollatsek et al. (1999), étude B. La principale différence entre ces deux études est la fréquence des mots utilisés. Dans l'étude A, les mots utilisés sont de haute fréquence d'occurrence, ce qui semble écraser l'effet de fréquence du voisinage dans l'expérience comportementale. Dans l'étude B, les mots utilisés ont soit une faible fréquence d'occurrence, soit une moyenne fréquence d'occurrence.

## 5.4 Discussion

Dans cette section nous avons étudié l'effet de la densité du voisinage orthographique sur deux tâches classiques : la décision lexicale, en étudiant séparément les réponses correctes OUI et NON, et la reconnaissance visuelle de mot. Les résultats obtenus reproduisent en partie les résultats observés expérimentalement.

Pour la tâche de décision lexicale en condition *mot* (Simulations B07), *BRAID* simule avec succès l'effet facilitateur lié à un voisinage orthographique dense, et ce résultat attendu est reproduit dans trois études différentes. Mais comme dans l'effet de fréquence vu Section 4, cet effet s'inverse sur les courbe de décours temporel après un certain nombre d'itérations. Ceci signifie que, lorsque l'on décide de la nature du stimulus avec un niveau de certitude assez faible, avoir un grand nombre de voisins accélère la décision. Mais dès que l'on se donne un seuil de décision plus élevé (0,85 dans les exemples présentés), avoir un grand nombre de voisins

<sup>14</sup>D'ailleurs, le modèle reproduit bien les données expérimentales sans mise à l'échelle avec un RMSE de 5,23 pour l'étude B et un RMSE de 1,66 pour l'étude C. Ces excellents résultats sont sans doute liés à la nature de la tâche de ces deux études (tâche de lecture) qui correspond à la méthode de calibrage du modèle.



devient pénalisant. Ce résultat est reproduit dans les Simulations B08 en condition *mot*. Dans cette simulation, un faible avantage est déjà observé avec un unique voisin, mais 3/4 voisins sont nécessaires pour que la différence de performance soit marquée. L'explication de l'effet de la densité du voisinage est similaire à l'explication de l'effet de fréquence de voisinage de la Section 4. Dans ce dernier cas, comme on l'a montré précédemment, c'est le voisin très fréquent qui porte l'essentiel de la masse de probabilité durant les premières itérations. Pour l'effet de la densité du voisinage, c'est la « somme » des voisins qui porte l'essentiel de la masse de probabilité. Et dans les deux cas, une fois que l'identité des lettres se précise suffisamment, c'est le mot qui correspond au stimulus qui accapare la masse de probabilité.

Pour la tâche de décision lexicale en condition *non-mot* (Simulations B08), *BRAID* simule avec succès l'effet inhibiteur de la densité du voisinage sur la réponse correcte NON. On constate sur les courbes de décours temporel (Figure 6.10) que *BRAID* commet beaucoup d'erreurs liées à la limite de 1500 ms imposée dans les simulations. Ce résultat est dû à notre choix de considérer comme non-reconnus, les non-mots qui ne font pas l'objet d'une décision en moins de 1500 ms. Il est probable que les participants n'adoptent pas la même stratégie. Par exemple, lorsque le seuil de décision prédéfini n'est pas atteint mais que les 1500 ms de présentation sont écoulées, il pourraient faire un tirage aléatoire suivant leur distribution sur la variable d'appartenance lexicale  $D^t$ , et répondre en conséquence. Il se pourrait également que le seuil évolue, et diminue avec le temps (Boehm, Hawkins, Brown, van Rijn, & Wagenmakers, 2016 ; Drugowitsch & Pouget, 2012). Quelle que soit la stratégie adoptée, elle conduirait à produire moins d'erreur que ne le prédit *BRAID*. Cependant, ces deux stratégies reposant sur la probabilité  $P([D^t = \text{emphfaux}] | \dots)$ , cela ne devrait pas changer les effets observés, mais seulement les « lisser ».

Enfin, pour la tâche de reconnaissance visuelle de mots (Simulations B08), *BRAID* simule avec succès l'effet inhibiteur de la densité du voisinage de l'étude B et l'absence d'effet de l'étude C, mais échoue à montrer l'effet facilitateur de l'étude A. Dans cette série de simulations, on constate que *BRAID* simule un effet inhibiteur lié à une forte densité de voisinage, sauf pour l'étude C. Dans cette étude, les mots utilisés sont de fréquence moyenne et les écarts de densités de voisinages n'apparaissent pas suffisamment différents, ce qui pourrait expliquer l'absence d'effet observée. En effet, si l'on compare les données utilisées dans l'étude C avec celles de l'étude A par exemple (voir Tableau 6.2), on observe des fréquences moyennes des mots plus faibles pour l'étude A, facilitant l'émergence des effets, et des différences de densité de voisinage plus importantes dans l'étude A : 4,7 en FD et 14,8 en HD, contre 1,8 en FD et 6 en HD pour l'étude C. Pour comprendre l'effet inhibiteur produit par *BRAID*, on peut s'appuyer sur la Figure 6.12. Le décours temporel de la reconnaissance du mot *VERTE* (à gauche) montre que la présence de nombreux voisins orthographiques limite la progression de l'identification du stimulus. En revanche, la reconnaissance du mot *VERTS* (à droite), qui n'a que trois voisins, n'est que peu ralentie. On voit également que ce ralentissement est observable tout au long du processus de reconnaissance, la pente restant assez faible, comparativement au mot *VERTS*. Cette dernière remarque explique que l'effet inhibiteur produit par *BRAID* est un effet robuste du modèle.

## 6 Effet d'interaction entre la fréquence des stimuli et leur voisinage

Nous avons vu, Section 2, que la fréquence avait un effet massif sur les performances en décision lexicale (simulation B02). De plus, nous avons montré, Section 4, que cet effet pouvait être modulé par la présence d'un voisin plus fréquent ou, Section 5, par une forte densité de voisinage.

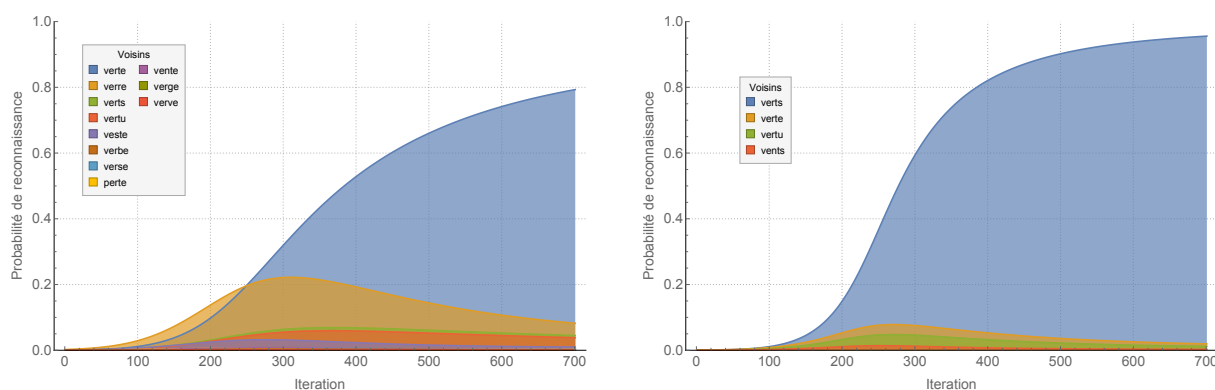


Figure 6.12 – **Illustration de l’effet de densité du voisinage en reconnaissance de mots.** À gauche, simulation du déroulement temporel de la reconnaissance du mot *VERTE* qui possède 10 voisins orthographiques. À droite, simulation du déroulement temporel de la reconnaissance du mot *VERTS* qui possède seulement 3 voisins orthographiques.

Siakaluk et al. (2002) ont manipulé ces variables dans une même étude que nous présentons ici. Nous cherchons à savoir si *BRAID* est également capable de rendre compte des interactions qui se produisent entre ces différentes variables.

## 6.1 Simulation B10 : les effets d’interaction fréquence-voisinage

Avec cette simulation nous faisons une synthèse des résultats que nous avons présentés sur l’influence du voisinage du stimulus en décision lexicale, puis nous nous intéressons aux interactions entre fréquence du stimulus, densité et fréquence du voisinage.

### 6.1.1 Expérience comportementale : Siakaluk et al. (2002, expérience 1)

Cette expérience comportementale a pour objectif initial de mettre en évidence l’influence des non-mots sur les performances en décision lexicale. Pour cela, les auteurs utilisent quatre contextes de non-mots, variant sur la densité de leur voisinage. Les trois premiers contextes, sans voisin (1A), faible voisinage (1B) et voisinage apparié entre mot et non-mot (1C), ne changent pas le sens des effets observés. Seule le contexte forte densité (1D) modifie le sens des effets. Nous avons choisi d’utiliser les résultats obtenus avec le contexte 1B, celui dans lequel les non-mots ont une densité de voisinage faible, entre 1 et 5 voisins. Cette condition étant fixée, nous nous intéressons aux performances en décision lexicale, dans les différentes conditions expérimentales proposées.

**Matériel et procédure** La tâche demandée aux sujets de cette expérience est une tâche de décision lexicale. Les mots et les non-mots utilisés sont, dans chaque condition, pour moitié des mots de 4 lettres, et pour moitié des mots de 5 lettres. Les stimuli sont regroupés suivant trois caractéristiques : leur fréquence (haute HF ou basse BF), la densité de leur voisinage (faible FD ou haute HD) et enfin la présence d’un voisin plus fréquent (absent 0HF ou présent 1HF). Les caractéristiques des stimuli sont présentées dans le Tableau 6.3. Le matériel est fourni par Siakaluk et al. (2002). Deux mots ne sont pas présents dans le lexique BLP et ont été exclus des simulations : *TRUE* et *CLUE*.

Tableau 6.3 – **Récapitulatif des caractéristiques des mots utilisés dans l’expérience comportementale et les simulations de la section B10.** HF : Haute Fréquence, BF : Basse Fréquence. La fréquence du voisinage correspond à la moyenne sur tous les mots du voisin le plus fréquent.

Stimulus	mots BF				mots HF			
	voisinage FD		voisinage HD		voisinage FD		voisinage HD	
	Data	Braid	Data	Braid	Data	Braid	Data	Braid
Absence de voisin de plus haute fréquence								
Fréquence	19,7	16,2	24,4	24,3	113	102,2	105,8	120,5
N	3,3	3,3	8,6	8	3,1	3	9,5	10
NF	8,2	9,5	13,9	15,5	28,1	43,4	58	67,4
Présence d’un voisin de plus haute fréquence								
Fréquence	20,2	19,9	20,4	18,2	98,3	133,9	105,3	103,7
N	3,5	4,2	10,2	10,5	3,3	3,9	10,8	11,2
NF	288,1	435,6	315,5	330,3	261,7	289,8	328,9	417,7

**Résultats comportementaux** Les résultats comportementaux reproduisent les résultats observés dans les études des précédentes sections. Ainsi, l’effet principal de fréquence des stimuli est significatif par sujet et par item, et les stimuli HF sont identifiés 43 ms plus rapidement que les stimuli BF. L’effet principal de la densité du voisinage est significatif par sujet, mais pas par item, et les stimuli HD sont identifiés 8 ms plus rapidement que les stimuli FD. De même, l’effet principal de la fréquence du voisinage est significatif par sujet, mais pas par item, et les stimuli 1HF sont identifiés 7 ms plus rapidement que les stimuli 0HF. L’interaction entre la fréquence du stimulus et la densité de son voisinage est significative uniquement par sujet. En basses fréquences, un avantage de 14,5 ms est observé en condition HD alors qu’il n’y a pas d’effet pour les hautes fréquences. Les résultats sont identiques si l’on considère l’interaction entre la fréquence du stimulus et la fréquence du voisinage, avec un avantage de 12,5 ms pour les stimuli ayant au moins un voisin plus fréquent, significatif par sujet. Enfin, il n’y a pas d’interaction entre densité du voisinage et fréquence du voisinage.

### 6.1.2 *BRAID* : Simulation B10

Cette simulation étant, pour partie, une réplique des simulations précédentes, on s’attend à retrouver l’effet de fréquence, l’effet de densité et de fréquence du voisinage. De plus, nous avons montré précédemment que la fréquence des stimuli avait un effet massif sur les performances. En particulier, on s’attend à observer des effets d’interaction entre la fréquence du stimulus et le voisinage (fréquence et densité).

**Matériel et procédure** Les stimuli utilisés dans cette simulation sont les mêmes que ceux utilisés lors des expériences comportementales. La tâche réalisée par *BRAID* est la tâche de décision lexicale (voir Chapitre 4, Section 1.4). Les résultats présentés Figure 6.13 ont été obtenus avec un seuil fixé à 0,75.

**Résultats des simulations** L’analyse des courbes de décours temporel (Figure 6.13, en haut) nous permet de retrouver les résultats observés précédemment. Durant les premières itérations, la présence d’un voisin de plus haute fréquence donne un net avantage. Nous observons ainsi un effet de la fréquence, très marqué autour du seuil de décision 0,8. Passé ce seuil, nous obser-

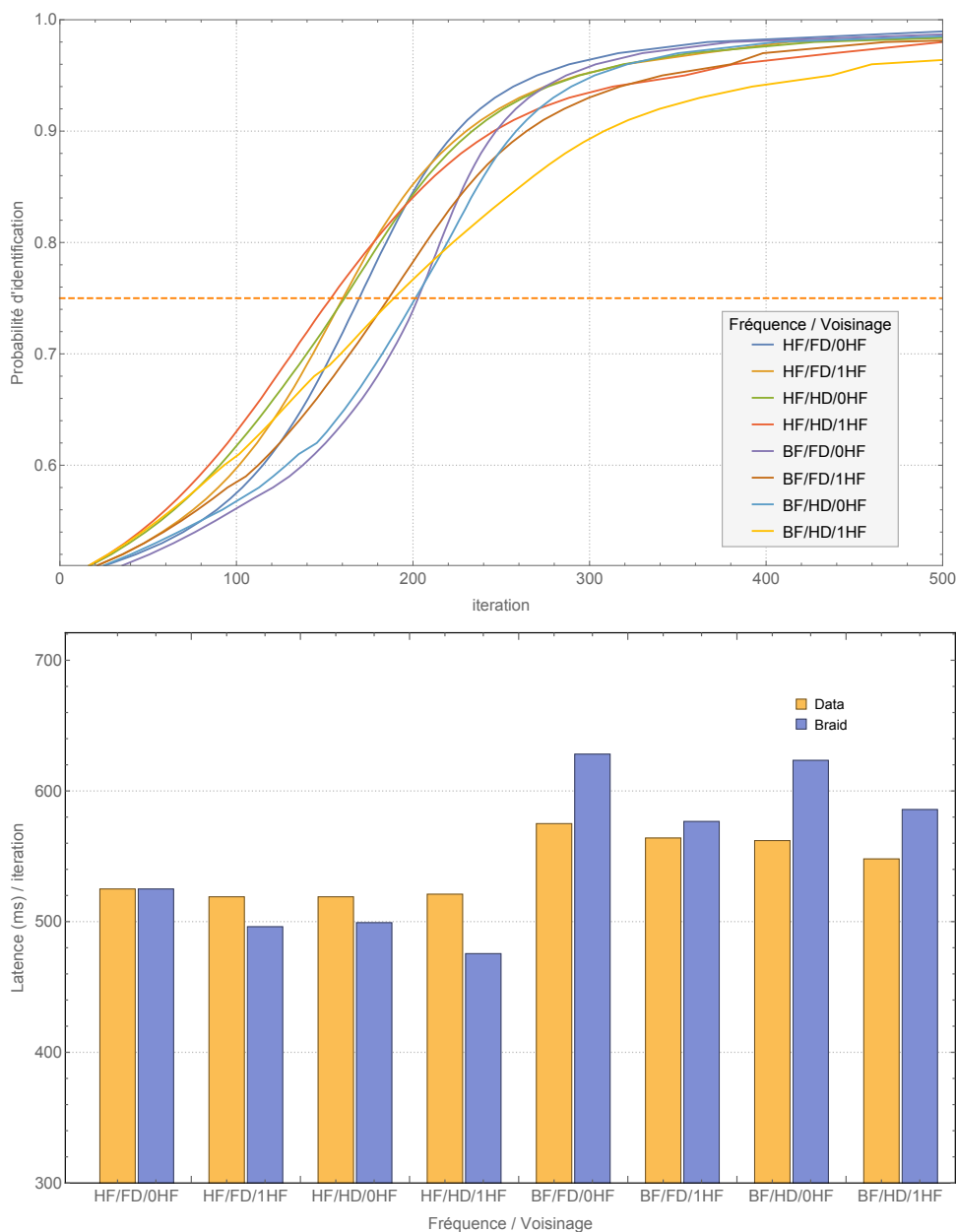


Figure 6.13 – **Simulation B10, effet d’interaction entre la fréquence des stimuli et leur voisinage en décision lexicale.** En haut, on présente les décours temporels de la décision lexicale pour chacune des huit conditions expérimentales, et la ligne en pointillés orange matérialise le seuil de décision choisi (0,75). En bas, on présente conjointement les résultats des simulations de *BRAID* (en bleu) et les données expérimentales (en jaune).

vons une inversion de performance avec un avantage pour les mots sans voisin plus fréquent et avec une faible densité de voisinage. Nous comparons (Figure 6.13, en bas) les performances de *BRAID* au seuil de décision 0,75 avec celles observées expérimentalement. Pour rappel, ce seuil est choisi en minimisant la racine carrée de l’écart quadratique moyen entre les données des simulations et les données comportementales. Nous observons une interaction entre la fréquence des stimuli et la présence d’un voisin plus fréquent. En basse fréquence, la différence de performance liée à la présence d’un voisin plus fréquent est plus marquée qu’en haute fréquence. De même, on observe une interaction entre la fréquence des stimuli et la densité du voisinage,

avec un effet nul en haute fréquence qui n'est présente ni dans les données comportementales, ni dans les valeurs de seuils inférieures au seuil choisi.

**Conclusion** Les résultats présentés dans cette simulation nous montrent que *BRAID* rend compte des effets principaux de fréquence du stimulus, de densité et de fréquence du voisinage. Ces résultats sont robustes même si l'effet de fréquence des stimuli est plus important dans les simulations que dans les observations expérimentales. *BRAID* rend également compte des interactions observées expérimentalement, comme l'interaction entre la fréquence des stimuli et la fréquence du voisinage, ainsi que l'interaction entre la fréquence des stimuli et la densité du voisinage. Cependant, *BRAID* ne montre pas d'interaction entre la fréquence et la densité du voisinage, ce qui correspond aux résultats expérimentaux.

## 7 Discussion sur la reconnaissance de mots et la décision lexicale

Dans ce chapitre, nous avons simulé dix-huit résultats d'expériences comportementales, regroupés en dix simulations, et portant sur la reconnaissance visuelle de mots et sur la décision lexicale. Nous avons commencé avec l'effet le plus robuste, et probablement le plus communément admis de la littérature traitant de la reconnaissance de mot : l'effet de fréquence. Avec la simulation B01 nous nous sommes intéressés à l'effet de fréquence dans une tâche de reconnaissance visuelle de mots et avec la simulation B02 nous nous sommes intéressés à l'effet de fréquence dans une tâche de décision lexicale. Dans ces deux simulations, notre modèle rivalise avec les performances du BR (Norris, 2006) et montre que le nombre d'itérations nécessaire pour reconnaître un mot diminue linéairement avec le logarithme de la fréquence d'occurrence, même si *BRAID* ne fait pas de différences et montre un plateau pour les fréquences inférieure à 5. De plus, grâce aux courbes de décours temporel, nous avons montré que ce résultat était robuste et pratiquement indépendant du seuil choisi. Pour finir, nous avons précisé le rôle de la fréquence en montrant qu'elle agit en donnant un avantage temporel aux mots les plus fréquents et en facilitant leur traitement.

Ensuite nous nous sommes intéressés spécifiquement à la décision lexicale. Dans la simulation B03, nous avons simulé avec succès l'effet de statut lexical, pour lequel on met moins de temps à reconnaître un mot qu'à rejeter un non-mot. Ce résultat a été obtenu pour des mots très peu fréquents, mettant en avant la robustesse de cet effet. Dans la simulation B04, nous avons simulé avec succès l'effet de légalité des non-mots, pour lequel on rejette plus rapidement une chaîne de caractère orthographiquement illégale qu'une chaîne de caractères légale. Cependant, nous nous sommes rendus compte que *BRAID* était moins performant à rejeter des non-mots que ne le montrent les résultats comportementaux. Cette relative lenteur est à imputer aux choix de modélisation que nous avons faits. En effet, la décision de rejeter un stimulus comme étant un mot se base sur un mécanisme de détection d'erreur, mais qui ne considère qu'une erreur au plus dans le stimulus. Nous perdons ainsi une partie du bénéfice que pourrait apporter la présence de plusieurs erreurs, comme c'est le cas dans la plupart des stimuli non-mots utilisés. Néanmoins, ces bons résultats nous ont permis d'établir la fiabilité de la décision lexicale dans *BRAID*.

Bien que l'effet de fréquence soit un effet massif, il peut être modulé par le voisinage du stimulus. Dans un premier temps, nous avons considéré l'effet de la présence d'un voisin orthographique plus fréquent sur les performances d'identification. Dans la simulation B05, nous avons utilisé la tâche de décision lexicale. Nous avons montré que la présence d'un voisin

orthographique plus fréquent a un effet facilitateur durant les premières itérations (autour de 200 ms), ce qui implique un choix qui repose sur un seuil de décision bas. Mais cet effet s'inverse avec l'augmentation du nombre d'itérations : la présence d'un voisin plus fréquent a un effet inhibiteur lorsque la décision est prise avec un seuil haut. Enfin, nous avons vu que les mots de haute fréquence sont moins sensibles à la présence d'un voisin plus fréquent. Dans la simulation B06, nous avons utilisé une tâche de reconnaissance visuelle de mot. Nous avons montré que la présence d'un voisin plus fréquent a systématiquement un effet inhibiteur sur les performances. Nous avons également montré que *BRAID* rend compte de cet effet même lorsque les stimuli sont de haute fréquence, ce qui ne correspond pas aux résultats comportementaux de Sears et al. (1995).

Dans un second temps, nous avons considéré l'effet de la densité du voisinage sur les performances d'identification. Dans la simulation B07, nous avons utilisé la tâche de décision lexicale et considéré le cas où le stimulus est un mot. nous avons montré qu'un voisinage dense a le même effet qu'un voisin plus fréquent. Dans les premières itérations (autour de 200 ms), un voisinage dense a un effet facilitateur sur les performances, puis l'effet s'inverse pour devenir inhibiteur. Nous avons ainsi simulé avec succès les résultats de quatre expériences comportementales (en incluant la condition mot de la simulation B08). Dans la simulation B08, nous avons utilisé la tâche de décision lexicale et considéré le cas où le stimulus est un non-mot. Nous avons montré que l'augmentation de la densité du voisinage des non-mots a un effet inhibiteur sur le rejet de ces stimuli. Enfin, dans la simulation B09, nous avons utilisé une tâche de reconnaissance visuelle de mot. Nous avons montré qu'un voisinage dense a systématiquement un effet inhibiteur sur les performances. Nous avons également montré que, dans les simulations, cet effet semblait plus important pour les stimuli de basse fréquence que pour les stimuli de haute fréquence, ce qui ne semble pas être le cas dans l'expérience de Andrews (1989).

Avec *BRAID*, nous avons mis en évidence, dans des tâches de décision lexicale, des interactions entre fréquence des stimuli et fréquence ou densité du voisinage. Cependant, ces résultats ne semblent pas refléter la réalité expérimentale. Nous avons donc réalisé une dernière simulation, la simulation B10, dans laquelle nous étudions spécifiquement ces interactions. Les résultats obtenus correspondent cette fois aux données expérimentales et mettent en évidence les interactions entre la fréquence des stimuli et la fréquence du voisinage, ainsi que les interactions entre la fréquence des stimuli et la densité du voisinage. Cependant, l'écart entre les résultats pour des stimuli de basse fréquence et ceux de hautes fréquences paraissent plus importants qu'ils ne sont expérimentalement.

**Le choix de la méthode d'ajustement des données simulées** Cette dernière observation pose en fait la question du choix de la méthode d'ajustement des données simulées aux données comportementales. Dans ce chapitre (voir Section 1), nous avons utilisé une simple mise à l'échelle par homothétie. Pour cela nous nous sommes donnés une valeur de seuil, ce qui nous permet d'avoir une valeur pour chaque condition expérimentale. Puis, nous choisissons arbitrairement une condition pour calculer le facteur d'homothétie (la condition de calibrage), et nous l'appliquons aux autres conditions (les conditions prédites).

Norris (2006) fait un autre choix. Il part du principe que dans les tâches d'identification que nous simulons avec nos modèles, il y a une partie du temps qui est alloué à la tâche d'identification a proprement parler, et une partie du temps qui est alloué à la partie « motrice » de la tâche, pour planifier le déplacement du bras et appuyer sur le bouton, par exemple. Pour cette raison, il adopte dans ces simulations une mise à l'échelle affine, c'est-à-dire qu'il ajoute une constante qui correspond à la partie « motrice », en plus du facteur multiplicatif de l'ajustement homothétique. Nous n'avons pas fait ce choix. En effet, pour les petits jeux de données ou des

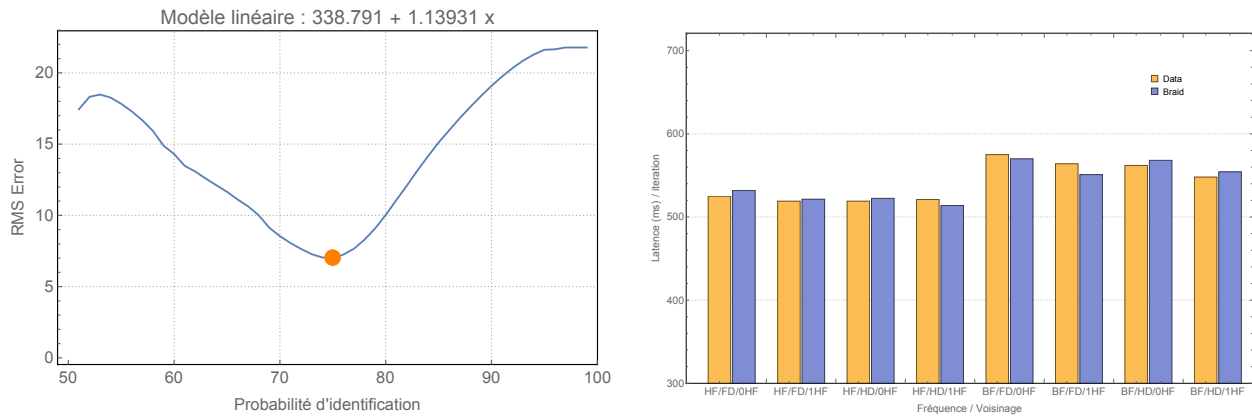


Figure 6.14 – **Simulation B10, résultats avec mise à l'échelle affine des données simulées.** À gauche, on présente la racine de l'erreur quadratique moyenne en fonction du seuil de décision (le point orange matérialisant le seuil choisi). À droite, on présente conjointement les résultats des simulations de *BRAID* (en bleu) et les données expérimentales (en jaune).

jeux de données dans lesquels il y a peu de variabilité, cette méthode est trop performante et conduit à des valeurs de paramètres aberrantes. Considérons un exemple, dans lequel on a quatre temps observés expérimentalement assez proches, comme dans l'expérience de Sears et al. (1995). Dans ce cas, appliquer une correction affine suggère des pentes négatives, et donc une inversion du sens de l'effet. En revanche, dans une simulation comme la simulation B10, dans laquelle nous avons à la fois des effets massifs (fréquence du stimulus) et des effets fins (densité du voisinage), et suffisamment de données (8 dans ce cas), il peut être intéressant d'utiliser une transformation affine. La Figure 6.14 montre le résultat que l'on aurait obtenu pour cette simulation. On constate que pour l'itération qui minimise l'écart entre les données simulées et les données comportementales, la constante vaut 338,8, ce qui suggère un « temps moteur » réaliste, cohérent avec des données récentes (Pylkkänen & Marantz, 2003), et la pente est proche de 1, ce qui suggère que les écarts obtenus restent une conséquence des mécanismes de *BRAID*. Dans ce cas, les données simulées sont plus proches des données comportementales, sans pour autant changer le sens des effets simulés. L'apport de cette nouvelle méthode d'ajustement des données est donc essentiellement cosmétique.

**Analyse globale des résultats** Comme dans le chapitre sur la perceptibilité des lettres (Chapitre 5), nous nous intéressons au pouvoir prédictif global de *BRAID*, mais cette fois pour les tâches de reconnaissance de mot et de décision lexicale. Pour cela, nous regroupons les résultats de toutes les simulations. Nous excluons cependant les simulations sur l'effet de fréquence, simulation B01 et simulation B02, car leur données de références n'étaient pas comportementales, mais reposaient sur les simulations du modèle BR. Nous considérons donc 16 expériences comportementales, 33 résultats en simulation de la tâche de décision lexicale (29 réponses OUI et 4 réponses NON) et 11 résultats en simulation de la tâche de reconnaissance visuelle de mots.

Pour rappel, à chaque expérience comportementale que nous avons simulé avec *BRAID*, nous avons fixé un seuil pour lequel les résultats sont comparés aux résultats comportementaux. Ce seuil est fixé en minimisant, sur toutes les itérations, la différence RMSE (la racine carrée du carré moyen des erreurs) entre les résultats des simulations et les données comportementales. Puis, en utilisant systématiquement la première condition, nous calculons un coefficient multiplicateur que nous appliquons aux résultats simulés. Ce faisant, nous contraignons les résultats

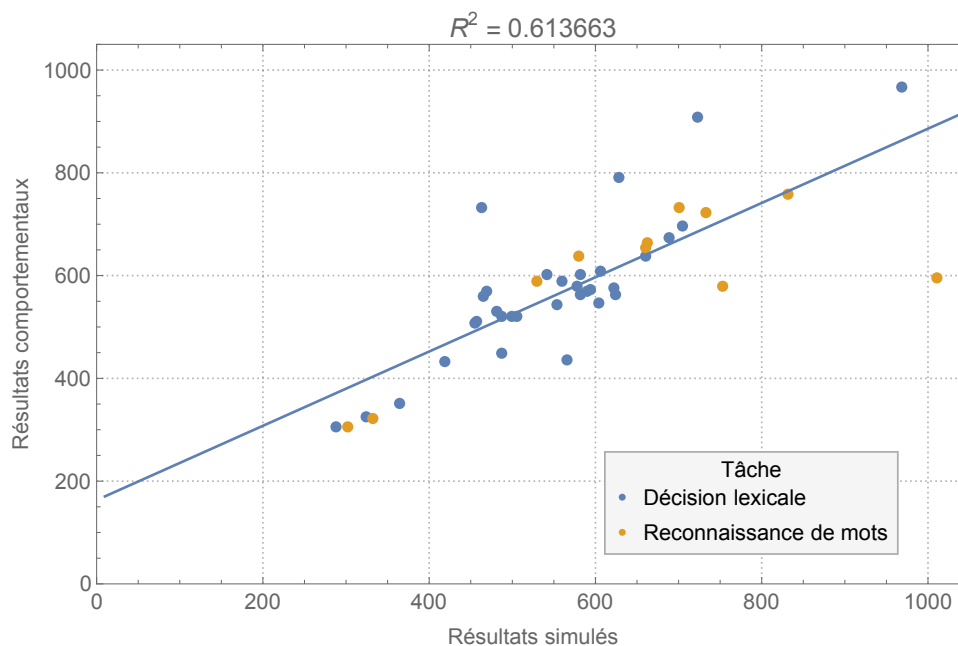


Figure 6.15 – **Analyse globale des résultats de reconnaissance de mots et de décision lexicale.** Corrélation entre les performances de reconnaissance de mots prédites par *BRAID* (en abscisse) dans les neuf premières simulations, et les observations comportementales correspondants (en ordonnée).

simulés et les données comportementales à coïncider sur la première condition, mais les autres conditions sont considérées prédites et c'est uniquement sur les résultats prédits que portent cette analyse.

L'analyse consiste à réaliser une régression linéaire avec comme facteur prédictif les résultats des simulations et comme facteur dépendant les données comportementales associées. Les résultats de l'analyse de régression sont présentés Figure 6.15. La droite de régression entre les données comportementales et nos simulations a un coefficient d'ajustement  $R^2 = 0,61$ , ce qui est bon étant donné la diversité des conditions expérimentales considérée ici. Cela montre que nos simulations ont une cohérence d'ensemble, et que *BRAID* est capable de capturer et reproduire une grande variété d'effets liés à l'identification des mots.





# Chapitre 7

## Conclusion générale

### 1 Principales contributions

Dans ce manuscrit de thèse, nous nous sommes intéressé à la modélisation computationnelle de la reconnaissance visuelle de mots. Nous rappelons ici les points forts de notre contribution.

**Une synthèse de la littérature** Notre première contribution est une analyse, que nous espérons exhaustive, de la littérature portant sur les modèles computationnels de reconnaissance visuelle de mots au cours de ces quarante dernières années. Des éléments de cette synthèse sont la base d'une publication d'un chapitre d'ouvrage (Phénix, Diard, & Valdois, 2016), qui l'étend en incluant une revue des modèles de lecture à haute voix.

Précédemment, une revue de littérature sur les modèles computationnels avait été proposée par Norris (Norris, 2013) avec pour objectif de discuter des différents styles de modélisation, des questions relatives à l'encodage positionnel et des études portant sur de larges corpora de données. Une autre synthèse un peu plus ancienne (Rayner & Reichle, 2010) s'était focalisée sur les différents types de modèles dits de lecture (modèles de reconnaissance de mots, de dénomination orale et de contrôle oculo-moteur). Notre apport est ici, pour chacun des dix modèles retenus, d'analyser en détail leur fonctionnement et d'étudier leurs performances dans la simulation de tâches comportementales classiques du domaine. Nous avons essentiellement souligné les similarités et les différences entre d'une part les cadres conceptuels qu'ils implémentent, et, d'autre part, les solutions d'implémentation proposées.

**Un modèle théorique conceptuel de reconnaissance visuelle de mots** Notre analyse critique de la littérature sur les modèles computationnels de reconnaissance de mots a conduit à mettre en évidence les disparités des méthodes mathématiques et des cadres théoriques utilisés. Nous avons pu identifier trois grandes problématiques par rapport auxquelles tout modèle de reconnaissance de mots doit se positionner : la première concerne leur architecture (trois niveaux sont-ils nécessaires ?) et le type de traitement de la séquence de lettres (parallèle ou sériel ?) ; la deuxième concerne le choix de l'encodage de la position des lettres du stimulus (doit-il être « strict », « relatif » ou « distribué » ?) ; la dernière concerne les propriétés du système visuel qui modulent la perception des mots (doit-on prendre en compte l'acuité et l'attention visuelle ?).

Nous avons pu identifier, pour chacune de ces questions d'intérêt, les réponses les plus fréquentes ainsi que les alternatives explorées par les modèles précédents. Ces alternatives ont ensuite été évaluées à la lumière des données comportementales disponibles, ou de données simulées. Nous avons ainsi pu définir un « cahier des charges » sous la forme de cinq hypothèses permettant de proposer et de justifier théoriquement un nouveau modèle conceptuel.

L'originalité de ce modèle réside principalement dans le fait d'intégrer une variété de composants visuels (position du regard, gradient acuité et interférences latérales) et un composant attentionnel. *BRAID* est le premier modèle, à notre connaissance, qui intègre l'ensemble de ces dimensions.

**Notre principale contribution, le modèle *BRAID*** Pour implémenter mathématiquement ce modèle conceptuel et obtenir le modèle *BRAID*, nous appliquons un formalisme probabiliste original, la modélisation bayésienne algorithmique (Diard, 2015), qui hérite de la programmation bayésienne (Bessière et al., 2013) et de la programmation bayésienne des robots Lebeltel et al. (2004). Ce formalisme nous permet d'exprimer les connaissances à intégrer au modèle, sous la forme de distributions de probabilités.

Une première particularité de ce formalisme est sa flexibilité pour exprimer des connaissances hiérarchiques et structurées. Cela contraste avec d'autres formalismes probabilistes, comme la modélisation bayésienne optimale, par exemple utilisée pour le BR (Norris, 2006). Dans ce cadre, définir un modèle repose essentiellement sur une unique distribution de probabilités pour exprimer les connaissances du modèle : la fonction de vraisemblance. Dans le modèle BR, celle-ci décrit la vraisemblance des mots du lexique pour une entrée visuelle donnée,  $P(I | W)$ . Étendre et enrichir le modèle nécessite alors de modifier la fonction de vraisemblance, la rendant ainsi compliquée à interpréter (Norris & Kinoshita, 2012). Par opposition, le cadre de la modélisation bayésienne algorithmique retenu ici est plus flexible, et permet de manipuler facilement des modèles hiérarchiques et structurés (Patri, Perrier, & Diard, 2016). Chaque bloc de connaissance peut-être représenté par un sous-modèle, lui-même décomposé en un ensemble de termes probabilistes. Il aurait été malaisé d'atteindre le niveau de complexité de *BRAID* en utilisant une seule fonction de vraisemblance.

Une autre particularité de ce formalisme est la dissociation qui y est faite entre connaissances et processus. Définir le modèle consiste à encoder, sous forme probabiliste, des connaissances d'un domaine. Ensuite, à partir de ce modèle de connaissances, le calcul probabiliste appliqué au modèle génère des modèles de processus, à même de simuler des tâches comportementales. Cela a deux avantages. Le premier est que la génération de ces processus se fait de manière mathématiquement contrainte par les règles du calcul probabiliste (l'inférence bayésienne). C'est un réel point fort, car l'implémentation de ces modèles de processus n'est plus sous le contrôle du modélisateur, et traduit directement les connaissances du modèle. Le second avantage est une fois de plus lié à la flexibilité du formalisme : le modèle n'est pas dédié à une unique tâche, et peut donc être utilisé pour implémenter différents processus. Ainsi, nous avons utilisé *BRAID*, dans cette thèse, pour simuler des tâches d'identification de lettres, de reconnaissance de mots et de décision lexicale, mais il pourrait également être utilisé pour simuler d'autres tâches, comme simuler le contrôle attentionnel ou le contrôle oculo-moteur, par exemple.

**La procédure de calibrage de *BRAID*** Dans les présentations des modèles computationnels de reconnaissance visuelle de mots étudiés dans notre revue de la littérature, les détails du processus de calibrage des paramètres sont souvent très limités. Dans le modèle IA (McClelland & Rumelhart, 1981), le calibrage a été réalisé de sorte que les résultats des simulations du modèle correspondent autant que possibles aux données comportementales présentées dans les deux articles de référence (McClelland & Rumelhart, 1981 ; Rumelhart & McClelland, 1982). Les modèles ayant une filiation avec IA héritent de ces paramètres. Pour les autres modèles, le plus souvent, les valeurs des paramètres sont données sans beaucoup de justifications.

La démarche que nous avons adoptée est tout autre. Nous nous sommes inspirés d'un jeu de données électrophysiologiques et comportementales, pour calibrer les paramètres impliqués

dans chacune des tâches simulées par *BRAID*. Ainsi, nous avons calibré les paramètres de la reconnaissance visuelle de lettres,  $Leak_{petScale_I}$ , sur des données électrophysiologiques de reconnaissance de lettres isolées. Puis nous avons calibré la pente de l'acuité  $Slope_G$  et le paramètre d'interférence  $\theta_I$  sur des données comportementales. De même, pour calibrer le paramètre  $Leak_W$  qui caractérise la dynamique de la reconnaissance visuelle de mots, nous nous sommes basés sur des données électrophysiologiques, et pour  $Leak_D$  sur des données comportementales. Notons que, d'un point de vue empirique, le calibrage nous paraît robuste, puisqu'aucun paramètre ne nécessite une valeur particulière, ou ne se situe dans un intervalle très contraint, pour produire les effets comportementaux étudiés. Enfin, contrairement au modèle IA (McClelland & Rumelhart, 1981), les études comportementales utilisées pour calibrer *BRAID* sont toutes différentes de celles présentées dans les chapitres de simulations (Chapitre 5 et Chapitre 6).

**Sa capacité à capturer des effets comportementaux** Nous avons utilisé *BRAID* pour simuler un large panel d'expériences comportementales. Ces expériences ont été regroupées en deux séries : les dix premières portent sur la perceptibilité des lettres et les dix-huit suivantes sur la reconnaissance de mots et la décision lexicale. Nous avons choisi de nous focaliser sur deux types de situations expérimentales et de rentrer dans le détail de chacune, nous confrontant ainsi à la simulation d'effets fins. Concernant la perceptibilité des lettres, on trouve un détail similaire dans les premières publications à l'appui du modèle IA (McClelland & Rumelhart, 1981 ; Rumelhart & McClelland, 1982) mais dans aucun des modèles suivants. Concernant l'effet de fréquence, nous avons choisi la méthode proposée par Norris (2006), car c'est à notre connaissance l'étude la plus fouillée des effets de fréquence. Concernant la décision lexicale, nous nous sommes inspirés du travail de Davis (2010b), le plus complet dans le domaine.

Les résultats de nos simulations montrent que *BRAID* reproduit fidèlement les effets observés dans chacune des études comportementales retenues, ce qui apporte une validation expérimentale au modèle. La capacité de *BRAID* à rendre compte des effets liés à la perceptibilité des lettres, observés dans des conditions expérimentales très variées, est excellente, avec un coefficient de détermination acceptable ( $R^2 = 0,58$ ) pour la corrélation générale entre résultats comportementaux et résultats des simulations. De la même façon, en décision lexicale et reconnaissance de mots, *BRAID* a une capacité remarquable à rendre compte des effets observés dans des conditions expérimentales très variées, avec un coefficient global de détermination également acceptable ( $R^2 = 0,61$ ) sur l'ensemble des simulations.

Ainsi, globalement, le modèle *BRAID* paraît être performant pour rendre compte des effets principaux observés. Nous revenons ici sur quelques unes de ces simulations. *BRAID* est à notre connaissance, le premier modèle computationnel de reconnaissance visuelle de mots à simuler, outre les effets classiques pré-cités, l'effet de prédictibilité des lettres et l'effet d'espacement des lettres. En ce qui concerne l'effet de fréquence, *BRAID* est capable de reproduire les simulations très sophistiquées effectuées avec le modèle BR Norris (2006), en reconnaissance visuelle de mots comme en décision lexicale. Ce résultat est d'autant plus remarquable que le modèle BR a été conçu spécifiquement pour rendre compte de cet effet. Mais *BRAID* apporte un nouvel éclairage sur la compréhension de ce phénomène. Comme dans le modèle BR (Norris, 2006), on utilise la fréquence pour donner un avantage aux mots fréquents (ce qui s'apparente au principe de pré-activation adopté dans IA (McClelland & Rumelhart, 1981)). Cependant, nos courbes de décours temporel montrent que la fréquence facilite également la vitesse d'accumulation d'information perceptive sur laquelle repose la reconnaissance des mots.

La capacité de simulation fidèle des résultats comportementaux est une condition première, mais l'intérêt de la modélisation est d'apporter un éclairage sur l'interprétation de résultats

complexes. *BRAID* remplit tout à fait cette fonction. En perceptibilité des lettres par exemple, les résultats des simulations apportent une explication originale au phénomène d’amorçage par un contexte pseudo-mots (voir Simulation A07).

Un autre point fort de *BRAID* est de nous aider à comprendre l’origine de résultats comportementaux en apparence contradictoires. La plupart des simulations effectuées dans les autres cadres théoriques concernent l’effet étudié à un temps donné, estimé à partir des caractéristiques de l’expérience comportementale et selon des critères pré-définis de seuil de décision par exemple. Au contraire, dans les simulations effectuées avec *BRAID*, nous présentons systématiquement non seulement les effets observés selon les critères pré-définis, mais également le déroulement du traitement indépendamment des critères choisis pour la simulation de l’effet. Ainsi, dans nos simulations des effets de fréquence du voisinage, voir Simulation B05, et densité du voisinage, voir Simulation B07), en décision lexicale, les courbes de déroulement temporel montrent que la valeur de seuil de décision est critique pour déterminer le sens de l’effet. En effet, le sens de l’effet s’inverse dans les simulations pour des seuils relativement proches, suggérant que l’effet expérimentalement observé n’est pas robuste. Ainsi, des variations de conditions expérimentales à première vue insignifiantes entre différentes études ont pu conduire aux résultats contradictoires qu’elles ont obtenus.

**Plan de notre discussion** La section qui suit aura pour but, dans un premier temps, de rappeler et discuter les hypothèses théoriques qui ont été retenues pour générer le modèle conceptuel à la base de *BRAID*. Nous analyserons la pertinence de ces choix par rapport aux modèles concurrents de reconnaissance de mots et plus largement par référence aux autres types de modèles de lecture (incluant les modèles de dénomination orale et de contrôle oculomoteur lors de la lecture de textes). Dans un second temps, nous discuterons des limites du modèle *BRAID* et tenterons de proposer des pistes d’amélioration. Enfin, nous aborderons les évolutions possibles du modèle, en évoquant la possibilité d’ajout d’un composant phonologique pour lui conférer des capacités de dénomination orale, en évoquant son potentiel d’extension en modèle de contrôle des mouvements oculaires, et en évoquant des pistes pour l’apprentissage de mots nouveaux et la simulation de l’apprentissage de la lecture chez les débutants lecteurs.

## 2 Discussion

Dans le cadre de ce travail de thèse, nous avons proposé un nouveau modèle de reconnaissance visuelle de mots, le modèle *BRAID*. L’objectif de ce nouveau modèle est d’apporter une réponse la plus unifiée possible aux problématiques actuelles de ce champ d’études. Une revue de la littérature portant sur les quarante dernières années de modélisation dans ce domaine nous a permis d’identifier deux questions centrales, qui sont au cœur des débats actuels : la question de la structure du modèle et la question du codage positionnel des lettres. Nous avons ajouté une troisième question qui fait moins débat aujourd’hui, mais qui est au cœur des différents champs d’études liés à la lecture : c’est la question des propriétés visuelles qui modulent la perception des mots écrits.

Pour répondre à ces trois problématiques, nous avons formulé cinq hypothèses théoriques, qui ont servi de base à la conception du modèle *BRAID*. Nous avons utilisé ce modèle dans de nombreuses simulations, le plus souvent avec succès. La discussion aura pour objectif de discuter la capacité du modèle *BRAID* à remplir notre « cahier des charges » théorique et répondre aux problématiques identifiées. Nous tenterons également de montrer que le modèle *BRAID* intègre des propriétés présentes dans certains des modèles computationnels précédents, en utilisant un cadre de description unifié, la modélisation bayésienne algorithmique. Nous reviendrons

également sur les simulations effectuées, les limites du modèle et les perspectives ouvertes par ce nouveau cadre théorique.

## 2.1 Quelle structure pour un modèle de reconnaissance visuelle de mots ?

Lorsque l'on construit un modèle de reconnaissance visuelle de mots, la première question que l'on se pose est celle de la structure du modèle. Cette question renferme en fait deux sous-questions. La première concerne l'architecture du modèle à proprement parler : quels sont les niveaux de représentation nécessaires et leurs connexions ? La seconde concerne le traitement des lettres du stimulus : doit-il être parallèle ou sériel ? Ce sont deux questions distinctes que nous avons traitées séparément.

### 2.1.1 Quelle structure pour un modèle de reconnaissance visuelle de mots ?

Nous avons répondu à cette question avec l'Hypothèse 1 : *L'architecture du modèle repose sur une structure à trois niveaux : le niveau sensoriel des lettres, le niveau perceptif des lettres et le niveau des connaissances lexicales*. Le choix d'une architecture à trois niveaux est motivé tout d'abord par le large consensus que l'on trouve dans la littérature. Cette architecture a été implémentée initialement dans le modèle IA (McClelland & Rumelhart, 1981). On la retrouve dans tous les modèles qui reconnaissent une filiation directe avec le modèle IA, comme le modèle MROM (Grainger & Jacobs, 1996) et le modèle SCM (Davis, 2010b). Mais cette architecture a fédéré bien d'autres modèles, et on la retrouve dans des modèles sans filiation directe, comme les modèles AV (Paap et al., 1982), MORSEL (Mozer & Behrmann, 1990) et SERIOL (C. Whitney, 2001).

Ainsi, tous les modèles descriptifs de reconnaissance de mots (Phénix, Diard, & Valdois, 2016) adoptent une architecture à trois niveaux. Par opposition, on ne retrouve pas cette architecture dans les modèles abstraits (Adelman et al., 2010 ; Gomez et al., 2008 ; Norris, 2006), qui se focalisent sur des propriétés spécifiques du processus de reconnaissance visuelle de mots : l'effet de fréquence pour le modèle BR (Norris, 2006), la distribution de la position des lettres pour le modèle Overlap (Gomez et al., 2008) et la sérialisation du processus d'identification des lettres et de leur position pour le modèle LTRS (Adelman et al., 2010). Ainsi, BR (Norris, 2006) et LTRS (Adelman et al., 2010) ne comportent que les niveaux lettres et mots ; Overlap (Gomez et al., 2008) ne comporte qu'un niveau lettre.

Le choix d'adopter une architecture à trois niveaux est également motivé par les possibilités d'ouverture offertes vers les autres champs d'études de la lecture. Cette architecture se retrouve dans les modèles de dénomination orale, comme le modèle DRC (Coltheart et al., 2001) et le modèle CDP (Perry, Ziegler, & Zorzi, 2007). On la retrouve également dans le modèle computationnel de contrôle des mouvements oculaires E-Z reader (Reichle, Pollatsek, Fisher, & Rayner, 1998 ; Reichle, Pollatsek, & Rayner, 2006).

### 2.1.2 Le traitement des lettres est-il parallèle ou séquentiel ?

Nous avons répondu à cette question avec l'Hypothèse 2 : *le traitement des lettres du stimulus est réalisé en parallèle*. Là encore, ce choix repose sur un quasi-consensus dans la littérature. Seuls deux modèles ont opté pour un traitement sériel des lettres du stimulus, le modèle SERIOL (C. Whitney, 2001) et le modèle SCM (Davis, 2010b). Le modèle SERIOL propose une modélisation détaillée des processus d'analyse sensorielle permettant de générer un pattern de décharge temporelle gauche-droite qui code la position relative des lettres pour l'encodage en bigrammes

ouverts de la séquence du mot. Cette idée est partiellement reprise dans le modèle SCM (Davis, 2010b) qui ne considère que le résultat du processus (pattern de décharge temporelle gauche-droite) pour le codage distribué de la position des lettres. L'hypothèse d'encodage sériel dans ces modèles est donc en partie la conséquence des choix de codage positionnel qu'ils adoptent. *BRAID*, comme les autres modèles à traitement parallèle, repose sur des choix d'encodage positionnel différents formulés indépendamment de la nature du traitement (parallèle ou sériel).

L'option de traitement parallèle des lettres de la séquence à lire est en outre compatible avec les données comportementales disponibles. Adelman et al. (2010) et Marzouki et Grainger (2014) montrent que le traitement des lettres dans la séquence d'un mot ou d'un pseudo-mot, respectivement, commence simultanément sur toutes les lettres dès les premières vingt millisecondes du traitement. Ils montrent néanmoins que traitement parallèle ne signifie pas une identification uniforme sur l'ensemble des lettres à traiter. Ainsi, certaines lettres sont mieux identifiées que d'autres : il y a un avantage des lettres initiales pour les mots et un avantage des lettres extrêmes (début et fin) pour les pseudo-mots. Tous les modèles qui adoptent un traitement parallèle retiennent la notion de traitement débutant simultanément sur l'ensemble des lettres du mot.

Par contre, certains modèles computationnels à traitement parallèle n'offrent aucune possibilité de modulation du traitement de l'identité des lettres en fonction de leur position. C'est le cas notamment du modèle IA, dans lequel la probabilité d'identifier la lettre A en position 1 ou en position 7 de la séquence du mot ne dépend que de l'extraction des traits et des influences lexicales, ce qui conduit à des réponses identiques sur les deux positions. On a donc un traitement parallèle uniforme. IA montre qu'il est très performant pour rendre compte d'un grand nombre de données expérimentales, ce qui semble justifier son choix de traitement parallèle uniforme mais, en réalité, le modèle ne se confronte qu'à des simulations sur des mots très courts. S'il était doté d'un lexique de mots plus complet, le modèle conduirait à traiter un mot long et un mot court à l'identique, et il ne pourrait donc pas simuler les effets de longueur observés en décision lexicale (New, Ferrand, Pallier, & Brysbaert, 2006). Tous les autres modèles parviennent à moduler l'identification des lettres dans la séquence du mot dans le cadre d'un traitement parallèle débutant simultanément sur chacune des lettres. Certains de ces modèles jouent sur l'acuité visuelle, comme le modèle VA (Paap et al., 1982), d'autres sur l'attention visuelle, comme le modèle MORSEL (Mozer & Behrmann, 1990), pour expliquer que l'identification de l'ensemble des lettres ne soit pas identique malgré un traitement parallèle du stimulus.

Dans *BRAID*, la perception des lettres est modulée à la fois par l'acuité visuelle et par l'attention : l'acuité induit une décroissance linéaire des capacités d'identification des lettres, avec l'éloignement du point de fixation ; l'attention agit comme un filtre qui limite fortement l'identification des lettres excentrées. *BRAID* se situe donc dans la lignée de la plupart des modèles précédents en adoptant un traitement parallèle sur l'ensemble des lettres tout en modulant l'identification des lettres de la séquence.

## 2.2 L'encodage de la position des lettres doit-il être strict, relatif ou distribué ?

Nous avons répondu à cette question avec l'Hypothèse 3 : *l'encodage de la position de chaque lettre est distribué sur l'ensemble des positions du stimulus*. Cette idée d'introduire de l'incertitude sur la position des lettres à identifier, en encodant leur identité sur plusieurs positions, a été formulée pour la première fois par Ratcliff (1981), mais la première implémentation dans un modèle de reconnaissance visuelle de mots est proposée par le modèle Overlap (Gomez et al., 2008). Deux modèles descriptifs récents ont intégré cette idée : le modèle SCM (Davis, 2010b)

et le modèle BR dans sa dernière version (Norris & Kinoshita, 2012). L'objectif de ces modèles est de rendre compte des effets d'amorçage en décision lexicale, l'amorce étant constituée des lettres du stimulus auxquelles on a appliqué une transformation (transposition, duplication, insertion, suppression, substitution, etc.). Cependant, cette proposition d'encodage distribué de la position est loin d'être la plus répandue. Le codage le plus populaire est le codage strict que l'on retrouve dans les modèles IA (McClelland & Rumelhart, 1981), MROM (Grainger & Jacobs, 1996), VA (Paap et al., 1982) et BR dans sa version initiale (Norris, 2006). Trois modèles ont adopté un codage relatif. Le modèle SERIOL (C. Whitney, 2001) et le *parallel open bigram model* (Grainger & van Heuven, 2003) optent pour un codage en bigrammes ouverts tandis que le modèle MORSEL (Mozer & Behrmann, 1990) adopte un codage en trigrammes ouverts.

L'option du codage « strict » consiste à encoder l'identité des lettres dans la position exacte où elles apparaissent dans la séquence. Ce codage a été mis en défaut dans des tâches de décision lexicale avec amorçage, où l'amorce était créée en manipulant les lettres du stimulus : transposition de lettres adjacentes ou suppression d'une lettre (Schoonbaert & Grainger, 2004), ou transposition de lettres non adjacentes (Perea & Lupker, 2004). Un codage strict ne peut rendre compte facilement des effets d'amorçage par transposition, sauf à imaginer des processus ad-hoc spécifiques à ces cas.

L'option du codage « relatif » consiste à encoder la position d'une nouvelle lettre par rapport aux lettres précédentes. Bien que plus flexible, ce codage a également été mis en défaut dans différentes expériences comportementales. En décision lexicale avec amorçage par transposition, Kinoshita et Norris (2013) ont obtenu un amorçage robuste de bigrammes inverses comme of - FO ou ob - ABOLISH qui ne correspondent à aucun des bigrammes ouverts de la séquence présentée. De même, dans une tâche « même-différent » en condition de transposition par blocs, l'amorce FULPAIN n'a pas d'effet facilitateur sur le stimulus PAINFUL (Kinoshita & Norris, 2009), contrairement à ce que prédit l'hypothèse des bigrammes ouverts. Enfin, en décision lexicale, le non-mot GASFUL est rejeté plus lentement que son contrôle orthographique GASFIL, mais ce n'est pas le cas pour les stimuli FULGAS et FILGAS. Dans les deux cas, la première paire de stimuli partage le même nombre de bigrammes ouverts que la seconde paire si bien qu'un codage relatif en bigrammes ouverts prédirait des effets équivalents dans les deux cas. Ce constat a conduit d'autres auteurs à opter pour un codage distribué dans les modèles SCM (Davis, 2010b) et BR (Norris & Kinoshita, 2008, 2012) et à montrer la capacité de ces modèles à simuler les résultats comportementaux problématiques pour les modèles à codage positionnel relatif.

En résumé, un codage strict ne saurait rendre compte des données d'amorçage avec transposition de lettres. Un codage relatif est capable de simuler la plupart des résultats comportementaux sur ce type de tâches. Néanmoins, les modèles dotés d'un encodage distribué sont plus performants : ils rendent compte des données simulées par les modèles à encodage relatif et de certains résultats spécifiques qui ne peuvent être simulés par ces derniers. Ils démontrent ainsi leur bonne capacité de simulation dans le cadre d'un modèle plus parcimonieux qui ne postule pas de niveau supplémentaire d'encodage orthographique, comme les bigrammes ouverts.

Si on se focalise maintenant sur les modèles à encodage distribué de la position des lettres du stimulus, les implémentations proposées varient selon les modèles. Dans le modèle Overlap (Gomez et al., 2008), une distribution est calculée pour chaque position par ajustement aux données comportementales. La distribution correspondant à la première position a la variance la plus faible, puis la variance de la distribution augmente avec la position. En revanche, dans les modèles SCM (Davis, 2010b) et BR (Norris & Kinoshita, 2012), l'incertitude positionnelle est la même sur toutes les positions. Dans les deux cas, la distribution est une distribution



gaussienne, centrée sur la lettre, la variance permettant de contrôler son influence sur les lettres voisines. Dans le modèle SCM (Davis, 2010b), la variance dépend de l'acuité tandis que dans le modèle BR (Norris & Kinoshita, 2012), la variance diminue au cours du temps jusqu'à la disparition de toute incertitude. Dans *BRAID*, l'encodage distribué ne prend en compte que les lettres adjacentes et est identique sur l'ensemble des positions. Ces différences d'encodage sont liées plus largement à la nature des mécanismes de traitement visuel postulés par les différents modèles. L'implémentation de l'encodage distribué adopté dans *BRAID* est simple mais suffisant pour rendre compte des données, car l'incertitude positionnelle est modulée par des influences indépendantes, qui sont l'acuité visuelle et l'attention. En l'absence de ces mécanismes de modulation, les autres modèles (Overlap (Gomez et al., 2008) et SCM (Davis, 2010b)) ont dû adopter des choix différents, variables selon la position de la lettre dans la séquence.

Les simulations réalisées dans le cadre de cette thèse montrent que *BRAID*, muni de ce type d'encodage distribué, est capable de simuler avec succès nombre de données expérimentales. Cependant, aucune de ces simulations ne porte spécifiquement sur les enjeux positionnels. Pour démontrer l'efficacité de l'encodage que nous avons choisi, il faut simuler une grande variété d'expériences d'amorçage. Un travail en ce sens a été entrepris, dans le cadre d'un stage de Master que j'ai encadré (Ginestet, 2016). Les résultats témoignent de la capacité de *BRAID* à simuler les données d'amorçage avec succès.

## 2.3 Quelles sont les propriétés du système visuel qui modulent la perception des mots ?

La reconnaissance visuelle de mots est influencée par plusieurs phénomènes : la position du regard, l'acuité visuelle et l'encombrement visuel (*crowding*) (Grainger et al., 2016). Si les mécanismes cognitifs dits de haut niveau sont largement représentés dans les modèles computationnels de reconnaissance de mots, ces derniers implémentent rarement les traitements visuels associés. De la même façon, les processus attentionnels bien qu'omniprésents dans la littérature en cognition visuelle sont le plus souvent absents de ces modèles.

### 2.3.1 Doit-on intégrer dans un modèle de reconnaissance visuelle de mots l'acuité visuelle ?

Nous avons répondu à cette question avec l'Hypothèse 4 : *le modèle contient un mécanisme permettant de simuler l'effet de l'acuité sur l'identification des lettres*. Cette réponse est motivée par le large consensus qui existe autour de cette question. En effet, il est largement admis que la probabilité d'identification d'une lettre dans un mot diminue avec l'augmentation de sa distance au point de fixation, et que cela est dû à une baisse de l'acuité visuelle. Des modélisations de la diminution de l'acuité avec l'augmentation de l'excentricité ont été proposées dans le contexte de la perception des mots (McConkie, Kerr, Reddix, Zola, & Jacobs, 1989 ; Nazir, 1991) ou dans des contextes non linguistiques (Watson & Ahumada, 2012).

Pourtant, l'acuité n'est intégrée que dans trois modèles parmi ceux présentés dans ce travail. Dans le modèle VA (Paap et al., 1982), il existe une matrice de confusion par position, ces matrices tiennent compte de la distance par rapport à la position du regard et sont symétriques dans les hémichamps droit et gauche. Dans le modèle SERIOL, l'acuité intervient au niveau de la couche d'entrée du modèle et se traduit par une diminution de l'activation des pixels avec l'augmentation de la distance au point de fixation (c'est le gradient d'acuité). Enfin, dans le modèle SCM, l'acuité n'influence pas directement le processus d'identification des lettres, mais celui qui détermine la position des lettres, ce qui a une influence indirecte sur leur identification.

La variance de la distribution de la position augmente linéairement avec l'excentricité. Dans ce modèle, c'est la variance de la position qui augmente, et donc la quantité d'interférence qu'une lettre reçoit des autres lettres. Au point de fixation, les variances sont petites, donc la position des lettres est connue avec une précision suffisante pour que seules la lettre centrale et les lettres adjacentes coexistent sur la même position. Mais lorsqu'on s'éloigne de la position centrale, les variances sont plus grandes, donc les lettres sont présentes sur plusieurs positions et de nombreuses lettres interfèrent avec la lettre à identifier. En conclusion, dans les modèles VA (Paap et al., 1982) et *BRAID*, l'extraction des informations sensorielles est pénalisée par la diminution de l'acuité visuelle, alors que dans le modèle SCM (Davis, 2010b), c'est la capacité à discriminer les lettres qui diminue avec l'excentricité.

Dans *BRAID*, l'acuité visuelle est implémentée de manière similaire à l'implémentation du modèle VA (Paap et al., 1982). Pour construire la représentation d'une lettre au niveau sensoriel, nous prenons en compte la lettre dans la bonne position, ainsi que la ou les lettres adjacentes. Leurs contributions sont ensuite pénalisées proportionnellement à leur éloignement au point de fixation (acuité). Cette implémentation diffère de l'implémentation du modèle SCM (Davis, 2010b).

Si la baisse de l'acuité visuelle avec l'augmentation de l'excentricité n'est jamais remise en question, comment se fait-il que tant de modèles ne l'implémentent pas ? Et comment font-ils pour avoir de bons résultats ? Si l'acuité visuelle peut ne pas être intégrée dans certains modèles sans pénaliser les performances, c'est qu'elle a peu d'influence sur les expériences comportementales simulées par ces modèles. En effet, dans ces expériences, les stimuli ont en général entre 3 et 6 lettres, et la position du regard est systématiquement centrée sur le stimulus. Donc, la lettre la plus éloignée est au plus à 3 lettres du point de fixation. Si l'on ajoute le fait que la baisse de l'acuité est compensée aux extrémités par la faiblesse des interférences latérales, on comprend qu'ignorer l'effet de l'acuité a peu de conséquences sur les résultats des simulations. D'ailleurs, l'essentiel des simulations de ce travail de thèse porte sur des mots courts centrés, ne permettant pas de mettre en évidence un rôle spécifique de l'acuité. Cependant, dans la Simulation A10, nous avons dû modifier le gradient d'acuité visuelle pour simuler les effets d'accroissement de l'espacement des lettres.

Pour que l'acuité impacte les performances en reconnaissance de mots ou de lettres, il faut des conditions expérimentales particulières. Les effets d'acuité sur l'identification des lettres se manifestent lorsque la taille des mots étudiés est plus grande (supérieure à 6 lettres) ou que la position du regard sur le stimulus est décalée par rapport au centre du mot. Dans ces cas, les performances des modèles qui n'intègrent pas d'effet de l'acuité visuelle seraient bien supérieures à celles des humains. Ces conditions expérimentales sont précisément celles des expériences qui traitent de la position optimale du regard, ou OVP (*optimal viewing position*). Dans l'étude (Montant, Nazir, & Poncet, 1998), les performances en décision lexicale sont mesurées en faisant varier la longueur des mots (de 5 à 9 lettres) et la position du regard. Les auteurs montrent que le taux de reconnaissance du mot varie selon la position de la séquence. Dans un travail préliminaire, (Phénix, Valdois, & Diard, 2016), nous avons simulé avec succès ces expériences et reproduit l'effet de la position optimale du regard. Les paramètres utilisés étaient les mêmes que ceux utilisés dans ce travail de thèse. La tâche réalisée était celle décrite Chapitre 4, Section 1.4. Nous avons notamment reproduit les résultats classiques : la position optimale de reconnaissance du mot est légèrement à gauche du centre du mot, les performances diminuent avec l'excentricité, de plus à excentricité égale, on reconnaît plus facilement le mot lorsque la fixation porte sur les lettres initiales que sur les lettres finales.

En conclusion, la modélisation du gradient d'acuité visuelle confère au modèle *BRAID* le potentiel de simuler les effets d'espacement et les effets d'OVP, en plus des effets classiques

dont rendent compte la plupart des modèles concurrents.

### 2.3.2 Quel est le rôle de l'attention visuelle dans la reconnaissance de mots ?

Nous avons répondu à cette question avec l'Hypothèse 5 : *le modèle contient un composant attentionnel dont le rôle est de faciliter le traitement des portions d'intérêt du stimulus, au détriment des autres*. L'attention visuelle est implémentée dans un seul des modèles computationnels de reconnaissance visuelle de mots parmi ceux présentés dans ce travail, le modèle MORSEL (Mozer & Behrmann, 1990). Dans ce modèle, le stimulus est représenté sous forme d'une image en deux dimensions, permettant de représenter un ou deux mots. Le composant attentionnel a deux fonctions : isoler le mot qui fait l'objet du traitement et faciliter l'extraction des traits caractéristiques de ce mot dans la zone couverte par l'attention, en filtrant les informations en dehors de la zone attentionnelle. Aucun des autres modèles de reconnaissance de mots ne postule de composant attentionnel.

En revanche, les modèles de contrôle des mouvements oculaires en situation de lecture de texte postulent l'existence de composants attentionnels dont le rôle est essentiellement de focaliser l'attention, soit de façon sérielle sur chaque mot successivement, dans le modèle E-Z reader (Reichle et al., 1998 ; Reichle, Rayner, & Pollatsek, 1999), soit de déployer un gradient d'attention sur deux mots simultanément, dans le modèle SWIFT (Engbert, Longtin, & Kliegl, 2002). On note également la présence d'un composant attentionnel dans un des modèles existants de dénomination orale de mot, le modèle MTM, *Multiple Trace Memory model* (Ans, Carbonnel, & Valdois, 1998). Dans ce dernier cadre théorique, l'attention a essentiellement pour fonction de délimiter la quantité d'information orthographique sur laquelle porte le traitement. Elle détermine ainsi si le traitement porte sur l'ensemble des lettres du mot ou au contraire, sur les unités qui le composent. Ce modèle délimite la zone de déploiement de l'attention (sur tout le mot ou des sous-parties) mais la distribution de l'attention est supposée uniforme pour la zone ainsi définie.

Dans *BRAID* comme dans MORSEL (Mozer & Behrmann, 1990), le composant attentionnel module l'identification des lettres de la séquence suivant une distribution gaussienne. Dans les simulations présentées dans cette thèse, la moyenne de cette distribution coïncide avec la position du regard, même si ces deux variables sont manipulables indépendamment. La variance de la distribution attentionnelle est modulée par la longueur de la séquence à traiter.

On peut, comme précédemment, se demander pourquoi la plupart des modèles de reconnaissance de mots n'intègrent pas de composant attentionnel, et, malgré cela, parviennent à simuler avec succès un large panel de résultats comportementaux. D'une part, on a longtemps considéré que la reconnaissance de mots était automatique chez le lecteur expert et qu'elle ne mettait donc en jeu aucun traitement attentionnel. Pourtant, ces considérations ont été remises en question il y a plus de quarante ans, par Becker (1976), qui a démontré que les performances en décision lexicale étaient diminuées en condition de double tâche, ce qui est généralement interprété comme démontrant l'implication de l'attention. Très récemment, Besner et al. (2016) a montré que la décision lexicale devenait impossible lorsque le mot était présenté à une position inattendue. Ces résultats suggèrent que l'attention est impliquée dans la reconnaissance des mots même fréquents. Son rôle a également été mis en évidence dans les premières étapes du traitement du stimulus (Brown, Gore, & Carr, 2002 ; Brown, Roos-Gilbert, & Carr, 1995), pendant lesquelles elle facilite l'extraction des traits en réduisant les effets d'interférence (Risko, Stolz, & Besner, 2010). Par ailleurs, les simulations sont effectuées le plus souvent sur des mots courts, donc dans des conditions qui minimisent, voire annulent, le rôle de l'attention. Ceci explique pourquoi les résultats des simulations peuvent être conformes aux données expérimentales malgré l'absence de composant attentionnel.

Le pendant de la question précédente est de se demander, au contraire, pourquoi les auteurs de deux modèles, un modèle de reconnaissance de mots et un modèle de dénomination orale, ont jugé nécessaire d'intégrer un composant attentionnel ? En fait, ces deux modèles avaient pour objectif non seulement de rendre compte du comportement normal mais également du comportement pathologique. Le modèle MORSEL (Mozer & Behrmann, 1990) se fixe pour objectif de simuler les performances de lecteurs cérébrolésés présentant une dyslexie par négligence. Simuler le fait que ces patients ne traitent qu'une portion spatialement définie de la séquence du mot implique de perturber le fonctionnement du composant attentionnel du modèle. Le modèle MTM (Ans et al., 1998) a également pour cahier des charges de rendre compte tant du comportement normal que pathologique. La simulation de cas de dyslexie acquise implique de restreindre le déploiement de la fenêtre attentionnelle du modèle. La modulation du composant attentionnel ouvre également la perspective de rendre compte des formes de dyslexies développementales secondaires à un trouble de l'attention visuelle. Comme ces deux derniers modèles, le modèle *BRAID* a été conçu dans le souci de rendre compte à la fois du comportement normal et du comportement pathologique. L'implémentation d'un composant attentionnel est indispensable pour rendre compte des pathologies de la lecture secondaires à un déficit de l'attention visuelle.

Même si la distribution de l'attention demeure fixe dans la plupart des simulations effectuées dans le cadre de cette thèse, nous avons modulé l'attention dans deux tâches portant sur l'effet de supériorité des mots et pseudo-mots. La première simulation où la distribution de l'attention a été manipulée, la Simulation A07, avait pour but de rendre compte des phénomènes d'amorçage du contexte de pseudo-mots. Dans cette simulation, le contexte *pseudo-mot* est présenté brièvement en amorce avant le pseudo-mot complet, la tâche consistant à identifier la lettre absente du contexte. Dans cette tâche, on s'attend à ce que l'amorçage ait un effet délétère sur les performances d'identification car l'amorce active les lettres de tous les mots du lexique qui partagent le même contexte. Or, expérimentalement, on observe un effet facilitateur de l'amorçage. Pour expliquer ce paradoxe, nous avons postulé un déplacement attentionnel sur la lettre cible durant la seconde partie de la présentation. Ce déplacement repose sur l'idée que la lettre apparaît subitement et donc capte l'attention, et sur le fait que la distribution construite à partir du contexte ne correspond pas au stimulus. Dans le modèle IA (McClelland & Rumelhart, 1981), les auteurs font le même constat d'un effet facilitateur inattendu. Pour le simuler, ils modifient les paramètres de façon ad-hoc, en supprimant la rétro-activation du niveau mot vers le niveau lettre, durant la phase de présentation du contexte.

La seconde simulation effectuée avec modification de la distribution de l'attention, la Simulation A09, avait pour but de rendre compte de l'effet de la prédictibilité de la position de la lettre cible. Dans cette simulation, nous montrons que le fait de connaître avec certitude la position de la lettre cible permet de déplacer le regard et le pic attentionnel sur la position cible, ce qui donne un avantage suffisant pour annuler l'effet de supériorité du mot. C'est la première fois, à notre connaissance, que cet effet est simulé dans le cadre d'un modèle de reconnaissance de mots.

En conclusion, introduire un composant attentionnel dans *BRAID* permet de rendre compte aisément de résultats expérimentaux qui posent problème aux modèles concurrents. Le modèle a également, de ce fait, le potentiel de rendre compte de comportements pathologiques. Plusieurs études ont d'ores et déjà été effectuées dans ce sens, même si elles ne sont pas exposées dans le cadre de cette thèse. Dans les travaux menés sur l'OVP (Phénix, Valdois, & Diard, 2016), nous avons simulé les performances décrites chez les faibles lecteurs comme relevant de ressources attentionnelles limitées (Aghababian & Nazir, 2000). Nous montrons qu'une réduction « pathologique » de la variance des distributions attentionnelles permet à *BRAID* de

simuler les courbes observées expérimentalement, c'est-à-dire des courbes de performances très piquées, « en V inversé », caractéristiques de la performance des faibles lecteurs.

### 2.3.3 Comment rendre compte des effets d'encombrement perceptif ou « crowding » ?

Il a été démontré expérimentalement que les capacités d'identification des lettres d'un mot varient selon la position occupée par la lettre dans la séquence du mot. Nous venons de voir que l'acuité donne un avantage à la lettre fixée mais pénalise l'identification à mesure que l'on s'éloigne du point de fixation. Nous avons également vu précédemment que la distribution de l'attention sur la séquence à traiter module l'identification des lettres. Ainsi, la lettre située sous la moyenne de la gaussienne est identifiée plus efficacement que les autres lettres de la séquence. Un troisième phénomène que nous n'avons pas encore détaillé est le phénomène d'encombrement perceptif ou « crowding ».

Aucune hypothèse n'a été spécifiquement formulée pour rendre compte de ce phénomène, selon lequel la perceptibilité d'une lettre est réduite lorsque celle-ci est entourée d'autres lettres (D. Whitney & Levi, 2011). En fait, aucune nouvelle hypothèse n'était nécessaire. C'est, dans *BRAID*, le mécanisme d'interférences latérales qui rend compte des phénomènes de « crowding ». Ce mécanisme d'interférence, postulé pour les besoins d'encodage positionnel du modèle, a pour effet secondaire de rendre compte des interférences entre lettres adjacentes caractéristiques du « crowding ».

En effet, pour chaque lettre de la séquence du mot, l'encodage positionnel distribué induit une incertitude sur l'identification de la lettre cible et de ses voisines. Cette incertitude est plus forte lorsqu'une lettre est entourée de deux lettres dont elle subit les interférences que pour les lettres des extrémités du mot (initiale et finale), qui ne subissent l'interférence que de la seule lettre qui leur est adjacente. Ainsi, sans formulation d'hypothèse supplémentaire, *BRAID* donne un avantage à l'identification des lettres initiale et finale. On notera que certains modèles, comme SCM (Davis, 2010b) par exemple, ont dû postuler l'existence de mécanismes spécifiques de codage positionnel propres aux lettres des extrémités du mot (les « letter banks », en l'occurrence).

Ainsi, selon *BRAID*, l'identification des lettres dans la séquence du mot est modulée par le jeu combiné de l'acuité visuelle, de l'attention et des interférences latérales. Expérimentalement, les profils classiquement observés lors du traitement de séquences de cinq lettres centrées sur le point de fixation sont « en W » sur les scores et « en M » sur les temps d'identification. Ces profils reflètent les effets limités d'interférence latérale sur les lettres des extrémités (conduisant à une identification meilleure et plus rapide des lettres initiales et finales) et l'effet facilitateur du pic attentionnel en position centrale.

Une seule des simulations présentées dans le cadre de cette thèse, la Simulation A10, a manipulé les interférences latérales afin de simuler les effets dus à l'espacement des lettres du mot tels qu'ils sont observés expérimentalement. L'espacement des lettres a pour effet de réduire les interférences latérales (Martelli, Di Filippo, Spinelli, & Zoccolotti, 2009), ce qui a pu être reproduit dans cette simulation. À notre connaissance, aucun autre modèle n'a tenté de reproduire les effets d'espacement des lettres sur la reconnaissance du mot.

La manipulation des interférences latérales a également été nécessaire dans l'étude sur l'OVP (Phénix, Valdois, & Diard, 2016) que nous avons menée en parallèle de cette thèse. Tel que précisé dans la section précédente, l'objectif de cette simulation était de reproduire les courbes d'OVP caractéristiques des faibles lecteurs et des lecteurs dyslexiques. Nous avons montré que *BRAID* était capable de reproduire ces courbes moyennant une réduction de la variance de la distribution attentionnelle sur la séquence du mot. Il était donc tentant d'en conclure que cela

appuyait l'hypothèse d'un déficit de l'empan visuo-attentionnel chez les faibles lecteurs et que la relation était de nature causale : les courbes OVP sont atypiques parce que les enfants présentent un déficit de l'empan visuo-attentionnel. La simulation nous a permis de faire un contrôle supplémentaire : est-ce que nous obtiendrions les mêmes résultats par simple accroissement des interférences latérales ? Plusieurs études expérimentales montrent que les lecteurs dyslexiques présentent un encombrement visuel excessif (Martelli et al., 2009) et *BRAID* pouvait nous aider à discuter du rôle potentiel de ce déficit dans les courbes OVP observées. Nous avons montré que les courbes obtenues par simulation après accroissement des interférences latérales ne correspondent pas aux courbes observées expérimentalement, contrairement à celles obtenues par réduction de la distribution de l'attention.

Globalement, *BRAID* implémente des mécanismes de position du regard, d'acuité visuelle, d'interférence latérale et d'attention qui ont tous un effet sur l'identification des lettres dans la séquence d'un mot. Il offre ainsi la possibilité de mesurer indépendamment l'impact de chaque mécanisme sur l'identification des lettres du mot et leur action combinée. Cette propriété est de nature à ouvrir de nouvelles pistes d'interprétation des phénomènes étudiés et devrait permettre de faire de nouvelles prédictions donnant lieu à des expériences inédites.

## 2.4 Les limites actuelles de *BRAID*

**BRAID : un modèle probabiliste... déterministe** Bien que le cadre de modélisation adopté dans *BRAID* soit probabiliste, les résultats que nous obtenons dans nos simulations sont déterministes. En effet, une fois le jeu de paramètres, notamment les seuils de décision et le stimulus fixés, la courbe de décours temporel produite par *BRAID* est toujours strictement identique pour une tâche donnée. Ainsi, en l'état, muni d'un unique jeu de paramètres, *BRAID* ne modélise ni variabilité inter-essais, ni variabilité inter-individus.

Nous interprétons *BRAID* comme simulant un comportement moyen. Par exemple, la matrice de confusion, qui résume la proximité visuelle des lettres en termes de traits, est obtenue par moyennage des données comportementales de plusieurs sujets. L'ensemble des mots du sous-modèle de connaissance lexicale provient d'un lexique construit à partir de centaines de résultats comportementaux permettant de définir la fréquence relative des mots. *BRAID* est donc un modèle doté de connaissances correspondant à une moyenne des lecteurs. Il génère des résultats de simulations qui sont comparés aux données d'expériences comportementales, souvent résumées par leurs moyennes. Ainsi, dans cette thèse, nous ne rendons pas compte de la variabilité expérimentalement observée, comme par exemple les distributions de temps de réaction observées expérimentalement, exploitées par les modèles MROM (Grainger & Jacobs, 1996) ou BR (Norris, 2006). Ceci peut donc être identifié comme une limite du modèle *BRAID*.

Pour pallier cette difficulté, nous distinguons deux cas : la variabilité inter-essais et la variabilité inter-individus. Pour prendre en compte la variabilité inter-essais produite par un même individu, nous pourrions faire varier le seuil de décision associé à la tâche, pour chaque essai, en décrivant le seuil non pas par une valeur mais par une distribution de probabilité mono-modale. C'est l'option qui est choisie pour le modèle SIAM (Jacobs & Grainger, 1992). Nous pourrions également supposer que le mécanisme d'inférence probabiliste n'est pas exact, comme dans nos simulations, mais approximé, et procédant par tirages aléatoires intermédiaires pour l'approximation des marginalisations. Ainsi, même dans des conditions identiques, ces tirages aléatoires selon les distributions du modèle pourraient être interprétées comme des « choix intermédiaires », par exemple sur l'identité des lettres.

Pour rendre compte de la variabilité inter-individus, les possibilités sont plus nombreuses. On pourrait ajouter deux sources de variabilité supplémentaires : la première modification

concernerait le lexique, en jouant sur le nombre de mots connus et leur fréquence. On pourrait par exemple exclure les mots de fréquence faible, pour simuler des individus qui sont peu exposés aux textes écrits et développent un vocabulaire limité. De même, on pourrait faire varier la fréquence des mots en fonction de contraintes sémantiques, pour simuler par exemple le profil de personnes travaillant dans différents secteurs d'activités ou ayant des centres d'intérêt spécifiques (vocabulaire spécialisé). La seconde source de variabilité proviendrait du sous-modèle sensoriel des lettres : en modifiant les valeurs de la diagonale de la matrice de confusion, nous pourrions simuler des personnes avec des capacités d'identification de lettres isolées plus ou moins bonnes et, en modifiant le paramètre d'interférence  $\theta_i$ , nous pourrions jouer sur les capacités de discrimination de lettres adjacentes.

Nous pourrions également approcher la variabilité inter-individuelle observée expérimentalement en créant un modèle *BRAID* par sujet expérimental et en ajustant les paramètres de *BRAID* à ceux des individus testés. Par exemple, si on mesure l'acuité visuelle, les effets de « crowding » et les capacités d'empan visuo-attentionnel d'un individu, on devrait pouvoir créer un modèle *BRAID* doté des paramètres ajustés correspondants. Dans ce cas, on s'attendrait à obtenir des performances en reconnaissance de mots ou décision lexicale comparables à celles observées chez cet individu spécifiquement. En créant autant de modèles que de sujets expérimentaux, on devrait pouvoir reproduire la variabilité inter-individuelle observée. Nous avons procédé, avec une stratégie apparente mais à plus gros grain, dans nos simulations sur l'OVP : nous y simulons deux groupes de lecteurs différents, bien identifiés expérimentalement (les normo-lecteurs et les faibles lecteurs), par des jeux de paramètres différents de *BRAID* (avec variance attentionnelle normale ou limitée).

**Temps de calcul vs. précision de la modélisation : le bon compromis** Les modèles computationnels sont toujours confrontés, à un moment ou à un autre de leur développement, au problème du temps de calcul. Le modélisateur doit donc faire des compromis entre une implémentation rigoureuse du modèle et des temps de calcul raisonnables. Dans le domaine de la reconnaissance visuelle de mots, la stratégie consiste souvent à limiter le lexique. Ainsi, le modèle IA (McClelland & Rumelhart, 1981), pionnier des modèles de reconnaissance visuelle de mots, a choisi de n'utiliser qu'une partie du lexique, avec 1 179 mots de quatre lettres, pour réaliser ses simulations. Le modèle BR (Norris, 2006), bien qu'il soit capable de reconnaître des mots de toute taille, a choisi de restreindre le lexique aux mots de même taille que le stimulus : par exemple, pour un stimulus de 5 lettres, le lexique n'est composé que des 4 106 mots de 5 lettres. Enfin, le modèle MORSEL (Mozer & Behrmann, 1990) a choisi d'utiliser un lexique adapté pour chaque stimulus présenté (composant PO), lexique qui contient en moyenne 300 mots ou semlex, 350 trigrammes et 17 000 connexions, alors que plus d'1,2 millions de connexions auraient été nécessaires avec un lexique complet.

Dans *BRAID*, nous avons été confronté au même problème. Nous avons également choisi de limiter la taille du lexique utilisé dans les simulations aux mots de même taille que celle du stimulus. Ce choix nous a contraint à utiliser la notion de voisinage de Coltheart, Davelaar, Jonasson, et Besner (1977), consistant à ne considérer comme voisins orthographiques que les mots de même longueur qui ne diffèrent que d'une lettre du stimulus cible. Cette conséquence sur la définition du voisinage ne nous paraît pas dommageable dans la mesure où la même définition était adoptée pour la sélection des stimuli dans les expériences comportementales. Néanmoins, ce choix est limitatif dans les simulations, puisqu'on se prive de l'influence des mots de tailles différentes (à plus ou moins une lettre notamment) sur les performances du modèle en identification de mots.

Nous avons également fait des compromis en fonction de la tâche. Dans le calcul de l'identifi-

cation de lettres avec influence lexicale, nous avons fait le choix de n'intégrer l'influence lexicale qu'au dernier pas de temps. Pour cela, nous avons séparé le calcul de l'identification des lettres du calcul de la reconnaissance des mots. Sans cette simplification, nous aurions dû, à la place, réaliser l'inférence dans deux chaînes de Markov couplées, à chaque pas de temps. Une technique approximée de calcul aurait pu être employée, et l'inférence aurait propagé l'information probabiliste dans cette boucle, jusqu'à convergence numérique. Ce processus aurait eu un coût important sur les temps de calcul. De plus, les connaissances lexicales auraient été « rétro-propagées » sur les lettres à chaque pas de temps ; avec le jeu de paramètres actuel, cela accélérerait grandement les temps des simulations d'identification de lettres, les rendant peu réalistes. Néanmoins, nous pensons qu'un recalibrage du modèle, par exemple en diminuant la quantité d'information sensorielle acquise à chaque itération, pourrait totalement compenser cette accélération de la dynamique de reconnaissance.

De plus, ce choix que les connaissances lexicales n'influent qu'au dernier pas de temps sur l'identité des lettres pourrait être interprété : en présence du stimulus, le modèle *BRAID* actuellement implémenté fonctionne comme s'il essayait d'extraire les informations d'identité sur les lettres du stimulus, et l'information se propage de manière contingente sur les connaissances lexicales (et donc la distribution sur  $W$  évolue). Ensuite, le stimulus disparaît de l'écran. Le système utilise alors toute l'information dont il dispose en combinant l'information lexicale avec la distribution de probabilité sur les lettres, afin de reconnaître au mieux la lettre cible. Une question intéressante consisterait à savoir si ce mécanisme procéderait par rétro-propagation des informations lexicales sur la distribution des lettres  $P_n^t$ , ou bien si les deux seraient combinées sur un autre espace de lettres, possiblement distinct neurobiologiquement de  $P_n^t$ , laissant ainsi les distributions  $P_n^t$  inchangées.

Nous avons également « simplifié » le calcul de la réponse NON en décision lexicale. Ce calcul repose sur la recherche d'erreurs. Un non-mot peut être vu comme un mot dont certaines lettres sont différentes. Ainsi, si le stimulus diffère de tous les mots du lexique d'au moins une lettre, alors ce n'est pas un mot. Faire une recherche d'erreurs, systématiquement pour toutes les combinaisons d'erreurs possibles, induirait un temps de calcul beaucoup trop long. En effet, pour un mot de longueur  $N$ , il y a  $2^N - 1$  patterns d'erreurs possibles. Nous avons donc fait le choix de ne rechercher qu'une erreur au plus dans la séquence, ne considérant donc que les  $N$  patterns avec une erreur. Cette simplification induit un allongement du temps nécessaire pour la réponse NON dans les simulations (en effet, les mots avec plusieurs erreurs ne sont pas plus rapidement détectés que lorsqu'ils en ont une seule), déséquilibrant le rapport entre le temps nécessaire à la réponse NON et à la réponse OUI, à l'avantage de cette dernière. Cette approximation affecte certainement les temps précis simulés, mais ne modifie vraisemblablement pas les patterns de réponse.

Nous comprenons que les simulations n'auraient pas pu être effectuées, en tout cas pas sur l'éventail des simulations présentées, sans les compromis que nous avons acceptés de faire. Par ailleurs, les conséquences de ces compromis nous semblent acceptables dans la mesure où ils ne devraient pas modifier le sens des effets simulés. Il existe cependant des pistes d'amélioration de l'implémentation de *BRAID*, pour accélérer les temps de calcul. Ainsi, le goulet d'étranglement du modèle *BRAID* est parfaitement identifié : c'est le calcul de similarité entre les percepts et l'ensemble des mots du lexique, qui est réalisé à chaque pas de temps. De nombreuses solutions informatiques existent et pourraient être étudiées, comme par exemple profiter des capacités de parallélisation massives du calcul matriciel, en exploitant les architectures des cartes graphiques (calcul sur GPU).



### 3 Perspectives

Dans cette section, nous présentons certaines des possibilités du modèle *BRAID* qui n'ont pas été présentées dans le cadre de ce travail de thèse, mais qui font soit l'objet de travaux en cours, soit de perspectives à court et moyen terme.

#### 3.1 *BRAID* : un cadre pour la comparaison de modèles

Jusqu'ici nous avons considéré *BRAID* uniquement comme un modèle de reconnaissance visuelle de mots. Mais *BRAID* fournit également un cadre général de modélisation probabiliste, dont nous pensons qu'il pourrait englober les modèles théoriques proposés ces quarante dernières années. Ainsi, les modèles seraient décrits dans un même cadre mathématique et théorique, et leurs résultats ne dépendraient plus des spécificités de leurs implémentations. On pourrait alors les comparer sur des simulations identiques, leurs résultats ne dépendant que des choix théoriques différents qu'ils font. De plus, ces modèles étant tous exprimés dans le même formalisme probabiliste, nous pourrions les comparer grâce à des méthodes de comparaisons de modèles elles-mêmes probabilistes. Nous obtiendrions ainsi une uniformité mathématique pour les modèles et les outils de comparaison. Nous présentons trois pistes de réécriture, dans *BRAID*, de modèles existants : le modèle IA (McClelland & Rumelhart, 1981) comme modèle de référence, le modèle BR (Norris, 2006) pour sa simplicité et le modèle MORSEL (Mozzer & Behrmann, 1990), à l'inverse, pour sa complexité.

**Implémentation du modèle IA avec *BRAID*** La structure du modèle IA est assez proche du modèle *BRAID*. Pour implémenter les principes de ce modèle, en partant de l'implémentation actuelle de *BRAID*, il faut en faire disparaître la distribution attentionnelle. Pour cela, il suffit de fixer les  $P(C_{A_n}^t | A)$  à une valeur constante pour toutes les positions. Ensuite, il faut supprimer les interférences latérales. En fixant la valeur de  $\Theta_I$  à 1, on passe d'un encodage distribué à un encodage strict. Il faut enfin modifier la matrice de confusion. Elle doit rendre compte de la similarité des traits détectés, et comme la variabilité du modèle repose sur la détection des traits visuels, il faut enrichir *BRAID* d'une variable informant le modèle de la détection ou non des traits présents dans le stimulus. On peut remarquer que la règle de normalisation jouerait ici le rôle « d'inhibiteur latéral » au sein de chaque niveau. Fort de ces modifications, nous aurions une implémentation probabiliste du modèle théorique à la base du modèle IA (McClelland & Rumelhart, 1981), à comparer à l'interprétation probabiliste discutée par McClelland (2013).

**Implémentation du modèle BR avec *BRAID*** Pour implémenter le modèle théorique à la base du modèle BR (Norris, 2006), il faut simplifier *BRAID* au maximum. Nous supprimons le sous-modèle d'attention visuelle. Nous réduisons le sous-modèle sensoriel des lettres aux variables  $I_n^t$ , sans interférence (suppression des variables  $G^t$ ,  $\Delta I_n^t$ ) et les distributions de probabilités  $P(I_n^t | S_n^t)$  sont obtenues par un tirage sur une distribution gaussienne réalisée à chaque pas de temps. Enfin, nous supprimons toute forme de déclin dans le modèle en fixant  $Leak_P$  et  $Leak_W$  à 0. Fort de ces modifications, nous aurions une implémentation du modèle théorique à la base du modèle BR (Norris, 2006).

**Implémentation du modèle MORSEL avec *BRAID*** Enfin, pour implémenter le modèle théorique sous-jacent au modèle MORSEL (Mozzer & Behrmann, 1990) à partir de la version actuelle du modèle *BRAID*, la tâche est beaucoup plus complexe. Cette fois, il faut enrichir *BRAID* à de nombreux niveaux. Tout d'abord, il faut casser le sous-modèle sensoriel des lettres

et remplacer ce qui se passe entre  $S_n^t$  et  $I_N^t$  par les différentes couches de traitements du stimulus, qu'il faut connecter au composant attentionnel. En effet, notre sous-modèle sensoriel des lettres est simplifié ; d'autres modèles, par exemple neurobiologiquement inspirés, détaillent ici les processus de détection de traits, de contours, etc. (Dehaene et al., 2005). Cela demande par exemple de représenter le stimulus non plus directement dans l'espace abstrait des lettres, mais par une représentation de l'image visuelle, comme une matrice à deux dimensions de pixels ou de réponses de capteurs rétiniens. Il faut également modifier l'interface commune entre le sous-modèle perceptif des lettres et le sous-modèle de connaissances lexicales. Ce ne sont plus des lettres mais des trigrammes. Il faut donc ajouter un niveau « trigramme » au sous-modèle perceptif des lettres qui « s'alimente » à partir des lettres. En revanche, au niveau du sous-modèle de connaissances lexicales, il suffit de remplacer les variables  $L_n^t$  par des triplets. On mesure ici la complexité du modèle résultant de ces opérations. Par exemple, il faudrait que le nombre de trigrammes évolue en fonction de la taille du stimulus.

### 3.2 Extensions du modèle *BRAID*

Avec *BRAID*, nous implémentons un modèle de connaissances dont l'objectif principal est la reconnaissance visuelle de mots, c'est-à-dire le processus qui est au cœur de toutes les activités de lecture, qu'il s'agisse de lecture à haute voix ou de contrôle oculo-moteur lors de la lecture de texte (Norris, 2013). Cependant, un intérêt de la méthodologie de la modélisation bayésienne algorithmique est sa flexibilité, dans le sens où les modèles peuvent être étendus, en intégrant de nouveaux sous-modules, de manière facilement interprétable et contrainte mathématiquement (Patri, Perrier, & Diard, 2016). Le modèle *BRAID* peut dès lors être conçu comme une première étape dans le développement ultérieur de modèles étendus. Nous aborderons dans cette section les extensions envisageables de façon à dessiner les perspectives de modélisations futures.

**Extension par ajout de composants phonologiques** Une première piste d'extension concerne tout naturellement l'ajout de composants phonologiques, afin de transformer *BRAID* pour obtenir un modèle de dénomination orale de mots écrits. Trois types de modèles computationnels de ce type ont été proposés jusqu'ici. Les modèles de type « double-voie » (Coltheart et al., 2001 ; Perry et al., 2007) intègrent le modèle IA (McClelland & Rumelhart, 1981) de reconnaissance de mots, alors même qu'il est aujourd'hui démontré que cette conception de la reconnaissance de mots est obsolète – remise en question de l'encodage positionnel strict, absence de mécanismes de traitement visuel (acuité, interférences latérales et attention). Le modèle connexionniste en triangle (Seidenberg & McClelland, 1989) ignore les traitements visuels impliqués dans l'identification des lettres de la séquence à lire. Enfin, le modèle connexionniste MTM (Ans et al., 1998) incorpore bien un composant attentionnel (quoique fortement simplifié par rapport à celui de *BRAID*), mais aucun des autres composants (position du regard, acuité et interférences latérales). Enrichir *BRAID* par ajout de composants phonologiques permettrait d'implémenter le premier modèle de dénomination orale de mots écrits qui incorpore un modèle sophistiqué de reconnaissance de mots.

L'intégration d'un modèle sophistiqué de reconnaissance de mots à un modèle de dénomination orale aura pour conséquence de rééquilibrer les dimensions orthographiques et phonologiques du modèle étendu. En effet, la plupart des modèles existants de dénomination orale de mots écrits mettent l'accent sur les processus phonologiques, largement détaillés, au détriment des processus d'analyse orthographique propres à la reconnaissance de mots. Un modèle intégrant des traitements visuels détaillés et des traitements phonologiques devrait permettre de mieux comprendre l'importance relative des effets orthographiques et des effets phonologiques

en dénomination de mots écrits. C'est également la condition *sine qua non* pour simuler la variété des profils de lecture atypiques observés en contexte dyslexique, qu'il s'agisse de dyslexies acquises ou de dyslexies développementales. Notons qu'un travail de thèse débute actuellement, au laboratoire, dans cette direction.

**Modélisation du contrôle oculo-moteur pour la lecture de textes** Il existe actuellement une dichotomie entre les modèles de reconnaissance de mots ou de dénomination orale et les modèles de contrôle des mouvements oculaires en lecture de texte. Cette dichotomie est liée d'une part à l'objet d'étude, que sont les mots isolés d'un côté, et les mots en contexte de l'autre, et, d'autre part, à l'absence de composants visuels et attentionnels dans les modèles du premier type. En particulier, les modèles de contrôle oculo-moteur incorporent systématiquement un composant attentionnel qui fait cruellement défaut aux modèles de reconnaissance de mots ou de dénomination orale.

*BRAID* semble plus apte que ces prédécesseurs à faire le pont entre les modèles de reconnaissance ou de dénomination de mots et les modèles de contrôle oculo-moteur, grâce au composant attentionnel qu'il inclut. De plus, la position du regard et la position de l'attention sont des composants du modèle qui peuvent être manipulés indépendamment, ce qui est utile dans la lecture de séquences de mots. On peut notamment très facilement imaginer une version du modèle permettant à l'attention de se déplacer sur le mot suivant, avant que la saccade vers ce mot (changement de position du regard) ne soit initiée. *BRAID* semble posséder a priori les composants nécessaires à la création d'un modèle plus unifié qui intègre reconnaissance de mots, dénomination orale et contrôle oculo-moteur au sein d'une même architecture.

Cela ne pourrait être par ailleurs envisagé qu'à relativement long terme, moyennant une adaptation du modèle au traitement des mots en contexte. Équipé d'un composant de reconnaissance de mot sophistiqué, le modèle devrait pouvoir rendre compte des effets de fréquence et de voisinage sur les mouvements oculaires. Il serait en mesure de simuler la reconnaissance de mots et donc de déterminer à quel moment l'attention devrait se reporter sur l'analyse du mot suivant. Par contre, le modèle n'est pas doté de connaissances sémantiques, dont on connaît pourtant le rôle non seulement dans l'apparition des saccades régressives mais aussi dans la prédictibilité du mot selon le contexte (Rayner, 1998).

Même si le formalisme est flexible, passer du modèle *BRAID* de reconnaissance de mots isolés à la simulation des mouvements oculaires lors de la lecture de texte devrait impacter les quatre niveaux structuraux du modèle. Il faudrait minimalement ajouter une variable « texte » (une séquence de mots organisés en deux dimensions) au sous-modèle sensoriel des lettres, et modifier la distribution des stimuli pour qu'ils dépendent de la distribution attentionnelle, comme dans le modèle MORSEL (Mozer & Behrmann, 1990). Il faudrait enrichir le sous-modèle attentionnel pour déplacer le centre de l'attention lorsque la probabilité de reconnaissance du mot précédent dépasse un seuil prédéfini. Il faudrait adapter les paramètres de déclin au sein des sous-modèles perceptif et de connaissances lexicales pour qu'ils s'ajustent aux conditions expérimentales de la tâche.

**Modélisation de l'apprentissage de la lecture** Concernant l'apprentissage de la lecture, *BRAID* permet dès à présent de s'interroger sur les mécanismes d'apprentissage de mots orthographiques nouveaux. Le modèle de reconnaissance de mots est capable de déterminer si un stimulus présenté fait partie des connaissances lexicales ou non. Cette fonction est assurée par le sous-modèle de décision lexicale qui a été implémenté dans le cadre de cette thèse dans le seul but de mesurer les temps nécessaires à l'acceptation ou au rejet d'un stimulus en tant que mot. La question est à présent de déterminer comment une séquence orthographique rejetée

(réponse NON en DL) parce que ne correspondant à aucun mot connu peut être apprise et intégrée aux connaissances lexicales.

La première question sera de déterminer si un mot nouveau peut être appris à partir du traitement parallèle de la séquence des lettres qui le composent. La réponse à cette question devrait être négative si le modèle est fidèle aux données comportementales. En effet, les débutants lecteurs traitent les mots lettre à lettre, ce qui reflète un traitement sériel systématique et laborieux. Apprendre la forme orthographique d'un nouveau mot semble donc impliquer d'extraire une information optimale sur l'identité des lettres dans chacune des positions de la séquence. Cela impliquerait de manipuler la distribution de l'attention de façon à « piquer » sur chacune des lettres successivement et à réduire voire annuler toute interférence latérale. Cette manipulation de l'attention permettra de passer d'un traitement parallèle (distribution de l'attention sur la séquence de lettres) à un traitement sériel (réduction de la variance de distribution de l'attention à une seule lettre cible à la fois et déplacement attentionnel). C'est là une originalité forte du modèle *BRAID* qui ressemble en cela fortement à ces prédécesseurs dotés d'un composant attentionnel, MORSEL (Mozer & Behrmann, 1990) et MTM (Ans et al., 1998). Dans ces trois modèles, la nature parallèle ou sérielle du traitement dépend de la distribution de l'attention alors que dans les modèles sans composant attentionnel, la distinction entre traitement parallèle des mots connus et traitement séquentiel des mots nouveaux implique de postuler des voies de lecture différentes (Coltheart et al., 2001 ; Perry et al., 2007). Notons qu'un travail de thèse est en cours actuellement, au laboratoire, dans cette direction, dans lequel nous explorons la capacité du modèle à basculer d'un mode « rapide et parallèle » pour les mots connus, à un mode « lent et sériel » pour les mots nouveaux, sur la base de l'optimisation du gain espéré d'entropie sur les distributions perceptives.

**Universalité du modèle** Le dernier point que nous souhaitons aborder dans le cadre de cette discussion concerne l'universalité du modèle *BRAID*, qui pourrait être exploitée à plus long terme. En effet, nous remarquons que, dans *BRAID*, il y a quelques hypothèses qui sont spécifiques du traitement visuel des lettres et des mots. Les stimuli sont supposés être des lettres, organisées spatialement selon une seule dimension (représentée, dans le modèle, par les indices  $1 : N$  des variables  $S, I, P, L$ , etc.) et les mots sont spécifiques à la langue dans laquelle sont effectuées les simulations mais aucun composant du modèle n'est spécifique à un système orthographique donné, contrairement aux modèles à encodage positionnel relatif par exemple. Ces derniers peuvent certes rendre compte de la flexibilité du codage positionnel observé dans les langues indo-européennes mais ils ne peuvent pas rendre compte du codage positionnel strict nécessaire au traitement des langues sémitiques (Frost, 2012). Nous avons déjà pu vérifier dans le cadre du travail de master que j'ai encadré (Ginestet, 2016) que *BRAID* peut facilement incorporer des lexiques de langues indo-européennes autres que le français pour la simulation de données obtenues dans ces langues. Nous faisons le pari que *BRAID* devrait s'avérer capable de simuler la reconnaissance de mots dans des langues de familles linguistiques très différentes, comme l'hébreu ou l'arabe ou même le chinois, dans la mesure où il est doté d'une majorité de composants qui ne sont spécifiques à aucune langue.

En fait, les composants implémentés ne sont même pas spécifiques au traitement des lettres et des mots. Par exemple, les interférences latérales traduisent vraisemblablement des propriétés neurobiologique du traitement des formes visuelles dans les couches successives du cortex visuel. Dans ces traitements, les objets différents mais très proches peuvent tomber dans le même champ récepteur d'un neurone ou d'une population de neurones, et donc interférer. La position du regard a évidemment une existence physique indépendamment de la lecture de lettres et de mots. L'attention visuelle contrôlée est également décrite dans de nombreux domaines de la

recherche visuelle ; elle y est par exemple décrite comme un « spot attentionnel », qui pourrait être une extension naturelle deux-dimensionnelle de notre répartition gaussienne de l'attention dans *BRAID*. Enfin, les connaissances lexicales ont pour rôle de prédire les configurations d'objets de base qui constituent les unités de plus haut niveau ; en ce sens, elles ont un rôle analogue à des mémoires de formes visuelles complexes, comme la mémoire des visages par exemple.

Il serait donc intrigant de vérifier si *BRAID* peut s'adapter à des domaines comme la reconnaissance de visages, la recherche visuelle d'objets, ou encore l'exploration de scènes visuelles naturelles. Nous pensons que de nombreux mécanismes et hypothèses de *BRAID* sont en réalité des descriptions générales des processus de traitement visuel, indépendamment du contexte spécifique de la lecture de lettres, et que, en conséquence, le modèle *BRAID* pourrait facilement y être adapté.

# Bibliographie

- Adelman, J. S. (2011). Letters in time and retinotopic space. *Psychological Review*, 118(4), 570–82.
- Adelman, J. S., Marquis, S. J., & Sabatos-DeVito, M. G. (2010). Letters in words are read simultaneously, not in left-to-right sequence. *Psychological Science*, 21(12), 1799–1801.
- Aghababian, V., & Nazir, T. A. (2000). Developing normal reading skills: aspects of the visual processes underlying word recognition. *Journal of Experimental Child Psychology*, 76, 123–150.
- Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ : Erlbaum.
- Andrews, S. (1989). Frequency and neighborhood effects on lexical access: Activation or search? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(5), 802–814.
- Andrews, S. (1992). Frequency and neighborhood effects on lexical access: Lexical similarity or orthographic redundancy? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(2), 234–254.
- Andrews, S. (1997). The effect of orthographic similarity on lexical retrieval: Resolving neighborhood conflicts. *Psychonomic Bulletin & Review*, 4(4), 439–461.
- Ans, B., Carbonnel, S., & Valdois, S. (1998). A connectionist multiple-trace memory model for polysyllabic word reading. *Psychological Review*, 105(4), 678–723.
- Balota, D. A., Yap, M. J., Cortese, M. J., Hutchison, K. A., Kessler, B., Loftis, B., . . . Treiman, R. (2007). The English Lexicon Project. *Behavior Research Methods*, 39(3), 445–459.
- Baron, J., & Thurston, I. (1973). An analysis of the word-superiority effect. *Cognitive Psychology*, 4(2), 207–228.
- Becker, C. A. (1976). Allocation of attention during visual word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 2(4), 556–566.
- Becker, C. A. (1980). Semantic context effects in visual word recognition: An analysis of semantic strategies. *Memory & Cognition*, 8(6), 493–512.
- Besner, D., & McCann, R. S. (1987). Word frequency and pattern distortion in visual word identification and production: An examination of four classes of models. In M. Coltheart (Ed.), *Attention and performance (vol. 12)* (pp. 201–220). Hillsdale, NJ : Lawrence Erlbaum Associates, Inc.
- Besner, D., Risko, E. F., Stolz, J. A., White, D., Reynolds, M., O'Malley, S., & Robidoux, S. (2016). Varieties of attention: Their roles in visual word identification. *Current Directions in Psychological Science*, 25(3), 162–168.
- Bessi re, P., Dedieu, E., Lebeltel, O., Mazer, E., & Mekhnacha, K. (1998a). Interpr tation vs. description II : Fondements math matiques. *Intellectica*, 26-27, 313-336.
- Bessi re, P., Dedieu, E., Lebeltel, O., Mazer, E., & Mekhnacha, K. (1998b). Interpr tation vs. description I : Proposition pour une th orie probabiliste des syst mes cognitifs sensori-moteurs. *Intellectica*, 26-27, 257-311.
- Bessi re, P., Laugier, C., & Siegwart, R. (Eds.). (2008). *Probabilistic reasoning and decision making in sensory-motor systems* (Vol. 46). Berlin : Springer.

- Bessière, P., Mazer, E., Ahuactzin, J. M., & Mekhnacha, K. (2013). *Bayesian programming*. Boca Raton, Florida : CRC Press.
- Boehm, U., Hawkins, G. E., Brown, S., van Rijn, H., & Wagenmakers, E.-J. (2016). Of monkeys and men: Impatience in perceptual decision-making. *Psychonomic Bulletin & Review*, *23*, 738–749.
- Brown, T. L., Gore, C. L., & Carr, T. H. (2002). Visual attention and word recognition in Stroop color naming: is word recognition “automatic”? *Journal of Experimental Psychology: General*, *131*(2), 220–240.
- Brown, T. L., Roos-Gilbert, L., & Carr, T. H. (1995). Automaticity and word perception: evidence from Stroop and Stroop dilution effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*(6), 1395–1411.
- Brysbaert, M., & New, B. (2009). Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, *41*(4), 977–990.
- Carr, T. H., Davidson, B. J., & Hawkins, H. L. (1978). Perceptual flexibility in word recognition: strategies affect orthographic computation but not lexical access. *Journal of Experimental Psychology: Human Perception and Performance*, *4*(4), 674–690.
- Carr, T. H., Lehmkuhle, S. W., Kottas, B., Astor-Stetson, E. C., & Arnold, D. (1976). Target position and practice in the identification of letters in varying contexts: A word superiority effect. *Perception & Psychophysics*, *19*(5), 412–416.
- Carroll, J. B., Davies, P., & Richman, B. (1971). *The american heritage word frequency book*. Houghton Mifflin Boston.
- Chase, C. H., & Tallal, P. (1990). A developmental, interactive activation model of the word superiority effect. *Journal of Experimental Child Psychology*, *49*(3), 448–487.
- Colas, F., Diard, J., & Bessière, P. (2010). Common Bayesian models for common cognitive issues. *Acta Biotheoretica*, *58*(2-3), 191–216.
- Coltheart, M., Davelaar, E., Jonasson, J. T., & Besner, D. (1977). Access to the internal lexicon. In S. Dornic (Ed.), *Attention and performance VI* (pp. 535–555). Hillsdale, NJ : Erlbaum.
- Coltheart, M., Rastle, K., Perry, C., Langdon, R., & Ziegler, J. C. (2001). DRC: A dual route cascaded model of visual word recognition and reading aloud. *Psychological Review*, *108*(1), 204–256.
- Davis, C. J. (2010a). SOLAR versus SERIOL revisited. *European Journal of Cognitive Psychology*, *22*(5), 695–724.
- Davis, C. J. (2010b). The spatial coding model of visual word identification. *Psychological Review*, *117*(3), 713.
- Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, *5*(3), 142–150.
- Dehaene, S., Cohen, L., Sigman, M., & Vinckier, F. (2005). The neural code for written words: a proposal. *Trends in Cognitive Sciences*, *9*(7), 335–341.
- Diard, J. (2015). *Bayesian algorithmic modeling in cognitive science* (Habilitation à Diriger des Recherches (HDR)). Université Grenoble Alpes.
- Drugowitsch, J., & Pouget, A. (2012). Probabilistic vs. non-probabilistic approaches to the neurobiology of perceptual decision-making. *Current Opinion in Neurobiology*, *22*, 963–969.
- Embick, D., Hackl, M., Schaeffer, J., Kelepir, M., & Marantz, A. (2001). A magnetoencephalographic component whose latency reflects lexical frequency. *Cognitive Brain Research*, *10*(3), 345–348.

- Engbert, R., Longtin, A., & Kliegl, R. (2002). A dynamical model of saccade generation in reading based on spatially distributed lexical processing. *Vision Research*, *42*, 621–636.
- Ferrand, L. (2007). *Psychologie cognitive de la lecture – reconnaissance des mots écrits chez l'adulte*. Bruxelles : De Boeck Supérieur.
- Forster, K. I., & Shen, D. (1996). No enemies in the neighborhood: absence of inhibitory neighborhood effects in lexical decision and semantic categorization. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*(3), 696–713.
- Frost, R. (2012). Towards a universal model of reading. *Behavioral and Brain Sciences*, *35*(5), 263–279.
- Geisler, W. S. (1989). Sequential ideal-observer analysis of visual discriminations. *Psychological Review*, *96*(2), 267–314.
- Gilet, E., Diard, J., & Bessière, P. (2011). Bayesian action-perception computational model: Interaction of production and recognition of cursive letters. *PLoS ONE*, *6*(6), e20387.
- Gilmore, G. C., Hersh, H., Caramazza, A., & Griffin, J. (1979). Multidimensional letter similarity derived from recognition errors. *Perception & Psychophysics*, *25*(5), 425–431.
- Ginestet, E. (2016). *Modélisation probabiliste de reconnaissance visuelle des mots : simulation d'effets d'amorçage et de compétition lexicale* (Mémoire de Master non publié). Univ. Grenoble Alpes.
- Gomez, P., Ratcliff, R., & Perea, M. (2008). The Overlap model: a model of letter position coding. *Psychological Review*, *115*(3), 577–601.
- Goolkasian, P., & King, J. (1990). Letter identification and lateral masking in dyslexic and average readers. *The American Journal of Psychology*, *103*(4), 519–538.
- Grainger, J., Bouttevin, S., Truc, C., Bastien, M., & Ziegler, J. (2003). Word superiority, pseudoword superiority, and learning to read: A comparison of dyslexic and normal readers. *Brain and Language*, *87*(3), 432–440.
- Grainger, J., Dufau, S., & Ziegler, J. C. (2016). A vision of reading. *Trends in Cognitive Sciences*, *20*(3), 171–179.
- Grainger, J., Granier, J.-P., Farioli, F., Van Assche, E., & van Heuven, W. J. B. (2006). Letter position information and printed word perception: the relative-position priming constraint. *Journal of Experimental Psychology: Human Perception and Performance*, *32*(4), 865–84.
- Grainger, J., & Holcomb, P. J. (2010). Neural constraints on a functional architecture for word recognition. In P. Cornelissen, P. Hansen, M. Kringelbach, & K. Pugh (Eds.), *The neural basis of reading* (pp. 3–32). New York : Oxford University Press.
- Grainger, J., & Jacobs, A. M. (1994). A dual read-out model of word context effects in letter perception: Further investigations of the word superiority effect. *Journal of Experimental Psychology: Human Perception and Performance*, *20*(6), 1158–1176.
- Grainger, J., & Jacobs, A. M. (1996). Orthographic processing in visual word recognition: A multiple read-out model. *Psychological Review*, *103*(3), 518–565.
- Grainger, J., Rey, A., & Dufau, S. (2008). Letter perception: from pixels to pandemonium. *Trends in Cognitive Sciences*, *12*(10), 381–387.
- Grainger, J., & van Heuven, W. (2003). Modeling letter position coding in printed word perception. In P. Bonin (Ed.), *Mental lexicon: “some words to talk about words”*. New York, NY, USA : Nova Science Publishers.
- Jackson, A., & Morton, J. (1984). Facilitation of auditory word recognition. *Memory & Cognition*, *12*(6), 568–574.
- Jacobs, A. M., & Grainger, J. (1992). Testing a semistochastic variant of the interactive activation model in different word recognition experiments. *Journal of Experimental*



- Psychology: Human Perception and Performance*, 18(4), 1174–1188.
- Jacobs, A. M., Rey, A., Ziegler, J. C., & Grainger, J. (1998). MROM-p: An interactive activation, multiple readout model of orthographic and phonological processes in visual word recognition. In J. Grainger & A. M. Jacobs (Eds.), *Localist connectionist approaches to human cognition* (pp. 147–188). L. Erlbaum Associates.
- Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge, UK : Cambridge University Press.
- Johnston, J. C. (1978). A test of the Sophisticated Guessing Theory of word perception. *Cognitive Psychology*, 10(2), 123–153.
- Johnston, J. C., & McClelland, J. L. (1974). Perception of letters in words: seek not and ye shall find. *Science*, 184(4142), 1192–1194.
- Johnston, J. C., & McClelland, J. L. (1980). Experimental tests of a hierarchical model of word identification. *Journal of Verbal Learning and Verbal Behavior*, 19(5), 503–524.
- Jordan, T. R., & Bevan, K. M. (1994). Word superiority over isolated letters: the neglected case of forward masking. *Memory & Cognition*, 22(2), 133–144.
- Jordan, T. R., & Bevan, K. M. (1996). Position-specific masking and the word-letter phenomenon: Reexamining the evidence from the Reicher-Wheeler paradigm. *Journal of Experimental Psychology: Human Perception and Performance*, 22(6), 1416–1433.
- Jordan, T. R., Redwood, M., & Patching, G. R. (2003). Effects of form familiarity on perception of words, pseudowords, and nonwords in the two cerebral hemispheres. *Journal of Cognitive Neuroscience*, 15(4), 537–548.
- Keuleers, E., Lacey, P., Rastle, K., & Brysbaert, M. (2012). The British Lexicon Project: lexical decision data for 28,730 monosyllabic and disyllabic English words. *Behavior Research Methods*, 44(1), 287–304.
- Kinoshita, S., & Norris, D. (2009). Transposed-letter priming of prelexical orthographic representations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(1), 1–18.
- Kinoshita, S., & Norris, D. (2013). Letter order is not coded by open bigrams. *Journal of Memory and Language*, 69, 135–150.
- Kucera, H., & Francis, W. N. (1968). Computational analysis of present-day american english. *American Documentation*, 19(4), 419.
- Laurent, R., Barnaud, M.-L., Schwartz, J.-L., Bessi re, P., & Diard, J. (2017). The complementary roles of auditory and motor information evaluated in a Bayesian perceptuo-motor model of speech perception. *Psychological Review*.
- Lebeltel, O., Bessi re, P., Diard, J., & Mazer, E. (2004). Bayesian robot programming. *Autonomous Robots*, 16(1), 49–79.
- Logan, G. D. (1996). The CODE theory of visual attention: An integration of space-based and object-based attention. *Psychological Review*, 103(4), 603–649.
- Madec, S., Rey, A., Dufau, S., Klein, M., & Grainger, J. (2012). The time course of visual letter perception. *Journal of Cognitive Neuroscience*, 24(7), 1645–1655.
- Marr, D. (1982). *Vision. a computational investigation into the human representation and processing of visual information*. New York, USA : W.H. Freeman and Company.
- Marshall, J. C., & Newcombe, F. (1973). Patterns of paralexia: A psycholinguistic approach. *Journal of Psycholinguistic Research*, 2(3), 175–199.
- Martelli, M., Di Filippo, G., Spinelli, D., & Zoccolotti, P. (2009). Crowding, reading, and developmental dyslexia. *Journal of Vision*, 9(4), 1–18.
- Marzouki, Y., & Grainger, J. (2014). Effects of stimulus duration and inter-letter spacing on letter-in-string identification. *Acta Psychologica*, 148, 49–55.

- Massaro, D. W. (1973). Perception of letters, words, and nonwords. *Journal of Experimental Psychology*, *100*(2), 349–353.
- Massaro, D. W. (1979). Letter information and orthographic context in word perception. *Journal of Experimental Psychology: Human Perception and Performance*, *5*(4), 595–609.
- McClelland, J. L. (2013). Integrating probabilistic models of perception and interactive neural networks: a historical and tutorial review. *Frontiers in Psychology*, *4*, 503.
- McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. an account of basic findings. *Psychological Review*, *88*(5), 375–407.
- McConkie, G. W., Kerr, P. W., Reddix, M. D., Zola, D., & Jacobs, A. M. (1989). Eye movement control during reading: II. frequency of refixating a word. *Perception & Psychophysics*, *46*(3), 245–253.
- Montant, M., Nazir, T. A., & Poncet, M. (1998). Pure alexia and the viewing position effect in printed words. *Cognitive Neuropsychology*, *15*(1-2), 93–140.
- Morton, J. (1969). Interaction of information in word recognition. *Psychological Review*, *76*(2), 165–178.
- Morton, J. (1979). Facilitation in word recognition: Experiments causing change in the logogen model. In P. A. Kollers, M. E. Wrolstad, & H. Bouma (Eds.), *Processing of visible language* (pp. 259–268). Springer.
- Morton, J. (1980). The logogen model and orthographic structure. In U. Frith (Ed.), *Cognitive processes in spelling* (pp. 117–133). Academic Press.
- Moulin-Frier, C., Diard, J., Schwartz, J.-L., & Bessière, P. (2015). COSMO (“Communicating about Objects using Sensory-Motor Operations”): a Bayesian modeling framework for studying speech communication and the emergence of phonological systems. *Journal of Phonetics (special issue “On the cognitive nature of speech sound systems”)*, *53*, 5–41.
- Moulin-Frier, C., Laurent, R., Bessière, P., Schwartz, J.-L., & Diard, J. (2012). Adverse conditions improve distinguishability of auditory, motor and perceptuo-motor theories of speech perception: an exploratory Bayesian modeling study. *Language and Cognitive Processes*, *27*(7–8), 1240–1263.
- Mozer, M. C., & Behrmann, M. (1990). On the interaction of selective attention and lexical knowledge: A connectionist account of neglect dyslexia. *Journal of Cognitive Neuroscience*, *2*(2), 96–123.
- Mueller, S. T., & Weidemann, C. T. (2012). Alphabetic letter identification: Effects of perceivability, similarity, and bias. *Acta Psychologica*, *139*, 19–37.
- Murphy, K. (2002). *Dynamic Bayesian networks: Representation, inference and learning* (Ph. D. thesis). University of California, Berkeley, Berkeley, CA.
- Nazir, T. A. (1991). On the role of refixations in letter strings: The influence of oculomotor factors. *Perception & Psychophysics*, *49*(4), 373–389.
- New, B., Ferrand, L., Pallier, C., & Brysbaert, M. (2006). Re-examining the word length effect in visual word recognition: New evidence from the english lexicon project. *Psychonomic Bulletin and Review*, *13*(1), 45–52.
- Norris, D. (2006). The Bayesian reader: Explaining word recognition as an optimal bayesian decision process. *Psychological Review*, *113*(2), 327–357.
- Norris, D. (2013). Models of visual word recognition. *Trends in Cognitive Sciences*, *17*(10), 517–524.
- Norris, D., & Kinoshita, S. (2008). Perception as evidence accumulation and Bayesian inference: Insights from masked priming. *Journal of Experimental Psychology: General*, *137*(3),

434–455.

- Norris, D., & Kinoshita, S. (2012). Reading through a noisy channel: Why there's nothing special about the perception of orthography. *Psychological Review*, *119*(3), 517–545.
- Norris, D., Kinoshita, S., & Casteren, M. v. (2010). A stimulus sampling theory of letter identity and order. *Journal of Memory and Language*, *62*, 254–271.
- Paap, K. R., Newsome, S. L., McDonald, J. E., & Schvaneveldt, R. W. (1982). An activation–verification model for letter and word recognition: the word-superiority effect. *Psychological Review*, *89*(5), 573–594.
- Patri, J.-F., Diard, J., & Perrier, P. (2016). Modélisation bayésienne de la planification motrice des gestes de parole : Évaluation du rôle des différentes modalités sensorielles. In *Actes de la conférence conjointe jep-taln-recital 2016* (pp. 419–427).
- Patri, J.-F., Perrier, P., & Diard, J. (2016). Bayesian modeling in speech motor control: a principled structure for the integration of various constraints. In *Proc. interspeech 2016* (pp. 3588–3592).
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA, USA : Morgan Kaufmann.
- Pelli, D. G., Farell, B., & Moore, D. C. (2003). The remarkable inefficiency of word recognition. *Nature*, *423*(6941), 752–756.
- Pelli, D. G., Palomares, M., & Majaj, N. J. (2004). Crowding is unlike ordinary masking: Distinguishing feature integration from detection. *Journal of Vision*, *4*, 1136–1169.
- Perea, M., & Lupker, S. J. (2004). Can *CANISO* activate *CASINO*? transposed-letter similarity effects with nonadjacent letter positions. *Journal of Memory and Language*, *51*, 231–246.
- Perea, M., & Pollatsek, A. (1998). The effects of neighborhood frequency in reading and lexical decision. *Journal of Experimental Psychology: Human Perception and Performance*, *24*(3), 767–779.
- Perry, C., Ziegler, J. C., & Zorzi, M. (2007). Nested incremental modeling in the development of computational theories: the CDP+ model of reading aloud. *Psychological Review*, *114*(2), 273–315.
- Phénix, T., Diard, J., & Valdois, S. (2016). Les modèles computationnels de lecture. In M. Sato & S. Pinto (Eds.), *Traité de neurolinguistique* (pp. 167–182). Louvain-la-Neuve, Belgium : De Boeck supérieur.
- Phénix, T., Valdois, S., & Diard, J. (2016). BRAID: A theoretical framework to investigate the role of VA span in reading. In *Proceedings of the british dyslexia association 10th international conference*. Oxford, UK.
- Pollatsek, A., Perea, M., & Binder, K. S. (1999). The effects of “neighborhood size” in reading and lexical decision. *Journal of Experimental Psychology: Human Perception and Performance*, *25*(4), 1142–1158.
- Pylkkänen, L., & Marantz, A. (2003). Tracking the time course of word recognition with MEG. *Trends in Cognitive Sciences*, *7*(5), 187–189.
- Pylkkänen, L., Stringfellow, A., & Marantz, A. (2002). Neuromagnetic evidence for the timing of lexical activation: an MEG component sensitive to phonotactic probability but not to neighborhood density. *Brain & Language*, *81*(1-3), 666–678.
- Rabiner, L. R., & Juang, B.-H. (1993). Fundamentals of speech recognition. In (pp. 321–389). Englewood Cliffs, New Jersey : Prentice Hall.
- Ratcliff, R. (1981). A theory of order relations in perceptual matching. *Psychological Review*, *88*(6), 552–572.
- Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, *124*(3), 372–422.

- Rayner, K., & Reichle, E. D. (2010). Models of the reading process. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(6), 787–799.
- Reicher, G. M. (1969). Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of Experimental Psychology*, 81(2), 275–280.
- Reichle, E. D., Pollatsek, A., Fisher, D. L., & Rayner, K. (1998). Toward a model of eye movement control in reading. *Psychological Review*, 105(1), 125–157.
- Reichle, E. D., Pollatsek, A., & Rayner, K. (2006). E–Z Reader: A cognitive-control, serial-attention model of eye-movement behavior during reading. *Cognitive Systems Research*, 7, 4–22.
- Reichle, E. D., Rayner, K., & Pollatsek, A. (1999). Eye movement control in reading: accounting for initial fixation locations and refixations within the E-Z Reader model. *Vision Research*, 39, 4403–4411.
- Risko, E. F., Stolz, J. A., & Besner, D. (2010). Spatial attention modulates feature crosstalk in visual word processing. *Attention, Perception, & Psychophysics*, 72(4), 989–998.
- Rubenstein, H., Garfield, L., & Millikan, J. A. (1970). Homographic entries in the internal lexicon. *Journal of Verbal Learning and Verbal Behavior*, 9(5), 487–494.
- Rubenstein, H., Lewis, S. S., & Rubenstein, M. A. (1971). Evidence for phonemic recoding in visual word recognition. *Journal of Verbal Learning and Verbal Behavior*, 10(6), 645–657.
- Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. the contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, 89(1), 60–94.
- Schoonbaert, S., & Grainger, J. (2004). Letter position coding in printed word perception: Effects of repeated and transposed letters. *Language and Cognitive Processes*, 19(3), 333–367.
- Sears, C. R., Hino, Y., & Lupker, S. J. (1995). Neighborhood size and neighborhood frequency effects in word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 21(4), 876–900.
- Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, 96(4), 523–568.
- Siakaluk, P. D., Sears, C. R., & Lupker, S. J. (2002). Orthographic neighborhood effects in lexical decision: the effects of nonword orthographic neighborhood size. *Journal of Experimental Psychology: Human Perception and Performance*, 28(3), 661–681.
- Simon, G., Bernard, C., Largy, P., Lalonde, R., & Rebai, M. (2004). Chronometry of visual word recognition during passive and lexical decision tasks: an ERP investigation. *The International Journal of Neuroscience*, 114(11), 1401–1432.
- Spoehr, K. T., & Smith, E. E. (1975). The role of orthographic and phonotactic rules in perceiving letter patterns. *Journal of Experimental Psychology: Human Perception and Performance*, 10(1), 21–34.
- Tarkiainen, A., Helenius, P., Hansen, P. C., Cornelissen, P. L., & Salmelin, R. (1999). Dynamics of letter string perception in the human occipitotemporal cortex. *Brain*, 122, 2119–2131.
- Townsend, J. T. (1971a). Alphabetic confusion: A test of models for individuals. *Perception & Psychophysics*, 9(6), 449–454.
- Townsend, J. T. (1971b). Theoretical analysis of an alphabetic confusion matrix. *Perception & Psychophysics*, 9(1), 40–50.
- van der Heijden, A. H. C., Malhas, M. S. M., & van den Roovart, B. P. (1984). An empirical interletter confusion matrix for continuous-line capitals. *Perception & Psychophysics*, 35(1), 85–88.
- Watson, A. B., & Ahumada, A. J. (2012). Modeling acuity for optotypes varying in complexity.

- Journal of Vision*, 12(10), 19–19.
- Wheeler, D. D. (1970). Processes in word recognition. *Cognitive Psychology*, 1, 59–85.
- Whitney, C. (2001). How the brain encodes the order of letters in a printed word: the SERIOL model and selective literature review. *Psychonomic Bulletin & Review*, 8(2), 221–43.
- Whitney, C., & Berndt, R. S. (1999). A new model of letter string encoding: Simulating right neglect dyslexia. In *Progress in brain research* (pp. 143–163). Elsevier.
- Whitney, C., & Cornelissen, P. (2005). Letter-position encoding and dyslexia. *Journal of Research in Reading*, 28(3), 274–301.
- Whitney, C., & Cornelissen, P. (2008). SERIOL reading. *Language and Cognitive Processes*, 23(1), 143–164.
- Whitney, C., & Lavidor, M. (2005). Facilitative orthographic neighborhood effects: the SERIOL model account. *Cognitive Psychology*, 51(3), 179–213.
- Whitney, D., & Levi, D. M. (2011). Visual crowding: A fundamental limit on conscious perception and object recognition. *Trends in Cognitive Sciences*, 15(4), 160–168.
- Yarkoni, T., Balota, D., & Yap, M. (2008). Moving beyond Coltheart's *N*: A new measure of orthographic similarity. *Psychonomic Bulletin & Review*, 15(5), 971–979.

# Annexe A

## Matériel utilisé dans les simulations présentées

### Simulation A01 : Effet de supériorité des mots

Liste des mots. Matériel issu de Johnston (1978, expérience 2)

- **Position 1** sink, hell, date, damp, bone, buck, dane, cent, foot, file, bake, have, dear, lent, hook, sore, feed, wink, sell, gate, ramp, gone, tuck, pane, dent, loot, wile, wake, wave, wear, sent, rook, tore, weed, ship, hewn, drip, boat, june, beam, deck, crag, fist, fife, bait, here, darn, long, hoof, seem, ford, whip, sewn, grip, goat, tune, team, peck, drag, list, wife, wait, were, warn, song, roof, teem, word
- **Position 2** hall, last, mall, find, dock, mist, lack, mess, bell, show, mist, shot, pill, slat, tall, fall, bend, hill, lost, mill, fond, duck, most, lick, miss, boll, slow, must, slot, poll, spat, toll, fell, bind, dart, cast, mane, nine, mock, live, lace, mend, belt, shed, silk, shop, sift, slur, warm, want, sell, dirt, cost, mine, none, muck, love, lice, mind, bolt, sled, sulk, slop, soft, spur, worm, went, sill
- **Position 3** sale, fake, lame, dice, gale, wane, lone, nose, race, mile, rise, role, male, back, wine, bans, hone, sane, fare, lave, dine, game, ware, lose, note, rare, mine, rite, rope, mare, bask, wise, bats, hose, hilt, poke, name, tack, help, hunt, mash, pact, milk, wish, pile, held, fact, tent, vans, vent, rank, hint, pore, nave, tank, hemp, hurt, math, part, mink, with, pipe, herd, fast, test, vats, vest, rack
- **Position 4** leaf, deal, harm, form, care, bead, seen, clap, slim, heal, bald, boon, lead, ting, seam, wind, bark, leap, dean, harp, fork, cart, bean, seep, claw, slip, hear, bale, boot, leak, tint, sear, wing, barn, gulf, meal, stem, loom, pose, raid, chin, crop, trim, foul, land, when, fold, rang, doom, hand, nook, gulp, mean, step, look, post, rain, chip, crow, trip, four, lane, whet, folk, rant, door, hang, noon

Liste des non-mots. Matériel construit à partir des mots de quatre lettres issus de l'article. La lettre critique est conservée à la même place. Les trois lettres restantes sont mélangées et assignées à un autre mot. Enfin une dernière passe garantit que les non-mots ainsi créés n'appartiennent pas au lexique utilisé (ici le BLP).

- **Position 1** slie, htea, deka, doto, bera, boer, dkoo, capm, fetn, fean, blel, hdee, dneo, lnte, hkni, svea, fcku, wukc, soen, gkae, rkni, getn, toer, pede, dmap, loto, wkoo, wena,

wnet, waev, saer, reil, tlel, weat, skec, hfie, drna, beme, juen, bnew, dema, cofo, fita, fdor, bree, hihp, dirp, lrga, htsi, stao, fogn, wtia, srga, gaot, ghpi, tekc, tofo, pfei, daem, ltsi, weme, wipr, wnue, wran, snig, rrdo, tenw, weer

- **Position 2** tams, badn, taso, cikd, somt, lifl, lahl, kelc, lelp, thsl, lilb, ohsw, sism, llst, tasa, fadn, leml, dikc, lolb, tims, dofn, sutl, toos, cikl, lilp, lolt, wlos, tusa, slsm, lolf, bpdn, tomd, lelh, lilm, namc, lasl, cale, nien, lotb, tird, makc, peos, fest, ehvl, rius, whtn, dimn, elds, tacs, mawr, kesl, eimn, uosr, ridt, eoel, euds, notw, cits, tilb, lols, slpo, sukl, elnn, comk, lpev, morw, fets, dinm,
- **Position 3** kbla, leka, aemw, ohce, mela, meni, idne, iesr, leco, efla, agse, enlo, sela, iwce, sbna, reno, erna, lena, deri, eavm, imne, arme, earw, eosn, elto, aerg, afne, bsta, eaps, oerr, absk, eizr, heto, iesw, utlh, ttke, edmh, peci, iklm, epno, ptsa, tacf, aklr, ensa, aklt, tilh, hacm, vsna, ehnp, vtne, hwni, vsna, eirp, kavt, tenv, nema, rkra, deth, etrt, hinw, utth, atpp, afrt, phse, amsh, epto, thsi, mkci,
- **Position 4** aedf, elal, islm, oobm, iwne, hrad, ntin, eesp, lacm, aegl, frad, ehan, acrd, ebag, blam, eald, orfk, islp, laen, itnp, iwnc, ahrt, oobn, blap, arcw, rabp, eesr, qace, edat, orfk, ahct, easr, abeg, alen, lugf, anrl, rcom, ehwm, irte, nlad, lofn, cihp, maem, onol, fuod, odon, sopd, oolg, nham, tesd, iark, tesp, anrn, lgup, odok, sopt, olon, olfp, alnw, tirp, hewr, onoe, riat, oufk, aemt, hnar, cihg, rocn

## Simulation A02 : Effet de supériorité des pseudo-mots

### Liste des mots

**Position 1** dark, park, date, late, deep, keep, four, your, look, took, lord, word, much, such, test, rest, were, here, what, that

**Position 2** came, come, fall, full, firm, form, gave, give, held, hold, list, lost, love, live, mass, miss, most, must, post, past

**Position 3** feet, felt, fine, fire, hand, hard, home, hope, knew, know, life, like, made, make, more, move, race, rate, then, than

**Position 4** band, bank, cool, cook, hear, heat, mail, main, mark, mare, mile, milk, pain, pair, pool, poor, rich, rice, wall, walk

### Liste des pseudo-mots

**Position 1** darl, parl, datu, latu, deet, keet, fous, yous, loob, toob, lort, wort, muct, suct, tesk, resk, wera, hera, whab, thab

**Position 2** camo, como, falk, fulk, firp, forp, gava, giva, helk, holk, lisk, losk, lovi, livi, masp, misp, mosp, musp, posr, pasr

**Position 3** chen, chan, fome, fope, hade, hake, heet, helt, jine, kire, mand, mard, nore, nove, pnw, pnw, sife, sike, tace, tate

**Position 4** cand, cank, cear, ceat, dail, dain, dall, dalk, dile, dilk, jain, jair, sich, sice, sool, sook, nol, noor, tark, tary

### Liste des non-mots

**Position 1** crea, trea, dmea, kmea, efte, lfte, ekwn, okwn, etnh, atnh, flei, klei, mheo, pheo, nfei, rfei, nhda, rhda, rmeo, vmeo

**Position 2** awth, atth, cmhu, cshu, edpe, ekpe, olko, otko, rdka, rpka, rldo, rwdo, rwee, rhee, stte, srte, tdea, tlea, ufro, uyro

**Position 3** ahre, ahte, crhi, crei, ilma, ilna, ipna, ipra, lmei, lmki, lwla, lwka, nbda, nbka, oclo, ocko, oplo, opro, rmka, rmya

**Position 4** lfla, lflu, lhde, lhdo, mcea, mceo, rfmi, rfmo, slti, slto, smsa, smsi, smto, smtu, spto, spta, vgea, vgei, vleo, vlei

## Simulation A03 : Effet de supériorité des pseudo-mots

**Liste des mots** (ils sont à la base de la construction des autres séries)

**Position 1** dark, park, lord, word, much, such, test, rest, what, that, fast, cast, fans, mans, bogs, dogs, made, sade, rock, lock

**Position 2** fall, full, firm, form, hell, hill, list, lost, mass, miss, most, must, post, past, lick, lock, bets, bits, find, fund

**Position 3** then, than, mark, mask, prim, prom, moms, mops, gags, gaps, muse, mute, laps, labs, risk, rink, want, wast, chap, chip

**Position 4** band, bank, mild, milk, rich, rick, wall, walk, chat, chap, drag, draw, shot, shop, grip, grid, pack, pact, find, fink

**Liste des pseudo-mots**

**Position 1** darl, parl, lort, wort, muct, suct, tesk, resk, whab, thab, fasl, casl, fanc, manc, bogr, dogr, madr, sadr, roct, loct

**Position 2** falk, fulk, firp, forp, helk, hilk, lisk, losk, masp, misp, mosp, musp, posr, pasr, lich, locb, betw, bitw, finl, funl

**Position 3** chen, chan, fark, fask, pril, prol, poms, pops, rags, raps, nuse, nute, faps, fabs, tisk, tink, kant, kast, thap, thip

**Position 4** fand, fank, vild, vilk, fich, fick, nall, nalk, drat, drap, frag, fraw, swot, swop, drip, drid, vack, sact, rind, rin

**Liste des non-mots** (consonnes)

**Position 1** dqrk, psrk, lfrd, wgrd, mjch, skch, tlst, rmst, whct, thct, fvst, cfst, fjns, mlns, bvgs, dcgs, mkde, shde, rbck, lck

**Position 2** fpll, fkl, fxrm, fqrm, hwll, hbll, lmst, lqst, mdss, mfss, mhst, mjst, pdst, pgst, ltck, ldck, bwts, bvts, ftnd, fgnd

**Position 3** thvn, then, mxrk, mds, prt, prjm, mdms, mgps, ghgs, gjps, mkse, mlte, lrps, ltbs, resk, rbnk, wpnt, wlst, chzp, chfp

**Position 4** bqnd, bsnk, mdld, mvlk, rfch, rgck, wll, wnlk, chjt, chkp, drhg, drlw, shmt, shpp, grtp, grrd, pqck, pxct, fdnd, fbnk

**Liste des non-mots** (QJX)

**Position 1** djqx, pjqx, lqxj, wjqx, mxjq, sqxj, tqjx, rqjx, wqjx, tjxq, fjqx, cqjx, fjxq, mxqj, bqxj, dxqj, mxjq, sjxq, rqjx, ljxq

**Position 2** jaqx, xujq, jiqx, qojx, xejq, xiqj, jixq, qoxj, jaxq, jixq, xoqj, juqx, qojx, xajq, jiqx, xojq, qexj, jixq, xijq, xuqj



**Position 3** jqex, jqax, jqrX, qjsx, qxij, xjoq, xjmq, jqpx, xjgq, xqpj, xqsj, xqtj, qxpj, xjbq, jqsx, qjnx, jqnx, jqsx, jxaq, xjiq

**Position 4** qjxd, qxjk, jxqd, xqjk, xjqh, xqjk, jxql, jqxk, xjqt, jqxp, jxqg, xjqw, qjxt, jqxp, jxqp, qxjd, jqxk, xqjt, jqxd, jqxk

## Simulation B01 : Effet du statut lexical

### Liste Mots non homographes concrets

**Fréquence 10** pond, lung, disc, pest, fern, tepid, moth, merge, ramp, sewer, filth, chef, broom, shawl, shrub, denim, foyer, twig, turf, cove

**Fréquence 100** cliff, lamp, barn, flesh, bird, grass, clay, rice, ocean, roof, tray, song, movie, wagon, cigar, gift, bacon, gang, wine, wrist

**Fréquence 1000** chair, city, girl, food, dark, thing, black, green, child, wife, lady, brown, large, small, door, white, money, town, woman, word

### Liste Mots non homographes abstraits

**Fréquence 10** risky, myth, mirth, trite, pact, veto, glee, defy, elite, farce, cult, whim, smug, trait, zeal, shirk, greed, lowly, bias, wary

**Fréquence 100** loss, pure, task, legal, lend, fury, ugly, debt, magic, grief, hint, occur, event, skill, solve, lazy, fate, quit

**Fréquence 1000** also, happy, year, alone, soon, less, today, fact, until, same, full, often, idea, along, quite, half, next, able, above

### Liste Mots homographes concrets

**Fréquence 10** dice, fuse, grill, crane, rake, stain, sling, tart, helm, snuff, mesh, stud, crumb, gorge

**Fréquence 100** lace, rock, sheet, chest, belt, pipe, shed, duck, organ, blow

**Fréquence 1000** light, place, water, table

### Liste Mots homographes abstraits/concrets

**Fréquence 10** peck, flap, raft, prone, pawn, pluck

**Fréquence 100** trace, yard, dirt, charm, nerve, block, dumb, star, alert, grave

**Fréquence 1000** post, late, power, feel, still, party, kind, mind, care, miss, head, mean, fire, case, left, hard

### Liste Mots homographes abstraits/concrets

pird, lant, dest, cest, parn, sepod, dath, farge, samp, lewer, pirth, crif, croom, dwarl, shrip, derom, fower, twac, cirf, nive, cruff, lany, bist, cresh, furd, fress, culy, lude, olian, seef, kaly, dong, polie, wadon, bogar, duft, bacud, gung, wose, wruct, choor, doty, verl, koot, dast, thung, plock, greni, field, wode, laty, chown, berge, stull, pror, whoss, murey, tron, weban, wost, adro, hally, wela, alite, soun, hust, tupay, frut, undel, sude, falt, odren, isaa, atung, quode, harf, nist, uple, edove

## Simulation B02 : Effet de légalité des non-mots

### Liste Non-mots illégaux non-prononçables

fuzg, glazb, tufk, cresf, graft, blacf, crudf, spokf, crepf, rudk, trucp, cravb, topk, railg, thonb, grovt, crabg, ratn, bravg, ponv, framk, winp , lamg

### Liste Non-mots illégaux prononçables

tritv, tubw, spitj, stuml, likj, flipb, rakv, latv, codg, stagv, safv, crazj, sagm, brakv, ghosj, crafj, stakg, stagb, crepw, grunw, scolor, flamr, spitk, skatb, tastd, solw, bonm, lamw, crusv, snakb

### Liste Non-mots légaux

drilk, clefe, melp, delm, losp, disp, drep, tors, trife, fronk, flet, pronk, blesp, drifs, bleg, nump, flan, jund, prot, drmlt, frilt, nosk, herv, lurt, marp, prind, teart, wesp, plam, flact, turb, mife, trene, plact, drawk, snact, flamt, smalt, sleem, prip, clert, lask, frese, clote, trumb, prind, threp, blean, moisp, stron, stune, firse, greel, stad, sland, jume, staim, sneap, crusp, jusk, norte, shire, slint, framp, hent, clipe, breep, fank, leask, saft, maind, treap, moune, hosk, grast, barp, rolt, strig, flink, quarp, blask, flad, glant, shart, plag, crad, blent, pask, clain, treap, strit, quib, tids, riot, preed, breal, plob, thirt, fure, grafe, gleap, quelt, crafe, hort, clup, scorp, Brust, sping, blask, hoist, charp, flast, blood, dreat, dend, tast, courp, samt, scarm, flane, clask, gloon, breat, curf, flast, sult, shelt, carm, learn, grafe, sweak, flirk, brait, vauld, weirk

## Simulation B03 et B04 : Effets de fréquence des mots

alone, onion, repel, veils, dales, papas, galas, snook, visas, sisal codex, bikes, mocha, sassy, bulge, swear, plugs, shyly, drily, belts guess, raped, vicar, snogs, llama, beads, brook, fugue, slept, molar snoop, erase, steel, bijou, squab, taxis, paint, wired, thorn, divan frizz, crank, moans, ropes, lingo, pylon, aides, wards, shops, cream muzak, wound, fared, opine, shore, rolfy, power, splay, sweat, temby eaves, grubs, foggy, japan, naira, agree, wrest, noise, herod, dealt price, minor, sonny, bills, maser, heels, rebut, foxes, semen, sends cabby, begot, venue, aloft, match, cacti, wheel, hecks, agues, poofy pates, delta, toils, flaps, limbo, kazoo, scree, gummy, sinks, stagy climb, these, slime, faith, shuck, waltz, boles, acorn, mural, tidal treks, lochs, guava, throb, wager, sales, amply, hoses, mange, avert, balms, reams, cushy, rusks, skirl, leash, trier, cheat, mayst, refer



# Annexe B

## Glossaire : définition des termes utilisés

Ce glossaire est inspiré du glossaire de Ferrand (2007), et rappelle les définitions usuelles du domaine de la reconnaissance visuelle de mots.

**AMORÇAGE** Technique expérimentale dans laquelle on étudie l'effet sur le traitement d'un mot cible de la présentation préalable d'une amorce. L'amorce est en général constitué en manipulant les lettres du mot, par exemple par transposition de deux lettres adjacentes ou non.

**BIGRAMME OUVERT** Paire de lettres ordonnées qui ne sont pas nécessairement contiguës. Par exemple pour *LIVRE*, *LI*, *IR*, etc.

**DÉCISION LEXICALE** Tâche dans laquelle les participants doivent décider si le stimulus visuel qui leur est présenté est un mot de leur lexique ou pas. Le plus souvent, ils doivent répondre le plus rapidement possible et le plus précisément possible

**DENSITÉ DE VOISINAGE** C'est le nombre de voisins orthographiques que possède un mot.

**CONCRÉTUDE** Les mots concrets sont les mots référant à des personnes, des substances, des lieux, et toute chose pouvant être perçue.

**L'EFFET DE FRÉQUENCE DES MOTS** C'est le fait que les mots fréquemment sont reconnus plus rapidement que les autres. On le trouve aussi bien en décision lexicale qu'en reconnaissance visuelle de mots.

**EFFET DE SUPÉRIORITÉ DES MOTS** C'est le fait qu'une lettre est identifiée plus précisément lorsqu'elle est présentée dans un mot (on parle de contexte *mot*), que lorsqu'elle est présentée isolément (contexte *lettre isolée*) ou entourée de lettres quelconques (contexte *non-mot*). La lettre à identifier est appelée la *lettre cible*.

**EFFET DE SUPÉRIORITÉ DES PSEUDO-MOTS** C'est le fait qu'une lettre est identifiée plus précisément lorsqu'elle est présentée dans une séquence prononçable de lettres (contexte *pseudo-mot*) que lorsqu'elle est présentée isolément (contexte *lettre isolée*) ou entourée de lettres quelconques (contexte *non-mot*).

**FRÉQUENCE D'OCCURRENCE** La fréquence d'occurrence d'un mot est le nombre de fois où ce mot apparaît dans un corpus donné de traces écrites. Ces traces peuvent être issues de livres comme pour le lexique *English Lexicon Project* (Balota et al., 2007), ou le *British Lexicon Project* (Keuleers et al., 2012), de sous-titres de films comme pour le lexique

**SUBLEX** (Brysbaert & New, 2009). Ces fréquences permettent d'estimer le nombre de fois où un lecteur a rencontré ce mot.

**GRAPHÈME** Représentation écrite d'un phonème. Les graphèmes de la langue française sont constitués le plus souvent de plusieurs lettres : ou, eau, ph, aient, etc.

**HOMOGRAPHE** Deux mots sont homographes lorsqu'ils s'écrivent de la même manière, mais n'ont pas la même signification. Exemple : fils (rouges) et fils (préféré).

**NON-MOT** C'est une chaîne de caractères qui, à la différence des pseudo-mots, n'est pas prononçable. Ils sont construits, par exemple, en remplaçant les voyelles d'un mot par des consonnes ou en inversant l'ordre des lettres.

**PSEUDO-MOT** C'est une chaîne de caractères qui est orthographiquement légale et prononçable mais qui ne correspond à aucune représentation lexicale du lecteur. Ils sont construits, par exemple, en changeant une lettre à un mot du lexique. Le mot *MAISON* devient le pseudo-mot *MAIRON*.

**RECONNAISSANCE VISUELLE DE MOT** Activité perceptive et cognitive conduisant à la reconnaissance consciente de mots écrits.

**TACHE 2AFC** C'est une tâche forcée à deux alternatives. Après la brève présentation du stimulus, ce dernier est masqué et deux propositions de réponses sont proposées. Dans une tâche de perceptibilité des lettres, les alternatives sont des lettres et elles sont positionnées le plus souvent de part et d'autres de la lettre à identifier. Dans une tâche de reconnaissance visuelle de mots, les alternatives sont des mots voisins.

**TEMPS DE RÉACTION** Mesure du temps que le sujet met pour répondre dans une tâche de psycholinguistique.

**TRIGRAMME OUVERT** Triplet de lettres ordonnées qui ne sont pas nécessairement contiguës. Par exemple pour *LIVRE*, *LIR*, *IVR*, etc.

**VOISINAGE ORTHOGRAPHIQUE** Dans la littérature, on trouve deux définitions du voisinage orthographique d'un mot. La première est la définition de Coltheart et al. (1977). Elle définit le voisinage orthographique d'un mot donné comme étant l'ensemble des mots du lexique qui ne diffèrent du mot considéré que d'une lettre, et ont la même longueur. La seconde définition (Yarkoni, Balota, & Yap, 2008), appelée OLD20, est basée sur la distance de Levenstein. Elle considère comme voisins d'un mot tous les mots qui peuvent être générés par suppression ou insertion d'une lettre, par inversion de deux lettres ou encore par substitution d'une lettre par une autre (cas précédent). Cette mesure est plus flexible et semble donner de meilleurs résultats pour la simulations d'effets en décision lexicale (Norris, 2013). Dans *BRAID*, nos choix d'implémentation nous limitent à l'usage de la première définition.



---

## **Titre — Modélisation bayésienne algorithmique de la reconnaissance visuelle de mots et de l’attention visuelle**

**Résumé** — Dans cette thèse, nous proposons un nouveau modèle conceptuel de la reconnaissance visuelle de mots implémenté sous forme mathématique de façon à évaluer sa capacité à reproduire les observations expérimentales du domaine. Une revue critique des modèles computationnels existants nous conduit à définir un cahier des charges sous la forme de cinq hypothèses qui sont à la base du modèle conceptuel proposé : le modèle est doté d’une architecture à trois niveaux (sensoriel, perceptif et lexical) ; le traitement est parallèle sur l’ensemble des lettres du stimulus ; l’encodage positionnel est distribué ; enfin, le traitement sensoriel des lettres intègre position du regard, acuité visuelle et distribution de l’attention visuelle. L’implémentation repose sur la méthodologie de la modélisation bayésienne algorithmique, conduisant au modèle *BRAID* (pour « *Bayesian word Recognition with Attention, Interference and Dynamics* »).

Nous vérifions la capacité du modèle à rendre compte des données obtenues en perceptibilité des lettres (par exemple, effets de supériorité des mots et des pseudo-mots, effets de contexte), en reconnaissance de mots et en décision lexicale (par exemple, effets de fréquence et de voisinage). Au total, nous simulons avec succès 28 expériences comportementales, permettant de rendre compte d’effets subtils dans chacun des domaines ciblés. Nous discutons les choix théoriques du modèle à la lumière de ces résultats expérimentaux, et proposons des perspectives d’extension du modèle, soulignant la flexibilité du formalisme choisi.

**Mots clés** — Modèle computationnel, modélisation bayésienne, reconnaissance visuelle de mots, décision lexicale, attention visuelle.

---

---

## **Titre — Bayesian algorithmic modeling of visual word recognition and of visual attention**

**Abstract** — In this thesis, we propose an original theoretical framework of visual word recognition, and implement it mathematically to evaluate its ability to reproduce experimental observations of the field. A critical review of previous computational models leads us to define specifications in the form of a set of five hypotheses, that form the basis of the proposed theoretical framework: the model is built on a three-layer architecture (sensory, perceptual, lexical); letter processing is parallel; positional coding is distributed; finally, sensory processing involves gaze position, visual acuity, and visual attention distribution. To implement the model, we rely on the Bayesian algorithmic modeling methodology, and define the *BRAID* model (for “*Bayesian word Recognition with Attention, Interference and Dynamics*”).

We verify the model’s ability to account for main experimental observations of studies of letter perceptibility (for instance, the word and pseudo-word superiority effects, the context effects), and of visual word recognition and lexical decision (for instance, the frequency effects and neighborhood effects). Overall, we successfully simulate 28 behavioral experiments, accounting for subtle effects in each domain. We discuss the theoretical hypotheses underlying the model in light of our experimental results, and propose various possible extensions of the model, highlighting the flexibility of the chosen framework.

**Keywords** — Computational modeling, Bayesian modeling, visual word recognition, lexical decision, visual attention.

---