# Réseaux de neurones profonds appliqués à la compréhension de la parole

Edwin Simonnet

# THÈSE DE DOCTORAT DE

LE MANS UNIVERSITÉ
**C**OMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N°601
*Mathèmatique et Sciences et Technologies de l'Information et de la Communication*
Spécialité : *Informatique*

Par


## « Edwin SIMONNET »


## « Réseaux de neurones profonds appliqués à la compréhension de la parole »


**Thèse présentée et soutenue à LE MANS UNIVERSITÉ, LIUM , le 12 février 2019**
**Unité de recherche : Laboratoire d'Informatique de l'Université du Mans (LIUM)**
**Thèse N° : 2019LEMA1006**


**Rapporteurs avant soutenance :**

Mme Sophie ROSSET, Directrice de recherche, LIMSI, CNRS, Université Paris-Sud

M. Frédéric BÉCHET, Professeur, LIS, CNRS, Université Aix-Marseille

**Composition du jury :**

Examinateurs : M. Marco DINARELLI, Chargé de recherche, LATTICE, CNRS, Université Sorbonne-Nouvelle
M. Bassam JABAIAN, Maître de conférences, CERI/LIA, Université d'Avignon

Dir. de thèse : M. Yannick ESTÈVE, Professeur, LIUM, Le Mans Université

Co-dir. de thèse : Mme Nathalie CAMELIN, Maître de conférence, LIUM, Le Mans Université

Invité(s)

M. Paul Deléglise, Professeur Émérite, LIUM, Le Mans Université

*À ma maman, qui s'est entièrement consacrée à ses enfants.*

*À ma bien-aimée, qui a traversé cette aventure à mes cotés.*

# Remerciements

Je tiens à remercier les membres de mon jury de thèse pour le temps qu'ils auront consacré à l'examen de mon manuscrit.

Je remercie mes encadrants de thèse : Yannick Estève, Nathalie Camelin et Paul Deléglise pour leur soutien, leurs conseils et leur direction précieuse au cours de cette thèse.

Je tiens également à remercier Renato De Mori et Sahar Ghannay avec qui j'ai eu la chance et le plaisir de collaborer dans mes travaux tout au long de ma thèse.

Je tiens enfin à remercier tous les membres du Laboratoire d'Informatique de l'Université du Mans pour leurs collaborations bénéfiques, leur accueil chaleureux et leur convivialité.

Pour finir, je remercie ma famille, ma compagne, ma belle-famille et mes amis pour tout le soutien et tous les conseils qu'ils auront pu m'apporter au cours de cette thèse, dans ses bon moments comme dans ses moments difficiles.

Nous remercions la région des Pays de la Loire pour son financement.


Nous remercions l'agence ANR pour son financement à travers *CHIST-ERA ERA-Net JOKER* sous le numéro de contrat *ANR-13-CHR2-0003-05*.

# TABLE DES MATIÈRES

# LISTE DES TABLEAUX

# TABLE DES FIGURES

# INTRODUCTION

La compréhension automatique de la parole a connu un vif intérêt au cours de ces dernières années dans le domaine de la recherche comme dans celui de l'industrie. La recherche en compréhension de la parole est centrale pour le développement des communications humain-machine comme dans les centres d'appels, les services téléphoniques, les robots compagnons ou encore les assistants personnels intégrés dans les smartphones mais aussi dans les maisons, les voitures, *etc.*

La compréhension de la parole peut être définie de différentes façons en aboutissant à des tâches différentes (résumé de discours, identification de thème, détection d'intention, extraction de concepts sémantiques ou d'entités nommées, *etc.*) On peut néanmoins la définir d'une manière générale comme l'extraction et la représentation automatique du *sens* contenu dans les mots d'une phrase parlée [De Mori *et al.*, 2008]. Elle peut dans de nombreux cas être résolue avec des méthodes d'apprentissage automatique supervisé. La compréhension de la parole peut être considérée comme une tâche de classification sémantique ou de segmentation et d'étiquetage en concept sémantique. La tâche d'étiquetage en concepts sémantiques est utilisée dans le cadre d'un système de dialogue oral. Les champs aléatoires conditionnels (CRF) sont réputés pour obtenir les meilleurs résultats sur cette tâche [Hahn *et al.*, 2011]. En 2015, les modèles neuronaux commencent d'y être mesurés [Mesnil *et al.*, 2015].

Les modèles neuronaux ont été envisagés et perfectionnés depuis les années 1950 [Rosenblatt, 1957]. Depuis une dizaine d'années, les modèles neuronaux prennent l'ascendant dans de nombreuses tâches de traitement du langage naturel grâce à des avancées algorithmiques et à la mise à disposition d'outils de calcul puissants comme les processeurs graphiques permettant alors d'exploiter tout leur potentiel. Ils connaissent aujourd'hui un grand succès dans de nombreux domaines comme la modélisation acoustique [Hinton *et al.*, 2012], la modélisation du langage [Mikolov *et al.*, 2011], la traduction automatique [Bahdanau *et al.*, 2014] et la compréhension de la parole avec les résultats prometteurs apportés par [Mesnil *et al.*, 2015]. Cette thèse, débutée en 2015, se place dans la cadre applicatif de la compréhension de la parole tout en s'inscrivant dans l'émergence de l'appren-

tissage profond : elle a pour but d'explorer le potentiel des réseaux de neurones artificiels dans un système de dialogue oral.

De nombreux obstacles rendent la tâche de compréhension de la parole complexe. L'un des principaux obstacles est l'interprétation difficile des transcriptions automatiques de la parole. En effet, le module de compréhension de la parole s'appuie sur un module de reconnaissance automatique de la parole en amont qui est lui même sujet aux erreurs. Ces erreurs de reconnaissances se répercutent alors sur le processus de compréhension. Elles peuvent être dues à la qualité du discours ou à des problèmes inhérents à la communication orale spontanée comme le bruit environnant ou les variabilités liées aux différents types de locuteurs.

D'autres obstacles à la compréhension de la parole sont dus au sens de la phrase. Cela peut être lié à des phrases agrammaticales ou encore à des disfluences verbales telles que les hésitations, répétitions et reprises encore une fois à cause d'une communication orale spontanée. Il arrive aussi qu'une phrase puisse avoir plusieurs sens possibles. La désambiguïsation se fera alors grâce au contexte.

Ces effets d'erreurs et d'imprécisions peuvent être en partie contrés grâce à des modèles numériques d'interprétation probabiliste comme les CRF ou les réseaux de neurones. De nouveaux perfectionnements doivent être recherchés afin de limiter encore l'impact des erreurs des transcriptions automatiques.

## Organisation du document

Ce document est structuré en deux grandes parties de quatre chapitres chacune. Dans un premier temps, l'état de l'art présente la compréhension et la reconnaissance de la parole puis les méthodes d'apprentissage automatique supervisé pouvant effectuer cette tâche. Nous commençons par des systèmes classiques pour finir avec des techniques d'apprentissage profond. Dans un second temps, les contributions présentent le travail réalisé durant cette thèse, à savoir l'utilisation du mécanisme d'attention dans le contexte de la compréhension de la parole, la gestion des erreurs de transcriptions automatiques et enfin des recherches sur l'optimisation des performances du système de compréhension en proposant une classification hiérarchique des étiquettes sémantiques.

Dans le premier chapitre, la compréhension de la parole est décrite à travers les différentes définitions qui peuvent y être associées et les différentes tâches qui peuvent en découler. Un inventaire de différents corpus de compréhension est présenté dans un deuxième temps.

Dans le deuxième chapitre, nous définissons la reconnaissance de la parole dont dépend directement la compréhension de la parole. Nous introduisons également des mesures de confiance de reconnaissance de la parole qui sont utilisées dans cette thèse.

Dans le troisième chapitre, nous abordons certains des modèles traditionnels utilisés avec succès avant l'essor des méthodes neuronales pour le traitement automatique du langage naturel. Dans un premier temps, nous apportons une présentation

théorique de ces modèles. Dans un second temps, nous étudions leur application concrète avec une présentation de résultats obtenus en compréhension de la parole.

Dans le quatrième chapitre, nous présentons les modèles fondés sur les réseaux de neurones artificiels faisant l'objet principal de notre étude. Parmi ces domaines, figure également la compréhension de la parole avec l'étude de Grégoire Mesnil [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015], des travaux que nous avons choisis comme base d'étude préliminaire. Ce chapitre s'organise de la même façon que le précédent en apportant une présentation théorique des modèles neuronaux dans un premier temps et en étudiant ensuite leur application concrète avec une présentation de résultats obtenus en compréhension de la parole.

Dans le cinquième chapitre, et premier chapitre des contributions, nous étudions l'apport du mécanisme d'attention à la compréhension de la parole. Cette technologie employée sur les modèles neuronaux constitue le point de départ de cette thèse en 2015. Le mécanisme d'attention est utilisé avec succès dans des domaines tels que la traduction automatique et nous présentons dans ce chapitre les motivations qui nous poussent à vouloir l'utiliser de façon innovante pour la compréhension de la parole à un moment où les modèles neuronaux sont relativement peu représentés dans ce domaine. La première section de ce chapitre définit les motivations liées à notre approche. La seconde section aborde l'obtention d'un réseau de neurones récurrent bidirectionnel à l'état de l'art inspiré de celui de l'étude de Grégoire Mesnil sur le corpus ATIS. Une fois ce système acquis, nous souhaitons l'appliquer au corpus MEDIA qui nous intéresse plus particulièrement. Nous parlerons également du système CRF de référence que nous utilisons au cours de cette thèse afin de disposer d'une base de comparaison à l'état de l'art. La dernière section décrit les premières contributions de cette thèse, avec l'apport du mécanisme d'attention à la compréhension de la parole. Nous y abordons également l'apport des descripteurs de mots dans notre système avec mécanisme d'attention. Dans ce chapitre, nous travaillons sur des sorties de transcriptions manuelles afin de nous placer dans un cadre de compréhension théorique pure dans un premier temps : nous dissocions les erreurs liées à la reconnaissance de la parole de celles directement liées à notre système de compréhension.

Dans le sixième chapitre, nous étudions l'impact des erreurs de compréhension liées aux erreurs du module de reconnaissance automatique en amont du module de compréhension. Il est en effet nécessaire de travailler sur des sorties de transcriptions automatiques afin de se placer dans le cadre pratique de l'application visée, soit un système de dialogue oral. Ainsi, ce chapitre étudie l'impact entraîné sur nos performances par la transition de transcriptions manuelles à automatiques dans nos systèmes de compréhension. Les erreurs de reconnaissance, inévitables lors du traitement de transcriptions automatiques, sont un problème majeur en compréhension de la parole. Cela nous impose de chercher des stratégies pour diminuer cet impact défavorable sur nos performances. Nous proposons d'explorer les pistes suivantes pour y parvenir : l'utilisation de mesures de confiance de reconnaissance, une détection d'erreur de reconnaissance au cours du processus de compréhension de la parole, la manipulation et l'enrichissement du corpus d'apprentissage de compréhension et

enfin la simulation d'erreurs de reconnaissance.

Dans le septième chapitre, nous abordons une désambiguïsation de la tâche de compréhension passant par une classification hiérarchique des étiquettes sémantiques. Nous cherchons à réduire la confusion entre les étiquettes sémantiques du corpus MEDIA en réduisant notre jeu d'étiquettes original. Des définitions d'ensembles de méta-étiquettes moins fines sont proposées dans le but d'obtenir des systèmes de compréhension plus performants à un niveau sémantique plus général. Nous présentons ensuite une nouvelle méthode pour améliorer les résultats de notre modèle neuronal standard en intégrant les sorties des différents systèmes de compréhension enrichis qui découlent de l'usage des méta-étiquettes. Plusieurs techniques sont explorées pour tirer parti de cette information généralisée, comme son intégration dans les descripteurs de mots ou le vote multi-système.

Enfin, en conclusion, un résumé des points clés de cette thèse est présenté ainsi que quelques perspectives pour de futurs travaux de recherche.

# Première partie

# Contexte de travail et état de l'art

# CHAPITRE 1

# LA COMPRÉHENSION DE LA PAROLE

## Sommaire

Avec le développement des communications humain-machine (centres d'appels, services téléphoniques, robots compagnons, assistants personnels dans les smartphones, maisons ou voitures, *etc.*), la compréhension automatique de la parole a connu un intérêt croissant au cours de ces dernières années, dans la recherche comme dans le commerce.

Dans un premier temps, nous décrivons la compréhension de la parole à travers les différentes définitions qui peuvent y être associées et les différentes tâches qui peuvent en découler. Un inventaire de différents corpus de compréhension est abordé dans un deuxième temps.

## 1.1 Qu'est-ce-que la compréhension de la parole ?

### 1.1.1 Définition

La compréhension automatique de la parole (*Spoken Language Understanding* - SLU) est une tâche informatique complexe qui peut être appréhendée de plusieurs façons.

Le but est de comprendre un utilisateur, *i.e.* d'extraire le sens à partir du langage naturel [Tur et De Mori, 2011]. Cela peut revenir par exemple à identifier dans l'énoncé de l'utilisateur des mots/phrases clés [Gupta *et al.*, 2006].

Une définition simple et globale de la compréhension peut être *"l'interprétation des signaux transportés par un signal de parole"* ([De Mori, 2007, De Mori *et al.*, 2008]). Cette vision de la compréhension est habituellement assimilée à l'extraction et à la représentation automatique du *sens* contenu dans les mots d'une phrase énoncée à l'oral.

Afin qu'une machine puisse comprendre et interagir avec un utilisateur, une première problématique concerne la définition de ce que l'on entend par le *sens* ou la *sémantique d'un discours*.

La sémantique concerne l'organisation de la signification ainsi que les relations entre les signes sensoriels, ou symboles, et ce qu'ils signifient [Woods, 1975]. L'interprétation sémantique informatique consiste en une conceptualisation du monde en utilisant des processus informatiques pour créer une structure représentant le sens, à partir de différents signaux/symboles, et de leurs caractéristiques, présents dans des mots ou des phrases [De Mori *et al.*, 2008]. En pratique, le monde, ou l'ensemble de ce qui est à comprendre, est limité par la tâche à traiter. Ainsi, cela dépend généralement de l'application, donc de la tâche visée avec l'utilisateur.

Dans [Camelin, 2007], il est décrit que la compréhension consiste à donner un sens à l'information lexicale extraite du signal sonore pour qu'elle soit traitable par la machine, *i.e.* extraire le sens utile (ou la demande) véhiculée par le locuteur. Il convient donc de choisir une *représentation sémantique*. Une représentation sémantique correspond à la définition sémantique de l'information nécessaire à la machine pour la réalisation d'une tâche : c'est le formalisme utilisé par la machine pour stocker le sens extrait dans le discours. La représentation sémantique choisie doit être adaptée à l'application/aux données à analyser : chaque système adopte une

représentation qui lui est propre car il n'existe pas de représentation sémantique générique qui puisse répondre aux besoins de toutes les applications possibles.

Il est aussi mentionné dans [Béchet, 2007] que la représentation du sens peut être envisagée selon de nombreuses perspectives : philosophique, linguistique, cognitive, *etc.* Ainsi plutôt que de choisir une théorie sémantique, on préfère l'aborder sous l'angle pragmatique de son utilisation dans un cadre applicatif.

### 1.1.2 Différentes tâches liées à la compréhension de la parole

La compréhension de la parole concerne plusieurs tâches consistant chacune à extraire des informations (du sens) véhiculées par la langue. Les différents systèmes de compréhension qui réalisent cela ont pour but commun l'extraction d'informations sémantiques dans un discours. Cependant, une représentation générique pour une compréhension ouverte est un objectif très complexe à atteindre notamment car celle-ci peut s'effectuer à différents niveaux : mot, phrase, document, *etc.* Ainsi, la représentation sémantique se définit de manière *ad hoc* par la tâche à réaliser dans le cadre applicatif. Cette section décrit plusieurs de ces tâches.

#### 1.1.2.1 Résumé de discours

La quantité croissante de documents et d'enregistrements audios ont attiré les intérêts de la recherche dans le domaine du résumé de discours automatique. Dans ce contexte, la compréhension vise à générer un résumé à partir d'un document. Cela permet de réduire l'effort humain dans l'accès ou le tri de l'information documentaire [Tur et De Mori, 2011]. Le système doit déterminer l'importance des phrases et extraire les plus importantes pour les inclure dans un résumé. D'autres tâches sont étudiées dans ce domaine comme le résumé concentré sur les requêtes : à partir d'une requête (ou question) de l'utilisateur, le système doit construire un résumé y répondant à partir d'un ensemble de documents [Maskey, 2008]. Il y a encore le résumé abstractif où le système ne peut pas simplement *copier et coller* pour créer un résumé mais doit reformuler les phrases dans un souci de cohérence [Kleinbauer *et al.*, 2007, Murray *et al.*, 2010].

#### 1.1.2.2 Segmentation et identification de thème

La tâche de segmentation en thème vise à diviser un long document (audio ou transcrit) en plus courts segments de thème homogène. Cette tâche de compréhension se place au niveau du document dans sa globalité et vise à simplifier la recherche documentaire. Mêlée au résumé de discours, elle permet de fournir un résumé par section du document et tout cela de façon automatique.

Dans certaines applications, la définition d'un thème peut être relativement accessible comme dans la segmentation de journaux télévisés où il est possible de délimiter des sujets individuels [Tur et De Mori, 2011, Guinaudeau, 2011, Bouchekif, 2017].

Dans d'autres applications comme des discussions en table ronde, si le sujet global est mono-thématique, il peut être difficile de segmenter l'ensemble du document, même pour un humain. On peut dans ce cas considérer les segments comme des activités différentes (discussion ou présentation) ou bien des *intentions* [Passonneau et Litman, 1997] (question, réponse, prise de décision). Au sein du projet PASTEL par exemple, des recherches sont effectuées dans le domaine de la structuration/segmentation thématique du discours (lié à des objectifs pédagogiques) à partir de transcriptions automatiques. D'autre recherches sont menées sur le traitement de flux en temps réel qu'exige une utilisation en présentiel [Mdhaffar *et al.*, 2018].

En somme, le choix de la segmentation dépend réellement des intérêts de l'utilisateur et de l'application souhaitée [Niekrasz et Moore, 2009]. L'identification de thèmes diffère de la segmentation. Dans la segmentation en thèmes, on considère qu'un document peut être séparé en différents morceaux correspondant à des sujets différents. L'identification de thèmes, en revanche, cherche à déterminer quels sujets sont couverts par un document.

Si les thèmes ne sont pas prédéfinis, ils peuvent être appris de façon non supervisée : cela peut être assimilé à un regroupement (*clustering*) thématique [Iyer *et al.*, 1994, Seymore et Rosenfeld, 1997].

#### 1.1.2.3 Détection d'intention et routage d'appel

La détection d'*intention* (*i.e.* l'objectif de l'utilisateur) est une tâche globale de compréhension de la parole. Elle consiste à interpréter l'intention de l'utilisateur à partir de son discours. Dans ce cadre, il peut s'agir de détecter des mots clés dans les phrases énoncées [Tur et De Mori, 2011]. La complexité est essentiellement de regrouper dans une même intention des phrases pouvant varier dans leur formulation (*"Je veux prendre un avion vers Paris"* et *"Je suis à la recherche d'un vol pour Paris"*).

La détection d'intention peut intervenir dans un centre d'appel pour déterminer l'intention de l'utilisateur et diriger l'appel vers le département approprié [Juang et Rabiner, 2005]. On parle alors de routage d'appel. Le routage d'appel peut s'avérer être une tâche complexe dépassant l'analyse d'une seule phrase [Xu et Sarikaya, 2013]. Plusieurs interactions humain-machine, sous la forme d'un dialogue, peuvent être nécessaires afin de collecter les fragments d'information requis pour classer le type d'appel [Paek et Horvitz, 2004, Gorin *et al.*, 1997].

#### 1.1.2.4 Extraction de concepts sémantiques

La recherche en compréhension de la parole se développe beaucoup dans les interactions humain-machine et plus particulièrement dans les *dialogues transactionnels* [Tur et De Mori, 2011] où plusieurs types d'informations doivent être collectés auprès de l'utilisateur. Ces types de systèmes sont souvent limités à un domaine spécifique ayant un espace sémantique restreint qui est défini par une représentation sémantique *ad hoc*.

**Représentation Attribut/Valeur *à plat*** Une représentation sémantique simple *à plat* peut être utilisée pour représenter l'interprétation d'une phrase parlée [De Mori *et al.*, 2008]. Par exemple dans [Béchet, 2007], l'auteur décrit un *décodage conceptuel* consistant à expliquer le sens d'un message par une séquence de concepts où un concept représente une unité de sens minimale pouvant avoir une portée générale (entités nommées, classes de verbes) ou une portée limitée au cadre applicatif. L'extraction automatique de ces concepts est alors fréquemment associée à une tâche de *remplissage de champs (slot filling)* ou *étiquetage en concepts sémantiques* comme décrit dans [Pieraccini *et al.*, 1992, He et Young, 2003, Hahn *et al.*, 2011, Mesnil *et al.*, 2015].

Un concept sémantique se définit comme une unité minimale de sens. Il se compose d'un attribut (nom du concept défini de manière *ad hoc*) auquel est rattachée une valeur (une des valeurs possibles que l'on pourrait rattacher à cet attribut). Il peut être porté par plusieurs mots consécutifs. L'ensemble de mots est alors appelé *support de concept*. Certains autres mots ne supportent pas de sens en rapport avec la représentation sémantique visée et ne sont alors associés à aucun concept. Usuellement, la détection des attributs est faite à l'aide d'une méthode d'apprentissage automatique supervisée ou de segmentation/étiquetage comme cela sera décrit plus tard (chapitres 3 et 4). La valeur est extraite dans un deuxième temps à partir des mots du support puis est normalisée via un ensemble d'expressions régulières adaptées au corpus ciblé. Les concepts se limitent et se définissent par rapport à un cadre applicatif.

Dans l'exemple du corpus MEDIA (section 1.2.3) concernant le domaine des réservations hôtelières, les concepts sémantiques à reconnaître concerneront des entités relatives à cette tâche (date d'arrivée/de départ, équipement de chambre, localisation touristique, *etc.*). Une fois les mots supportant un concept détectés et étiquetés, la valeur du concept est extraite à partir du support de concept puis normalisée, comme dans l'exemple de la figure 1.1.


FIGURE 1.1 – Un exemple de concepts dans MEDIA [Hahn *et al.*, 2008]. Ici on a *temps-date* et *objetBB* les attributs et entre crochets les valeurs.

**Représentation par cadre sémantique** Une représentation sémantique plus riche de l'espace sémantique doit prendre en compte de nombreux éléments tels que le raisonnement, la composition des constituants sémantiques en structures (agent, action, thème), et les procédures pour les relier aux signaux [De Mori *et al.*, 2008]. La structure de l'espace sémantique peut être représentée par un ensemble de modèles appelés *cadres sémantiques (semantic frames)* [Fillmore, 1976, Baker *et al.*, 1998] dont les *champs (slots)* correspondent aux va-

riables qui les composent. Ces structures peuvent être vues comme des organisations hiérarchiques de concepts. Elles sont identifiées par un nom et un ensemble de paires attribut/valeur, ou champs, avec des procédures pouvant être rattachées aux champs. Le but est alors de choisir le cadre sémantique correct pour un énoncé utilisateur et d'extraire à partir de l'énoncé la valeur de ses champs. Concevoir une telle représentation capable de capturer la riche expressivité du langage est difficile et c'est pour cela que les représentations sémantiques ont tendance à être adaptées en fonction des applications visées en ayant pour contrepartie de limiter la portabilité du système vers de nouveaux domaines et applications. Dans cette optique, la bonne compréhension du système se mesure par les actions prises en réponse à l'énoncé de l'utilisateur. La figure 1.2 montre un exemple de trois cadres sémantiques pour le domaine de ATIS concernant le secteur des réservations de vol (section 1.2.2).

```
<frame name="ShowFlight" toplevel="1">
    <slot name="Flight" filler="Flight"/>
</frame>
<frame name="GroundTrans" toplevel="1">
    <slot name="City" filler="City"/>
</frame>
<frame name="Flight">
    <slot name="DCity" filler="City"/>
    <slot name="ACity" filler="City"/>
</frame>
```

FIGURE 1.2 – Un exemple de cadre sémantique pour le domaine ATIS [Wang *et al.*, 2006].

Chaque cadre contient des champs dont le type précise avec quoi les remplir. Dans la suite de cette thèse, par abus de langage, nous appellerons *concept/valeur* le couple attribut/valeur. Nous appelerons *concept* l'attribut seul.

#### 1.1.2.5 Extraction d'entités nommées

La tâche d'extraction d'entités nommées [Grishman et Sundheim, 1996] consiste à détecter dans un texte des entités (ou catégories) nommées comme les noms de gens ou d'organismes, localisations géographiques, dates, montants *etc.* Le but est donc d'extraire dans un texte toutes les chaînes de caractères correspondant à une de ces entités. La tâche n'inclut pas la résolution de références [Tur et De Mori, 2011] : par exemple *"New York"* peut être étiqueté avec *"localisation"* mais *"la ville"* ne peut pas l'être, même si elle réfère à New York.

Les principaux problèmes rencontrés dans cette tâche sont les problèmes de segmentation et de classification. La segmentation consiste à trouver le début et la fin de la chaîne de mots associée à une entité alors que la classification est le fait d'associer la chaîne à la bonne catégorie selon l'ensemble d'entités défini.

L'extraction d'entités nommées forme un cas particulier d'extraction de concepts sémantiques : la problématique de segmentation correspond à celle de la détection du support de mot et la classification à celle de trouver le bon concept.

### 1.1.3 La compréhension dans un contexte de dialogue oral

Nous avons présenté dans la section précédente différentes tâches de compréhension de la parole. Dans cette section, nous présentons plus en détail la compréhension dans un système de dialogue oral qui constitue la tâche centrale de cette thèse.

#### 1.1.3.1 Système de dialogue oral

**Définition** Un système de dialogue oral (*Spoken Dialogue System*) est défini comme une application où la machine communique avec l'humain sous la forme d'une discussion [De Mori, 1998, Minker et Bennacef, 2004]. Elle opère via une interaction vocale avec un utilisateur dans le but de lui procurer un service (information, réservation, conversation, *etc.*) [Gorin *et al.*, 1997]. Plusieurs étapes sont nécessaires, dont l'extraction des informations sémantiques dans l'énoncé de l'utilisateur. Cette tâche de compréhension est celle étudiée dans cette thèse.

Plusieurs modules interviennent dans cette interaction humain-machine. Classiquement, ces modules sont :

1. Reconnaissance de la parole : Produit une transcription automatique (ensemble de mots) à partir du signal audio du discours utilisateur.
2. **Compréhension automatique de la parole** : Extrait une représentation sémantique à partir des mots issus du module précédent.
3. Gestionnaire de dialogue : Recherche l'information appropriée dans une base de données (relativement à la sortie du module précédent) puis applique la stratégie définie par rapport à ce qui a été compris en effectuant l'action requise.
4. Générateur de langage : Formule la décision prise sous la forme d'une phrase.
5. Synthétiseur vocal : Génère le signal de parole de la réponse pour communiquer avec l'utilisateur à l'oral.

Ils sont présentés de manière schématique dans la figure 1.3.

Dans ce cadre, le système de compréhension est souvent réduit à la construction d'une représentation sémantique spécifique à une fonction particulière. Cela est possible pour des systèmes fournissant des services d'information ou de réservation dans un cadre précis (indications touristiques, réservation de vols, *etc.*).

**Problématiques** Plusieurs problématiques concernent les systèmes de compréhension déployés dans des applications industrielles. Premièrement, les systèmes

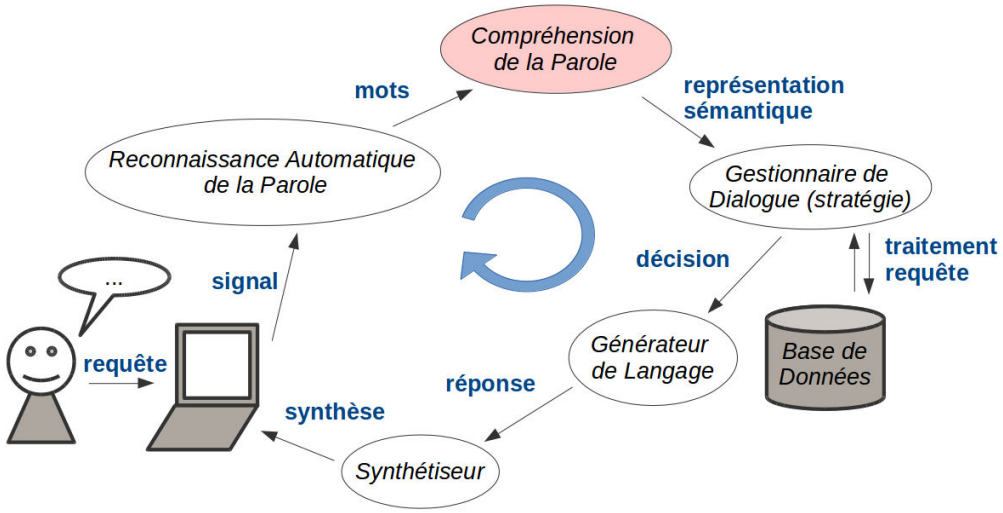



FIGURE 1.3 – Schéma d'un système de dialogue oral.

de compréhension sont généralement disponibles dans une seule langue et ne supportent donc pas le multilinguisme[1]. Ensuite, les systèmes existants sont très contraints et limités par la tâche ou le domaine d'application. Par conséquent, les systèmes de compréhension sont très dépendants de la langue et du domaine. Enfin la qualité de l'interaction utilisateur/système est toujours loin d'être aisée et naturelle.

**Système ouvert** Un système de dialogue oral peut être dit *ouvert* ou *multi-domaine*, signifiant que l'utilisateur peut dialoguer avec le système sans, ou avec peu, de restriction de sujet. Souvent dans une compréhension multi-domaine, un système global emploie différents sous-systèmes spécialisés sur une tâche sémantique particulière et l'utilisateur est redirigé vers le sous-système approprié sans que cela ne lui soit nécessairement explicité. La redirection vers le bon système s'apparente à un problème de routage d'appel (section 1.1.2.3). On peut citer par exemple le portail DialPort [Zhao *et al.*, 2016] qui propose différents types d'informations et d'interactions via une interface de type *chatbot* (messagerie avec un agent intelligent). Il met en relation des systèmes de recherche de différents groupes (CMU, Cambridge, USC, Santa Cruz) et l'utilisateur est dirigé vers le système approprié en fonction de sa demande.

Une autre branche de système de dialogue ouvert, ou en donnant l'illusion, sont les assistants personnels. Ceux-ci doivent coordonner d'autres types de données comme la localisation ou bien tenir en mémoire des informations de l'utilisateur pour

1. Cela est cependant de moins en moins un problème pour de grands acteurs tel que *Google* ou *Apple*.

que l'interaction lui soit personnalisée. La fonction domotique amène les systèmes de dialogue oraux à être également en connexion avec d'autres appareils tels que les lampes ou le thermostat. Cette branche connaît un essor commercial important avec l'intérêt croissant pour les assistants personnels tels que *Alexa (Amazon), Google, Siri (Apple)*. Cependant ce sont des systèmes propriétaires fermés à l'étude et dont les avancées technologiques ou les corpus employés sont peu rendus accessibles dans la recherche.

#### 1.1.3.2 Analyse sémantique du dialogue

**Humain-machine** D'autres champs d'études de la compréhension de la parole vont plus loin que la compréhension de ce qui est dit par l'utilisateur en s'étendant à la détection des émotions. Cet intérêt est porté par le développement d'interfaces humain-machine de plus en plus adaptée et réactive au comportement de l'utilisateur. Dans [Lee et Narayanan, 2005], la détection d'émotions se fait par l'utilisation combinée d'informations verbales (informations de langage et de discours) et non verbales (signaux de paroles liés à des émotions comme la prosodie, le stress, *etc.*). L'émotion peut également être intégrée dans la réponse de la machine comme dans le projet JOKER où un robot compagnon intègre de l'humour dans son interaction et évalue la réaction obtenue chez l'utilisateur [Dubuisson Duplessis *et al.*, 2015].

**Humain-humain** La compréhension de la parole est le plus souvent liée à une interaction avec une machine. La parole étant le vecteur privilégié de transmission d'informations entre humains [Tur et De Mori, 2011], les recherches en compréhension peuvent également s'étendre à la transcription automatique et à la compréhension de conversations entre humains : à deux locuteurs, en réunion, en conférence. Les recherches dans ce domaine s'orientent dans différentes catégories telles que la détection d'*actes de dialogue* comme les suggestions, questions, l'acquiescement, *etc.* Elles peuvent aussi s'apparenter à un problème de détection d'intention [Stolcke *et al.*, 2000, Tur *et al.*, 2006] (section 1.1.2.3). Il est délicat de traiter des conversations entre humains en raison de leur caractère privé mais il existe un registre de conversations qui sont sujettes à analyses comme dans les centres d'appels par exemple (*cf.* DECODA, section 1.2.5).

### 1.1.4 Représentation sémantique choisie

D'une manière générale, nous avons vu dans la section 1.1.2 que les tâches de compréhension de la parole sont souvent ciblées sur l'extraction d'informations sémantiques telles que des entités, des thèmes, des intentions, *etc.* Cette thèse a pour but l'étude de la compréhension appliquée dans un cadre de dialogue oral. Nous avons choisi d'appliquer notre étude sur le corpus MEDIA et cela nous restreint à un domaine sémantique particulier lié aux réservations hôtelières et informations touristiques (section 1.2.3).

La représentation sémantique choisie sera réduite à un ensemble de concepts

sémantiques associés à des valeurs : nous choisirons de nous focaliser sur la détection des concepts annotés dans ce corpus et l'extraction de leur valeur, utilisant pour cela une représentation sémantique à plat.

**Obtenir un étiquetage en concept/valeur** Nous choisissons dans cette thèse de nous restreindre à une représentation sémantique à plat consistant en une tâche d'étiquetage en concept sémantique (section 1.1.2.4). L'étiquetage consiste à extraire automatiquement une séquence de concepts sémantiques à partir de la séquence de mots issus de la phrase énoncée par l'utilisateur et cela sous forme de couples concept/valeur. Cet étiquetage est une interprétation du sens de l'énoncé utilisateur et permet d'alimenter le gestionnaire de dialogue.

La tâche d'étiquetage en concepts sémantiques soulève principalement deux problématiques. Premièrement, la segmentation en concepts. Comme vu dans la section 1.1.2.4, un concept peut couvrir plusieurs mots. Ainsi, il s'agit de délimiter correctement les mots associés à un concept. Deuxièmement, la catégorisation qui consiste à trouver le bon concept, c'est-à-dire associer le support de concept à un élément de l'ensemble défini de concepts et en extraire la valeur. La catégorisation du concept est au premier plan de notre tâche mais la segmentation reste toutefois essentielle, car détecter le bon concept sur les mauvais mots nous entraînera potentiellement à commettre une erreur au niveau de la définition de la valeur. Il convient donc de faciliter la catégorisation du concept en optimisant la segmentation.

Afin d'aider d'avantage la délimitation des séquences de mots supportant le concept, un formalisme courant [Ramshaw et Marcus, 1995] consiste à associer à chaque mot une unique étiquette ayant le nom du concept et de lui ajouter une lettre suffixe selon qu'il s'agit du premier mot du support (*B* - *Beginning*) ou d'un mot dans le support (*I* - *Inside*). S'il s'agit d'un mot hors de tout concept, la lettre suffixe correspondante sera *O* (*Outside*).

Ainsi l'étiquetage en concepts sémantiques est vu comme un étiquetage mot à mot où il faut trouver pour chaque mot l'étiquette correspondante (concept+suffixe ou *O*). Après avoir obtenu un étiquetage automatique de ces étiquettes, les concepts sont regroupés grâce à leurs suffixes et les valeurs sont extraites à partir des supports de mots.

Le tableau 1.1 montre un exemple de la tâche de compréhension par étiquetage en concepts sémantiques.

### 1.1.5 Mesures d'évaluation

Afin d'estimer les performances d'un système de compréhension, nous devons choisir la mesure la plus adaptée en fonction de la tâche. Dans cette partie, nous exposons différentes mesures classiques utilisées en compréhension. La mesure choisie fournit un score de qualité de l'hypothèse donnée par le système de compréhension par rapport à une référence.

| MOT | CONCEPT | ETIQUETTE | VALEUR |
|---|---|---|---|
| je<br>veux | null | O<br>O | - |
| une | nombre-chambre | nombre-chambre-B | 1 |
| chambre<br>double | chambre-type | chambre-type-B<br>chambre-type-I | double |
| pour<br>deux<br>personnes | sejour-nbPersonne | sejour-nbPersonne-B<br>sejour-nbPersonne-I<br>sejour-nbPersonne-I | 2 |

TABLE 1.1 – Exemple d'étiquetage en concepts sémantiques issu du corpus MEDIA (section 1.2.3).

#### 1.1.5.1 F-mesure

La f-mesure combine les métriques de rappel et de précision, des mesures classiquement utilisées dans le domaine de la recherche d'information. Elle détermine l'efficacité globale du système. On obtient un score prenant en compte la présence ou l'absence d'un concept dans une phrase sans notion de séquentialité :

$$f\text{-}mesure = \frac{2(precision.rappel)}{precision + rappel}$$

La précision est le pourcentage de concepts corrects trouvés par le système sur la totalité des concepts émis par le système. Le rappel représente le pourcentage de concepts corrects retrouvés parmi tous ceux qu'il fallait retrouver effectivement selon la référence. Rappel et précision sont habituellement calculés sur chaque classe avant d'en faire une moyenne pondérée.

Dans un cadre de compréhension, au lieu de faire un calcul pour chaque concept, on peut préférer déterminer globalement si les concepts sont détectés correctement. C'est le cas notamment dans [Mesnil *et al.*, 2013] où l'auteur met un outil à disposition pour ce calcul. Précision et rappel sont définis [2] comme suit :

$$rappel = \frac{nombre\ de\ supports\ corrects}{nombre\ de\ supports\ dans\ la\ reference}$$

$$precision = \frac{nombre\ de\ supports\ corrects}{nombre\ de\ supports\ dans\ l'hypothese}$$

Un support de concept est correct s'il commence et finit avec les mêmes mots pour l'hypothèse et la référence. La notion de valeur n'est pas prise en compte. Nous emploierons cette version de la f-mesure dans nos travaux.

2. Calculés à l'aide du script *conlleval.pl* fourni par *Recurrent Neural Networks with Word Embeddings DeepLearning 0.1 documentation (2015)*, http://www.deeplearning.net/tutorial/rnnslu.html#rnnslu.

#### 1.1.5.2 Taux d'erreur sur les mots

Le taux d'erreur sur les mots (*Word Error Rate* - WER) est une mesure utilisée en reconnaissance automatique de la parole (chapitre 2). Il indique le taux de mots incorrectement reconnus par rapport à la référence. Après avoir obtenu un alignement optimal entre la référence et l'hypothèse, le WER est défini comme suit :

$$WER = \frac{S + D + I}{N} * 100$$

avec $N$ le nombre de mots de la référence, $S$ le nombre de substitutions (mots remplacés par un autre), $D$ (*Deletion*) le nombre de suppressions (mots omis dans l'hypothèse) et $I$ le nombre d'insertions (mots ajoutés dans l'hypothèse)[3]. Plus le taux est faible (minimum 0) et plus la reconnaissance est bonne. Le taux maximum n'est pas borné et peut dépasser 100 en cas de très mauvaise reconnaissance s'il y a beaucoup d'insertions par exemple.

Le WER est utilisé dans de rares cas en compréhension où cela revient à évaluer l'exactitude de l'étiquetage en concepts sémantiques au niveau de l'étiquette sémantique (concept+suffixe). Dans ce rare cas, nous appellerons cela un taux d'erreur sur les étiquettes (*Label Error Rate* - LER).

#### 1.1.5.3 Taux d'erreur sur les champs

Le taux d'erreur sur les champs (*Slot Error Rate* - SER) [Makhoul *et al.*, 1999] est une mesure utile en compréhension prenant en compte la notion de frontière des entités nommées ou champs.

On compare dans notre cas les concepts : un concept de l'hypothèse est correct par rapport à celui de la référence s'il a les mêmes frontières, c'est-à-dire que son support de mots commence et se termine par les mêmes mots que celui de la référence. La formule est donc la même que pour le WER mais sur les concepts en tenant compte de leurs frontières.

#### 1.1.5.4 Taux d'erreur sur les couples concept/valeur

Le taux d'erreur sur les concepts (*Concept Error Rate* - CER) et le taux d'erreur sur les couples concepts/valeurs (*Concept-Value Error Rate* - CVER) sont des mesures spécialement dédiées à l'étiquetage en concepts sémantiques.

CER et CVER se calculent sur le même principe que le WER mais au niveau du concept ou du couple concept/valeur. Le CER se concentre sur l'exactitude de la reconnaissance des concepts. Le CVER prend également en compte la détection correcte des valeurs liées aux concepts en regardant les couples concept/valeur. Cela signifie qu'un couple n'est correct que si on a détecté le bon concept et la bonne valeur.

3. L'alignement entre l'hypothèse et la référence peut être par exemple calculé avec l'utilitaire *sclite*, fourni par le NIST (National Institute of Standards and Technology) (`http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm`, NIST SCLITE Scoring Package Version 1.5).

| $M$ | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ | $w_7$ | $w_8$ | $w_9$ | $w_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $R_{CV}$ | x[$v_2$] | | y[$v_3$] | z[$v_4$] | - | x[$v_6$] | | y[$v_8$] | | |
| $R_{BIO}$ | x-B | x-I | y-B | z-B | O | x-B | x-I | y-B | y-I | y-I |
| $H_{BIO}$ | z-B | x-B | x-I | y-B | O | x-B | x-I | y-B | y-B | z-B |
| $H_{CV}$ | z[$v_1$] | x[$v_2$] | | y[$v_4$] | - | x[$v_6$] | | y[$v_8$] | y[$v_9$] | z[$v_{10}$] |

TABLE 1.2 – Exemple d'étiquetage en concept/valeur de référence avec une hypothèse fournie par un système d'étiquetage automatique.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $R_{BIO}$ | - | x-B | x-I | y-B | z-B | O | x-B | x-I | y-B | y-I | y-I |
| $H_{BIO}$ | z-B | x-B | x-I | y-B | - | O | x-B | x-I | y-B | y-B | z-B |
| | *I* | | | | *D* | | | | | *S* | *S* |

TABLE 1.3 – Alignement $R_{BIO}/H_{BIO}$ du tableau 1.2.

Les métriques de CER et CVER sont donc les métriques favorisées dans cette thèse car elles correspondent précisément aux entités recherchées dans notre tâche.

#### 1.1.5.5 Exemple

Nous présentons ici un exemple afin d'illustrer les mesures définies ci-dessus. Nous proposons une hypothèse factice fournie par un système d'étiquetage utilisant le formalisme BIO sur l'ensemble de concepts {x,y,z}.

Dans le tableau 1.2, il est à noter :

— $M$ Les mots $w_i$ de l'énoncé utilisateur
— $R_{CV}$ L'étiquetage en concept/valeur de départ obtenu par annotation d'un humain, sous le format *concept[valeur normalisée]*
— $R_{BIO}$ La transformation de $R_{CV}$ en étiquetage de concept pour chaque mot selon le formalisme BIO
— $H_{BIO}$ L'hypothèse générée par le système par étiquetage mot à mot
— $H_{CV}$ Le regroupement de concepts et l'extraction de valeur normalisée à partir de $H_{BIO}$ et $M$

Voyons maintenant ce que l'on obtient avec les différentes évaluations.

**F-mesure** On observe 5 supports de concepts dans la référence $R_{CV}$ et 7 dans l'hypothèse $H_{CV}$. Au total seul un est correct (sur les mots $w_6$ et $w_7$) en commençant et en finissant par les mêmes mots dans l'hypothèse et la référence. On a alors un rappel de $\frac{1}{5}$, une précision de $\frac{1}{7}$ et finalement une f-mesure d'environ 0,17.

**LER** Le tableau 1.3 montre l'alignement obtenue entre $R_{BIO}$ et $H_{BIO}$.

Quatre opérations sont comptabilisées pour dix étiquettes dans la référence. On calcule donc un LER de $\frac{4}{10} = 40\%$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $R_{CV}$ | $\mathrm{x}_{w_1,w_2}$ | $\mathrm{y}_{w_3}$ | $\mathrm{z}_{w_4}$ | $\mathrm{x}_{w_6,w_7}$ | $\mathrm{y}_{w_8,w_{10}}$ | - | - |
| $H_{CV}$ | $\mathrm{z}_{w_1}$ | $\mathrm{x}_{w_2,w_3}$ | $\mathrm{y}_{w_4}$ | $\mathrm{x}_{w_6,w_7}$ | $\mathrm{y}_{w_8}$ | $\mathrm{y}_{w_9}$ | $\mathrm{z}_{w_{10}}$ |
| | *S* | *S* | *S* | | *S* | *I* | *I* |

TABLE 1.4 – Alignement $R_{CV}/H_{CV}$ du tableau 1.2 en concepts avec frontières de mots.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $R_{CV}$ | - | x | y | z | x | y | - | - |
| $H_{CV}$ | z | x | y | - | x | y | y | z |
| | *I* | | | *D* | | | *I* | *I* |

TABLE 1.5 – Alignement $R_{CV}/H_{CV}$ du tableau 1.2 en concepts uniquement.

**SER** Pour le SER où on distingue les frontières, l'alignement est celui présenté dans le tableau 1.4. On note $concept_{w_a,w_b}$ le concept dont les frontières sont les mots $w_a$ et $w_b$.

On voit qu'avec ces critères, seul un concept est correct. Six opérations sont comptabilisées pour cinq concepts dans la référence. On obtient donc un SER très élevé de $\frac{6}{5} = 120\%$. Comme vu auparavant, le taux maximum n'est pas borné et peut dépasser 100 en cas de très mauvaise reconnaissance.

**CER** Le tableau 1.5 montre l'alignement obtenu entre $R_{CV}$ et $H_{CV}$ en ne tenant pas compte de la valeur.

Dans ce cas on a un CER de $\frac{4}{5} = 80\%$.

**CVER** Le tableau 1.6 montre l'alignement obtenu entre $R_{CV}$ et $H_{CV}$ en tenant compte de la valeur.

Dans ce cas on a un CVER de $\frac{5}{5} = 100\%$. CVER et SER sont assez proches dans leurs calculs. Ce qui les distingue c'est la notion de frontière pour le SER et de valeur normalisée pour le CVER.

## 1.2 Les corpus

L'entraînement et l'évaluation du système de compréhension dans un cadre d'apprentissage automatique supervisé nécessitent l'usage de corpus annotés pour chacune des différentes phases d'apprentissage et de test.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $R_{CV}$ | - | x[$v_2$] | y[$v_3$] | z[$v_4$] | x[$v_6$] | y[$v_8$] | - | - |
| $H_{CV}$ | z[$v_1$] | x[$v_2$] | y[$v_4$] | - | x[$v_6$] | y[$v_8$] | y[$v_9$] | z[$v_{10}$] |
| | *I* | | *S* | *D* | | | *I* | *I* |

TABLE 1.6 – Alignement $R_{CV}/H_{CV}$ du tableau 1.2 en couples concept/valeur.

Disposer de corpus annotés est peu courant. Peu sont librement mis à disposition et les entreprises possèdent souvent leur propres corpus. Il est donc possible dans le domaine de la recherche de travailler sur des corpus ayant une à plusieurs décennies d'ancienneté.

La première partie de cette section traite de l'utilisation que l'on peut faire d'un corpus dans un cadre d'apprentissage, tandis que les suivantes donnent des exemples de corpus classiquement utilisés dans le domaine de la compréhension de la parole.

### 1.2.1 Corpus et classification

Un corpus forme un ensemble fini de textes ou de documents choisis comme base d'étude. Chaque corpus définit un cadre applicatif avec une représentation sémantique (section 1.1.1) et une difficulté qui lui est propre. Des annotations manuelles sont habituellement produites pour étiqueter chaque texte avec des étiquettes sémantiques. La classification consiste alors à construire un modèle basé sur ce corpus qui permettra d'associer automatiquement chaque texte à son ensemble d'étiquettes sémantiques.

Un corpus se décompose classiquement en trois ensembles APP, DEV et TEST :

— L'ensemble APP est utilisé pour entraîner le système de compréhension. Disposant des mots et des couples concept/valeur associés, le système construit un modèle de classification en se fondant sur les données d'apprentissage annotées et l'algorithme employé. Le modèle consiste en l'ensemble condensé des informations pertinentes retenues pour la détermination d'une classe en fonction d'une donnée.

— L'ensemble DEV est utilisé pour l'ajustement des paramètres du système en vue de son optimisation. Les ajustements s'effectuent en fonction des résultats de classification obtenus sur l'ensemble DEV (corpus sur lequel le système n'a pas appris), afin de rendre le modèle plus robuste. La meilleure configuration est choisie en fonction des meilleurs résultats observés sur l'ensemble DEV.

— L'ensemble TEST est utilisé pour obtenir les performances finales du système une fois l'apprentissage fini. Il s'agit d'une portion de corpus que le système n'a jamais vu au cours de son apprentissage et de son ajustement. Ainsi, les résultats obtenus sur le corpus TEST donnent une évaluation légitime des performances du système et surtout estiment sa capacité de généralisation, c'est-à-dire de s'adapter à de nouvelles données jamais rencontrées.

Après avoir présenté l'utilisation d'un corpus ci-dessus, nous présentons dans les sections suivantes des exemples de corpus classiquement utilisés avec en section finale un tableau récapitulatif.

### 1.2.2 ATIS

Le corpus anglais ATIS (*Airline Travel Information System*) proposé en 1990 [Hemphill *et al.*, 1990] par la DARPA (*Defense Advance Research Projects Agency*)

est un corpus spécialisé dans les requêtes de réservation de billets d'avion.

Il est segmenté et annoté avec des étiquettes sémantiques concept/valeur et peut ainsi servir de corpus d'entraînement pour des systèmes supervisés de compréhension. Les tours de parole sont aussi classés selon l'intention et le domaine.

La collecte des tours de parole est faite entre des utilisateurs et un système simulé (protocole *Wizard of Oz*) où seuls les tours de parole des utilisateurs sont utilisés pour l'apprentissage et la classification. Il est composé de respectivement 4978 et 893 phrases annotées pour les ensembles APP et TEST et selon 64 concepts sémantiques pour un vocabulaire de 572 mots.

Par exemple dans la commande *"show flights from Boston to New York today"*, *"Boston"* est associé au concept *"departure"*, les mots *"New York"* avec le concept *"arrival"* et *"today"* avec *"date"*. L'intention est *"Find Flight"* tandis que le domaine est *"Airline Travel"*.

C'est le corpus international état de l'art. Il a notamment été utilisé par [He et Young, 2003, Raymond et Riccardi, 2007, Tur *et al.*, 2010, Mesnil *et al.*, 2013] pour la tâche d'extraction de concepts sémantiques.

### 1.2.3 MEDIA

Le corpus MEDIA (2003) est un corpus de dialogue état de l'art français collecté dans le projet Media/Evalda ([Devillers *et al.*, 2004, Bonneau-Maynard *et al.*, 2005, Bonneau-Maynard *et al.*, 2009]). Il contient 1257 dialogues téléphoniques entre des utilisateurs et un système Wizard of Oz dans le domaine de la réservation d'hôtel et des informations touristiques. Cet ensemble de tours est divisé en trois sous-corpus : l'ensemble APP qui contient 17,7k énoncés, l'ensemble DEV qui contient 1,3k énoncés et enfin l'ensemble TEST qui est composé de 3,5k énoncés (proportions données dans la figure 1.4).

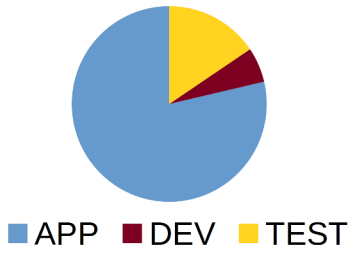



Figure 1.4 – Proportions du corpus MEDIA.

Le vocabulaire est plus riche que celui d'ATIS avec 2460 mots. Chaque énoncé a été manuellement transcrit et annoté avec 76 concepts sémantiques associés à leur valeur. Les concepts vont de simples réponses (*e.g.* le mot *"oui"* est associé au concept *"reponse"*) à des requêtes spécifiques à la tâche de réservation hôtelière (*e.g.* les mots *"avec baignoire"* sont associés au concept *"equipement_chambre"*). Il y a aussi des concepts plus généraux concernant la date ou la localisation (*"temps-mois"*, *"localisation-codePostal"*, ...). Enfin le corpus MEDIA contient des concepts

complexes tels que les co-références (tableau d'exemple 1.1, section 1.1.4).

Dans MEDIA, le but du dialogue pour l'utilisateur est d'obtenir des informations qui sont stockées dans une base de données. Par conséquent, les noms de rues, de villes ou d'hôtels, les listes d'équipement de chambre, les types de nourriture, *etc.* sont connus. De plus, des mots plus généraux représentant les nombres, les jours, les mois sont également connus. Tous ces mots (spécifiques à la tâche de compréhension ou généraux) ont été rassemblés dans un lexique sémantique qui permet d'associer un mot à une classe sémantique. Les concepts du corpus MEDIA sont également annotés selon un mode (positif, affirmatif, interrogatif ou optionnel) et un spécifieur (définit des relations entre les concepts). Tous les niveaux d'annotations ne sont pas à considérer obligatoirement. Le mode ou les spécifieurs ne sont pas considérés ici afin de se comparer aux plus grands nombres de travaux publiés sur MEDIA travaillant sur le concept et la valeur [Hahn *et al.*, 2011, Vukotic *et al.*, 2015, Dinarelli et Tellier, 2016].

En moyenne, on compte dans le corpus MEDIA environ 7 mots pour 3 concepts par tour de parole. Cependant le nombre de concepts annotés dans une phrase a une forte variabilité et peut inclure plus de 150 mots et plus de 30 concepts. Combiné à l'important nombre de concepts représentés (76 au total) et à la présence de concepts complexes à traiter (co-références), le corpus MEDIA est un corpus ambitieux sur lequel travailler.

### 1.2.4 PORTMEDIA

Les corpus PORTMEDIA (2009) ont été développés dans le cadre du projet ANR éponyme. Ils complètent le corpus MEDIA afin de favoriser le développement et l'évaluation de méthodes performantes pour la compréhension dans le cadre des systèmes de dialogues humain-machine [Jabaian *et al.*, 2010, Lefèvre *et al.*, 2012, Lefèvre *et al.*, 2012, Dediu *et al.*, 2013].

Le projet PORTMEDIA tente d'apporter des solutions à certaines difficultés rencontrées en compréhension de la parole par le développement de nouveaux corpus visant des objectifs distincts mais complémentaires. Il se focalise notamment sur la robustesse aux erreurs de reconnaissance, une représentation sémantique haut-niveau et une portabilité multilingue et multi-domaine.

Pour répondre à cette dernière problématique et adapter un système de compréhension vers un nouveau domaine ou langage, PORTMEDIA a lancé une collecte importante d'une base de données de dialogues ciblés : la collecte d'un nouveau corpus en italien pour la portabilité multilingue, et celle d'un nouveau corpus sur un nouveau domaine (réservation de billets de festival) pour la portabilité multi-domaine. Le corpus italien est la traduction du corpus MEDIA (réservation hôtelières et informations touristique) et suit les mêmes règles avec 10,4k énoncés pour un vocabulaire de 3253 mots annotés parmi les mêmes concepts sémantiques de MEDIA. Le corpus français sur une tâche inédite est constitué de 10k énoncés pour un vocabulaire de 3065 mots annotés parmi 35 concepts sémantiques.

| | ATIS | MEDIA | PORTMEDIA | DECODA |
|---|---|---|---|---|
| ***année*** | 1990 | 2003 | 2009 | 2012 |
| ***langue*** | Anglais | Français | Français/Italien | Français |
| ***cadre*** | Billets d'avions | Tourisme | Billets de festival /Tourisme | Transports Parisiens |
| ***tâche*** | Étiquetage sémantique + Détection intention-domaine | Étiquetage sémantique | Étiquetage sémantique | Routage d'appel |
| ***phrases*** | 5,8k | 22,5k | 10k/10,4k | 96,1k |
| ***concepts*** | 64 | 76 | 35/76 | 12 |
| ***vocab.*** | 572 | 2 460 | 3 065/3 253 | 8 806 |
| ***transc-ription*** | Manuelle | Manuelle | Manuelle + Automatique | Manuelle |

TABLE 1.7 – Résumé comparatif des corpus présentés.

### 1.2.5 DECODA

Le corpus français DECODA (2012) [Béchet *et al.*, 2012, Lailler *et al.*, 2015] traite de données de centres d'appels de la Régie Autonome des Transports Parisiens (RATP). Ce cadre applicatif permet de collecter une grande quantité de données de différents types de locuteurs et avec peu de données personnelles révélées. La présence de données privées est un frein à la disponibilité publique. Néanmoins, dans DECODA, le type d'utilisateurs de ce service demande des informations de bus ou de métro sans parler d'eux personnellement. Ainsi l'anonymisation des données est simplifiée car il n'y a pas beaucoup de signal à retirer.

Ce corpus peut servir à l'entraînement de classification de tours de parole (routage d'appel) où le but est d'associer une étiquette parmi 12 à un tour complet (*Info trafic, Itinéraire, Objets trouvés* ...). L'étiquetage ne se fait donc plus au niveau du mot comme dans les corpus précédents.

Le corpus a donc été anonymisé, segmenté, transcrit manuellement, puis annoté à différents niveaux linguistiques : disfluences verbales, étiquette morphosyntaxique (*Part Of Speech* - POS), entités nommées, dépendances syntaxiques.

Il est composé de 1514 dialogues (durée moyenne de 3 minutes) pour 74 heures de signal enregistrées sur deux jours de trafic afin d'avoir une bonne représentation des requêtes faites tout au long de la journée. Cela consiste en un total de 96,1k tours utilisateurs étudiables. Le vocabulaire est riche de 8806 mots.

### 1.2.6 Résumé

Cette section présente un résumé des caractéristiques principales des corpus cités dans les sections précédentes, regroupées dans le tableau 1.7.

## 1.3 Conclusion

Ce premier chapitre a donné une présentation de la tâche de compréhension de la parole. La compréhension de la parole est une tâche informatique vaste et complexe : elle peut être conçue de plusieurs façons, trouver plusieurs définitions et concerner différentes tâches.

La compréhension de la parole a finalement été réduite à notre propre cadre applicatif, c'est-à-dire une compréhension dans un cadre d'étiquetage en concepts sémantiques dans un système de dialogue oral avec une représentation sémantique concept/valeur à plat.

Ce chapitre a également fait la présentation de différentes mesures d'évaluation pouvant être utilisées en compréhension dont le CER/CVER qui sera retenu. Néanmoins la f-mesure, assez répandue dans certaines études présentées dans cet état de l'art, sera parfois utilisée à des fins de comparaison. Des corpus classiquement utilisés ont été présentés ensuite.

CHAPITRE 2

# LA RECONNAISSANCE DE LA PAROLE

**Sommaire**

Ce chapitre traite de la reconnaissance de la parole dont dépend directement la compréhension de la parole.

Nous souhaitons décrire ici le module de reconnaissance automatique de la parole (*Automatic Speech Recognition* - ASR) qui se trouve en amont du module de compréhension automatique de la parole, comme décrit dans la section 1.1.3.1.

De nos jours, les systèmes de reconnaissance de la parole sont construits avec une approche guidée par les données [Sarikaya *et al.*, 2014, Mesnil *et al.*, 2015, Hakkani-Tür *et al.*, 2016]. Le système utilisé dans cette thèse, et développé par le LIUM, est décrit dans la section 6.1.1. On peut citer également d'autres systèmes tels que *CMU Sphinx* [Lee *et al.*, 1990], *Microsoft Whisper* [Huang *et al.*, 1995], *CUED-HTK* [Woodland *et al.*, 1998], *Julius* [Lee et Kawahar, 2009], *Kaldi* [Povey *et al.*, 2011], *RASR* [Wiesler *et al.*, 2014], ou encore des systèmes de reconnaissance français : *LIMSI* [Gauvain *et al.*, 1994] par le LIMSI, *SPEERAL* [Nocera *et al.*, 2002] par le LIA, *ANTS* [Brun *et al.*, 2005] par le LORIA.

Les sections suivantes décrivent la constitution et les enjeux d'un système de reconnaissance de la parole ainsi que différentes mesures associées pouvant être utilisées dans le cadre de la compréhension de la parole.

## 2.1 Description

La reconnaissance de la parole a pour but d'extraire l'information lexicale contenue dans le signal de parole. Plus particulièrement, nous nous intéressons à la *reconnaissance de la parole continue*. Elle permet de traiter un flux continu de parole pour que le locuteur s'exprime de façon naturelle, à la différence d'une reconnaissance de mots isolés (pause entre chaque mot prononcé) ou de mots connectés (un ou plusieurs mots prédéfinis).

L'objectif des systèmes de reconnaissance automatique de la parole continue est de produire la séquence de mots prononcés à partir du signal acoustique. Cela se fonde en général sur une approche statistique introduite dans [Jelinek, 1976] qui recherche la séquence de mots $W^* = w_1 w_2 ... w_k$ à partir des observations acoustiques $X = x_1 x_2 ... x_t$ en cherchant à maximiser :

$$W^* = \arg\max_W P(W|X)$$

où $P(W|X)$ est la probabilité d'émission de la séquence de mots $W$ sachant $X$.

Suivant le théorème de Bayes, cela peut se décrire ainsi :

$$W^* = \arg\max_W \frac{P(X|W)P(W)}{P(X)}$$

La probabilité $P(X)$ est une constante indépendante de la séquence $W^*$ reconnue. Par conséquent, on obtient :

$$W^* = \arg\max_W P(X|W)P(W)$$

ne laissant plus à considérer que deux paramètres requérant des modèles probabilistes :

— La probabilité $P(X|W)$ d'observer la séquence acoustique $X$ par l'énonciation des mots $W$. Elle est estimée par un *modèle acoustique*.

— La probabilité $P(W)$ que $W$ soit énoncé dans le langage. Elle est donnée par un *modèle de langage*.

La figure 2.1 représente les différents composants en interaction dans un module de reconnaissance et les sections suivantes décrivent chacun de ces composants.

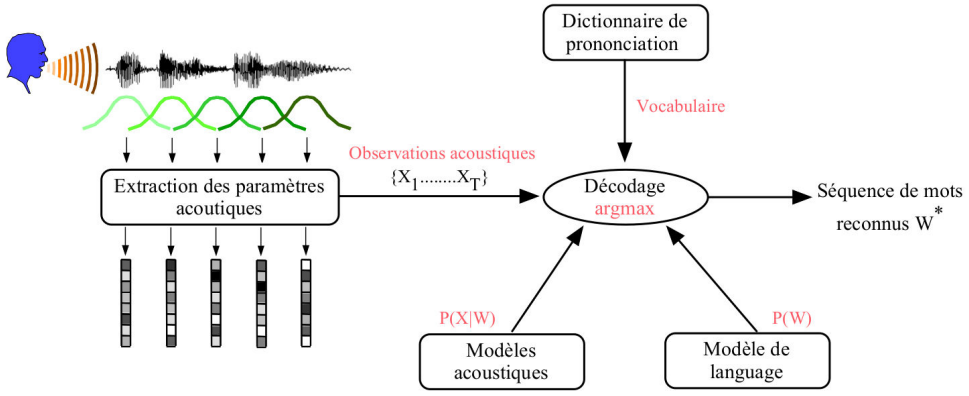



FIGURE 2.1 – Architecture d'un système de reconnaissance de la parole [Ghannay, 2017].

### 2.1.1 Modèle acoustique

Le modèle acoustique sert à estimer la vraisemblance du signal acoustique, soit $P(X|W)$.

Les modèles de Markov cachés (*Hidden Markov Model* - HMM) figurent parmi les méthodes proposées pour la modélisation acoustique du signal [Jelinek, 1976, Rabiner, 1989]. Les HMM sont des automates probabilistes à états finis. Dans le cadre de la reconnaissance, l'unité est le phonème (la plus petite unité de son pour distinguer un mot). Ainsi chaque phonème est modélisé par un HMM distinct et la modélisation d'un mot consiste à considérer l'ensemble des modèles de phonèmes successifs le composant. La figure 2.2 montre un exemple de HMM.

Les cercles représentent des états $E_i$ capables de générer des observations $o_j$ avec une densité de probabilité $b_i(o_k)$ (soit $P(o_k|E_i)$. La transition entre deux états a la probabilité $P(E_{i+1}|E_i)$ et seules les transitions de gauche à droite sont autorisées (par respect des contraintes temporelles du signal de parole). La concaténation de plusieurs HMM (phonèmes) permet d'obtenir des mots puis des phrases.

Pour entraîner le modèle acoustique, on concatène les HMM des phonèmes à partir d'un corpus d'apprentissage de différentes réalisations de phonèmes de la

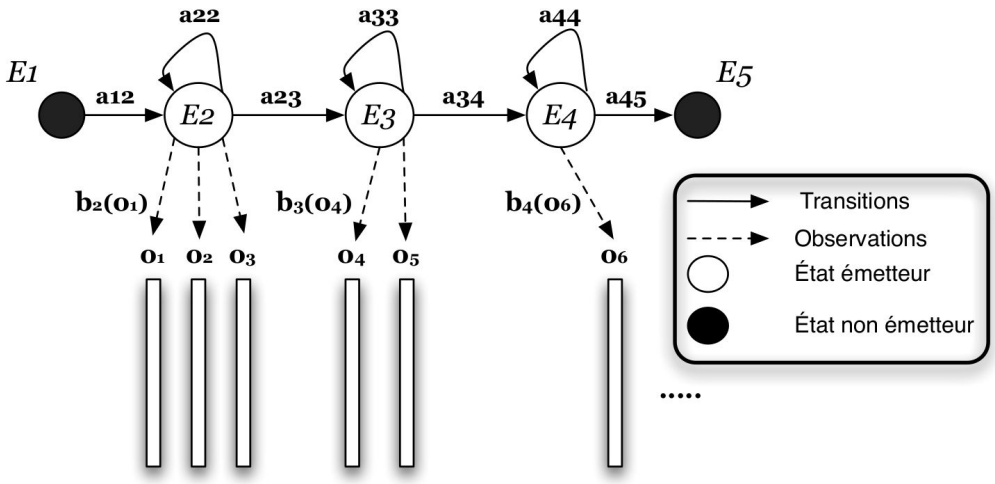



FIGURE 2.2 – Exemple d'un HMM à 5 états, dont 3 émetteurs [Bougares, 2012].

langue considérée. Un chemin dans cet automate représente toutes les chaînes de mots possibles et celui qui obtient la plus forte probabilité donne l'alignement optimal du signal acoustique. La résolution du meilleur chemin peut être obtenue par différents algorithmes (e.g. : algoritme de Viterbi [Forney, 1973], méthode itérative de Baum-Welch [Baum, 1972]).

L'estimation des probabilités d'observation associées aux états $b_i(o_k)$ a longtemps été obtenue par mélange de modèles gaussiens. Ces modèles estiment des densités de probabilité avec une somme pondérée de fonctions de densité gaussienne, d'espérance et de covariance.

Plus récemment, les réseaux de neurones profonds (*Deep Neural Network* - DNN) ont été introduits dans ce domaine où ils ont apporté de meilleurs résultats que les mélanges de modèles gaussiens [Hinton *et al.*, 2012]. La figure 2.3 montre une architecture d'un modèle acoustique HMM/DNN.

La dernière couche du réseau calcule la probabilité de chaque état des HMM sachant une observation.

### 2.1.2 Modèle de langage

Le modèle de langage, généralement de nature probabiliste, sert à évaluer les contraintes linguistiques pour choisir la suite de mots la plus probable selon $P(W)$. Pour modéliser ces contraintes, il attribue une probabilité à chaque séquence de mots $W$ de longueur $k$, tel que :

$$P(W) = P(w_1) \prod_{i=2}^{k} P(w_i|w_1...w_{i-1})$$

avec $w_1...w_{i-1}$ l'historique du mot $w_i$.

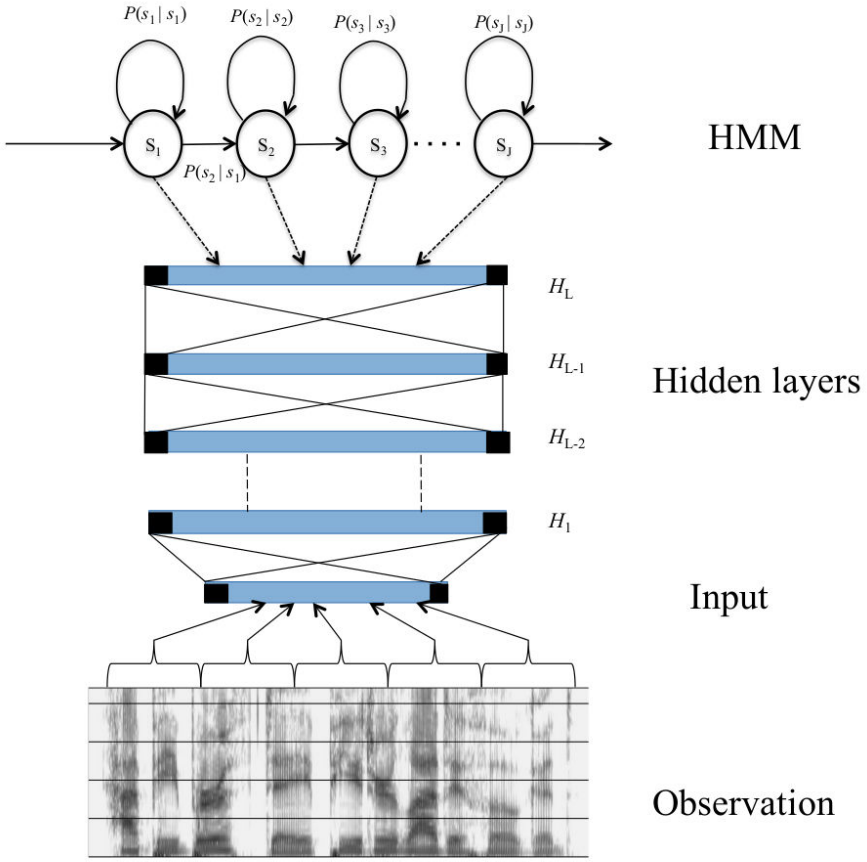



FIGURE 2.3 – Une architecture HMM/neuronale pour la modélisation acoustique [Samson Juan, 2015].

Un modèle n-gramme permet d'estimer cette probabilité avec un historique des $n$-1 mots le précédant. L'ordre de grandeur $n$ est généralement de trois ou quatre et dépasse rarement cette valeur. En effet, l'augmentation de l'ordre du modèle de langage n-gramme augmente de manière exponentielle sa complexité [Chen et Goodman, 1999].

Avec un ordre supérieur à un, l'équation devient :

$$P(W) = P(w_1) \prod_{i=2}^{k} P(w_i | w_{i-n+1} ... w_{i-1})$$

avec $P(w_1)$ la probabilité d'observer le mot $w_1$ et $P(w_i|w_{i-n+1}...w_{i-1})$ celle du mot $w_i$ sachant son historique $w_{i-n+1}...w_{i-1}$. Cette dernière probabilité se calcule avec la méthode du *maximum de vraisemblance* de façon à ce que la distribution des probabilités du modèle de langage soit celle qui maximise la vraisemblance du

corpus d'apprentissage :

$$P(w_i|h) = \frac{C(h, w_i)}{C(h)}$$

avec $h$ l'historique $(w_{i-n+1}...w_{i-1})$ et $C$ le nombre d'occurrences d'une séquence de mots dans le corpus d'apprentissage.

Des modèles de langages neuronaux émergent ces dernières années. Leur principe est de projeter les mots du contexte ($n$-1 mots précédents) dans un espace continu pour exploiter la similarité entre les mots. Les vecteurs résultants correspondent aux représentations continues des mots (voir section 4.2.1). Ces représentations confèrent aux modèles neuronaux une meilleure capacité de généralisation leur permettant de mieux gérer le problème des n-grammes absents du corpus d'apprentissage et de prendre en compte un contexte large [Bengio *et al.*, 2003]. Les modèles de langages neuronaux se fondent sur différents types d'architectures : propagation avant [Bengio *et al.*, 2003, Schwenk, 2007], récurrents [Mikolov *et al.*, 2010, Mikolov *et al.*, 2011]. Ces types de réseaux sont décrits dans les sections 4.1.1 et 4.1.3.1.

## 2.2 Impact des transcriptions automatiques dans la compréhension de la parole

La compréhension de la parole demeure une tâche très complexe même en étant restreinte à une tâche d'étiquetage en concepts sémantiques d'une part et réduite à une représentation sémantique spécifique d'autre part. En dépit d'importantes progression dans la recherche, les systèmes de compréhension font toujours des erreurs. Elles peuvent être liées à une difficulté de délimitation du support de concept, des ambiguïtés intrinsèques dans les supports localisés ou encore une faiblesse à identifier suffisamment de contraintes contextuelles pour résoudre les ambiguïtés d'interprétation.

Les transcriptions manuelles sont étudiées dans un premier temps avant les transcriptions automatiques afin de se focaliser uniquement sur les erreurs liées à la compréhension et non à celles liées à la reconnaissance de la parole. Cela a néanmoins l'inconvénient de placer le système de compréhension dans un cadre purement théorique et de ne pas l'entraîner à se placer dans la chaîne finale de modules décrits plus tôt (section 1.1.3.1) composée d'un système de reconnaissance puis d'un système de compréhension où la reconnaissance constitue l'entrée de la compréhension. Pour se rapprocher de la configuration finale visée (si on s'intègre dans un système de dialogue oral), il est nécessaire de travailler avec des transcriptions automatiques auxquelles on aura automatiquement aligné les concepts de référence.

Cependant, ce passage aux transcriptions automatiques impacte significativement les résultats en compréhension [Wang *et al.*, 2003, Hahn *et al.*, 2011]. En effet, des erreurs supplémentaires sont introduites par l'interaction entre un système de compréhension et un système de reconnaissance lui-même sujet à l'erreur et ayant un effet bruitant sur l'entrée de la compréhension. Tandis que certaines erreurs de

compréhension sont inhérentes à la tâche de compréhension (segmentation du support, identification du concept), d'autres en revanche seront liées à une répercussion des erreurs de reconnaissance selon qu'elles concernent :

— Un mot dans le support de concept. Cela risque d'impacter la délimitation ou bien l'identification du concept. De plus, dans le cas où le système de compréhension parviendrait néanmoins à segmenter et identifier le concept correctement, la valeur du concept (issue des mots du support) serait altérée, produisant ainsi une erreur en CVER non visible selon le CER.

— Un mot hors du support de concept. Dans ce cas la répercussion sur le concept pourrait être moindre, sauf s'il s'agit d'une information contextuelle que le module de compréhension pourrait utiliser pour désambiguïser la détection du concept courant. Dépendant de la tâche de compréhension visée, certains mots outils (ayant une portée plus syntaxique que sémantique) peuvent avoir une réelle utilité pour la détection du concept. Par exemple dans le cadre de la réservation de vol (corpus ATIS, section 1.2.2), les mots *"de","pour",...* auront une grande importance pour la détection des villes de départ et d'arrivée (*"de Paris vers Berlin","un vol pour Tokyo"...*). Une erreur de reconnaissance sur ces mots peut donc avoir beaucoup d'incidence. Dans le domaine de la détection d'entités nommées, on peut partir du même principe que certains mots sont utiles sans pour autant être présents dans les entités nommées. Les travaux de [Ben Jannet, 2015] proposent notamment une métrique qui permet de comparer la qualité de différentes transcriptions automatiques pour la détection d'entités nommées et capable de prendre en compte l'importance de ces mots.

Lorsqu'un système de compréhension travaille sur des transcriptions manuelles *i.e.* sans erreur, les erreurs de compréhension produites sont alors uniquement liées au système de compréhension. Lorsqu'un système de compréhension travaille sur des transcriptions automatiques en revanche, les erreurs de compréhension produites sont liées à la fois au système de compréhension et au système de reconnaissance. Ainsi, la bonne performance du système de compréhension est fortement liée à la bonne performance du système de transcription : plus le système de reconnaissance sera performant et plus on tendra vers les résultats que l'on obtiendrait sur des transcriptions manuelles.

## 2.3 Estimation de la qualité d'une transcription automatique

### 2.3.1 Principe

La précision de la reconnaissance automatique de la parole a été grandement améliorée au cours des trois dernières décennies. Néanmoins, les erreurs restent inévitables, en particulier dans des conditions bruyantes [Gong, 1995, Ghannay, 2017]. Nous sommes ainsi conscients de la présence d'erreurs dans les transcriptions au-

tomatiques. Il n'est donc pas judicieux de considérer ces transcriptions comme des transcriptions manuelles sans erreur, à fournir en entrée à d'autres systèmes de traitement de la parole (tel que le système de compréhension). Pour cette raison, on préférera attribuer un poids aux mots d'une transcription automatique témoignant de la probabilité de ce mot d'être juste ou erroné par le système de reconnaissance de la parole : il reviendra ensuite aux systèmes en aval de prendre cette information en compte comme une mesure de confiance par exemple.

En général, les erreurs faites par le module de reconnaissance de la parole sont réduites par une optimisation lors de l'estimation de ses paramètres qui minimise le WER au lieu de maximiser le score de vraisemblance [Mangu *et al.*, 2000].

Les erreurs faites sur les mots peuvent être en partie corrigées en associant une phrase hypothèse avec des mesures de confiance de mots. Dans [Yu *et al.*, 2011], des méthodes sont proposées pour construire des descripteurs de confiance afin d'améliorer la qualité de différents types de mesures de confiance. Les méthodes proposées pour optimiser les mesures de confiance sont basées sur un modèle de maximum d'entropie avec des contraintes de distribution, des réseaux de neurones artificiels [Zhang *et al.*, 2005], et des réseaux de connaissance profonds (*Deep Belief Network*). Plus récemment, de nouvelles fonctionnalités et des réseaux de neurones récurrents ont été utilisés pour la détection d'erreurs de reconnaissance [Ogawa et Hori, 2015].

### 2.3.2 Mesures retenues

Deux mesures de confiance sont utilisées dans cette thèse pour l'estimation d'erreurs de reconnaissance. Ces mesures de confiance de reconnaissance peuvent prétendre à fournir une information pertinente afin de mieux gérer les erreurs de reconnaissance dans un cadre de compréhension.

La première, la plus classique, est la probabilité de mot *a posteriori (pap)* calculée avec des réseaux de confusion comme décrit dans [Mangu *et al.*, 2000]. En calculant la séquence de mots la plus vraisemblable pour proposer une hypothèse de reconnaissance, un système de reconnaissance de la parole collecte des informations qui peuvent être utiles pour calculer la probabilité *a posteriori* d'un mot hypothèse [Ghannay, 2017]. Celle-ci donne une estimation de fiabilité sur le fait que le mot correspond à l'information lexicale portée par les observations acoustiques [Moreau *et al.*, 2000].

La seconde est une variante d'une nouvelle approche, introduite dans [Ghannay *et al.*, 2015a, Ghannay *et al.*, 2016a]. Cette mesure est calculée avec un perceptron multi-couches multi-flux (*Multi-Stream Multi-Layer Perceptron* - MS-MLP) prenant en entrée différents types d'informations de confiance. Parmi elles, les plus appropriées pour la compréhension sont : le plongement (*cf.* section 4.2.1) du mot courant et de ses voisins, sa longueur, un modèle de langage à mécanisme arrière (*backoff behavior*), le POS du mot courant, son étiquette de dépendance syntaxique avec le gouverneur et le plongement de son gouverneur. Les autres informations telles que les informations prosodiques et les plongements de mots acoustiques dans [Ghannay *et al.*, 2015b] et [Ghannay *et al.*, 2016a] pourraient aussi être utili-

| | ***pap*** | ***cm*** |
|---|---|---|
| NCE | 0,147 | 0,462 |

TABLE 2.1 – Comparaison des capacités de prédiction d'erreurs de reconnaissance de la parole de la *pap* et de la *cm* en terme de NCE sur le corpus MEDIA TEST.

sées mais n'ont pas été considérées. Une attention particulière a été portée sur le calcul des plongements de mot résultant d'une combinaison de différents plongements de mots connus (*CBOW*, *Skip-gram* fournis par *word2vec* [Mikolov *et al.*, 2013a], *GloVe* [Pennington *et al.*, 2014]) à travers l'utilisation d'un auto-encodeur neuronal et ce pour améliorer les performances du système de détection d'erreurs de reconnaissance [Ghannay *et al.*, 2016b]. Le MS-MLP utilisé pour la détection d'erreurs de reconnaissance a deux unités de sortie. Elles calculent les scores pour les étiquettes *Correct* et *Erreur* associées à une hypothèse générée par le module de reconnaissance de la parole. Cette hypothèse est évaluée par la valeur *softmax* de l'étiquette *Correct* marquée par le MS-MLP. On nomme *cm* la mesure de confiance dérivée de la valeur fournie par le système MS-MLP de détection d'erreurs pour l'étiquette *Correct*.

### 2.3.3 Comparaison des mesures

#### 2.3.3.1 Contribution d'information

Les expériences ont montré que la *cm* est plus efficace que la *pap* quand la comparaison est basée sur une entropie-croisée normalisée (*Normalized Cross Entropy* - NCE) [Ghannay *et al.*, 2015b], qui mesure la contribution d'information fournie pour chaque mesure de confiance. Le tableau 2.1 montre les valeurs NCE obtenues par ces deux mesures de confiance sur le corpus TEST de MEDIA.

#### 2.3.3.2 Capacités prédictives

La figure 2.4 montre les capacités prédictives de la *cm* comparées à celles de la *pap* sur le corpus MEDIA TEST.

La courbe montre le pourcentage prédit de mots corrects en fonction de l'intervalle de confiance. La meilleure mesure est celle dont les pourcentages sont les plus proches de la ligne diagonale.

## 2.4 Mesure de similarité pour la simulation d'erreurs

### 2.4.1 Principe

La section 2.2 a abordé l'idée qu'un système de compréhension devait travailler sur des transcriptions automatiques : il est nécessaire de disposer de transcriptions automatiques et non seulement manuelles.

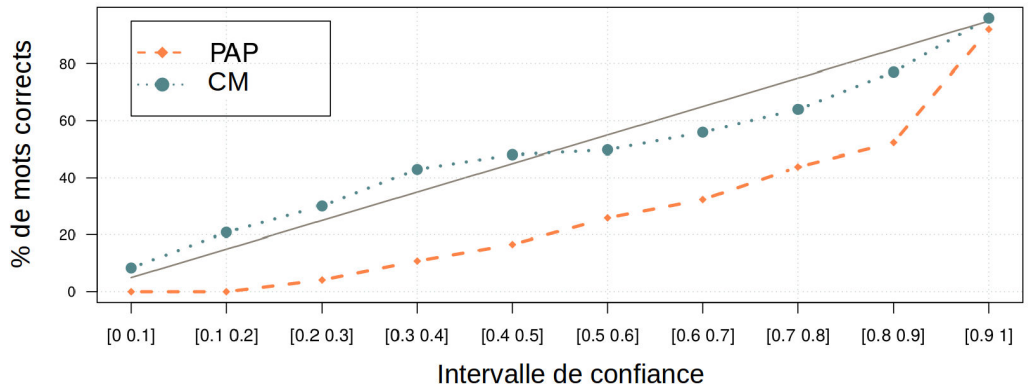



FIGURE 2.4 – Capacités prédictives de la *cm* comparées à la *pap* sur le corpus MEDIA TEST.

Dans certains cas, les corpus requis pour l'apprentissage des systèmes de dialogue sont rares. Ainsi, certaines méthodes ont été proposées pour simuler les erreurs de transcription dans ce cadre [Pietquin et Beaufort, 2005, Schatzmann *et al.*, 2007] pour l'optimisation de stratégie de dialogue oral ou pour la génération de données d'entraînement pour des systèmes de dialogues. La simulation d'erreurs de transcription a également été utilisée pour l'entraînement de modèles de langage discriminatifs afin d'améliorer les performances en WER des systèmes de reconnaissance [Jyothi et Fosler-Lussier, 2010].

### 2.4.2 Mesures de similarités linguistiques et acoustiques

Cette thèse utilise des nouvelles mesures de similarité dans une optique de simulation d'erreurs de reconnaissance. Ces mesures sont issues des travaux de Sahar Ghannay [Ghannay *et al.*, 2015b, Ghannay *et al.*, 2015a, Ghannay *et al.*, 2016a, Ghannay *et al.*, 2016b].

Dans ces travaux, on suppose que les mots susceptibles d'être confondus par un système de reconnaissance sont ceux acoustiquement proches. Cette hypothèse a également été retenue dans [Fosler-Lussier *et al.*, 2002, Stuttle *et al.*, 2004], où la simulation des erreurs est basée sur la similarité phonétique des mots. De plus, on considère que ces mots confondus peuvent également être linguistiquement proches.

Ainsi, les mesures de similarités s'appuient sur l'utilisation des plongements linguistiques et acoustiques pour prédire une liste de mots qui pourraient être substitués par un système de reconnaissance de la parole à un mot effectivement prononcé. On nomme cette liste *"liste de confusion"*. Elle se compose des mots les plus proches du mot analysé selon une mesure de similarité qui s'appuie sur la combinaison des similarités cosinus des plongements de types linguistiques et acoustiques.

Les plongements linguistiques de mots correspondent à la combinaison par analyse en composante principale de différents types de plongements de mots :

*word2vecf* [Levy et Goldberg, 2014], *Skip-gram* et *GloVe* comme décrit dans [Ghannay *et al.*, 2016b].

Les plongements acoustiques de mots correspondent à la projection de séquences acoustiques de longueur variable dans un espace de faible dimension de telle sorte que les mots qui se prononcent de la même manière sont projetés dans la même zone, tandis que les mots qui se prononcent différemment sont projetés dans des zones différentes. L'approche utilisée pour construire ces représentations s'inspire de celle proposée dans [Bengio et Heigold, 2014].

Les similarités linguistiques et acoustiques $L_{Sim}$ et $A_{Sim}$ entre deux mots $w_1$ et $w_2$ sont calculées avec la similarité cosinus appliquée respectivement aux plongements linguistiques et acoustiques de $w_1$ et $w_2$.

### 2.4.3 Interpolation linéaire des similarités

La mesure de similarité finale est obtenue en combinant l'interpolation linéaire des similarités cosinus linguistique et acoustique. La similarité résultante est appelée $LA_{SimInter}$, et est définie comme suit :

$$LA_{SimInter}(\lambda, w_1, w_2) = (1 - \lambda) \times L_{Sim}(w_1, w_2) + \lambda \times A_{Sim}(w_1, w_2)$$

où $w_1$ et $w_2$ sont les deux mots à comparer et $\lambda$ est le coefficient d'interpolation.

Comme l'objectif est de prédire ou corriger les erreurs du système de reconnaissance, la valeur $\lambda$ est optimisée à cette fin. Pour estimer $\lambda$, une liste connue d'erreurs de substitution générées par le système de reconnaissance est utilisée. Dans cette liste, on définit $h$ comme étant l'hypothèse de mot erronée et $\overline{r}$ le mot de référence qui a été substitué par $h$. Pour chaque paire de mots $(h, \overline{r})$ dans la liste, la probabilité que le mot $h$ soit reconnu lorsque le mot de référence $\overline{r}$ est erroné est calculée :

$$P(h|\overline{r}) = \frac{\#(h, \overline{r})}{\#\overline{r}}$$

où $\#(\mathrm{h}, \overline{r})$ est le nombre de substitutions de $\overline{r}$ par $h$ et $\#\overline{r}$ le nombre d'erreurs sur le mot de référence $\overline{r}$.

On retient alors le coefficient d'interpolation $\hat{\lambda}$ qui minimise l'erreur quadratique moyenne (MSE) entre la valeur proposée par $LA_{SimInter}(\lambda, h, \overline{r})$ et la valeur effective de $P(h|\ \overline{r})$. On définit $\hat{\lambda}$ tel que :

$$\hat{\lambda} = \arg\min_{\lambda} MSE(\forall(h, \overline{r}) : P(h|\overline{r}), LA_{SimInter}(\lambda, h, \overline{r}))$$

où $LA_{SimInter}(\lambda, h, \overline{r})$ et $P(h|\ \overline{r})$ sont calculés sur tous les couples $(h,\ \overline{r})$ possibles.

En utilisant $LA_{SimInter}$ avec $\hat{\lambda}$, il est maintenant possible de proposer pour un mot donné sa liste de confusion contenant ses voisins les plus proches linguistiquement et acoustiquement. La valeur de $LA_{SimInter}(\hat{\lambda}, x, y)$ est considérée comme une mesure de similarité entre les mots $x$ et $y$ et nous la notons plus simplement $confus(x, y)$ (elle sera utilisée en contribution dans la section 6.3.2).

## 2.5 Conclusion

Ce deuxième chapitre a apporté une description de la reconnaissance automatique de la parole. La reconnaissance de la parole n'est pas un sujet d'étude de cette thèse mais est néanmoins essentielle dans ses travaux étant donné qu'elle constitue l'entrée du module de compréhension de la parole dans un cadre de système de dialogue oral. Nous avons également défini deux mesures de confiance et une mesure de similarité pour la reconnaissance de la parole qui seront utilisées dans nos travaux sur la compréhension.

CHAPITRE 3

# PRINCIPAUX MODÈLES D'ÉTIQUETAGE ET DE CLASSIFICATION

**Sommaire**

La compréhension de la parole réduite à une tâche d'étiquetage en concepts sémantiques décrite dans le premier chapitre (section 1.1.4) peut être résolue avec des méthodes d'apprentissage automatique supervisé ou de segmentation/étiquetage.

Ce troisième chapitre aborde certains des modèles traditionnels utilisés avec succès pour la compréhension de la parole avant l'essor des méthodes neuronales (elles-mêmes décrites dans le prochain chapitre).

Comme vu dans la section 2.2, l'interprétation des transcriptions automatiques de la parole est difficile. Des modèles numériques d'interprétation probabiliste ont été introduits afin de résoudre cette tâche. Il s'agit d'apprendre un modèle qui associe automatiquement une étiquette sémantique à un tour de parole (classification) ou aux différents supports de mots (segmentation/étiquetage).

Dans un premier temps, nous apportons une présentation théorique de ces modèles. Dans un second temps, nous regardons leur application concrète avec une présentation de résultats obtenus en compréhension de la parole.

## 3.1 Description théorique

### 3.1.1 Grammaires

Les premiers systèmes de compréhension se sont concentrés sur les connaissances linguistiques afin de produire des systèmes à base de règles [Tur et De Mori, 2011] ou *grammaires*. Leur objectif est de transformer le langage naturel en une représentation logique afin d'obtenir une grammaire formelle sur laquelle on peut effectuer une analyse syntaxico-sémantique [De Mori *et al.*, 2008]. Si on considère le langage *fini*, il est possible de lister l'ensemble des chaînes le composant. Ainsi une grammaire formelle peut générer un langage donné avec un nombre fini de règles.

Par exemple on peut trouver dans [Boite *et al.*, 1999] une proposition de grammaire simplifiée :

$$phrase = groupe\ nominal + verbe\ conjugué$$

$$groupe\ nominal = déterminant + nom\ [+préposition + groupe\ nominal]$$

Une grammaire sémantique pourrait consister en :

$$phrase = agent + action + theme$$

$$agent = determinant + nom$$

$$action = verbe$$

$$theme = determinant + nom$$

Pour la phrase *"Le client accepte le contrat" ("the customer accepts the contract")* on aurait le résultat d'analyse décrit dans la figure 3.1.

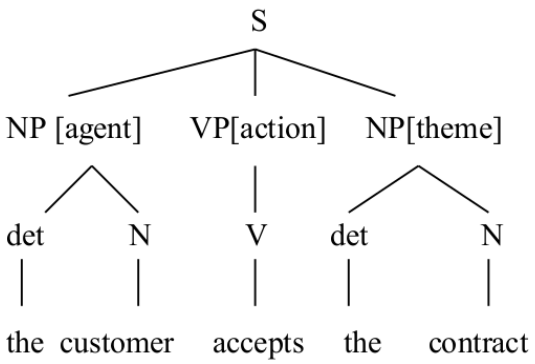



FIGURE 3.1 – Arbre d'analyse sémantique avec des étiquettes sémantiques attachées [De Mori *et al.*, 2008].

#### 3.1.1.1 Différentes classes de grammaires

Chomsky [Chomsky, 1957, Chomsky, 1965] donne la définition d'une grammaire ainsi :

$$G = (S_T, S_N, R, A)$$

tel que :

— $S_T$ est l'ensemble des symboles terminaux.
— $S_N$ est l'ensemble des symboles non-terminaux.
— $R$ est l'ensemble des règles pour passer d'un symbole à un autre.
— $A$ est l'axiome, le symbole de départ (inclus dans $S_N$).

Il a également défini une hiérarchie des grammaires formelles en plusieurs types selon la complexité des langages engendrés et les contraintes imposées aux règles de production. Ces grammaires (ou langages engendrés) sont classées de 0 à 3 et sont strictement incluses les unes dans les autres tel que :

$$Classe\ 3 \subset Classe\ 2 \subset\ Classe\ 1 \subset Classe\ 0$$

Elles sont définies telles que :

— **Classe 0** : Les grammaires générales qui définissent les langages *récursivement énumérables*. Ce type de grammaire n'a pas de restriction sur les règles. Cela inclut tous les langages définis par une grammaire formelle.
— **Classe 1** : Les grammaires *contextuelles* ou *sensibles au contexte* qui définissent les langages *contextuels*. Le membre de droite doit contenir au moins autant de symboles que le membre de gauche. Ces grammaires sont contextuelles car le traitement d'un élément non-terminal peut dépendre des éléments se trouvant autour de lui.
— **Classe 2** : Les grammaires *hors-contexte* (ou *algébriques*) qui définissent les langages *algébriques*. Le membre de gauche de chaque règle doit être constitué d'un seul symbole non-terminal. Ces grammaire sont hors-contexte

car, inversement à une grammaire contextuelle, les symboles non-terminaux sont traités indépendamment de la place où ils se trouvent.

— **Classe 3** : Les grammaires *régulières* qui définissent les langages *rationnels*. C'est un type de grammaire hors contexte où chaque règle doit au moins générer un symbole terminal. Les grammaires sont soit régulières *à gauche* où chaque membre droit de règle peut commencer par un non-terminal ($A \rightarrow B\mu$ ou $A \rightarrow \mu$ avec $A, B \in S_N$ et $\mu \in S_T$), soit régulière *à droite* où chaque membre droit de règle peut finir par un non-terminal ($A \rightarrow \mu B$ ou $A \rightarrow \mu$).

Les grammaires hors-contexte sont les plus utilisées pour le traitement du langage naturel. Dans le cadre d'un système de dialogue où les phrases sont finies et de taille limitée, on utilise habituellement des grammaires régulières [Mohri et Nederhof, 2001] dont la puissance de description permet de décrire une grande partie de la structure d'un langage.

#### 3.1.1.2 Grammaire probabiliste

Si un langage peut être reconnu par une grammaire, des résultats inattendus peuvent survenir lorsqu'une phrase est polysémique. Une analyse sémantique plus profonde est ainsi requise et peut nécessiter des informations contextuelles pour distinguer les différentes intentions de l'utilisateur. Dans l'exemple de [Allen *et al.*, 2007], la requête *"Peut-on déplacer les gens par hélicoptère ?"* énoncée dans un scénario de gestion de crise, peut représenter deux intentions différentes :

1. Un changement de programme : utiliser un hélicoptère à la place d'un camion
2. Une question de faisabilité : est-ce-possible ?

Des grammaires probabilistes (ou stochastiques) sont utilisées pour pondérer les hypothèses afin d'éliminer les ambiguïtés les plus courantes. À la différence d'une grammaire classique, une grammaire probabiliste est une grammaire où un poids est associé à chaque règle de production. Cela est fait dans le but de distinguer les dérivations possibles d'un même mot ou les interprétations d'une même expression. Une approche probabiliste utilise des poids pour choisir le sens le plus vraisemblable (ou le plus fréquent) dans le cas d'une ambiguïté où un mot peut avoir plusieurs sens. C'est le cas par exemple du système d'analyse TINA [Seneff, 1989], qui transforme une grammaire en donnant plus de poids aux structures de phrase les plus utilisées.

### 3.1.2 Automate à états finis

Avec un langage naturel fini représenté par une grammaire régulière, il est alors possible d'employer des automates à états finis (*Finite State Machine* - FSM) pour représenter les connaissances linguistiques issues de cette grammaire.

Un FSM, dont un exemple est donné dans la figure 3.2, représente le langage produit par une grammaire et peut déterminer l'appartenance d'une phrase à ce langage en prenant en entrée une chaîne de symboles et en effectuant un algorithme de reconnaissance de la chaîne [Raymond, 2005].

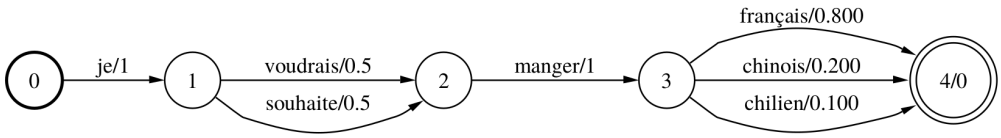



FIGURE 3.2 – Exemple de FSM représentant une grammaire [Raymond, 2005].

Le FSM reconnaît une chaîne s'il peut lire tous ses symboles en partant de l'état initial et en arrivant dans un état final. Le langage accepté par le FSM est l'ensemble des chaînes dont les symboles font passer de l'état initial jusqu'à un de ses états finaux par une suite de transitions utilisant les symboles disponibles dans l'ordre.

Un transducteur est un FSM capable de faire le lien entre deux ensembles de symboles. Il peut reconnaître des formes prédéfinies en faisant une analyse grammaticale à partir du langage reconnu comme le montre la figure 3.3.

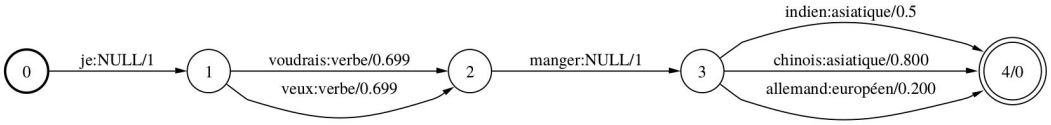



FIGURE 3.3 – Exemple de FSM transducteur [Raymond, 2005].

Le passage d'observations vers des classes peut se faire comme dans [Hahn *et al.*, 2008] par la combinaison de transducteurs $\lambda_{w_1^N} \circ \lambda_{w2c} \circ \lambda_{ML}$ avec :

— $\lambda_{w_1^N}$ la représentation par FSM du graphe de mot de la séquence d'observation $w_1 ... w_n$ généré par la reconnaissance automatique.
— $\lambda_{w2c}$ groupant les transducteurs traduisant les observations en classes.
— $\lambda_{ML}$ représentant le modèle de langage stochastique qui donne la probabilité conjointe avec $w_1^N$ la séquence d'observations, $c_1^N$ la séquence de classes et $h_n = \{w_{n-1}c_{n-1}, w_{n-2}c_{n-2}\}$ :

$$P(w_1^N, c_1^N) = \prod_{n=1}^{N} P(w_n c_n | h_n)$$

Ainsi, il s'agit d'un modèle génératif capable de générer des exemples à partir de la distribution conjointe entre la séquence de mots et la séquence de concepts.

### 3.1.3 Machines à vecteur de support

Les machines à vecteur de support (*Support Vector Machines* - SVM) ou séparateurs à vastes marges [Vapnik, 1982, Vapnik, 1995] forment une classe d'algorithmes

d'apprentissage pour construire un classifieur à valeurs réelles. Ils permettent de résoudre des problèmes de discrimination non-linéaire.

La classification consiste en deux tâches :

**La transformation non-linéaire des entrées** Il est nécessaire de se placer dans un espace où les données sont linéairement séparables. Cette transformation est requise dans le cas où les données ne le sont pas. On projette les données, dans un espace pouvant être de plus grande dimension, grâce à une transformation basée sur un noyau comme illustré dans la figure 3.4. Le noyau est une fonction retournant le produit scalaire de deux arguments et pouvant être linéaire, polynomiale ou gaussienne.

**Le choix d'une séparation linéaire optimale** Une fois que l'on dispose d'un espace linéairement séparable, les classes sont séparées par des classifieurs linéaires déterminant un hyperplan optimal, comme le montre la figure 3.5. Cet hyperplan doit séparer correctement les données et maximiser la distance du point le plus proche, appelée marge (représenté par $d$ dans la figure). Les hyperplans sont déterminés grâce à un certain nombre de points qui forment les *vecteurs supports*.

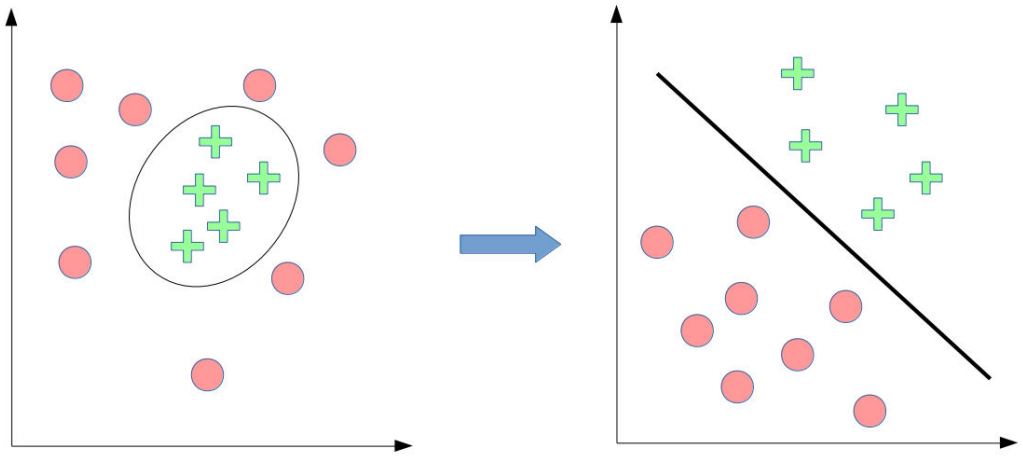


FIGURE 3.4 – Projection des données d'entrée dans un espace où elles sont linéairement séparables.

La plupart des problèmes de catégorisation de textes sont linéairement séparables. Afin d'utiliser un SVM sur du texte, on emploie la technique du *sac de mots*. Les mots sont représentés par des chiffres. Le lexique complet représente un vecteur et chaque phrase sera codée par ce vecteur [Joachims, 1998]. Afin de procéder à une tâche d'étiquetage, le problème est considéré comme une suite de classifications : une pour chaque élément de la séquence. Étant donné que les SVM sont des classifieurs binaires, une classification parmi $K$ classes est possible en procédant à une classification par paires. Dans [Hahn *et al.*, 2011], on construit $(K(K-1)/2)$ classifieurs pour considérer toutes les paires de classes. La décision finale est obtenue

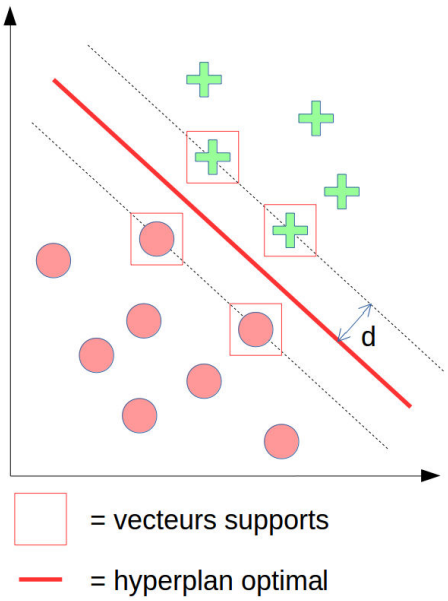



FIGURE 3.5 – Hyperplan optimal et marge maximale.

grâce à un vote pondéré entre les classifieurs.

### 3.1.4 Champs aléatoires conditionnels

#### 3.1.4.1 Description

Les champs aléatoires conditionnels (*Conditionnal Random Fields* - CRF) [Lafferty *et al.*, 2001] sont des modèles graphiques non orientés et discriminants suivant une méthode probabiliste permettant à la fois de segmenter et d'annoter une séquence de données.

Un modèle discriminant est un modèle statistique qui cherche à prédire l'appartenance d'un ensemble d'observations à une classe prédéfinie à partir d'un ensemble de descripteurs. Les méthodes discriminantes considèrent le problème à résoudre comme un ensemble de contraintes à prendre en compte et ne nécessitent pas d'hypothèse d'indépendance préalable.

Soit $X = (w_1, ..., w_N)$ la séquence d'observations et $Y = (c_1, ..., c_N)$ la séquence de classe, les CRF calculent la probabilité conditionnelle $P(Y|X)$, c'est-à-dire la probabilité d'obtenir la séquence de classe $Y$ sachant la séquence d'observations $X$. Leur principe est de maximiser la probabilité conditionnelle d'obtenir une séquence de classe $Y$ pour une séquence de mots $X$ donnée.

Soit $G = (V, E)$ un graphe (avec $V$ l'ensemble des noeuds ou sommets et $E$ l'ensemble des arêtes, soit des paires d'éléments de $V$) tel que $Y = (Y_v)_{v \in V}$ afin que $Y$ soit indexé par les sommets de $G$. Nous obtenons alors la formule générale

suivante [Hammersley et Clifford, 1971] :

$$P_\theta(Y|X) \propto exp(\sum_{e \in E,k} \lambda_k t_k(e, Y|_e, X) + \sum_{v \in V,k} \mu_k s_k(v, Y|_v, X))$$

où $\lambda_k$ et $\mu_k$ sont des vecteurs de poids, $Y|_S$ est l'ensemble de $Y$ associé aux sommets du sous-graphe $S$, $t_k(e, Y|_e, X)$ et $s_k(v, Y|_v, X)$ sont des vecteurs de caractéristiques supposés donnés et fixes.

Il s'agit alors, lors de l'apprentissage, de déterminer le vecteur de poids $\theta = (\lambda_1, \lambda_2, ...; \mu_1, \mu_2, ...)$ qui maximise la log-vraisemblance conditionnelle sur les données d'apprentissage tel que :

$$L(\theta) = \sum_{i=1}^{n} log P_\theta(y_i|x_i)$$

Plusieurs méthodes de résolutions existent alors (*Viterbi*, *Improved Iterative Scaling*,...) [Malouf, 2002].

#### 3.1.4.2 Avantages

L'avantage principal des CRF comparé à des modèles génératifs tels que les FSM est la possibilité d'utiliser l'ensemble des observations d'une séquence pour prédire une étiquette et de réduire les fortes dépendances faites par ces modèles. L'attribution d'une classe à une observation n'est pas contrainte par le seul historique immédiat mais par toutes les observations précédentes et suivantes. Cela est particulièrement intéressant quand la détermination d'une classe peut se faire avec des éléments situés avant ou après l'observation dans la chaîne, ou dans les autres chaînes.

En revanche, les modèles génératifs assignent une probabilité jointe à la paire de séquences observation-classe ($P(X, Y)$), cherchant donc à maximiser la probabilité conjointe des exemples rencontrés dans l'apprentissage.

Les CRF dépassent une limitation fondamentale des modèles génératifs qui peuvent être biaisés lorsque des états ont peu d'états successeurs.

**Biais des étiquettes** Le problème du *biais des étiquettes* vient du fait que les transitions entre états ne dépendent que des états mis en cause dans la transition et non de l'ensemble des états du modèle comme le montre l'exemple de la figure 3.6 [Lafferty *et al.*, 2001].

Supposons une séquence d'observations *rib* :

— L'observation $r$ fonctionne avec les deux transitions partant du premier état qui ont donc la même probabilité.

— L'observation $i$ n'a pas de conséquence sur les probabilités de transitions aux états suivants car les états 1 et 4 n'ont qu'une transition vers un autre état.

Les chemins 0123 et 0453 sont donc équivalents pour le modèle génératif et cela indépendamment de la séquence observée. Comme les transitions ne génèrent pas les observations mais sont conditionnées par elles, les états avec un seul état

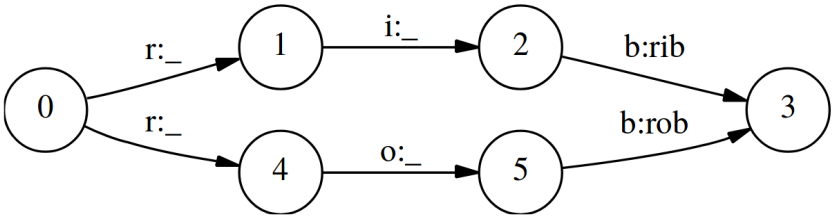



FIGURE 3.6 – Exemple de biais des étiquettes avec un modèle à états-finis conçu pour distinguer les mots *rib* et *rob* [Bottou, 1991]

suivant ignorent l'observation. De manière générale, les états avec peu de transitions prennent peu en compte l'observation. A cause de cela, si le corpus d'apprentissage contient plus de *rob* que de *rib*, il sera privilégié et le mot *rob* sera toujours décodé.

Les CRF résolvent ce problème de manière probabiliste en garantissant la convergence vers le *maximum de vraisemblance global* : l'ensemble des observations d'une séquence est prise en compte dans la prédiction d'une étiquette/classe. Toutes les observations précédentes et suivantes sont prises en comptes pour attribuer une classe et non pas seulement l'historique immédiat.

Pour conclure il a été montré dans des études comme [Rubinstein et Hastie, 1997] qu'un apprentissage de modèle discriminant est coûteux mais robuste et ne nécessite pas de connaissances spécifiques sur la distribution réelle des informations. En revanche, l'apprentissage d'un modèle génératif est peu coûteux mais le modèle doit s'adapter correctement à la distribution réelle.

## 3.2 Implémentation en compréhension de la parole

Cette section présente des résultats état de l'art obtenus par l'application des FSM, SVM et CRF sur la tâche d'étiquetage en concepts sémantiques (section 1.1.2.4) sur le corpus MEDIA (section 1.2.3).

Nous considérons l'étude menée dans [Hahn *et al.*, 2008, Hahn *et al.*, 2011] présentant des résultats état de l'art au début de cette thèse comme base de référence pour la suite de nos travaux.

Dans cette étude, les auteurs appliquent sur le corpus plusieurs méthodes d'étiquetage en concepts sémantiques dont les FSM, SVM et les CRF tels que décrits dans les sections 3.1.2, 3.1.3 et 3.1.4. Ils utilisent le corpus français MEDIA ainsi que d'autres corpus de langues différentes mais comparable à MEDIA en taille et détails d'annotation afin d'évaluer la portabilité de ces systèmes. Nous ne nous intéressons néanmoins qu'au corpus MEDIA ici. De plus, ils utilisent un système de reconnaissance de la parole (obtenant un WER de 31,4 sur le corpus MEDIA TEST) afin de fournir des tests à la fois sur les transcriptions manuelles et automatiques dans le but de mesurer l'impact de la reconnaissance automatique sur les systèmes (section 2.2).

| Descripteurs utilisés | CER |
|---|---|
| mot uniquement | 88,9 |
| config-descripteur-0 | **10,6** |

TABLE 3.1 – Comparaison des performances des CRF en fonction de l'utilisation de descripteurs sur le corpus MEDIA TEST (transcriptions manuelles) [Hahn *et al.*, 2011].

### 3.2.1 Connaissance *a priori* et descripteurs de mots

Les systèmes de compréhension peuvent effectuer leur tâche de compréhension plus efficacement à l'aide de connaissances *a priori* sur les mots [Hahn *et al.*, 2008, Hahn *et al.*, 2011].

**FSM** Les auteurs placent un transducteur $\lambda_{gen}$ entre les transducteurs $\lambda_{w_1^N}$ et $\lambda_{w2c}$ (décrits dans la section 3.1.2) qui convertit les mots en catégories sémantiques (*e.g.*, *VILLE*, *MOIS*, ...) représentant une connaissance *a priori* de la tâche de compréhension et permettant une meilleure généralisation des données d'apprentissage.

**SVM** L'algorithme utilise des SVM séquentiels en passes avant et arrière qui utilisent leurs précédentes décisions sous forme de descripteurs.

**CRF** Pour les CRF, les auteurs emploient un ensemble de descripteurs de mots sémantiques et syntaxiques. L'ensemble total de descripteurs utilisés est défini ensuite. Nous appellerons cette configuration de descripteurs **config-descripteur-0** :

- le mot
- les catégories sémantiques prédéfinies du mot qui sont :
  - des catégories spécifiques à MEDIA, telles que : noms des rues, villes ou hôtels, listes d'équipements de chambre, type de nourriture, ... *e.g.* : VILLE pour Paris
  - des catégories plus générales, telles que : nombres, jours, mois, ... *e.g.* : NOMBRE pour trente-trois
- un ensemble de descripteurs morphologiques : les premiers et derniers n-grammes de lettres (pour n de 1 à 4) du mot et un descripteur binaire indiquant si la première lettre est en majuscule (capitalisation).

Ces descripteurs sont activés par les fonctions caractéristiques définies dans la section 3.1.4.1 Les CRF intègrent des informations de contexte et prennent des valeurs discrètes en entrée.

L'information représentée par ces descripteurs de mots permet une amélioration significative des résultats des CRF comme le montre le tableau 3.1.

| | Transcriptions | | | |
|---|---|---|---|---|
| | manuelles | | automatiques | |
| **Modèle** | **CER** | **CVER** | **CER** | **CVER** |
| FSM | 14,1 | 16,6 | 27,5 | 31,3 |
| SVM | 13,4 | 15,9 | 25,8 | 29,7 |
| CRF (*cf.* tab. 3.1) | **10,6** | **12,6** | **23,8** | **27,3** |
| Combinaison ROVER | **10,2** | **12** | **23,1** | **26** |

TABLE 3.2 – Comparaison des performances des FSM, SVM et CRF sur le corpus MEDIA TEST [Hahn *et al.*, 2011].

### 3.2.2 Résultats expérimentaux

Le tableau 3.2 montre les résultats en CER/CVER obtenus avec les FSM, SVM et CRF.

Dans cette étude, les CRF obtiennent les meilleurs résultats sur MEDIA avec un CER de 10,6 et un CVER de 12,6 sur transcriptions manuelles et un CER de 23,8 et un CVER de 27,3 sur transcriptions automatiques. Les CRF obtiennent de manière systématique les meilleurs résultats dans toutes les configurations testées par rapport aux autres méthodes (la langue ou le type de transcription). Ils sont donc considérés comme l'approche état de l'art pour la compréhension automatique telle que définie dans la tâche d'étiquetage en concepts sémantiques. Cela n'est pas nécessairement le cas pour d'autres tâches de compréhension comme l'analyse d'opinion sur des messages de longueurs variables [Camelin *et al.*, 2010]. D'autres expériences ont déjà par le passé montré la supériorité des CRF sur des problèmes réels ([Lafferty *et al.*, 2001]).

Enfin, les auteurs observent une certaine variabilité entre les méthodes présentées pour les valeurs d'insertion/suppression/substitution ce qui laisse imaginer une possible complémentarité des méthodes. C'est pourquoi les auteurs proposent une combinaison par vote pondéré des hypothèses obtenues par les différentes méthodes présentées dans l'étude. Cette combinaison est réalisée par le system ROVER (*Recognizer output voting error reduction*) connu pour avoir montré de bonnes performances en reconnaissance de la parole [Fiscus, 1997a]. Les poids des systèmes sont optimisés sur le corpus DEV avec la méthode de Powell [Powell, 1978]. Cette combinaison apporte une performance encore meilleure que celle obtenue par les CRF état de l'art.

## 3.3 Conclusion

Ce troisième chapitre a donné une présentation de modèles classiques d'apprentissage automatique supervisé dans le domaine de la compréhension de la parole et permettant d'accomplir un étiquetage en concepts sémantiques décrit dans le premier chapitre.

Les systèmes à base de CRF sont les plus performants jusqu'alors sur cette tâche et ils constituent le système état de l'art au commencement de cette thèse. En effet, cette thèse débute en 2015, à un moment où les réseaux de neurones ont fait leurs preuves dans beaucoup de tâches de traitement du langage naturel mais n'ont pas encore été beaucoup employés en compréhension de la parole.

Cette thèse qui a pour but l'étude des réseaux de neurones pour la compréhension de la parole gardera donc les CRF comme référence future au cours de ses contributions.

CHAPITRE 4

# MODÈLES NEURONAUX

**Sommaire**

Ce quatrième chapitre présente les modèles fondés sur les réseaux de neurones artificiels, connaissant un essor grandissant aujourd'hui et faisant l'objet principal de notre étude.

Cette thèse, débutée en 2015, s'inscrit dans l'émergence de l'apprentissage profond (*deep learning*) du début des années 2010 dans de nombreux domaines comme :

— La reconnaissance et la classification d'image [Tompson *et al.*, 2014, Zbontar et LeCun, 2015].
— La modélisation acoustique [Hinton *et al.*, 2012, Jaitly *et al.*, 2012, Abdel-Hamid *et al.*, 2012].
— La modélisation du langage [Mikolov *et al.*, 2011]
— La traduction automatique [Bahdanau *et al.*, 2014, Cho *et al.*, 2014a].

Parmi ces domaines, figure également la compréhension de la parole avec [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015], des travaux qui nous servirent de base d'étude préliminaire.

Cette thèse a pour but l'étude des réseaux de neurones en compréhension de la parole appliquée à la tâche d'étiquetage en concepts sémantiques (section 1.1.2.4) en gardant comme comparaison les CRF, un système état de l'art sur cette tâche jusque-là (section 3.2.2).

Comme dans le chapitre précédent, nous apportons une présentation théorique des modèles neuronaux dans un premier temps. Dans un second temps, nous regardons leur application concrète avec une présentation de résultats obtenus en compréhension de la parole.

## 4.1 Description théorique

### 4.1.1 Neurones et réseaux de neurones

Les réseaux de neurones (*Neural Networks* - NN) sont une méthode largement répandue de nos jours et obtenant des résultats performants dans de nombreux domaines, en classification supervisée ou non supervisée. Bien que remontant aux années 1950 [Rosenblatt, 1957, Rosenblatt, 1958] (appelés alors *perceptron* et composés d'un seul neurone), les NN se sont développés un peu plus tard suite à la mise en place de nouveaux types de NN [Hopfield, 1982], et de nouvelles méthodes d'apprentissage [LeCun, 1985, Rumelhart *et al.*, 1985, Rumelhart *et al.*, 1988]. L'apprentissage profond a continué d'être perfectionné par la suite [Hinton *et al.*, 2006, Salakhutdinov et Hinton, 2009], mais il a surtout révélé son potentiel grâce à la mise à disposition d'outils de calcul puissants (tels que les processeurs graphiques) permettant d'exploiter le potentiel des NN.

Un neurone *artificiel* (ou neurone *formel*) s'inspire d'un neurone *biologique* auquel il donne une inspiration mathématique comme le montre la figure 4.1.

Dans un neurone formel, on observe :

— des entrées ($X = x_1...x_n$) auxquelles sont associés des poids ($W = w_1...w_n$) relatifs à l'importance de l'information véhiculée

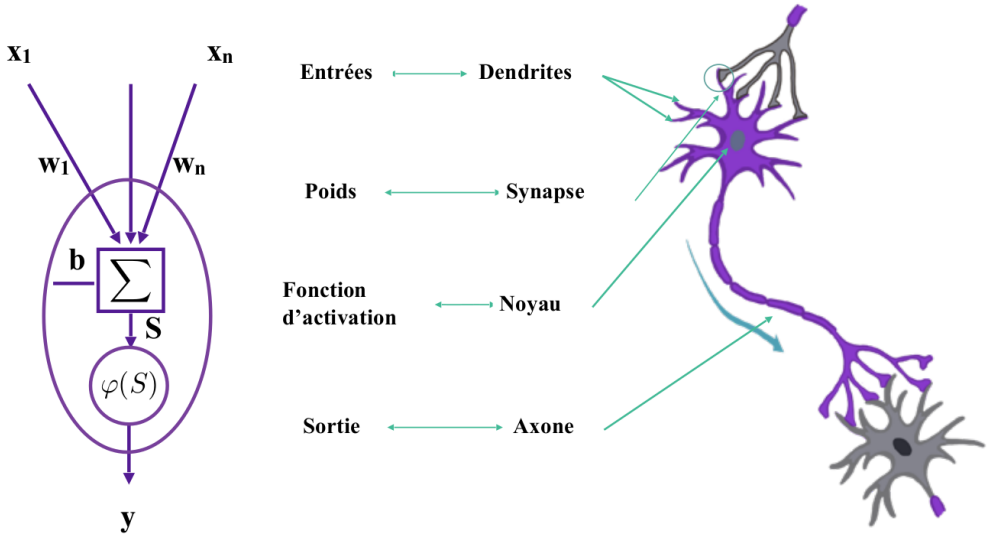



FIGURE 4.1 – Comparaison entre un neurone biologique et un neurone formel [Ghannay, 2017].

- un biais ($b$) constituant le poids d'une entrée constante permettant d'ajouter de la flexibilité au réseau en agissant sur la position de la frontière de décision [Rosenblatt, 1957].
- une fonction d'activation ($\varphi$) appliquée sur les entrées et le biais comme par exemple :
  - Sigmoïde : $\varphi(Z) = \dfrac{1}{1+e^{-Z}}$
  - Tangente hyperbolique ($tanh$) : $\varphi(Z) = \dfrac{1-e^{-Z}}{1+e^{-Z}}$
  - ReLu (*Rectified linear unit*) : $\varphi(Z) = max(0, Z)$
  - Identité (annulation de l'activation) : $\varphi(Z) = Z$
- une sortie ($y$) pouvant être utilisée comme l'entrée d'autres neurones telle que :

$$y = \varphi(W.X + b)$$

Un NN est alors l'association de plusieurs neurones groupés en couches reliées par des connexions pondérées comme le montre la figure 4.2 (exemple appliqué à la production d'étiquettes à partir de mots).

L'architecture d'un NN détermine la manière dont les neurones sont ordonnés et connectés au sein d'un même réseau. En général, un NN est composé de plusieurs couches de neurones successives : les entrées ($x$), les couches cachées ($h$, qui ne sont pas accessibles en dehors du réseau) jusqu'à la couche de sortie ($s$). La profondeur du NN est déterminée par le nombre de couches cachées.

Une des architectures les plus simples (que représente la figure 4.2)

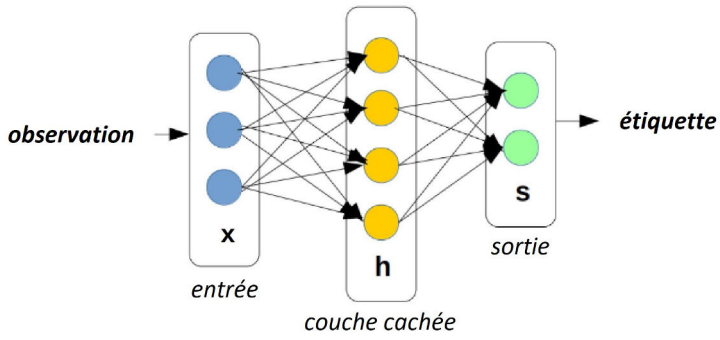



FIGURE 4.2 – Schéma d'un NN.

est le NN à propagation avant (*feed forward*) ou perceptron multi-couches [Minsky et Papert, 1969]. Il véhicule l'information de l'entrée vers la sortie. Le NN propage l'information de cette couche d'entrée vers une ou plusieurs couches cachées et enfin dans la couche de sortie donnant les scores de chaque élément de sortie. L'élément au score le plus élevé est celui qui sera retenu. Pour chaque observation de la séquence d'entrée, le système propose un élément de sortie. Cette séquence d'éléments en sortie constitue l'*hypothèse* du système.

Des architectures plus complexes sont présentées dans la section 4.1.3

### 4.1.2 Apprentissage

L'entraînement du NN consiste en l'ajustement de ses paramètres (matrices de poids, biais) pour classer une séquence d'observations en se basant sur un corpus d'apprentissage donnant des exemples d'association entrée/sortie. Plusieurs méthodes d'apprentissage existent. Elles dépendent de l'architecture, des données d'apprentissage, de la tâche à traiter, *etc.*

L'algorithme de rétro-propagation du gradient [Rumelhart *et al.*, 1986] est l'un des algorithmes les plus utilisés. Il s'appuie sur la minimisation d'une fonction de coût dérivable ou d'erreur $E$ par la méthode de descente de gradient. Cette fonction de coût est calculée entre la sortie du NN et la référence désirée. Plusieurs types de coût peuvent être envisagés comme l'erreur moyenne quadratique ou l'entropie croisée [Stemmer *et al.*, 2002]. Cette fonction de coût utilisée pour l'apprentissage ne se montre pas forcément pertinente pour les phases de validation et de test où l'on préférera des métriques reflétant plus l'objectif de notre tâche comme vu dans la section 1.1.5. Ces métriques ne peuvent pas être utilisées comme fonction de coût car elles ne sont pas dérivables.

L'algorithme commence par une *propagation avant* au cours de laquelle les valeurs de sortie du NN sont calculées (utilisant la matrice de poids $W$) et ainsi l'erreur entre les valeurs de sortie estimées et les valeurs désirées. On effectue ensuite la *propagation arrière* qui consiste à rétro-propager la dérivée partielle de l'erreur ($\frac{\partial E}{\partial W}$) par rapport aux poids du réseau. Enfin, les poids ($W'$) sont mis à jour en fonction

de cette dérivée partielle selon la formule :

$$W' = W - \lambda \Delta W \ , \ \lambda > 0$$

$$\Delta W = \frac{\partial E}{\partial W}$$

avec $\lambda$ le taux d'apprentissage (*learning rate*). Le taux d'apprentissage détermine la variabilité des poids entre deux mises à jour. Un taux faible équivaut à des variations faibles rendant l'apprentissage lent mais qui peut garantir une certaine stabilité. Ce paramètre d'ajustement du NN est à fixer. Il fait partie des hyper-paramètres de l'apprentissage comme nous le verrons plus loin. Certaines méthodes existent pour faire varier ce taux de façon optimale au cours de l'apprentissage (AdaDelta (Adaptive learning rate) [Bottou, 2010, Bergstra et Bengio, 2012, Zeiler, 2012, Schaul *et al.*, 2013]).

Trois variantes d'apprentissage sont utilisées en général :

**La rétro-propagation stochastique** met à jour les poids à chaque présentation d'un exemple d'apprentissage.

**la rétro-propagation par lot (*batch*)** met à jour les poids selon la moyenne des gradients sur tous les exemples d'apprentissage qui forment un lot (plus rapide que pour chaque élément). La moyenne diminue le risque de sur-apprentissage.

**la rétro-propagation par mini-lot (*mini-batch*)** est la plus utilisée car elle combine les deux variantes précédentes en utilisant plusieurs exemples sans pour autant en prendre la totalité.

Il est important que les exemples soient mélangés. En effet avec beaucoup d'exemples consécutifs de la même classe, la convergence risque d'être lente. Pour cela, on permute aléatoirement les exemples afin d'éliminer les dépendances entre les exemples successifs. Avec certaines architectures qui captent des dépendances temporelles (signaux, musique, parole, séries chrono, vidéo) on n'a néanmoins pas le choix que de présenter des séquences dont les éléments sont fortement dépendants, mais on peut toujours mélanger les séquences. Dans ce cas, l'ensemble d'apprentissage est une suite de séquences.

L'apprentissage consiste en plusieurs phases appelées *époques* elles-mêmes constituées d'*itérations*. Une itération est l'application de l'algorithme d'apprentissage sur un exemple/lot/mini-lot : les paramètres du système sont mis à jour à chaque itération. Une époque en est l'application sur tous les exemples d'apprentissage. Des époques sont exécutées pour faire diminuer l'erreur $E$. Le nombre d'époques est un autre hyper-paramètre à fixer. Un grand nombre d'époques a l'avantage de laisser le système apprendre plus longtemps et de potentiellement s'améliorer davantage. En revanche cela entraîne une augmentation du temps et du coût d'exécution. Il y a également un risque de sur-apprentissage. Une méthode classique consiste à fixer une *patience* en nombre d'époques. Si le système ne s'est pas amélioré en un nombre d'époques inférieur à sa patience, il s'arrêtera. S'il trouve une amélioration, il réinitialise son compteur de patience.

Le paramétrage retenu du NN est celui qui aura obtenu la meilleure évaluation sur l'ensemble de données de développement DEV. La disposition d'un corpus de validation pendant l'apprentissage est donc requise pour les NN (contrairement aux CRF). En effet, certains types de systèmes de compréhension (tels que les réseaux de neurones) ont besoin de procéder à des étapes de validation *pendant* leur apprentissage durant lesquelles ils procèdent à une auto-évaluation de leur performance. La calibration finalement conservée du système sera celle ayant obtenue la meilleure validation au cours de l'apprentissage.

En conclusion, de nombreux hyper-paramètres sont à optimiser dans un NN pour améliorer son apprentissage : taux d'apprentissage, taille de mini-batch, taille et nombre de couches cachées, nombre d'époques, *etc.* Cette optimisation des hyper-paramètres nécessite d'exécuter de nombreuses expériences afin de tester les différentes combinaisons possibles entraînant une forte complexité en termes de temps d'exécution. Cela fait partie des difficultés liées à la manipulation des réseaux de neurones.

### 4.1.3 Architectures

Il existe de nombreuses architectures neuronales allant au-delà d'un réseau à propagation avant. Cette section présente les architectures utilisées dans cette thèse : les réseaux de neurones récurrents, bidirectionnels, encodeur-décodeur avec mécanisme d'attention. Il existe de nombreux autre types d'architectures non abordées dans cette section telles que les réseaux de neurones convolutifs (Convolutional Neural Network - CNN) [LeCun *et al.*, 1990], le *Deep Belief Network* [Hinton *et al.*, 2006] ou encore le *Restricted Boltzmann Machine* [Salakhutdinov et Hinton, 2009].

#### 4.1.3.1 Récurrence Avant/Arrière

Cette section définit comment introduire de la récurrence dans un NN. Un réseau de neurones récurrent (*Recurrent Neural Network* - RNN) réutilise l'information des autres éléments de la séquence d'observation en entrée en considérant la séquence comme un tout lié [Medsker et Jain, 1999, Mesnil *et al.*, 2013]. Ceci est fait dans le but de découvrir des dépendances longue distance au sein de de la séquence.

**Principe** La récurrence consiste à injecter lors de l'étape courante, une information extraite lors d'une autre étape (suivante/précédente). La question est de déterminer où l'information est injectée et où elle est extraite. La récurrence de type *elman* [Elman, 1990] consiste à extraire l'information de la couche cachée (lorsqu'il n'y a qu'une couche) d'une autre étape et à la réinjecter dans la couche cachée courante. Une variante appelée *jordan* [Jordan, 1997] consiste cette fois à extraire l'information de la couche de sortie d'une autre étape et à la réinjecter dans la couche cachée courante. Il est possible de combiner elman et jordan dans une variante hybride en extrayant à la fois couche caché et couche de sortie d'une autre

étape pour la réinjecter dans la couche cachée courante. Ces variantes sont résumées dans la figure 4.3.

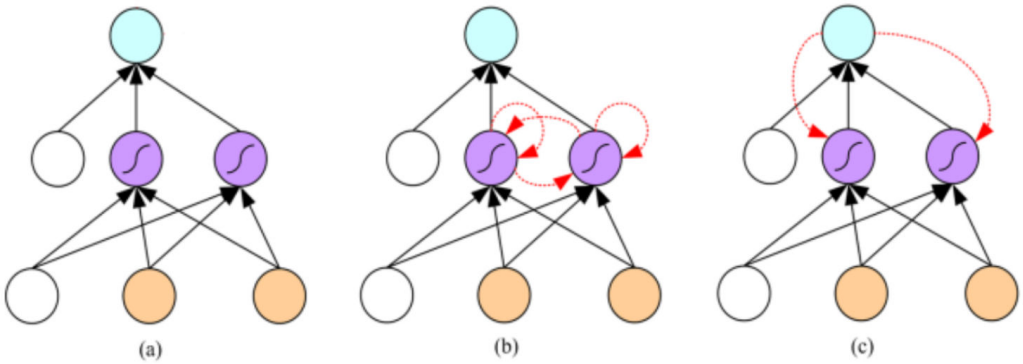



FIGURE 4.3 – Schémas de NN à propagation avant (a), RNN elman (b) et RNN jordan (c) [Mesnil *et al.*, 2015].

Nous nous référons à la variante elman dans la suite.

**Avant** La récurrence avant (*forward*) consiste à récupérer de l'information des couches cachées précédentes en réinjectant la couche cachée de l'étape précédente dans la couche cachée courante. Les neurones des couches cachées gardent de l'information contextuelle des couches cachées précédentes. Ainsi en *avançant* dans la séquence d'observations, on bénéficie d'informations du *passé* de la séquence rendant le RNN capable d'effectuer des prédictions au-delà des capacités d'un simple NN à propagation avant.

La figure 4.4 en donne un exemple.

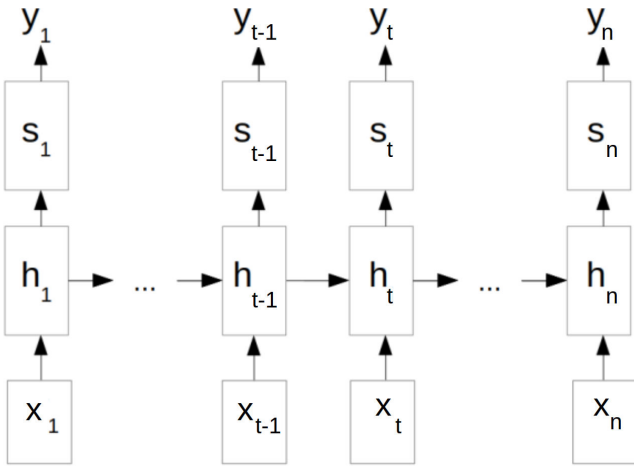



FIGURE 4.4 – Schéma d'un RNN avant.

La récurrence avant est définie comme suit :

$$couche\ cachee : h(t) = sigmoid(W_x.x(t) + W_h.h(t-1) + b_h))$$

$$couche\ de\ sortie : s(t) = softmax(W.h(t) + b))$$

où $x(t)$ est l'élément d'entrée du RNN à l'instant $t$ et $h(t-1)$ la sortie de la couche cachée à $t-1$. Les paramètres $W_x$, $W_h$ et $W$ du RNN sont les matrices de poids respectivement entre la couche d'entrée et la couche cachée courante, entre la couche cachée précédente et la couche cachée courante, et entre la couche cachée courante et la couche de sortie. Les paramètres $b_h$ et $b$ sont les biais, et $h_0$ la couche cachée initiale de l'étape précédente pour le premier élément de la séquence pour lequel rien n'a encore été calculé (lorsque $t = 0$, on ne dispose de rien à $t-1$). Ces paramètres sont ajustés au cours des époques d'apprentissage (section 4.1.2).

**Arrière** Un RNN arrière (*backward*) fonctionne de la même manière mais dans l'autre sens : la prédiction est de la fin vers le début de la séquence d'observations, soit du futur vers le passé. On réinjecte la couche cachée suivante dans la courante pour récupérer l'information du futur de la séquence en reculant dedans. Cela revient à appliquer un RNN avant sur la phrase à l'envers.

La récurrence arrière est définie comme suit :

$$h(t) = sigmoid(W_x.x(t) + W_h.h(t+1) + b_h))$$

$W_h$ représente la matrice de poids entre la couche cachée de l'étape prochaine et la courante. $h_0$ est la couche cachée initiale de l'étape suivante pour le dernier élément de la phrase *i.e.* le premier élément donné au RNN (lorsque $t = n$, on ne dispose de rien à $t+1$). La couche de sortie est toujours calculée de la même manière.

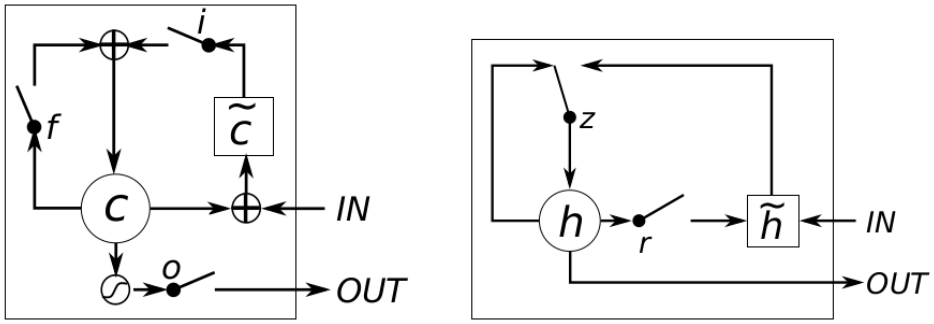
**Limites** Un RNN peut rencontrer des difficultés pour capturer des dépendances à long terme à cause de la variation exponentielle de l'erreur $E$. L'influence d'une entrée sur les couches cachées varie exponentiellement en passant par les connexions récurrentes. En effet, l'erreur locale à un instant $t$ rétro-propagée dans le temps s'exprime de manière récursive en fonction des erreurs rétro-propagées aux instants passés [Bengio *et al.*, 1994].

Pour résoudre ce problème, une solution consiste à remplacer l'unité récurrente classique par une unité récurrente utilisant des *portes*. Ces portes sont des fonctions d'activation modulant le flux d'information dans l'unité. On en distingue deux types :

— Une unité récurrente à mémoire à court et long termes (*Long-Short Term Memory* - LSTM) [Hochreiter et Schmidhuber, 1997, Graves *et al.*, 2013]. Cette unité est composée d'une mémoire $c$ (et $\tilde{c}$ le nouveau contenu de la mémoire) et de trois portes. L'entrée $i$ (*input*) choisit les informations pertinentes transmises à la mémoire. La sortie $o$ (*output*) protège le réseau du contenu de sa mémoire. L'oubli $f$ (*forget*) permet à l'unité de remettre à zéro le contenu de sa mémoire.

— Une unité récurrente à portes (*Gated Recurrent Units* - GRU) [Cho *et al.*, 2014b]. Cette unité n'est composée que de deux portes. La réinitialisation $r$ (*reset*) décide si l'état précédent de l'unité est ignoré ou non. La porte de modification $z$ permet de décider si l'état caché $h$ doit être mis à jour avec le nouvel état caché $\tilde{h}$ ou non.

La figure 4.5 illustre ces deux types d'unités.




(a) Long Short-Term Memory (b) Gated Recurrent Unit

FIGURE 4.5 – Illustration des unités LSTM (a) et GRU (b). LSTM : $c$ et $\tilde{c}$ sont la mémoire et le nouveau contenu de la mémoire. GRU : $h$ et $\tilde{h}$ sont l'activation et l'activation candidate [Cho *et al.*, 2014b].

#### 4.1.3.2 Récurrence Bidirectionnelle

Un RNN bidirectionnel (biRNN) effectue des prédictions prenant en compte à la fois les informations provenant du passé et du futur [Schuster et Paliwal, 1997]. Il s'agit ainsi de la combinaison des RNN avant et arrière. En effet, des RNN avant et arrière déjà entraînés sont utilisés conjointement, comme le montre la figure 4.6.

Il y a deux matrices de poids $W_h$ : la première $W_{hFW}$ entre la couche cachée de l'étape précédente et la courante ; et la seconde $W_{hBW}$ entre la couche cachée de l'étape suivante et la courante. Il en va de même pour les biais $b_{hFW}$ et $b_{hBW}$. Finalement il n'y a pas de couches cachées initiales $h_0$ étant donné que ces dernières sont récupérées depuis les RNN avant et arrière déjà entraînés.

La couche cachée est alors calculée comme suit (dans un réseau de type elman [1]) :

$$h_{BD}(t) = sigmoid(W_x.x(t)+W_{hFW}.h_{FW}(t-1)+b_{hFW}+W_{hBW}.h_{BW}(t+1)+b_{hBW})$$

Il est aussi possible d'envisager des dépendances à plus long terme en fournissant au réseau la somme des étapes précédentes/suivantes, c'est-à-dire avec un nombre $T$ d'étapes temporelles supérieur à 1 selon l'équation :

$$h_{BD}(t) = sigmoid(W_x.x(t) + T_{FW} + T_{BW})$$

1. l'opération serait la même pour un jordan excepté que la couche de sortie est réinjectée dans la couche cachée à $t+1$.

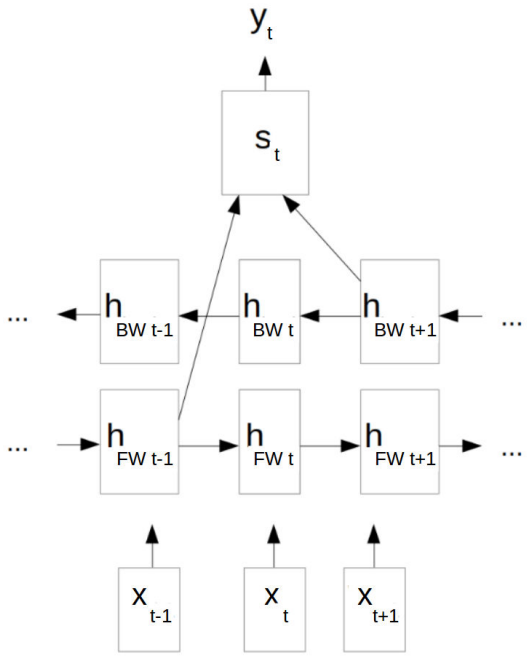



FIGURE 4.6 – Schéma d'un biRNN.

avec :

$$T_{FW} = \sum_{k=1}^{T} (W_{hFW,k}.h_{BW}(t-k) + b_{hFW})$$

$$T_{BW} = \sum_{k=1}^{T} (W_{hBW,k}.h_{FW}(t+k) + b_{hBW})$$

A chaque position de la séquence, on dispose des informations venant du futur et du passé. Cela donne au biRNN une vision globale supérieure comparée aux RNN avant et arrière seuls [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015, Vukotic *et al.*, 2015].

#### 4.1.3.3 Auto-encodeur

Un encodeur-décodeur classique (ou auto-encodeur, voir figure 4.7) ne cherche pas à passer d'une représentation en entrée à une représentation différente en sortie, comme les systèmes que nous venons de voir.

Ses prédictions consistent à reproduire ses propres entrées [Rumelhart *et al.*, 1985]. L'intérêt de ce type d'architecture se trouve dans les représentations latentes apprises par l'auto-encodeur au niveau de la couche cachée, capables de capturer suffisamment d'informations pour reconstruire les entrées. Ces représentations peuvent être alors utilisées comme pré-entraînement de réseaux de neurones profonds [Bengio, 2009, Vincent *et al.*, 2010]. Initialiser les poids d'un réseau à l'aide d'un auto-encodeur peut aider à accélérer l'apprentissage par rapport à une simple initialisation aléatoire.

Un auto-encodeur est constitué de deux modules :

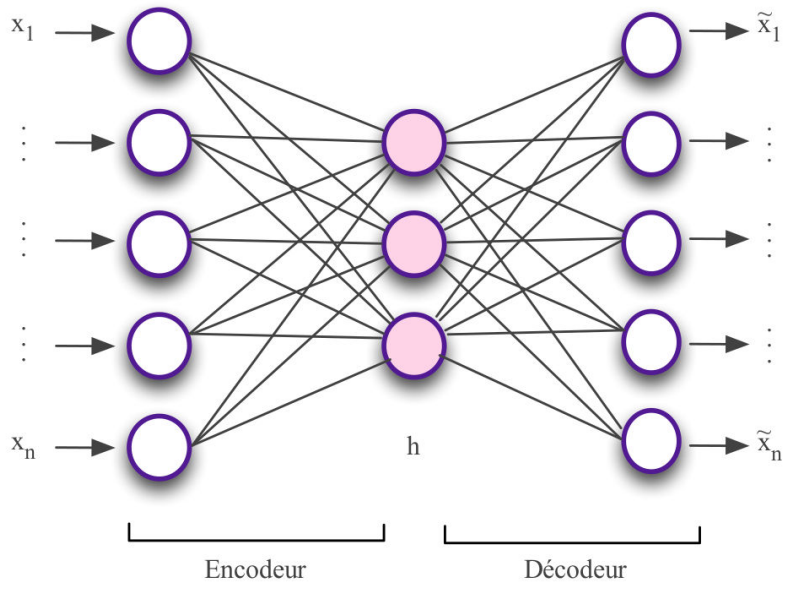



FIGURE 4.7 – Schéma d'un auto-encodeur [Ghannay, 2017].

— Un **encodeur** projetant l'entrée $X$ dans un espace de faible dimension tel que :

$$h = f_e(W_e.X + b_e)$$

avec $W_e$ la matrice de poids, $b_e$ le biais et $f_e$ une fonction d'activation. La couche de projection $h$ est susceptible de conserver les informations utiles de $X$. Dans le cas où la taille de $h$ est inférieure à celle de $X$, l'auto-encodeur applique une compression des entrées pour réduire la redondance dans le but d'apprendre une représentation compacte conservant l'information la plus pertinente. Dans le cas où la taille de $h$ est supérieure, le réseau apprend une représentation sur-complète mais qui peut tout de même apprendre quelque chose d'utile en rapport avec la distribution des données [Bengio *et al.*, 2007].

— Un **décodeur** qui reconstruit l'entrée $X$ à partir de la représentation $h$ selon :

$$\widetilde{X} = f_d(W_d.h + b_d)$$

Tous les paramètres sont optimisés pour réduire l'erreur de reconstruction. Le but du décodage est de vérifier si l'encodeur a capturé l'information utile contenue dans les données fournies en entrée. Le choix des fonctions de reconstruction $f_d$ et de coût $E$ dépend de la tâche et de la distribution des données [Memisevic, 2011, Rudy et Taylor, 2014].

Un auto-encodeur débruitant (*denoising*) est un cas particulier consistant à apprendre une représentation robuste d'une entrée partiellement corrompue. L'objectif est de rendre l'auto-encodeur capable de capter des informations suffisantes à la reconstruction de l'entrée malgré sa corruption [Vincent *et al.*, 2008].

#### 4.1.3.4 Encodeur-décodeur

**Principe** Dans le cadre d'un système de séquence à sequence, un système encodeur-décodeur peut permettre d'aller plus loin que la récurrence seule. Si un RNN permet à une position donnée de la séquence d'avoir une connaissance globale du passé et/ou du futur, la connaissance reste néanmoins concentrée autour de l'élément. Or, une hypothèse est qu'il est parfois important pour classer certains éléments de disposer de connaissances utiles présentes à d'autres emplacements de la séquence d'observations [Cho *et al.*, 2014a].

Étant donné que les couches cachées des RNN tendent à mieux représenter les entrées récentes, les informations dont dispose un RNN à la position $i$ se concentrent sur les éléments autour de $x_i$, et si d'autres informations peuvent se montrer utiles ailleurs dans la séquence, ces informations seront moins bien représentées, surtout dans de grandes séquences.

Un RNN, basé sur un GRU, remplit le rôle d'encodeur. Supposons ce RNN bidirectionnel. Il calcule une annotation $h_i$ pour chaque élément $x_i$ de la séquence d'entrée $\{x_1, \ldots, x_n\}$. Cette annotation est la concaténation des couches cachées correspondantes avant et arrière obtenues respectivement par le RNN avant et le RNN arrière constituant le biRNN. Chaque annotation $h_i$ contient le résumé à la fois des éléments précédents et des éléments suivants se concentrant autour de $x_i$.

Après avoir appliqué cet encodeur bidirectionnel, pour chaque élément à l'intérieur de la séquence d'entrée, la séquence d'annotations $\{h_1, \ldots, h_n\}$ est utilisée pour calculer un vecteur $c$. Ce calcul prend en compte une somme pondérée de toutes les annotations calculées par l'encodeur. Dans ce cas les poids sont figés. Le vecteur $c$ contient alors une représentation utile pondérée et de dimension fixe de l'information contenue dans toute la séquence d'observations, quelle que soit sa taille. Il sera utilisé par le décodeur afin de calculer la sortie.

La figure 4.8 illustre un RNN avant encodeur-décodeur.

**Mécanisme d'attention** Le mécanisme d'attention est une amélioration apportée au biRNN encodeur-décodeur précédent. Cela permet de constituer une structure biRNN encodeur-décodeur avec mécanisme d'attention (biRNN-EDA).

Le mécanisme d'attention fut intuitivement conçu afin de prendre en compte la position des éléments dans la séquence d'observations en entrée en utilisant une approche encodeur-décodeur à l'aide d'un biRNN. Il consiste à donner plus ou moins de poids à certains éléments en fonction de leur importance pour classer l'élément courant. Pour cela, il attribue des poids à chaque élément d'entrée. Ces poids sont ré-estimés après chaque génération d'une sortie. Cela permet au décodeur de décider à quelles parties de la séquence d'entrée il doit prêter attention et de ne pas avoir à prendre automatiquement en considération toute l'information. L'architecture biRNN-EDA est décrite dans la figure 4.9.

Après avoir appliqué cet encodeur bidirectionnel, pour chaque élément à l'intérieur de la séquence d'entrée, la séquence d'annotations $\{h_1, \ldots, h_n\}$ est utilisée par le décodeur pour calculer un vecteur de contexte $c_t$ (représenté par une croix

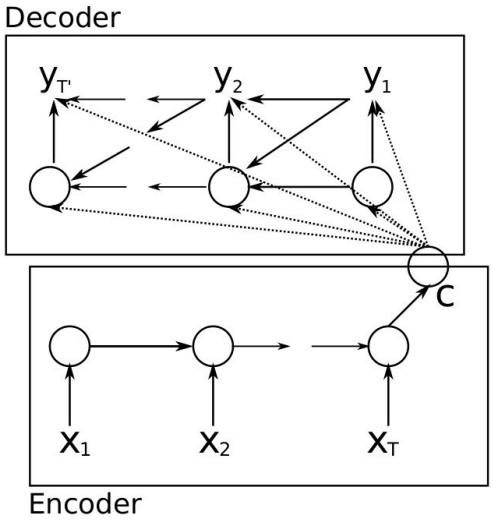



FIGURE 4.8 – Illustration d'un RNN avant encodeur-décodeur de [Cho *et al.*, 2014a].

dans la figure 4.9). Ce calcul prend en compte une somme pondérée de toutes les annotations calculées par l'encodeur :

$$c_t = \sum_{i=1}^{n} \alpha_{ti} h_i$$

avec $\alpha_{ti}$ le poids (ou l'attention) accordé à l'élément $i$ lorsque que l'on calcule la sortie de l'élément à $t$, et $i$ parcourant les entrées.

Le vecteur de contexte contient donc pour un élément observé à $t$, une représentation utile de l'information contenue dans toute la séquence d'observations prenant en compte l'importance plus où moins grande de chaque élément. Un vecteur de contexte est recalculé après chaque émission d'une étiquette en sortie.

Cette pondération dépend de la cible en sortie courante et constitue le cœur du mécanisme d'attention : une bonne estimation des poids $\alpha_{ti}$ permet au décodeur de choisir les parties de la séquence d'entrée auxquelles il doit prêter attention (avec $a$ une fonction d'alignement évaluant la correspondance de l'entrée autour de $i$ et de la sortie à $t$ [Bahdanau *et al.*, 2014]) :

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{n} exp(e_{tk})} \quad ; \quad e_{ti} = a(s_{t-1}, h_i)$$

Le vecteur de contexte sera utilisé par le décodeur conjointement avec l'étiquette émise en sortie précédemment $y_{t-1}$ et l'état précédent $s_{t-1}$ de la couche de sortie du RNN afin de calculer la couche de sortie courante $s_t$ :

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

avec $f$ une fonction linéaire telle qu'un réseau LSTM [Sutskever *et al.*, 2014].

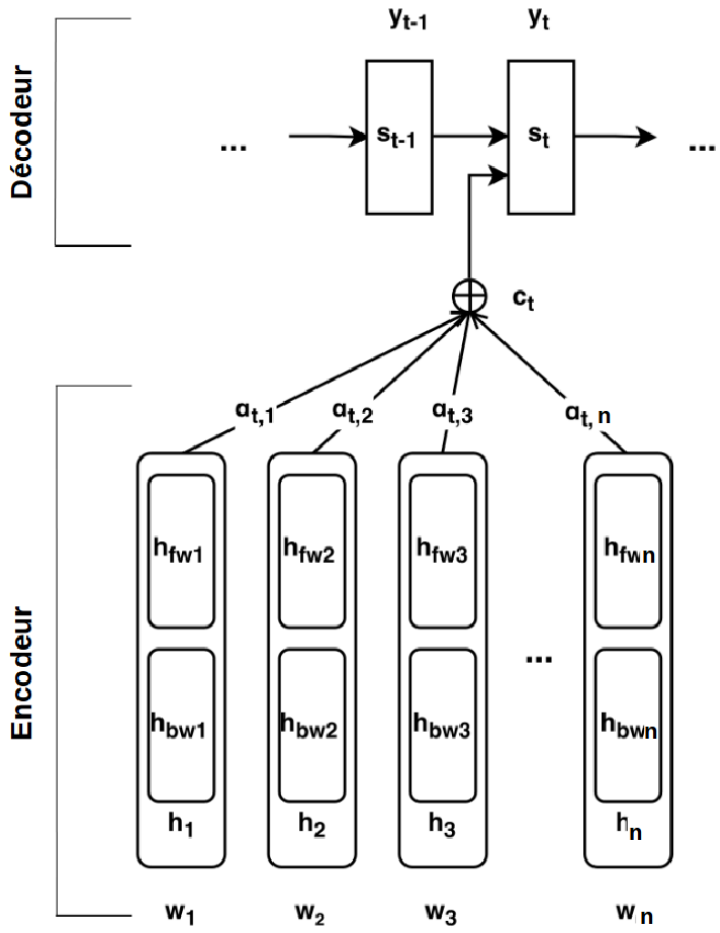



FIGURE 4.9 – Illustration d'un biRNN encodeur-décodeur avec mécanisme d'attention de [Bahdanau *et al.*, 2014].

## 4.2 Implémentation en compréhension de la parole

### 4.2.1 Représentation de l'entrée

En compréhension de la parole, les observations en entrée sont des mots. Les réseaux de neurones de leur coté prennent des vecteurs en entrée. Ces vecteurs peuvent être faits de valeurs continues.

Afin de représenter les mots par des vecteurs, qui pourront être acceptés par les NN, une solution intuitive est la représentation *one-hot*. Son principe est d'associer à chaque élément d'un vocabulaire un vecteur de la taille du vocabulaire où chaque composante est à zéro sauf celle qui indexe l'élément à représenter. Ce type de représentation a les inconvénients d'être volumineuse (plus le vocabulaire est grand) et de ne pas être riche d'informations supplémentaires (telles que des informations sémantiques et syntaxiques).

Les plongements de mots (*word embeddings*) sont une méthode récente et efficace pour la représentation des mots dans un système neuronal. Les plongements sont une projection des mots du vocabulaire dans un espace de faible dimension de manière à préserver les similarités sémantiques et syntaxiques. Le mot est représenté par un vecteur de valeurs réelles dense et de faible dimension. Chaque dimension représente une caractéristique latente du mot, qui peut capter des propriétés syntaxiques et sémantiques [Ghannay, 2017]. Cette représentation plus intéressante sera celle employée dans les NN. Les plongements de mots utilisés peuvent être pré-calculés par un autre système neuronal ou bien initialisés aléatoirement puis affinés au cours de l'apprentissage.

Les plongements de mots ont été introduits à travers la construction de modèles de langage neuronaux [Bengio *et al.*, 2003, Schwenk *et al.*, 2006, Schwenk, 2013]. Ils ont été utilisés avec succès en tant qu'informations supplémentaires dans plusieurs tâches liées au traitement du langage : l'étiquetage morpho-syntaxique, le regroupement en syntagme, la reconnaissance d'entités nommées, la détection de mention [Collobert *et al.*, 2011, Turian *et al.*, 2010, Bansal *et al.*, 2014] et en compréhension de la parole [Mesnil *et al.*, 2013, Yao *et al.*, 2014, Mesnil *et al.*, 2015, Liu et Lane, 2016].

De nombreuses méthodes neuronales existent pour la construction de plongements de mots parmi lesquelles : les modèles de langage neuronaux [Bengio *et al.*, 2003, Schwenk, 2007], *C&W* [Collobert et Weston, 2008], *Word2vec, CBOW, Skip-gram* [Mikolov *et al.*, 2013a, Mikolov *et al.*, 2013b], *Word2vec-deps* [Levy et Goldberg, 2014] ou encore *GloVe* [Pennington *et al.*, 2014].

Il est possible de visualiser les plongements de mots avec une technique non linéaire de réduction de dimensions permettant de projeter des données à haute dimension dans un espace de deux ou trois dimensions. On peut alors observer les plongements de mots sous la forme d'un nuage de points comme le montre la figure 4.10 : il est possible d'y voir se former des îlots de mots portant sur le même type d'information.

Afin de pouvoir comparer des plongements, une méthode consiste à utiliser le calcul de la similarité cosinus pour rechercher les mots les plus proches (similaires) d'un plongement donné. De cette façon, si les vecteurs de mots sont proches les uns des autres en termes de distance, alors ils doivent être sémantiquement et/ou syntaxiquement proches.

### 4.2.2 Utilisation de modèles neuronaux en compréhension de la parole

Comme vu dans la section précédente, des architectures de plus en plus complexes ont été introduites afin d'améliorer les performances de classification neuronale, en essayant de capturer de l'information utile dans différentes parties de la séquence d'entrée. Ces architectures neuronales se sont répandues en compréhension de la parole.

Dans [Mesnil *et al.*, 2013], les auteurs évaluent les performances de RNN avant

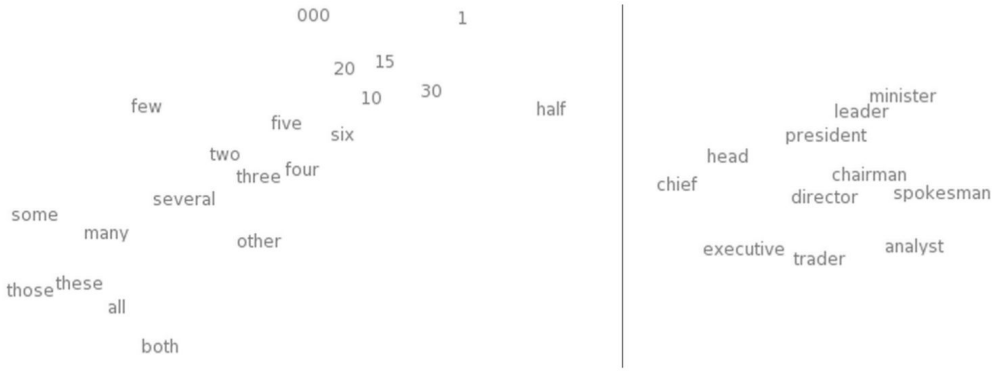



FIGURE 4.10 – Visualisation en deux dimensions de plongements de mots [Turian *et al.*, 2010]. A gauche : des mots portant une information numérique. A droite : des mots portant une information sur l'emploi.

et arrière sur le corpus ATIS et obtiennent d'encore meilleurs résultats par l'utilisation d'un biRNN permettant au système d'avoir une vision sur le passé et le futur de la phrase à chaque position. Dans [Mesnil *et al.*, 2015], les mêmes auteurs approfondissent leur étude avec des réseaux hybrides elman/jordan.

Les auteurs de [Dinarelli et Tellier, 2016, Dinarelli *et al.*, 2017] dépassent les architectures RNN elman et jordan en établissant une nouvelle façon de réinjecter l'information en sortie dans le système récurrent comme le montre la figure 4.11. Ils obtiennent ainsi des améliorations sur le corpus MEDIA.

Des biRNN LSTM sont utilisés dans [Hakkani-Tür *et al.*, 2016] pour de l'étiquetage de cadres sémantiques (section 1.1.2.4).

Dans [Serdyuk *et al.*, 2018], les auteurs développent un système de bout en bout (*end to end*) avec la combinaison de systèmes neuronaux de reconnaissance automatique de la parole et de compréhension. Ce système neuronal complet s'affranchit donc de la représentation textuelle classique entre la sortie du module de reconnaissance et l'entrée de celui de compréhension.

D'autres améliorations peuvent être faites en se focalisant sur la tâche, en essayant de la simplifier. Dans [Hinton *et al.*, 2015], un ensemble de modèles neuronaux spécialistes apprennent à distinguer des classes affinées que le modèle complet à tendance à confondre.

Au commencement de cette thèse (2015), le mécanisme d'attention n'est pas employé pour la compréhension de la parole. Initialement dédié à la reconnaissance d'écriture manuelle [Graves, 2013], il est utilisé avec succès pour la reconnaissance de la parole [Chorowski *et al.*, 2014, Chorowski *et al.*, 2015]. Le mécanisme d'attention est également répandu en traduction automatique [Bahdanau *et al.*, 2014, Cho *et al.*, 2014a]. Suivant [Ma *et al.*, 2015], dans [Chen *et al.*, 2016] un CNN est proposé pour encoder la représentation de la connaissance exprimée dans une phrase

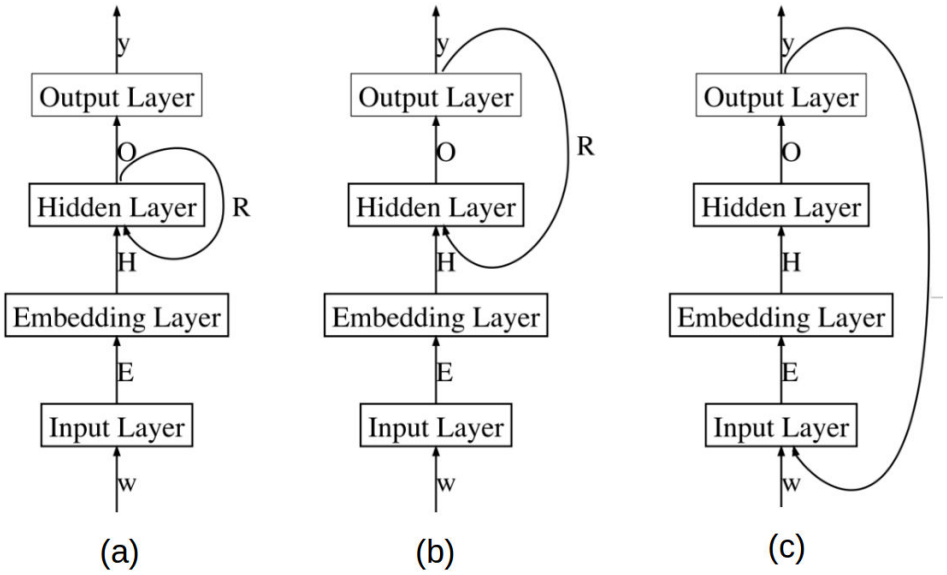



FIGURE 4.11 – Nouvelle architecture proposée par [Dinarelli et Tellier, 2016, Dinarelli *et al.*, 2017] (a : elman, b : jordan, c : nouvelle approche).

parlée. Cet encodage est utilisé comme un mécanisme d'attention pour limiter la génération d'étiquettes exprimées dans la même phrase. Dans [Reddy *et al.*, 2016], des LSTM ont été proposés avec un mécanisme d'attention pour faire l'analyse de phrases en langage naturel vers des représentations logiques. Enfin, dans [Liu et Lane, 2016], le mécanisme d'attention est employé pour la tâche de détection d'intention.

Comme mentionné auparavant, cette thèse, débutée en 2015, s'inscrit dans l'émergence de l'apprentissage profond parmi de nombreux domaines dont la compréhension de la parole avec [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015]. Nous nous référons donc à cette étude qui sert de base de travail et de référence pour les travaux de cette thèse et en présentons les résultats dans la section suivante.

### 4.2.3 Résultats expérimentaux

Les expériences passées décrites dans [Hahn *et al.*, 2008, Hahn *et al.*, 2011] ont montré que les meilleures performances en annotation sémantique sur les transcriptions manuelles et automatiques du corpus MEDIA (section 1.2.3) ont été obtenues avec les CRF comme le montrent les résultats de la section 3.2.2.

Récemment, [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015] introduit des réseaux de neurones à la tâche d'étiquetage en concepts sémantiques et montre que les RNN peuvent atteindre de meilleures performances que les CRF sur cette tâche appliquée au corpus ATIS (section 1.2.2). Dans ces deux études, les auteurs utilisent des RNN et biRNN dans des versions elman, jordan ou hybride. Ils utilisent également des

| | | Transcriptions | |
|---|---|---|---|
| | | **manuelles** | **automatiques** |
| **Étude** | **Modèle** | **f-mesure** | |
| [Mesnil *et al.*, 2013] | RNN elman forward | 93,65 | - |
| | RNN elman backward | 92,12 | - |
| | RNN jordan forward | 93,77 | - |
| | RNN jordan backward | 93,31 | - |
| | biRNN jordan | **93,98** | - |
| [Mesnil *et al.*, 2015] | RNN hybride | **95,06** | **84,74** |
| [Mesnil *et al.*, 2013] [Mesnil *et al.*, 2015] | CRF | 93 | 81,15 |

TABLE 4.1 – Comparaison sur ATIS (TEST) entre CRF et RNN [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015].

| **Modèle** | **Corpus** | **f-mesure** |
|---|---|---|
| CRF | ATIS manuel | 95,23 |
| meilleure configuration RNN | | **96,16** |
| CRF | MEDIA manuel | **86** |
| meilleure configuration RNN | | 83,25 |

TABLE 4.2 – Comparaison sur ATIS et MEDIA (TEST) entre CRF et biRNN [Vukotic *et al.*, 2015].

fenêtres contextuelles de mots comme pour les CRF, consistant à passer en entrée le mot avec ceux qui l'entourent. Les résultats sont présentés dans la table 4.1 sur des transcriptions manuelles et automatiques : pour les transcriptions automatiques, les auteurs utilisent un système de reconnaissance atteignant un WER de 13,8 sur le corpus TEST.

Néanmoins, [Vukotic *et al.*, 2015] a démontré que ces meilleures performances des RNN ne se renouvellent pas sur un corpus de compréhension plus complexe tel que le corpus MEDIA. En effet dans cette dernière étude, les CRF ont obtenu des résultats significativement meilleurs que ceux des RNN [2]. Les résultats sont reportés dans le tableau 4.2 [3].

La tâche MEDIA semble plus difficile à traiter que celle d'ATIS. On remarque notamment que la taille du vocabulaire est plus grande dans le corpus MEDIA que dans le corpus ATIS (2460 mots contre 572) pour un nombre de concepts sémantiques proche (75 contre 64). La proportion de mots du corpus étant associés à un

2. Il est à noter que les améliorations de [Dinarelli et Tellier, 2016, Dinarelli *et al.*, 2017] présentées en figure 4.11 ont permis aux RNN de surpasser les CRF sur MEDIA mais cela n'était pas d'actualité au commencement de cette thèse en 2015

3. L'auteur fournit également les temps d'exécution de ses systèmes. Il est intéressant de noter que le temps d'exécution est beaucoup plus faible pour les CRF avec un temps d'apprentissage de l'ordre du quart d'heure contre une à trois heures pour les RNN.

concept est également plus importante chez MEDIA (72% des mots contre 49%). De plus, MEDIA comporte des contenus sémantiques dont les segmentations sont plus difficiles à localiser et désambiguïser que celles d'ATIS. La variabilité est donc plus importante pour MEDIA que pour ATIS, ce qui rend la tâche de compréhension d'autant plus complexe. Les CRF semblent exploiter plus efficacement les contextes complexes. D'une manière générale il a été montré que les limites du corpus ATIS ont été atteintes en termes de qualité d'évaluation de la compréhension de la parole pour les systèmes d'apprentissage profond actuels [Béchet et Raymond, 2018], raison pour laquelle le corpus MEDIA est à préférer pour effectuer des comparaisons pertinentes.

Il est donc difficile d'établir si les CRF ou les RNN constituent l'état de l'art pour la compréhension de la parole et plus précisément pour la tâche d'étiquetage en concepts sémantiques. Les CRF sont au début de cette thèse un système réputé et solide pour cette tâche, mais qui ont atteint un plateau dans leurs performances. Les architectures RNN de leur coté montrent des résultats prometteurs sans surpasser totalement les CRF.

## 4.3 Conclusion

Ce quatrième chapitre a donné une présentation des modèles de classification neuronaux.

La première section en a fait une description théorique en abordant leur principe de fonctionnement, leur mode d'apprentissage et enfin en exposant certaines de leurs architectures. Ces architectures neuronales de plus en plus complexes permettent d'aller à chaque fois plus loin dans leur capacité à analyser leurs données et à déceler des dépendances utiles entre les éléments.

La deuxième section a présenté l'utilisation récente des modèles neuronaux en compréhension de la parole. Les systèmes les plus performants sont les réseaux récurrents hybrides elman/jordan ou bidirectionnels. Ils ont permis d'arriver à la conclusion que les RNN étaient capables de surpasser les modèles CRF, état de l'art jusqu'alors, sur une tâche d'étiquetage en concepts sémantiques. Néanmoins cette conclusion obtenue sur le corpus ATIS ne se renouvelle pas sur le corpus plus complexe MEDIA. Cela montre que les réseaux de neurones, bien qu'étant prometteurs, requièrent une étude approfondie et de nouvelles améliorations sur la tâche de compréhension de la parole.

# Deuxième partie

# Contributions

CHAPITRE 5

# PORTER LE MÉCANISME D'ATTENTION À LA COMPRÉHENSION

**Sommaire**

Ce premier chapitre de contributions concerne l'apport du mécanisme d'attention dans notre tâche de compréhension automatique de la parole.

Comme nous l'avons vu dans la partie état de l'art, cette thèse a pour objet la compréhension de la parole (chapitre 1) plus précisément dans la tâche d'étiquetage en concepts sémantiques. De plus, cette thèse s'inscrit dans l'émergence de l'apprentissage profond dans de nombreux domaines (chapitre 4). L'étude de Grégoire Mesnil [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015] a notamment constitué une base d'étude importante pour les travaux préliminaires de cette thèse. Cette étude décrit l'apport de réseaux de neurones récurrents pour la tâche d'étiquetage en concepts sémantiques sur le corpus ATIS avec une amélioration particulière obtenue par l'utilisation conjointe de récurrences avant et arrière formant un réseau de neurones récurrent bidirectionnel. Cette amélioration a alors permis de surpasser le modèle CRF (chapitre 3) état de l'art jusqu'alors. Un des premiers objectifs de cette thèse a par conséquent été de reproduire et de maîtriser un tel réseau sur le corpus ATIS.

Une deuxième étude importante ayant orienté les travaux de cette thèse est l'étude de Vedran Vukotic [Vukotic *et al.*, 2015] apportant une modération sur les conclusions de l'étude précédente. En effet si un biRNN parvient à surpasser les CRF sur la tâche d'étiquetage en concepts sémantiques sur ATIS, cette conclusion ne se renouvelle pas sur un corpus plus complexe comme le corpus MEDIA. Cela nous amène donc d'une part à nous focaliser sur le corpus MEDIA pour évaluer la suite de nos travaux, la tâche de compréhension MEDIA semblant plus complexe et plus ambitieuse à traiter que celle du corpus ATIS. D'autre part, cela nous amène à chercher des améliorations à apporter au biRNN pour progresser tout en continuant de nous mesurer aux CRF : le mécanisme d'attention est la première de ces améliorations et est par conséquent l'objet principal de ce chapitre. Nous étions en effet les premiers à porter cette technique à la compréhension de la parole en 2015 ([Simonnet *et al.*, 2015], *cf.* publications 8.1).

Le but de cette thèse n'est pas d'établir si les CRF ou les réseaux de neurones constituent l'état de l'art pour la compréhension. Nous sommes convaincus du potentiel des architectures neuronales qui montrent des résultats prometteurs et nous avons pour objectif d'étudier leur utilisation pour la compréhension de la parole.

La première section de ce chapitre définit les motivations liées à notre approche.

La seconde section aborde l'obtention d'un biRNN état de l'art inspiré de celui de [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015] sur le corpus ATIS, puis son adaptation au corpus MEDIA qui nous intéresse plus particulièrement. Nous parlerons également du système CRF de référence que nous utiliserons au cours de cette thèse pour avoir une base de comparaison.

La dernière section décrit les premières contributions de cette thèse, c'est-à-dire l'apport du mécanisme d'attention à la compréhension de la parole. Nous étudions également l'apport de différents descripteurs de mots dans notre système biRNN avec mécanisme d'attention.

Dans l'optique finale de notre application de système de dialogue oral, nous travaillerons sur des sorties de transcriptions automatiques. Néanmoins dans ce

chapitre, les transcriptions manuelles sont utilisées dans un premier temps comme défini dans la section 2.2.

## 5.1 Motivations

Comme défini auparavant, cette thèse s'inscrit dans l'émergence des réseaux de neurones pour les traitements automatiques. L'application des réseaux de neurones pour la compréhension de la parole constitue donc l'originalité de cette thèse, étant donné qu'ils sont assez peu présents dans ce domaine à son début mais plus dans des domaines comme la reconnaissance de la parole, d'images, la traduction automatique, *etc.*

Nos études préliminaires nous ont amené à nous intéresser au mécanisme d'attention, une innovation qui s'annonçait prometteuse pour les systèmes neuronaux, alors appliquée à la reconnaissance de la parole [Chorowski *et al.*, 2014, Chorowski *et al.*, 2015] et à la traduction automatique [Bahdanau *et al.*, 2014, Cho *et al.*, 2014a] (section 4.1.3.4).

Nous sommes convaincus du potentiel des architectures neuronales et décidons que la première contribution doit concerner l'apport d'un réseau de neurones récurrent bidirectionnel encodeur-décodeur avec mécanisme d'attention pour une tâche de compréhension de la parole, initialement dédié à la reconnaissance d'écriture manuelle et ayant été utilisé avec succès pour la reconnaissance de la parole et la traduction automatique. Alors qu'il est utilisé dans d'autres domaines, le mécanisme d'attention porté à la compréhension constitue une innovation.

En effet, le principe du mécanisme d'attention semble également répondre à la problématique de l'étiquetage en concepts sémantiques et permettrait d'aller plus loin qu'un simple biRNN (section 4.2.3). Un biRNN peut à une position donnée de la phrase, avoir une connaissance globale du passé et du futur mais la connaissance reste néanmoins concentrée autour de l'élément courant. Or, nous pensons que dans le cadre de la compréhension de la parole, il peut parfois être utile de disposer de connaissances présentes à d'autres emplacements de la phrase pour classer certains éléments, c'est-à-dire de regarder certaines zones particulières dans la phrase pour désambiguïser le choix du concept.

Considérons par exemple la phrase *"celui qui est à cent trente euros la chambre"*. Afin d'associer le support de mots *"cent trente"* au concept *"paiement-montant-entier"*, il faut regarder l'information monétaire du mot (*"euros"*) à l'extérieur du support. En revanche pour associer le support de mot *"euros"* au concept *"paiement-monnaie"*, le support de mot seul suffit.

C'est également le cas pour la traduction automatique. Le RNN avec mécanisme d'attention s'inspire largement de l'architecture proposée dans [Bahdanau *et al.*, 2014] pour la traduction automatique.

Nous souhaitons utiliser cette méthode pour la compréhension en considérant le processus d'étiquetage en concepts sémantiques, *i.e.* la détection de mots supportant des concepts et l'association avec ce concept, similaire à un problème de traduc-

tion depuis des mots (langage source) vers des étiquettes de concepts sémantiques (langage cible) associés à leur valeur.

La mise en place d'un réseau de neurones récurrent bidirectionnel encodeur-décodeur avec mécanisme d'attention sur le corpus MEDIA nécessite dans un premier temps de maîtriser l'utilisation d'un réseau bidirectionnel, nécessitant lui même la maîtrise d'un réseau récurrent avant et arrière. La section suivante se concentrera sur ces points préliminaires. Le mécanisme d'attention pourra ensuite être abordé dans la section d'après.

Les éléments de comparaison dont nous disposons sont les travaux de [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015] présentant les résultats d'un biRNN sur le corpus ATIS ainsi que ceux de [Hahn *et al.*, 2008, Hahn *et al.*, 2011] présentant les résultats de CRF sur le corpus MEDIA. Nous devrons donc commencer notre étude neuronale sur le corpus ATIS avant de l'adapter au corpus MEDIA tout en nous évaluant par rapport aux CRF conservant jusque-là une avance sur cette tâche [Vukotic *et al.*, 2015].

## 5.2 Constitution d'un réseau bidirectionnel état de l'art

Le mécanisme d'attention s'appuie sur un biRNN encodeur-décodeur. Par conséquent, notre premier objectif doit être de reproduire correctement un tel système.

### 5.2.1 biRNN état de l'art sur ATIS

#### 5.2.1.1 Éléments de base

Dans cette section, nous souhaitons obtenir un biRNN état de l'art à partir de celui de [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015] testé alors sur le corpus ATIS. Afin de valider notre implémentation de ce réseau de neurones bidirectionnel, des premières expériences ont été faites sur le corpus ATIS afin de comparer nos résultats avec ceux présentés dans cette étude.

Comme vu dans la section 1.2.2, le corpus ATIS est composé d'un ensemble APP et TEST. Afin de disposer d'un ensemble DEV, nous divisons le corpus d'apprentissage comme suit : 80% pour le corpus APP et 20% pour le corpus DEV.

Notre implémentation se base sur des outils fournis par les auteurs de [Mesnil *et al.*, 2013] et plus précisément sur l'implémentation proposée par son premier auteur [1]. Cependant, l'implémentation RNN ne correspond pas à celle utilisée pour leurs expériences : seulement l'implémentation du RNN avant de type elman/jordan avec le paramètre de dépendance à long terme T (*cf.* section 4.1.3.2) fixé à 1 est disponible tandis que cette étude utilise un réseau de neurones bidirectionnel.

1. *Recurrent Neural Networks with Word Embeddings DeepLearning 0.1 documentation (2015)*, http://www.deeplearning.net/tutorial/rnnslu.html#rnnslu

| Modèle | f-mesure |
|---|---|
| biRNN *jordan* [Mesnil *et al.*, 2013] (tab. 4.1) | **93,98** |
| RNN avant *elman* | 94,13 |
| RNN arrière *elman* | 90,81 |
| biRNN *elman* | **94,22** |

TABLE 5.1 – Comparaison entre les performances du biRNN présenté dans [Mesnil *et al.*, 2013] et notre implémentation d'un biRNN état de l'art (ATIS TEST manuel).

#### 5.2.1.2 Implémentation

Nous utilisons un RNN de type elman, implémenté à partir de la version avant fournie dans l'implémentation de Grégoire Mesnil.

Nos hyper-paramètres s'inspirent de ceux proposés par l'auteur :

— taux d'apprentissage : 0.062
— taille de lot : 9
— graine pour l'initialisation aléatoire : 345
— critère de test : score f-mesure
— nombre d'époques : 100
— fenêtre contextuelle : 5
— nombre de neurones dans la couche cachée : 200
— dimension des plongements : 50

Nous présentons les résultats sur le corpus TEST. Comme mentionné dans la section 4.1.2, ceux-ci sont obtenus avec la meilleure configuration du système obtenue sur le corpus DEV.

La version arrière est implémentée à partir de la version avant : la phrase est donnée à l'envers pour simuler cet effet comme mentionné dans la section 4.1.3.1. Avec un RNN arrière acquis, le bidirectionnel peut ensuite être implémenté comme décrit dans la section 4.1.3.2. L'apprentissage de ce biRNN utilisant conjointement un RNN avant et arrière est un apprentissage *parallèle* qui consiste à entraîner les RNN avant, arrière et bidirectionnel en même temps c'est-à-dire que l'on met à jour à chaque époque le RNN avant, puis arrière puis enfin le bidirectionnel.

#### 5.2.1.3 Résultats obtenus

Le tableau 5.1 présente les résultats obtenus dans cette configuration avec pour rappel ceux obtenus par les auteurs de [Mesnil *et al.*, 2013] [2].

On peut ainsi constater que notre système biRNN état de l'art est conforme à

2. Bien que Grégoire Mesnil ait choisi la configuration jordan, nous avons préféré conserver la version elman. Elman et jordan présentent pour nous des performances comparables dans les expériences avec néanmoins un dérèglement irrégulier des résultats de la version jordan à la fin de l'apprentissage.

| Modèle | f-mesure | CER |
|---|---|---|
| RNN avant | 74,04 | 21,2 |
| RNN arrière | 77,42 | 19,3 |

Table 5.2 – RNN Avant/Arrière (MEDIA TEST manuel).

nos attentes sur le corpus ATIS car il atteint des résultats similaires à ceux proposés par Grégoire Mesnil.

### 5.2.2 Adaptation sur MEDIA

#### 5.2.2.1 Motivations

Maintenant que nous disposons d'un biRNN état de l'art sur le corpus ATIS en comparaison avec l'étude de [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015], nous souhaitons passer à une étude du corpus MEDIA, un corpus plus ambitieux que celui d'ATIS. Il est plus difficile d'obtenir d'aussi bons résultats sur MEDIA que sur ATIS et les RNN ne parviennent d'ailleurs pas à surpasser les CRF sur cette tâche [Vukotic *et al.*, 2015]. Le corpus MEDIA constitue donc notre objectif de perfectionnement de nos RNN.

À compter de cette étape, nous évaluons également en CER, une mesure prenant en compte la notion de *concept* (*cf.* section 1.1.5.4). Cette mesure est plus adaptée et plus représentative pour notre tâche de compréhension de la parole d'étiquetage en concepts sémantiques[3]. La f-mesure en revanche ne traite que de la recherche d'information.

#### 5.2.2.2 Résultats obtenus

**RNN** Nous appliquons donc sur le corpus MEDIA les RNN avant et arrière présentés dans le tableau 5.1. Les résultats obtenus sont visibles dans le tableau 5.2.

Comme précisé plus haut, nous observons des performances moins élevées en f-mesure sur la tâche MEDIA comparées à celles observées sur la tâche ATIS.

**biRNN avec plusieurs modes d'apprentissage** Munis de RNN avant et arrière, nous pouvons maintenant appliquer un biRNN sur MEDIA. Les RNN avant et arrière utilisés pour l'entraînement ou la classification du bidirectionnel sont entraînés individuellement auparavant. Cette fois-ci, nous testons également différents modes d'apprentissage en plus de l'apprentissage parallèle précédent.

**parallèle** L'apprentissage parallèle consistait à entraîner les RNN avant, arrière et ensuite bidirectionnel à chaque époque.

**param-opt** Ensuite, nous essayons un apprentissage dans lequel l'entraînement du RNN bidirectionnel se base sur les meilleurs paramètres (paramètres optimaux) à la fois du RNN avant et du RNN arrière entraînés auparavant.

3. Les intervalles de confiances retenus sont consultables en annexe.

| Mode d'apprentissage | f-mesure | CER |
|---|---|---|
| parallèle | **79,51** | **15,3** |
| param-opt | 78,55 | 18,1 |
| parallèle/param-opt | 79,15 | 16,2 |

TABLE 5.3 – biRNN avec plusieurs modes d'apprentissage testés (MEDIA TEST manuel).

Cela veut dire que nous procédons à l'apprentissage complet du RNN avant dont on récupère la meilleure configuration. Puis nous recommençons pour le RNN arrière. Enfin le biRNN est entraîné en utilisant à chaque époque les meilleures configurations des RNN avant et arrière. Nous avons testé ce mode pour voir si le biRNN tire meilleur parti des RNN avant et arrière au maximum de leurs performances plutôt que d'évoluer en parallèle avec eux.

**parallèle/param-opt** Enfin, un dernier apprentissage qui combine les deux approches précédentes : à chaque époque les RNN avant et arrière sont entraînés comme dans l'apprentissage *parallèle*. Ensuite le RNN bidirectionnel utilise les paramètres des dernières meilleures époques pour les RNN avant et arrière respectivement comme dans l'apprentissage *param-opt*. Par conséquent si le RNN avant/arrière donne un meilleur résultat à une époque, il sera utilisé comme dans un apprentissage *parallèle*. S'il ne s'améliore pas, on utilise la dernière configuration enregistrée qui donnait le meilleur résultat jusque-là.

Les résultats d'expériences visibles dans le tableau 5.3 ont montré que le meilleur apprentissage est l'approche *parallèle* suivie du *parallèle/param-opt* et enfin du *param-opt*.

Cela peut être expliqué par le fait que le biRNN apprend davantage avec des RNN avant et arrière qui ont une plus grande variabilité au cours des époques même s'ils ne donnent pas toujours les meilleurs résultats. L'apprentissage en est par conséquent plus diversifié. En effet l'approche *param-opt* utilisant des paramètres avant et arrière fixés à partir de leurs meilleures époques est celle qui donne les pires résultats. L'apprentissage parallèle sera celui utilisé à l'avenir.

L'architecture RNN bidirectionnelle (quel que soit le mode d'apprentissage) obtient de meilleurs résultats en comparaison avec un RNN arrière ou avant. Cela confirme l'utilité de faire intervenir des informations du contexte passé et futur ensemble.

**Dépendances à long terme** Notre objectif est également d'implémenter des dépendances à long terme comme défini dans [Mesnil *et al.*, 2013] (section 4.1.3.2) en fournissant au réseau la somme des étapes précédentes/suivantes, c'est-à-dire avec un T supérieur à 1. L'implémentation initiale de Grégoire Mesnil donnait une implémentation avec T fixé à 1.

| Modèle | Dépendance à long terme T | f-mesure | CER |
|---|---|---|---|
| RNN avant (*fw*) | 1 (*cf.* tab. 5.2) | 74,04 | 21,2 |
| | 2 | **75,87** | **20,8** |
| | 3 | 72,58 | 23,9 |
| | 4 | 73,01 | 23,5 |
| RNN arrière (*bw*) | 1 (*cf.* tab. 5.2) | **77,42** | **19,3** |
| | 2 | 76,8 | 21,6 |
| | 3 | 76,39 | 20,7 |
| | 4 | 75,37 | 21,7 |
| biRNN | 1(*fw*), 1(*bw*) (*cf.* tab. 5.3) | **79,51** | **15,3** |
| | 2(*fw*), 1(*bw*) | 79,01 | 17,07 |

TABLE 5.4 – Impact des dépendances à long terme (MEDIA TEST manuel).

Une fois cela implémenté, nous pouvons tester différentes valeurs de T pour les RNN avant et arrière afin de voir si cet ajout de contexte aide le système. Les résultats sont visibles dans le tableau 5.4.

Le RNN arrière fonctionne mieux avec la configuration par défaut de T fixé à 1. En revanche le RNN avant donne de meilleurs résultats avec un T de 2. Néanmoins un bidirectionnel utilisant un RNN avant avec T=2 et un RNN arrière avec T=1 ne fonctionne pas mieux. La meilleure configuration d'un biRNN est toujours celle par défaut avec T=1 pour ses deux RNN.

### 5.2.3 Implémentation d'un CRF état de l'art sur MEDIA

#### 5.2.3.1 Motivations

Nous disposons à présent d'un système biRNN robuste validé sur le corpus ATIS en comparaison avec l'étude de [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015] et testé sur le corpus MEDIA.

Parallèlement, les CRF constituent un système classiquement utilisé pour la tâche d'étiquetage en concepts sémantiques et leurs résultats ont une valeur de référence sur MEDIA comme l'a montré la section 3.2.2 exposant l'étude de Stephan Hahn [Hahn *et al.*, 2008, Hahn *et al.*, 2011]. Dans cette étude, plusieurs systèmes sont mis en compétition sur la tâche MEDIA où les CRF remportent les meilleurs résultats, que l'on se trouve en configuration de transcriptions manuelles ou automatiques. Comme précisé dans [Vukotic *et al.*, 2015], les CRF restent plus performants que les RNN sur le corpus MEDIA.

Ainsi, il est important pour la suite de cette étude de disposer d'un modèle CRF état de l'art pour effectuer en parallèle les expériences sur les modèles neuronaux et disposer d'une comparaison valable au fur et à mesure que nous améliorerons notre système neuronal. Pour cela, nous avons implémenté un système CRF comme base de comparaison. Nous pourrons le modifier en parallèle de notre système neuronal

| | CER | |
|---|---|---|
| **Modèle** | **DEV** | **TEST** |
| CRF mot uniquement | 16,7 | 14,7 |
| +fenêtre contextuelle -1..+1 | 14,2 | 12,7 |
| +fenêtre contextuelle -2..+2 | **14,1** | **12,3** |

TABLE 5.5 – Résultats de CRF avec/sans fenêtre contextuelle (MEDIA manuel).

afin de constater les différents effets de nos innovations sur les CRF comme sur les RNN.

Nous n'évaluons pas ici nos systèmes CRF en f-mesure, jugeant le CER suffisamment pertinent pour l'évaluation de notre tâche de compréhension de la parole. Nous montrons également nos scores obtenus sur le corpus DEV.

#### 5.2.3.2 Implémentation

S'inspirant du meilleur système proposé dans [Hahn *et al.*, 2011], la boîte à outils Wapiti [Lavergne *et al.*, 2010] a été utilisée.

**Fenêtre contextuelle** Des premiers résultats sont montrés dans le tableau 5.5 avec le mot seul en entrée, ainsi qu'avec l'utilisation de différentes fenêtres contextuelles en entrée, montrant les bénéfices d'une fenêtre plus grande.

**Descripteurs de mot** Afin d'améliorer les performances de compréhension des systèmes sur le corpus MEDIA, un ensemble de descripteurs, inspiré de [Hahn *et al.*, 2011], représente chaque occurrence de mot en entrée. Les descripteurs complètent le mot en entrée afin de faciliter l'association des mots avec un contenu sémantique et donc d'aider le système de compréhension à atteindre une meilleure compréhension.

L'ensemble complet de descripteurs utilisés est défini ensuite. Nous appellerons cette configuration de descripteurs **config-descripteur-1** :

— les descripteurs de **config-descripteur-0** (cf. descripteurs 3.2.1)

— le POS (que n'employait pas l'étude [Hahn *et al.*, 2011])

Nous obtenons le POS avec l'outil LIA-TAGG [4].

Après de nombreuses expériences effectuées sur le corpus DEV, notre modèle de descripteurs final inclut les instances précédentes et suivantes pour les mots et POS dans un unigramme ou un bigramme afin d'associer une étiquette sémantique avec le mot en cours. De plus sont associées avec le mot courant les catégories sémantiques des deux instances précédentes et des deux suivantes. Les autres descripteurs ne sont considérés qu'à la position courante.

4. LIA-TAGG : http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bechet/download_fred.html.

| Modèle | CER | |
|---|---|---|
| | DEV | TEST |
| CRF implémenté | 11,7 | 11,3 |
| CRF [Hahn *et al.*, 2011] (*cf.* tab. 3.2) | 12,3 | 10,6 |

TABLE 5.6 – Résultats de notre CRF comparé à celui de Stephan Hahn (MEDIA manuel).

Stephan Hahn emploie les deux instances précédentes et suivantes pour les mots, et les instances précédentes et suivantes pour les catégories sémantiques. Les autres descripteurs sont considérés à la position courante.

Le tableau 5.6 montre les résultats obtenus par notre système en comparaison avec celui de Stephan Hahn.

Les résultats nous indiquent que notre implémentation est conforme aux attentes de l'étude [Hahn *et al.*, 2008, Hahn *et al.*, 2011]. Elle servira de base de comparaison pour la suite des travaux sur les RNN.

## 5.3 Mise en place du mécanisme d'attention

Dans cette section, nous montrons l'apport de l'attention dans le domaine de la compréhension avec des propositions d'implémentation et des premiers résultats sur MEDIA. Nous mettons en place un réseau de neurones récurrent bidirectionnel encodeur-décodeur avec mécanisme d'attention : le biRNN-EDA est décrit dans la figure 5.1.

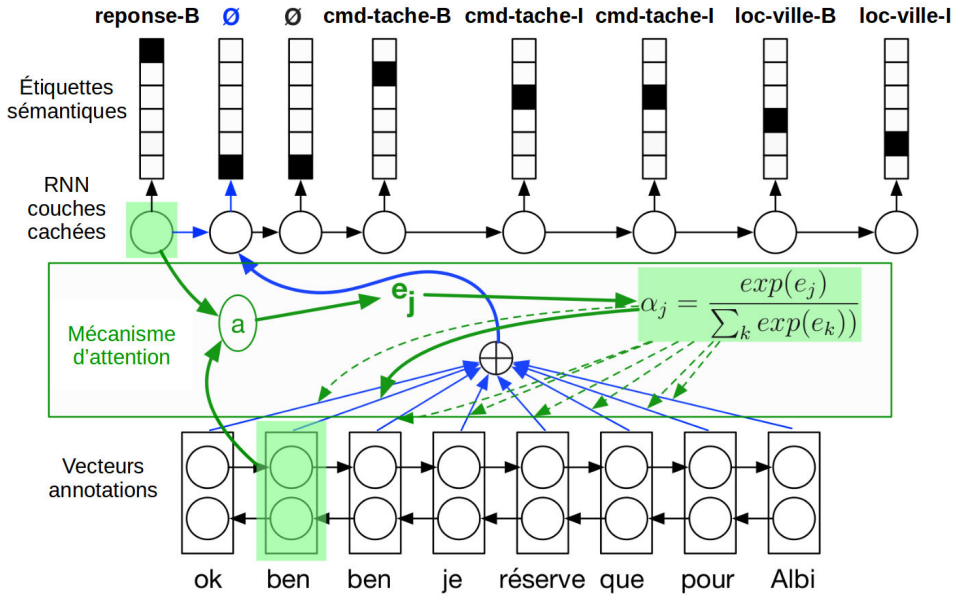



FIGURE 5.1 – Architecture du biRNN-EDA pour MEDIA.

| Modèle | f-mesure | CER |
|---|---|---|
| biRNN (*cf.* tab 5.3) | 79,51 | 15,3 |
| biRNN-EDA avec mécanisme d'attention | **80,27** | **12,9** |
| biRNN-ED sans mécanisme d'attention | 38,25 | - |

TABLE 5.7 – Ajout mécanisme d'attention (MEDIA TEST manuel).

Dans la section précédente, afin de valider notre propre implémentation d'un réseau de neurones bidirectionnel et comparer nos résultats avec ceux présentés dans [Mesnil *et al.*, 2013], nous avons mené des premières expériences sur le corpus ATIS. Les premières expériences faites sur le corpus ATIS ont bien confirmé la qualité du système biRNN état de l'art utilisé dans cette thèse, en comparant les résultats obtenus à ceux présentés par Grégoire Mesnil. Cela valide notre implémentation et nous a permis d'étudier son utilisation sur le corpus MEDIA que nous ciblons, plus complexe et plus exigeant que ATIS pour la compréhension (*cf.* section 4.2.3).

Avec ces outils préliminaires acquis et maîtrisés, nous sommes maintenant en mesure de chercher des améliorations supplémentaires telles qu'un RNN avec mécanisme d'attention comme nous le voyons dans cette section.

### 5.3.1 Implémentation du mécanisme d'attention

Notre implémentation d'un RNN avec mécanisme d'attention est conçue à partir de la boîte à outils *nmtpy*. Cet outil a été créé par des chercheurs du LIUM [Caglayan *et al.*, 2017] pour une tâche de traduction automatique. Dans cette tâche, les séquences d'entrée et de sortie ont souvent des longueurs différentes. L'approche avec un RNN encodeur-décodeur est particulièrement bien adaptée pour ce cas de figure.

Pour la tâche de compréhension que nous avons définie en revanche (*cf.* section 1.1.4), il y a une correspondance entre un mot et son étiquette sémantique. C'est cette correspondance entre les mots (entrées) et les étiquettes sémantiques (sorties) qui nous permet ensuite d'extraire la valeur du concept. Afin d'obtenir un alignement précis, nous avons modifié le processus de décodage du RNN bidirectionnel en imposant que la séquence d'étiquettes en sortie et la séquence de mots en entrée aient la même taille (une étiquette par mot).

**Résultats** Le tableau 5.7 montre les performances mesurées en termes de f-mesure et de CER liées au mécanisme d'attention sur le corpus MEDIA.

Les résultats montrent que le biRNN encodeur-décodeur avec mécanisme d'attention est plus performant qu'un RNN bidirectionnel classique.

De plus, il est aussi montré qu'un biRNN encodeur-décodeur (biRNN-ED) obtient de très faibles performances sans mécanisme d'attention lors de la production d'une séquence d'étiquettes en sortie ayant la même longueur que la séquence de

mots en entrée. Cela montre l'importance du mécanisme d'attention sans lequel un biRNN encodeur-décodeur ne peut se montrer utile.

Par la suite, nous arrêtons d'évaluer nos systèmes en f-mesure, jugeant le CER suffisamment pertinent pour l'évaluation de notre tâche de compréhension de la parole. Nous montrons également nos scores obtenus sur le corpus DEV et sur le corpus TEST.

### 5.3.2 Premières analyses

#### 5.3.2.1 Exemple d'observation des poids

La figure 5.2 montre un exemple d'application du biRNN-EDA sur une phrase de MEDIA.

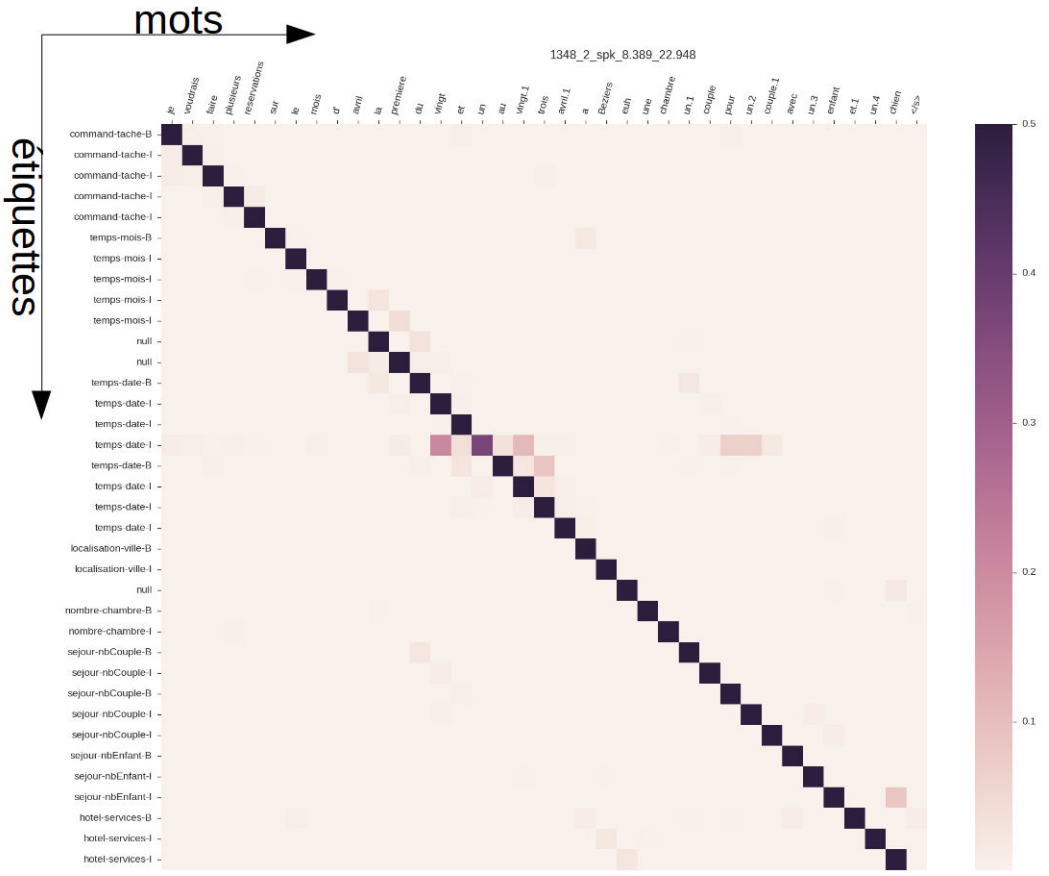



FIGURE 5.2 – Exemple d'application du biRNN-EDA sur une phrase de MEDIA : on voit pour les étiquettes (ordonnées) les poids du mécanisme d'attention attribués aux mots de la phrase (abscisses).

Pour chaque étiquette émise (ordonnées), on peut voir la répartition des poids sur les mots (abscisses) c'est-à-dire quels mots ont été regardés en particulier afin

d'émettre cette étiquette. On remarque globalement une diagonale signifiant que pour l'émission d'une étiquette à une position $t$, on donnera le plus d'importance au mot de la phrase à la position $t$ correspondante. Mais il arrive que le système regarde autour de $t$ La figure 5.3 montre un grossissement sur une zone intéressante de la figure précédente illustrant ce point.

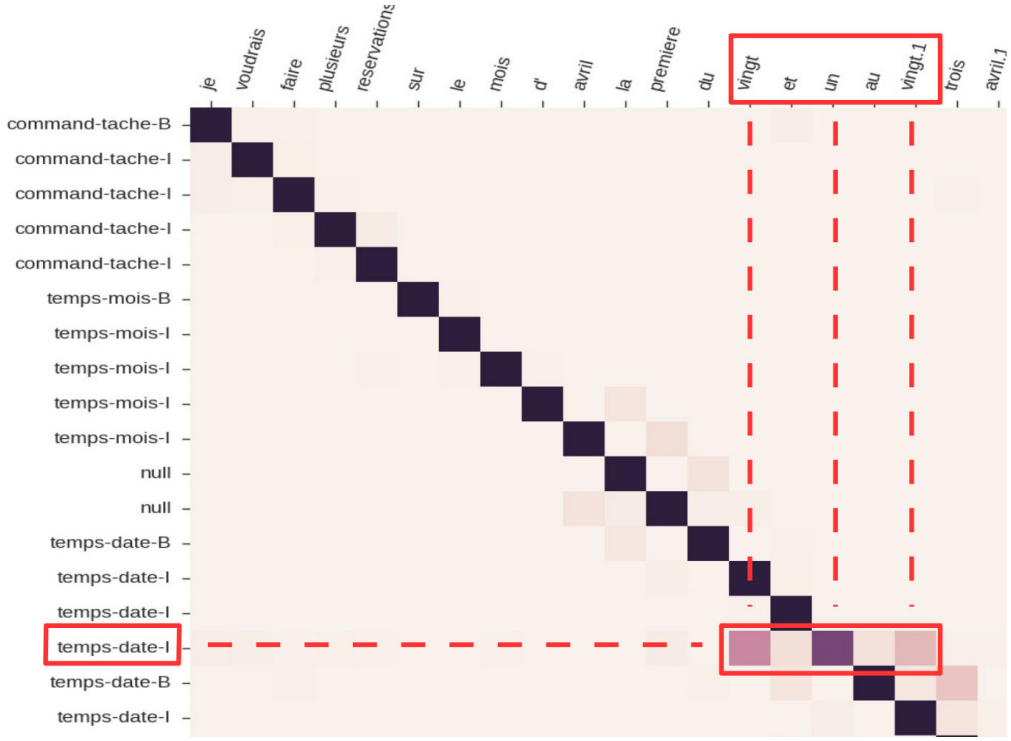



FIGURE 5.3 – Grossissement de la figure 5.2.

Ainsi, nous voyons que pour une étiquette « temps-date-I », le système a pris en compte les mots « vingt et un au vingt » qui constituent globalement la fourchette de date de la demande utilisateur. À partir de cet alignement, il serait possible de retrouver la valeur correspondant aux mots. Cet exemple illustre le principe du mécanisme d'attention et démontre l'importance de regarder certaines zones particulières dans la phrase pour désambiguïser le choix du concept.

#### 5.3.2.2 Fenêtre contextuelle

Nous avons également implémenté pour le mécanisme d'attention une fenêtre contextuelle afin de passer au système non pas seulement le mot $w_t$ mais $[w_{t-1}, w_t, wt + 1]$. Cette technique est employée pour les CRF ainsi que pour les RNN et biRNN. Cet ajout de contexte permet un gain supplémentaire à ces systèmes par rapport à l'utilisation du mot seul. Le tableau 5.8 montre les résultats obtenus.

| | CER | |
|---|---|---|
| **Fenêtre contextuelle** | **DEV** | **TEST** |
| sans (*cf.* tab. 5.7) | **13,9** | **12,9** |
| -1..+1 | 14,7 | 13,8 |

TABLE 5.8 – Application d'une fenêtre contextuelle sur un biRNN-EDA (MEDIA manuel).

L'usage de la fenêtre n'améliore pas le système dans le cadre d'un biRNN-EDA. Comme vu précédemment, le mécanisme d'attention doit s'occuper lui-même de rechercher les mots qui ont de l'importance dans la phrase. Par conséquent, il n'est pas utile de forcer le système à prendre plus de contexte en entrée avec l'usage d'une fenêtre.

### 5.3.3 Descripteurs de mots : intégration d'information de plus haut niveau

Après avoir appliqué avec succès le mécanisme d'attention et effectué des expériences préliminaires, présentées dans la section précédente, nous abordons dans cette section l'intégration de différents descripteurs de mots en entrée de notre biRNN-EDA.

Les descripteurs de mots (section 3.2.1), de nature syntaxique ou sémantique, sont utiles pour ajouter une information supplémentaire et complémentaire au mot en entrée du système de compréhension afin de l'aider dans sa tâche. Ainsi, le mot n'est pas représenté seulement par des lettres mais par d'autres informations de haut niveau, du contexte, etc...

Cette technique est employée par les CRF avec succès comme l'avait montré le tableau 5.6, qu'il s'agisse du système de Stephan Hahn ou bien de notre reproduction. Nous souhaitons observer ici si cette technique peut également apporter un gain pour les systèmes neuronaux, et plus particulièrement pour un biRNN-EDA.

Les deux architectures neuronales et CRF construisent leur modèle d'apprentissage sur le même ensemble de descripteurs en entrée, déjà présenté en section 5.2.3.2 (**config-descripteurs-1**), avec des valeurs continues pour le premier et des valeurs discrètes pour le second. Pour les NN, tous ces descripteurs additionnels sont représentés par des vecteurs encodés avec la méthode « one-hot » : les descripteurs sémantiques et syntaxiques représentent une information supplémentaire à ajouter dans le vecteur de plongement de mot d'entrée au mot avec une dimension totale de 5,4k unités quand ils sont tous pris en compte.

#### 5.3.3.1 Mot OU Classe sémantique en entrée

Comme vu dans la section 5.2.3.2, nous disposons de bases de données de vocabulaire liées à la tâche de compréhension que l'on nomme catégories sémantiques.

| | |
|---|---|
| Mots | j'aimerais réserver un hôtel pour les **trois premiers** jours de **Mai** à **Marseille**. |
| Mots *OU* Cat. sem. | j'aimerais réserver un hôtel pour les **UNIT ORDINAL** jours de **MOIS** à **VILLE**. |

TABLE 5.9 – Représentation d'un énoncé utilisateur par mots et par mots OU catégories sémantiques.

| | **CER** | |
|---|---|---|
| **Entrée** | **DEV** | **TEST** |
| Mot (*cf.* tab. 5.7) | **13,9** | **12,9** |
| Mot *OU* Catégorie sémantique | 14,6 | 14,3 |

TABLE 5.10 – Résultats obtenus sur un biRNN avec l'utilisation de catégories sémantiques en entrée (MEDIA manuel).

Dans MEDIA, on a les listes de villes, services proposés par un hôtel, types de nourriture servis par un restaurant, *etc.* Il y a aussi des catégories plus générales (chiffres, mois ...).

Une première possibilité d'intégration de cette information est la suivante : si le mot a une catégorie, il est alors substitué par celle-ci. Un exemple est disponible dans le tableau 5.9.

Ainsi, nous substituons les mots appartenant à une classe (nom de ville, *etc.*) par leur catégorie sémantique afin de voir si le système réagit mieux avec cette connaissance *a priori* au lieu du mot original. En plus d'apporter une connaissance *a priori*, cela consiste également à généraliser l'entrée du système, en espérant retirer ainsi des informations qui lui sont peu utiles ou favorisent des confusions. Le tableau 5.10 montre les résultats obtenus et indique qu'aucune amélioration n'est observée.

Il semble que la forme de surface des mots aide le système à prendre sa décision. La classe sémantique ne doit pas remplacer le mot mais certainement le compléter.

#### 5.3.3.2 Utilisation d'un descripteur à la fois en plus du mot

Le tableau 5.11 montre les résultats obtenus par l'utilisation d'un seul descripteur sémantique ou syntaxique à la fois en plus du mot.

Les informations syntaxiques et sémantiques se montrent utiles pour le système biRNN-EDA comme cela est le cas pour les CRF. En effet, elles apportent toutes une amélioration individuellement. Nous avons également testé l'ajout en descripteurs de plongements orthographiques de mots configurés par Sahar Ghannay. Ces plongements de mots capturent des informations de natures orthographiques pour un mot donné et ont donc pour but de substituer les descripteurs n-grammes de lettres au format one-hot. Les plongements de mots offrent une représentation plus riche que la représentation one-hot comme l'avait précisé la section 4.2.1 et cette

| | CER | |
|---|---|---|
| **Descripteur** | **DEV** | **TEST** |
| mot uniquement (*cf.* tab. 5.7) | 13,9 | 12,9 |
| mot+catégorie sémantique | **12,8** | **12,5** |
| mot+capitalisation | **13** | **12,7** |
| mot+POS | **13** | **11,9** |
| mot+lettres | **13,4** | **12,7** |
| mot+plongements orthographiques | 14,1 | 13,6 |

TABLE 5.11 – Contribution des descripteurs pour un biRNN-EDA individuellement (MEDIA manuel).

| | CER | |
|---|---|---|
| **Descripteur** | **DEV** | **TEST** |
| mot+catégorie sem., statique (*cf.* tab. 5.11) | **12,8** | **12,5** |
| mot+catégorie sem., appris | 13,8 | 13,1 |

TABLE 5.12 – Optimisation de descripteurs par un biRNN-EDA (MEDIA manuel).

expérience a été faite dans le but de constater si leur utilisation surpasse celle des one-hot. Les résultats ont montré que ce n'était pas le cas.

#### 5.3.3.3 Optimisation des descripteurs lors de l'apprentissage

Lors de l'apprentissage du système, les descripteurs sémantiques et syntaxiques sous forme one-hot ne varient pas tandis que les autres paramètres du système sont optimisés (plongements de mots, matrices de poids, *etc.*, *cf.* section 4.1.2) Nous avons essayé de laisser le système biRNN-EDA optimiser ses descripteurs de la même manière que ses autres paramètres en envisageant que le système puisse rendre ses descripteurs plus performants (les enrichir en information) au cours de l'apprentissage.

À l'initialisation, les descripteurs (mot mis à part, représenté par un plongement aléatoire) sont initialisés avec leur valeur one-hot. C'est après la première itération d'apprentissage que le système peut les modifier. Le tableau 5.12 montre les résultats obtenus avec les descripteurs de catégorie sémantique.

Comme le montrent les résultats, la meilleure configuration est avec un descripteur additionnel statique, son optimisation entraînant une perte de leur information. Pour la suite, nous choisirons donc de laisser le système optimiser le plongement de mot en entrée mais pas les descripteurs additionnels qui seront fixés à leur valeur one-hot.

| Descripteurs | | | | | CER | |
|---|---|---|---|---|---|---|
| **mot** | **cat. sem.** | **POS** | **cap.** | **lettres** | **DEV** | **TEST** |
| ***mot + 2 descripteurs : 6 cas*** | | | | | | |
| X | x | x | | | **12,3** | **12,7** |
| X | x | | x | | 13 | 12,9 |
| X | x | | | x | 12,7 | 12,1 |
| X | | x | x | | 12,5 | 11,9 |
| X | | x | | x | **12,4** | **11,4** |
| X | | | x | x | 13,2 | 12,9 |
| ***mot + 3 descripteurs : 4 cas*** | | | | | | |
| X | x | x | x | | 12,8 | 12 |
| X | x | | x | x | 12,4 | 11,7 |
| X | x | x | | x | **12,1** | **11,8** |
| X | | x | x | x | **12,3** | **12,5** |
| ***mot + 4 descripteurs : 1 cas*** | | | | | | |
| X | x | x | x | x | **12,3** | **11,9** |

TABLE 5.13 – Contribution des descripteurs pour un biRNN-EDA simultanément (MEDIA manuel).

#### 5.3.3.4 Utilisation de plusieurs descripteurs

Après avoir constaté le succès des descripteurs additionnels utilisés individuellement, nous en arrivons à vouloir constater les améliorations que nous pouvons obtenir par leur utilisation simultanée. Le tableau 5.13 montre les résultats obtenus avec une combinaison de descripteurs.

Parmi les quatre descripteurs additionnels retenus apportant des améliorations individuellement, toutes les combinaisons possibles sont testées pour deux, trois et quatre descripteurs simultanément, en plus du mot. Il est intéressant de noter que toutes les combinaisons apportent des améliorations. Parmi celles-ci, les cinq meilleures en considérant le corpus DEV sont :

1. catégorie sémantique + POS + lettres
2. catégorie sémantique + POS + capitalisation + lettres (toutes)
3. POS + capitalisation + lettres
4. catégorie sémantique + POS
5. POS + lettres

On notera que le POS est particulièrement intéressant car il est présent à chaque fois. Les combinaisons de descripteurs permettent d'obtenir des améliorations en comparaison à leur utilisation individuelle. Cependant la combinaison de descripteurs n'apporte pas systématiquement d'amélioration étant donné que l'on obtient de meilleurs résultats avec une combinaison de trois descripteurs additionnels parmi quatre devant celle des quatre ensemble.

Néanmoins, le descripteur de capitalisation est un descripteur utile capable d'apporter des améliorations individuellement et dans d'autres combinaisons. Nous choisissons donc de ne pas l'exclure et d'utiliser tous les descripteurs à notre disposition par la suite.

### 5.3.4 Optimisations

Les sections précédentes ont présenté différentes expériences desquelles résulte un système biRNN encodeur-décodeur avec mécanisme d'attention performant fonctionnant sur le corpus MEDIA et utilisant avec succès un ensemble de descripteurs additionnels sémantiques et syntaxiques.

Les architectures DNN doivent également être calibrées sur leurs hyper-paramètres respectifs afin de donner les meilleurs résultats (*cf.* section 4.1.2).

De base, la configuration de notre système est la suivante :

— pas de décroissance du taux d'apprentissage (*decay*) : 0,0005
— taille de lot : 25
— dimension des plongements de mots/d'étiquettes : 80
— dimension des couches cachées : 100
— nombre d'époques : 100
— mélange des lots (pour simuler un plus gros corpus) : désactivé

C'est cette configuration qui a donné les résultats précédemment présentés.

Afin d'optimiser notre système, nous avons essayé différentes combinaisons testées avec des variations autour de ces valeurs avec la meilleure combinaison déterminée sur le corpus DEV :

— pas de décroissance du taux d'apprentissage : entre 0,0002 et 0,0008 avec un pas de 0,0001
— taille de lot : entre 10 et 40 avec un pas de 5
— dimension des plongements de mots/d'étiquettes : entre 50 et 110 avec un pas de 10
— dimension des couches cachées : entre 70 et 130 avec un pas de 10
— nombre d'époques : augmentation de 100
— mélange des lots : activé/désactivé

La configuration d'hyper-paramètres la plus favorable est la suivante :

— pas de décroissance du taux d'apprentissage : 0,0002
— taille de lot : 25
— dimension des plongements de mots/d'étiquettes : 90
— dimension des couches cachées : 100
— nombre d'époques : 200
— mélange des lots (pour simuler un plus gros corpus) : désactivé

Les résultats sont visibles dans la table 5.14.

Cette optimisation a permis un gain additionnel intéressant (gain relatif de 0,7%) nous permettant de nous aligner au niveau des CRF sur la tâche d'étiquetage en concepts sémantiques de MEDIA.

| Hyper-paramètres | CER | |
|---|---|---|
| | DEV | TEST |
| configuration standard (*cf.* tab. 5.13) | 12,3 | 11,9 |
| configuration optimisée | **11,7** | **10,7** |
| CRF implémenté (*cf.* tab. 5.6) | 11,7 | 11,3 |

TABLE 5.14 – Incidence de l'optimisation des hyper-paramètres pour un biRNN-EDA avec tous les descripteurs de mots (MEDIA manuel).

## 5.4 Conclusion

Dans ce chapitre, nous avons mis en place avec succès un système neuronal état de l'art de compréhension de la parole. Ce système s'inspire d'un réseau de neurones récurrent bidirectionnel intégrant des informations de contexte ([Mesnil *et al.*, 2013]). De plus, nous y avons apporté un mécanisme d'attention, une technologie issue de la traduction automatique [Bahdanau *et al.*, 2014] qui montre des résultats prometteurs. Le biRNN extrait des informations contextuelles du passé et du futur de la phrase afin de mieux traiter la tâche d'étiquetage en concepts sémantiques. Le biRNN-EDA permet d'aller encore plus loin en encourageant le réseau de neurones à se focaliser sur certaines parties utiles de la phrase pour l'étiquetage en concept du mot en cours. Les expériences montrent que les biRNN avec mécanisme d'attention obtiennent de meilleures performances que les RNN récemment proposés pour la tâche d'étiquetage en concepts sémantiques. Les systèmes ont été étudiés sur les transcriptions manuelles du corpus MEDIA.

L'approche classique d'un RNN bidirectionnel a été introduite et présentée comme l'approche état de l'art pour la compréhension en 2013 par [Mesnil *et al.*, 2013] sur le corpus ATIS. Même si [Vukotic *et al.*, 2015] a montré que les CRF obtiennent toujours de meilleurs résultats que ce RNN bidirectionnel sur des données plus complexes comme le corpus MEDIA, nos résultats montrent que des approches prometteuses comme le mécanisme d'attention peuvent toujours améliorer les RNN pour la compréhension. Les expériences montrent que l'on peut obtenir de meilleurs résultats en construisant une architecture neuronale plus complexe. Cette architecture atteint de meilleurs résultats qu'une approche plus classique avec un RNN bidirectionnel seul sur un corpus de compréhension complexe.

Un système à base de CRF a été implémenté afin de disposer d'une comparaison fiable avec un système réputé et performant sur la tâche d'étiquetage en concepts sémantiques sur le corpus MEDIA.

Les deux systèmes biRNN-EDA et CRF bénéficient de descripteurs syntaxiques et sémantiques leur apportant un soutien important. Cela nous conforte à continuer de les utiliser et de nouveaux descripteurs pourront être considérés dans les études suivantes. De plus, une autre piste pourrait consister à détecter de nouvelles classes sémantiques automatiquement par *clustering* ou par regroupement distributionnel de nos plongements d'étiquettes (section 4.2.1).

Après une optimisation finale du biRNN-EDA, nous avons pu montrer que le système à base de réseaux de neurones peut atteindre des résultats tout aussi performants que les CRF.

CHAPITRE 6

# GÉRER LES ERREURS DE TRANSCRIPTIONS AUTOMATIQUES

## Sommaire

Les transcriptions manuelles ont été étudiées dans un premier temps afin de mieux analyser le système dans un cadre de compréhension théorique pure, c'est-à-dire de ne pas ajouter aux erreurs de compréhension les répercussions des erreurs de transcription automatique (section 2.2). Il reste néanmoins nécessaire de travailler sur des sorties de transcription automatique afin de se placer dans le cadre pratique de l'application visée, soit un système de dialogue oral (section 1.1.3.1).

Ainsi, ce chapitre traite de la transition de transcriptions manuelles à automatiques et de l'impact que cela entraîne sur nos systèmes de compréhension. Les erreurs de reconnaissance, inévitables lors du traitement de transcriptions automatiques, sont un problème majeur en compréhension de la parole (section 2.2). Cela nous impose de chercher des stratégies pour diminuer cet impact défavorable sur nos performances.

Les deux systèmes biRNN-EDA et CRF cherchent à étiqueter la meilleure séquence d'hypothèses de mots générées par le même système de reconnaissance de la parole décrit dans la première section. Ces mots sont associés à des mesures de confiance calculées par une architecture neuronale intégrant de nouveaux types de descripteurs de confiance, décrits dans la section 2.3.

Notre objectif principal après cela sera la gestion des erreurs de reconnaissance automatique. Nous proposons de parvenir à ce résultat par l'utilisation des mesures de confiance de reconnaissance, par une détection d'erreurs de reconnaissance au cours du processus de compréhension de la parole, par la manipulation et l'enrichissement du corpus d'apprentissage de compréhension et enfin par la simulation d'erreurs de reconnaissance.

## 6.1 Impact des transcriptions automatiques

### 6.1.1 Le système de reconnaissance de la parole du LIUM dédié à MEDIA

Pour ces expériences, un système de reconnaissance de la parole dédié à MEDIA est utilisé. Il s'agit d'une variante du système développé par le LIUM qui a remporté la campagne d'évaluation sur la langue française REPERE [Rousseau *et al.*, 2014]. Ce système est basé sur la boîte à outils de reconnaissance de la parole Kaldi [Povey *et al.*, 2011]. Cela a permis de fournir des transcriptions automatiques pour le corpus MEDIA qui étaient initialement peu accessibles (section 1.2.3).

Le jeu de données d'apprentissage utilisé pour estimer les paramètres des modèles acoustiques des DNN consiste en 145 781 segments de paroles provenant de plusieurs sources : les corpus de diffusions radiophoniques ESTER [Galliano *et al.*, 2006] et ESTER2 [Galliano *et al.*, 2009] qui contiennent environ 100 heures de discours chacun, le corpus de diffusion télévisée ETAPE [Gravier *et al.*, 2012] contenant environ 30 heures de discours, le corpus d'apprentissage de diffusion télévisée REPERE contenant environ 35 heures de discours et d'autres données de diffusion radiophonique et télévisée du LIUM pour environ 300 heures de discours. Au total 565 heures de discours composent le corpus d'apprentis-

| APP | DEV | TEST |
|---|---|---|
| 23,7 | 23,4 | 23,6 |

TABLE 6.1 – WER des transcriptions produites par le système de reconnaissance automatique pour MEDIA.

sage. Ces enregistrements ont été convertis dans des formats de types téléphoniques (exemple : 8kHz) avant d'entraîner les modèles acoustiques afin de les rendre plus appropriés aux données téléphoniques de MEDIA.

Les DNN intègrent (pour l'apprentissage et le décodage) des coefficients cepstraux de l'échelle de Mel (*Mel-Frequency Cepstrum Coefficients*) concaténés à des $i$-vecteurs, afin d'adapter les modèles acoustiques aux locuteurs.

Le vocabulaire du système de reconnaissance contient tous les mots présents dans les jeux de données APP et DEV de MEDIA, afin de limiter le nombre de mots hors vocabulaire, soit environ 2,5K mots. Un premier modèle de langage bigramme est appliqué durant le décodage pour générer des treillis de mots. Ces treillis sont ensuite recomposés en appliquant un modèle de langage trigramme.

Comme nous le verrons dans la section suivante, nous souhaitons disposer, pour la compréhension de la parole, d'un corpus d'apprentissage transcrit automatiquement afin de le rendre plus robuste aux erreurs de reconnaissance automatique. Nous souhaitons donc disposer d'un corpus APP contenant des erreurs de reconnaissance automatique de même type et en même quantité que celles que nous pourrions rencontrer dans les corpus DEV et TEST. Il est donc important que l'écart en WER entre les corpus APP, DEV et TEST soit réduit. Or, si le modèle de langage est entraîné sur le corpus APP, nous risquons d'avoir moins d'erreurs de reconnaissance dans la transcription automatique du corpus APP que nous devrions en avoir en réalité. Pour éviter des erreurs faites par un modèle de langage sur-entraîné sur le corpus d'apprentissage MEDIA, une validation croisée (*leave one out*) est effectuée : nous avons concaténé les corpus APP et DEV puis construit aléatoirement à partir de cela quatre sous-ensembles. Chaque sous-ensemble est transcrit en utilisant un modèle de langage entraîné sur les transcriptions manuelles présentes dans les trois autres groupes et linéairement interpolé avec un modèle de langage "générique" entraîné sur un grand ensemble de journaux français extraits du web, contenant 77 millions de mots. Les données de test sont transcrites avec un modèle de langage entraîné sur le corpus d'apprentissage MEDIA et le même modèle de langage générique.

Comme présenté dans le tableau 6.1, les performances en WER pour les corpus APP, DEV et TEST sont approximativement de 23,5%.

À titre de comparaison, le WER du système de reconnaissance utilisé dans [Hahn *et al.*, 2011] est de 30,3% pour le corpus DEV et de 31,4% pour le corpus TEST.

| CORPUS | CER | |
|---|---|---|
| APP & DEV | TEST manuel | TEST automatique |
| manuel | **10,7** (*cf.* tab. 5.14) | 35 |
| automatique | 27 | **24** |

TABLE 6.2 – Répercussions du passage aux transcriptions automatiques pour un biRNN-EDA, avec utilisation de descripteurs (MEDIA).

### 6.1.2 Résultats

Dans cette section, nous présentons les résultats obtenus par nos systèmes de compréhension appliqués aux sorties du système de reconnaissance du LIUM décrit ci-dessus.

Dans le chapitre précédent sur corpus manuel, nous avons pu obtenir une amélioration significative pour le biRNN-EDA en optimisant les hyper-paramètres (section 5.3.4). Nous essayons donc également d'optimiser le biRNN-EDA sur le corpus automatique. Comme dans le chapitre précédent, nous partons de la configuration par défaut du système déjà définie et procédons à la même optimisation que celle décrite dans la section 5.3.4. Dans ce cas l'optimisation sur transcriptions automatiques ne permet pas de trouver une configuration plus favorable au traitement de la tâche. Nous conservons donc cette configuration par défaut.

#### 6.1.2.1 Importance d'apprendre sur des transcriptions automatiques

Nous présentons dans le tableau 6.2 les résultats obtenus en testant sur des transcriptions automatiques le système biRNN-EDA entraîné sur transcriptions manuelles comme défini dans le chapitre précédent : nous utilisons également les mêmes descripteurs de mots définis dans le chapitre précédents (**config-descripteurs-1**).

Ces résultats nous montrent qu'un entraînement sur données manuelles est très insuffisant pour traiter des données automatiques. Un apprentissage sur des données automatiques contenant des erreurs de reconnaissance automatique de même type et en même quantité que celles que nous pouvons rencontrer lors du test est important pour rendre le système plus robuste à ces erreurs : nous passons d'un CER de 35 à 24.

Inversement, il est intéressant de constater que le fait d'apprendre sur des transcriptions automatiques plus difficiles à traiter que des manuelles, ne permet pas pour autant de mieux traiter la tâche manuelle : nous passons en effet de 10,7 à 27.

Par la suite, nous employons des transcriptions automatiques pour les corpus APP, DEV et TEST.

#### 6.1.2.2 Importance d'utiliser des descripteurs de mot

Nous présentons dans le tableau 6.3 des résultats obtenus selon que nous travaillions sur des transcriptions manuelles ou automatiques, avec ou sans utilisation de descripteurs de mots (**config-descripteurs-1**).

| Transcriptions (APP, DEV, TEST) | Descripteurs | CER | |
|---|---|---|---|
| | | DEV | TEST |
| manuelles | sans (*cf.* tab. 5.7) | 13,9 | 12,9 |
| | avec (*cf.* tab. 5.14) | **11,7** | **10,7** |
| automatiques | sans | 43,3 | 41,8 |
| | avec (*cf.* tab. 6.2) | **24,2** | **24** |

TABLE 6.3 – Incidence de l'utilisation de descripteurs de mots sur un biRNN-EDA, selon que l'on travaille sur des transcriptions manuelles ou automatiques (MEDIA).

Les descripteurs de mots sont un atout sur des transcriptions automatiques comme nous l'avions conclu sur des transcriptions manuelles dans le chapitre précédent. En revanche, il est intéressant de remarquer que le gain apporté par les descripteurs est bien plus intéressant sur des transcriptions automatiques où le CER est réduit de moitié alors que nous ne gagnions que quelques points sur les transcriptions manuelles. Les informations apportées par les descripteurs apportent notamment une correction cruciale qui compense les mots erronés par la reconnaissance de la parole.

Nous envisageons également dans cette section de nouvelles configurations de descripteurs :

**config-descripteurs-2** Nous utilisons une version de l'outil MACAON [Nasr *et al.*, 2010], particulièrement adaptée aux transcriptions automatiques, et nous permettant de récupérer de nouveaux descripteurs syntaxiques. Pour chaque mot nous avons les étiquettes suivantes : *le lemme, le mot gouverneur et la relation du gouverneur avec le mot courant.* Ces nouveaux descripteurs sont ajoutés à la précédente configuration *config-descripteurs-1.*

**config-descripteurs-3** Grâce à la disponibilité du lemme pour chaque mot, nous avons également essayé de retirer le mot de la liste des descripteurs de la précédente configuration. Ceci est fait dans l'hypothèse que le système neuronal n'ait pas nécessairement besoin de la conjugaison des verbes pour traiter une requête et donc que la forme généralisée du mot donnée par le lemme soit suffisante. En ne se servant que du lemme, on espère obtenir une entrée mieux généralisée et aussi réduire la complexité liée à la taille du vocabulaire. La généralisation apportée par l'utilisation du lemme à la place du mot (**config-descripteurs-3**) réduit le vocabulaire d'un tiers (2460 à 1650).

Les résultats pour le biRNN-EDA sont présentés dans le tableau 6.4 (avec les CRF à titre de comparaison sur la meilleure configuration du biRNN-EDA).

Les nouveaux descripteurs (**config-descripteurs-2**) apportent un gain supplémentaire permettant d'obtenir un CER de 22,3 sur le corpus TEST.

Malgré la généralisation apportée par l'utilisation du lemme uniquement (**config-descripteurs-3)**, nous n'obtenons pas de meilleurs résultats. Cette expé-

| | | CER | |
|---|---|---|---|
| **Système** | **Descripteurs** | **DEV** | **TEST** |
| biRNN-EDA | config-descripteurs-1 (*cf.* tab. 6.2) | 24,2 | 24 |
| | **config-descripteurs-2** | **24** | **22,3** |
| | config-descripteurs-3 | 24,2 | 23,2 |
| CRF | config-descripteurs-2 | - | **20,9** |

TABLE 6.4 – Résultats du biRNN-EDA et des CRF sur des transcriptions automatiques et montrant l'influence des différentes configurations de descripteurs de mots (MEDIA automatique).

rience similaire à celle menée sur les catégories sémantiques dans la section 5.3.3.1 nous amène à la même conclusion : les informations sémantiques sont très utiles en descripteurs additionnels mais ne substituent pas le mot.

Au final, même avec notre meilleure configuration de descripteurs, nous observons comme attendu que la transition de transcription manuelle à automatique rend la tâche de compréhension bien plus complexe à cause de l'imperfection de la reconnaissance automatique, ce qui impacte lourdement les performances en étiquetage sémantique.

En comparaison, les CRF sur la meilleure configuration de descripteurs obtiennent un CER de 20,9.

#### 6.1.2.3 Analyse d'erreurs de reconnaissance automatique

Comme nous l'avons vu dans la section 2.2, tandis que certaines erreurs de compréhension sont inhérentes à la tâche de compréhension en elle-même (segmentation du support, identification du concept), d'autres en revanche seront liées à une répercussion des erreurs de reconnaissance selon qu'elles concernent un mot dans ou hors d'un support de concept.

En regardant les corpus de développement et de test, on constate que 77% des erreurs de reconnaissance automatique sont situées sur des mots supports de concept (contre 23% sur des mots n'en supportant aucun (*null*)). Le corpus est constitué à 61% de supports de concept (contre 39% de hors support). On constate également qu'en cas d'erreur de reconnaissance automatique sur un mot, il y aura erreur de compréhension dans un tiers des cas sur l'étiquette.

On constate aussi que 62% des erreurs de reconnaissance automatique concernent des mots outils[1]. Dans la plupart des tâches de compréhension, on considère que ces mots forment la phrase et ont un rôle plus syntaxique que sémantique, mais ont néanmoins une importance dans le processus de compréhension de la parole.

Le tableau 6.5 montre les vingt mots les plus erronés par la reconnaissance

1. Exemple : *à, de, que, et, pas, etc.* Nous nous référons à la liste fournie par le système *Snowball* (`http://snowball.tartarus.org/algorithms/french/stop.txt`).

| # | MOT | % |
|---|---|---|
| *1* | est | 7,4 |
| *2* | c’ | 6,6 |
| *3* | et | 3,9 |
| *4* | oui | 3,8 |
| *5* | un | 3,8 |
| *6* | je | 2,7 |
| *7* | de | 2,6 |
| *8* | à | 2,4 |
| *9* | non | 2,3 |
| *10* | le | 2,2 |
| *11* | il | 2,0 |
| *12* | a | 1,9 |
| *13* | que | 1,9 |
| *14* | ce | 1,7 |
| *15* | y | 1,4 |
| *16* | s’ | 1,3 |
| *17* | l’ | 1,2 |
| *18* | en | 1,2 |
| *19* | d’ | 1,1 |
| *20* | pas | 1,1 |

TABLE 6.5 – Classement des vingt mots les plus erronés par la reconnaissance automatique de la parole dans le corpus MEDIA.

de la parole (par exemple, sur la totalité des mots erronés par la reconnaissance automatique de la parole, 7,4% sont le mot *"est"*).

Les mots les plus erronés sont plus souvent des mots courts et constitués de seulement une syllabe et qui peuvent être donc facilement mal interprétés (*"est/et", "à/a", etc.*).

Le tableau 6.6 donne un classement des dix concepts les plus concernés par les erreurs de transcriptions automatiques (par exemple, sur la totalité des mots erronés par la reconnaissance automatique de la parole, 12,7% sont associés au concept *"reponse"*).

Le tableau 6.7 donne un classement des dix concepts les plus erronés par la compréhension automatique (par exemple, sur la totalité des concepts erronés par la compréhension automatique de la parole, 13,4% sont le concept *"connectProp"*).

Un croisement de ces deux tableaux, présenté dans le tableau 6.8, montre de nombreux concepts en commun tels que des références, des connecteurs entre des entités de domaine et des noms propres pouvant être des valeurs d’attributs différents : cela indique bien une incidence de la reconnaissance sur la compréhension.

Ces concepts sont souvent supportés par des mots courts et pouvant être confondus. Nous en retrouvons beaucoup dans le tableau 6.5 des mots les plus erronés

| # | CONCEPT | % |
|---|---|---|
| *1* | reponse | 12,7 |
| *2* | command-tache | 11,6 |
| *3* | connectProp | 7,9 |
| *4* | localisation-ville | 6,8 |
| *5* | objet | 5,5 |
| *6* | nom-hotel | 4,6 |
| *7* | lienRef-coRef | 4,2 |
| *8* | nombre-chambre | 3,1 |
| *9* | hotel-services | 3,1 |
| *10* | localisation-lieuRelatif | 3,0 |

TABLE 6.6 – Classement des dix concepts les plus concernés par les erreurs de reconnaissance automatiques de la parole dans le corpus MEDIA.

| # | CONCEPT | % |
|---|---|---|
| *1* | connectProp | 13,4 |
| *2* | reponse | 12,2 |
| *3* | lienRef-coRef | 8,8 |
| *4* | objet | 7,3 |
| *5* | command-tache | 3,0 |
| *6* | nombre | 1,3 |
| *7* | localisation-ville | 1,3 |
| *8* | nom-hotel | 0,9 |
| *9* | localisation-lieuRelatif | 0,7 |
| *10* | temps-jour-mois | 0,6 |

TABLE 6.7 – Classement des dix concepts les plus erronés par la compréhension automatique dans le corpus MEDIA.

| CONCEPT | MOTS ASSOCIÉS |
|---|---|
| reponse | ***oui***, ***non***, *d'accord* |
| command-tache | ***je***, *réserver*, *voudrais* |
| connectProp | ***et***, ***est***, *donc* |
| localisation-ville | ***à***, *Paris*, ***de*** |
| objet | *prix*, *chambre*, *chambres* |
| nom-hotel | *hôtel*, ***l'***, *du* |
| lienRef-coRef | *la*, ***le***, ***l'*** |
| localisation-lieuRelatif | ***un***, ***de***, *la* |

TABLE 6.8 – Concepts communs aux erreurs les plus fréquentes de compréhension et de reconnaissance de la parole dans le corpus MEDIA (croisement des tableaux 6.7 et 6.6) avec les mots qui les supportent le plus fréquemment. Les mots en gras sont ceux faisant partie des vingt mots les plus erronés par la reconnaissance de la parole (*cf.* tableau 6.5).

par la reconnaissance de la parole. Leur désambiguïsation nécessite des relations de contexte complexes qui ne peuvent pas être caractérisées automatiquement (du moins avec la quantité de données d'apprentissage disponible) par les CRF ni par le type de mécanismes d'attention utilisés dans les NN.

Pour surpasser cette complexité induite par les erreurs de reconnaissance automatique, la suite des recherches sera orientée sur la gestion et la détection d'erreurs de reconnaissance automatique au sein du processus de compréhension.

## 6.2 Gestion et détection d'erreurs de reconnaissance

Cette partie concerne le problème de la détection et de la gestion d'erreurs de reconnaissance automatique et leur utilité pour améliorer les systèmes de compréhension de la parole.

Nous avons constaté que les transcriptions automatiques apportent une lourde contribution d'erreurs en compréhension de la parole. Il est donc intéressant de se concentrer sur la gestion de ces erreurs liées à la reconnaissance automatique. Nous envisageons dans cette section des systèmes de compréhension capables d'utiliser des informations concernant les erreurs de reconnaissance automatique ou bien encore des systèmes de compréhension capables de détecter eux-mêmes les erreurs de reconnaissance.

Nous verrons qu'en utilisant des scores de détection d'erreurs de reconnaissance automatique présentés dans la section 2.3 et en enrichissant l'ensemble des étiquettes sémantiques avec des étiquettes spécifiques aux erreurs de reconnaissance, les systèmes de compréhension peuvent apprendre conjointement à trouver les concepts et à détecter les erreurs de reconnaissance les impactant. D'autres améliorations peuvent également être apportées en utilisant plusieurs systèmes (CRF et NN) pour produire une hypothèse plus forte par combinaison de sorties de compréhension.

| | DEV | | TEST | |
|---|---|---|---|---|
| **Descripteurs** | **CER** | **CVER** | **CER** | **CVER** |
| mot+syntaxiques/sémantiques (Configuration 2, *cf.* tab. 6.4) | 24 | 30,6 | 22,3 | 28,8 |
| *+pap* | 22,9 | 29,7 | 21,3 | 28,1 |
| *+cm* | 23,1 | 29,9 | 22 | 28,8 |
| *+pap+cm* | **23,1** | **29,3** | **21,4** | **27,7** |

TABLE 6.9 – Impact de l'intégration des mesures de confiance sur un biRNN-EDA (MEDIA automatique).

Nous évaluons à présent également en CVER pour nous intéresser plus précisément à la tâche finale souhaitée d'un système de dialogue oral qui, en plus de détecter le concept doit extraire la valeur correspondante afin de pouvoir traiter la requête de l'utilisateur. Cela est également important car la valeur peut aussi être impactée par les erreurs de reconnaissance automatique. L'optimisation des systèmes sera également en CVER et non plus en CER.

### 6.2.1 Mesure de confiance de reconnaissance

Dans cette section, nous nous fixons comme objectif d'aider les systèmes de compréhension à mieux appréhender les erreurs de reconnaissance automatique qui peuvent subvenir dans ou hors des supports de concept.

Nous souhaitons fournir au système de compréhension une information sur la qualité de la transcription automatique qui pourrait l'aider dans sa tâche. Pour cela nous utilisons les deux mesures de confiance décrites dans la section 2.3. Elles sont introduites pour aider à la localisation des mots erronés qui peuvent affecter les performances de la compréhension. Il s'agit de la probabilité *a posteriori* (*pap*) et de la mesure de confiance calculée avec un MS-MLP prenant en entrée différents types d'informations de confiance (*cm)*. Elles estiment la fiabilité du mot reconnu par le système de reconnaissance *i.e.* la probabilité d'être justement transcrit.

Nous utilisons ces mesures comme des descripteurs de compréhension supplémentaires à combiner avec ceux syntaxiques et sémantiques déjà définis dans la Configuration 2, utiles pour la détection des concepts, leur délimitation *etc...* Les deux mesures de confiance de reconnaissance sont ajoutées en tant que valeurs continues (étant une probabilité) pour les NN et discrétisées pour les CRF avec l'outil discretize4CRF [2].

Les tableaux 6.9 et 6.10 montrent respectivement les résultats obtenus avec l'utilisation de ces descripteurs.

Les résultats expérimentaux montrent une réduction significative en CER et CVER sur le corpus MEDIA et confirment l'avantage attendu par l'introduction de descripteurs de confiance de reconnaissance. Ces résultats montrent que les mesures

2. https://gforge.inria.fr/projects/discretize4crf/

| Descripteurs | CER | CVER |
|---|---|---|
| mot+syntaxiques/sémantiques | 20,9 | 26 |
| *+pap* | 20,5 | 25,7 |
| *+pap+cm* | **19,9** | **25,1** |

TABLE 6.10 – Impact des mesures de confiance sur un CRF (MEDIA automatique TEST).

de confiance, donnant une information relative à la position de potentielles erreurs de reconnaissance, sont un atout pour les systèmes de compréhension. De plus ces deux mesures qui aident le système séparément peuvent également se montrer encore plus utiles une fois combinées. Le même comportement est observé par les CRF.

### 6.2.2 Système de compréhension à détection d'erreurs

Dans la section précédente, nous avons réussi à apporter aux systèmes de compréhension une information sur la qualité de la transcription automatique produite par un sous-système à détection d'erreurs de reconnaissance automatique.

Dans cette section, nous souhaitons construire un seul système qui résolve les deux tâches : détection d'erreurs de reconnaissance et compréhension de la parole. Nous pensons que l'apprentissage joint peut éventuellement renforcer les performances sur les deux tâches.

#### 6.2.2.1 Évaluation de la compréhension MEDIA

Nous considérons maintenant un ensemble de classes constitué d'une part des étiquettes conceptuelles MEDIA et d'autre part de deux étiquettes indiquant si le mot transcrit automatiquement est erroné.

Ces étiquettes remplacent l'étiquette habituelle lorsque le mot hypothétique est erroné. Si le mot hypothétique erroné supporte un concept, il est associé à l'étiquette *ERROR_C* (pour Concept), *ERROR_N* (pour *Null*) dans le cas contraire.

De cette façon, le système de compréhension recherche les zones erronées par la transcription automatique en associant aux mots des étiquettes *ERROR*. S'il n'y a pas d'erreur, le système produit alors l'étiquette sémantique classique.

Afin de pouvoir procéder à l'évaluation de compréhension classique sur MEDIA, les étiquettes hypothèses *ERROR_C* et *ERROR_N*, ne faisant pas parties du corpus, sont remplacées par *null*, informant que le mot ne transmet aucune information MEDIA. Cela permet d'exécuter le protocole d'évaluation MEDIA habituel. Les résultats obtenus par le biRNN-EDA sont reportés dans le tableau 6.11.

Un résultat est également donné représentant la capacité pure de détection d'erreurs de reconnaissance du système. Cette mesure est un taux d'erreur de détection de mots mal reconnus (ou encore un taux d'erreur de détection des erreurs de reconnaissance - TDE). Elle est calculée à partir de deux états uniquement : *ERROR* (*ERROR_C* ou *ERROR_N*) ou *CORRECT* (une étiquette MEDIA sans erreur de

| | DEV | | | TEST | | |
|---|---|---|---|---|---|---|
| **Descripteurs** | **CER** | **CVER** | **TDE** | **CER** | **CVER** | **TDE** |
| mot+ syntaxiques/sémantiques | 24,3 | 30,3 | 15 | 23,3 | 29,6 | 14,2 |
| *+pap* | **23,5** | **29,1** | **11,8** | **22,6** | **28,4** | **11,8** |
| *+cm* | 23,4 | 29,5 | 12,5 | 21,7 | 27,8 | 11,8 |
| *+pap+cm* | 24 | 29,5 | 12 | 21,4 | 27,2 | 12 |

TABLE 6.11 – Impact des mesures de confiance sur un biRNN à détection d'erreurs (MEDIA automatique).

transcription automatique). Les étiquettes *ERROR_C* ou *ERROR_N* sont converties en *ERROR* et les autres (détectées comme non erronées) en *CORRECT*. Le calcul entre la référence et l'hypothèse correspond donc à un LER uniquement sur des étiquettes *CORRECT/ERROR*.

Les résultats sont similaires à ceux du tableau 6.9, avec une légère amélioration sur le corpus DEV. Le système a également été évalué sur sa capacité pure en TDE à détecter les erreurs : nous notons que les mesures de confiance *pap* et *cm* se montrent particulièrement utiles pour la détection d'erreurs en TDE par rapport à un système n'en utilisant aucune. Notre système de compréhension est capable de donner une bonne estimation de la position des erreurs de reconnaissance.

Il est intéressant aussi de noter que sur ce système, le meilleur résultat est obtenu avec la *pap* uniquement et non avec les deux mesures comme sur les systèmes sans détection d'erreurs. L'apport des mesures de confiance utilisées seules ou en combinaison peut dépendre de la configuration des systèmes, qu'ils soient neuronaux ou à base de CRF, avec ou sans détection d'erreurs *etc.*... Par conséquent, dans la suite des travaux, les architectures de compréhension prennent tous les descripteurs syntaxiques et sémantiques mais les deux mesures de confiance peuvent être prises partiellement : l'une ou l'autre ou les deux selon la configuration la plus puissante sur le corpus DEV.

#### 6.2.2.2 Discrétisation des mesures de confiance

Comme nous l'avons vu précédemment, les systèmes neuronaux sont capables d'intégrer en entrée des valeurs continues tandis que les CRF doivent les discrétiser. Nous cherchons ici à vérifier quelle forme de mesure de confiance se montre la plus utile pour les NN entre continue et discrétisée.

Le résultats présentés dans le tableau 6.12 montrent que la forme continue est bien la forme la mieux acceptée par le système neuronal que ce soit avec ou sans détection d'erreurs de reconnaissance.

| Configuration | | DEV | | TEST | |
|---|---|---|---|---|---|
| **Mesure de confiance** | **Erreur** | **CER** | **CVER** | **CER** | **CVER** |
| continue (*cf.* tab. 6.9 & 6.11) | sans | **23,1** | **29,3** | **21,4** | **27,7** |
| | avec | **23,5** | **29,1** | **22,6** | **28,4** |
| discrète | sans | 24 | 29,5 | 22,7 | 28,7 |
| | avec | 23,8 | 29,1 | 23,8 | 29,5 |

TABLE 6.12 – Impact de la discrétisation de la mesure de confiance de reconnaissance de la parole pour un biRNN-EDA avec ou sans détection d'erreurs (MEDIA automatique).

#### 6.2.2.3 Différents modes d'évaluation

Les résultats précédemment obtenus par notre biRNN-EDA à détection d'erreurs ont été obtenus avec une optimisation en CVER sur l'ensemble d'étiquettes MEDIA uniquement en convertissant les étiquettes d'erreurs (*ERROR_C* ou *ERROR_N*) en étiquettes *null*. Cette évaluation est la plus importante car il s'agit de notre objectif final. On notera cette évaluation **A**.

Nous cherchons ici de nouvelles mesures d'évaluation prenant en compte la détection d'erreurs et dont le but est de mieux comprendre le comportement de notre système de compréhension à détection d'erreurs. On peut également espérer trouver une mesure d'évaluation plus adaptée à la compréhension et détection d'erreurs jointe, et qui permettrait d'obtenir de meilleurs résultats sur l'évaluation **A** que ceux précédemment obtenus.

Les nouvelles mesures sur lesquelles nous essayons d'optimiser le biRNN-EDA à détection d'erreurs sont les suivantes :

**B - Compréhension uniquement sur les concepts portés par des mots supports bien reconnus** En CVER par rapport à une référence ne contenant pas les concepts supportés par des mots erronés dans la transcription automatique. Ceci est fait afin de ne pas impacter le système en lui imposant de trouver des étiquettes de mots erronés dans la reconnaissance qu'il ne peut trouver puisqu'il les convertira en *null* s'il les détecte.

**C - Compréhension et détection d'erreurs de reconnaissance** En CVER par rapport à une référence mêlant concept et détection d'erreurs. Les étiquettes ERROR_N et ERROR_C sont intégrées à la référence classique afin d'évaluer simultanément la détection de concept et la détection d'erreurs de reconnaissance de la parole.

**D - Recherche d'erreurs de reconnaissance** En TDE sur la recherche d'erreurs pure (ERROR ou CORRECT).

Le tableau 6.13 montre un exemple applicatif de ces différentes évaluations.

Les résultats obtenus par le biRNN-EDA sont présentés dans les tableaux 6.14 et 6.15 respectivement pour le corpus DEV et le corpus TEST.

En considérant les résultats sur le corpus de validation (tableau 6.14), les meilleurs résultats (sur référence classique **A**) sont obtenus par les systèmes à détec-

| | | | | | |
|---|---|---|---|---|---|
| **1** | je | veux | deux | chambre | double |
| **2** | je | vois | de | chambre | trouble |
| **3** | - | x | x | - | x |
| **4** | O | O | nombre-chambre-B | chambre-type-B | chambre-type-I |
| **5** | O | ERR._N | ERR._C | chambre-type-B | ERR._C |
| **A** | - | | nombre-chambre | chambre-type | |
| **B** | - | | | chambre-type | - |
| **C** | - | ERR._N | ERR._C | chambre-type | ERR._C |
| **D** | COR. | ERR. | ERR. | COR. | ERR. |

TABLE 6.13 – Tableau montrant un exemple de détection d'erreurs de reconnaissance de la parole dans la tâche de compréhension MEDIA avec (1) : Les mots originaux, (2) : La transcription automatique, (3) : La position des erreurs de reconnaissance, (4) : L'étiquetage idéal recherché sans détection d'erreurs, (5) : L'étiquetage idéal recherché avec détection d'erreurs, (A) : La référence classique, (B) :La référence sans erreur de reconnaissance, (C) : La référence avec erreurs de reconnaissance et (D) : La référence de détection d'erreurs de reconnaissance uniquement.

| | | **Évaluation DEV** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **A : étiquettes** | | **B : étiquettes -erreurs** | | **C : étiquettes +erreurs** | | **D : erreur/ correct** |
| **Err.** | **DEV** | **CER** | **CVER** | **CER** | **CVER** | **CER** | **CVER** | **TDE** |
| sans | A | 23,1 | 29,3 | 22,7 | 26,4 | 42,9 | 45,5 | - |
| avec | A | **23,5** | **29,1** | 22 | 24 | **29,4** | **37,5** | **11,8** |
| avec | B | 23,4 | 29,5 | **21,1** | **23,5** | 30,2 | 39,8 | 12,5 |
| avec | C | **23,5** | **29,1** | 22 | 24 | **29,4** | **37,5** | **11,8** |
| avec | D | 24,3 | 30,3 | 22,4 | 25,3 | 30,3 | 40 | 11,9 |

TABLE 6.14 – Différents modes d'évaluation pour un biRNN-EDA à détection d'erreurs (Err.), avec descripteurs (MEDIA DEV automatique). *Cf.* tab. 6.9 et 6.11.

| | | Évaluation TEST | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | A : étiquettes | | B : étiquettes -erreurs | | C : étiquettes +erreurs | | D : erreur/ correct |
| **Err.** | **DEV** | **CER** | **CVER** | **CER** | **CVER** | **CER** | **CVER** | **TDE** |
| sans | A | **21,4** | **27,7** | 21,8 | 24,8 | 41,3 | 43,4 | - |
| avec | A | 22,6 | 28,4 | 21,5 | 23,7 | **28,4** | **37,1** | 11,8 |
| avec | B | 21,7 | 27,8 | **20,6** | **22,8** | 28,5 | 38 | 11,8 |
| avec | C | 22,6 | 28,4 | 21,5 | 23,7 | **28,4** | **37,1** | 11,8 |
| avec | D | 22,1 | 28,4 | 20,9 | 23,2 | 28,7 | 38,5 | **11,5** |

TABLE 6.15 – Différents modes d'évaluation pour un biRNN-EDA à détection d'erreurs (Err.), avec descripteurs (MEDIA TEST automatique). *Cf.* tab. 6.9 et 6.11.

tion d'erreurs avec une optimisation classique ou avec prise en compte des erreurs [3] (**A** et **C**). Ils sont suivis du système sans détection d'erreurs.

Les systèmes à détection d'erreurs optimisés sur **A** et **C** donnent également les meilleurs résultats sur l'optimisation incluant les erreurs (**C**) et en détection d'erreurs pure (**D**). L'optimisation avec retrait des erreurs (**B**) ne donne les meilleurs résultats que sur cette même évaluation. A noter qu'un système sans détection d'erreurs donne de très mauvais résultats sur une évaluation incluant les erreurs (**C**) montrant encore la force de détection d'erreurs des autres systèmes.

Cet ordre de performance ne se renouvelle pas nécessairement sur le corpus de TEST (tableau 6.15) mais nous souhaitons néanmoins que nos décisions soient prises en accord avec les résultats de développement. L'évaluation classique (**A**) est celle qui nous importe le plus à terme. Par conséquent on considère pour la suite que nos meilleurs systèmes sont les systèmes avec et sans détection d'erreurs avec la même optimisation classique.

### 6.2.3 Combinaison multi-systèmes de compréhension

#### 6.2.3.1 Principe

Nous disposons de deux architectures de compréhension : l'une neuronale et l'autre à base de CRF. De plus, la nouvelle mise en place d'une compréhension à détection parallèle d'erreurs de reconnaissance nous donne accès à deux nouvelles versions de ces deux systèmes pour un total de quatre systèmes de compréhension. Ces quatre systèmes sont également aidés par des mesures de confiance de reconnaissance. On peut ainsi considérer que ces quatre systèmes disposent d'un point de vue différents sur la tâche et donc de forces et de faiblesses qui leur sont propres. Nous noterons système standard (*std.*) les systèmes effectuant uniquement la com-

3. Il est possible de trouver le même résultat pour deux optimisations différentes dans le cas où les deux optimisations ont atteint leur meilleur résultat à la même étape de validation.

| | DEV | | | | | | TEST | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | | | CV | | | C | | | CV | | |
| **Config.** | % | **P** | **R** | % | **P** | **R** | % | **P** | **R** | % | **P** | **R** |
| CRF std. | 21,3 | 89 | 84 | 26,6 | 84 | 79 | 19,9 | 90 | 85 | 25,1 | 85 | 80 |
| CRF err. | **21,7** | **90** | **83** | **26,5** | **85** | **78** | **20,6** | **91** | **84** | **25,4** | **86** | **79** |
| NN std. | 23,1 | 87 | 83 | 29,3 | 80 | 77 | 21,4 | 88 | 84 | 27,7 | 81 | 77 |
| NN err. | 23,5 | 89 | 81 | 29,1 | 83 | 76 | 22,6 | 90 | 82 | 28,4 | 84 | 77 |

TABLE 6.16 – Performances de nos quatre systèmes de compréhension NN/CRF standard et avec détection d'erreurs de reconnaissance de la parole (MEDIA automatique). *Cf.* tab. 6.9, 6.10 et 6.11. On donne les résultats (taux d'erreur (%), précision (P) et rappel (R)) sur les concepts (C) et les couples concept/valeur (CV).

préhension et système à erreurs (*err.*) les systèmes effectuant une compréhension et une détection d'erreurs de reconnaissance simultanée. Pour rappel, les performances de ces quatre systèmes sont reportés dans le tableau 6.16.

Le meilleur système est le CRF à détection d'erreurs suivi du CRF classique puis du NN à détection d'erreurs et enfin du NN classique. Les CRF ont une nette avance sur les NN. En revanche il n'y a pas une grande différence entre un système classique et sa version à détection d'erreurs de reconnaissance. Il est surtout intéressant de voir que dans un système à détection d'erreurs, la précision est meilleure, signifiant que les concepts trouvés sont globalement plus justes.

Ces systèmes peuvent alors se montrer complémentaires et la combinaison optimale de ces architectures peut apporter des améliorations supplémentaires par rapport aux systèmes seuls.

Nous proposons plusieurs protocoles de combinaisons :

— Le vote : pour un mot donné, on dispose de quatre propositions d'étiquettes. On conserve celle qui remporte le plus de voix. Il peut s'agir d'un vote égal où on donne le même poids de vote à chaque système ou bien un vote pondéré où les systèmes ont des poids différents. Ces poids sont estimés par rapport aux performances optimales sur l'ensemble DEV.

— Le consensus : dans ce cas, les quatre systèmes doivent être d'accord, soit fournir la même étiquette pour un mot pour qu'elle soit conservée, dans le contraire on étiquettera *null*. Un consensus partiel à 75% n'exige que trois avis identiques sur quatre pour garder l'étiquette.

#### 6.2.3.2 Résultats

Les résultats des combinaisons sont reportés dans la table 6.17.

Pour référence, la combinaison ROVER [Fiscus, 1997b] appliquée aux six sys-

| | DEV | | | | | | TEST | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | | | CV | | | C | | | CV | | |
| **Config.** | **%** | **P** | **R** | **%** | **P** | **R** | **%** | **P** | **R** | **%** | **P** | **R** |
| vote égal | 20,9 | 90 | 84 | 25,9 | 85 | 79 | 19,6 | 91 | 85 | 24,8 | 85 | 80 |
| vote pond. | **20,4** | **91** | **84** | **25,3** | **85** | **79** | **19,3** | **91** | **85** | **24,5** | **86** | **80** |
| cons. 75% | 23,6 | 93 | 80 | 28,3 | 88 | 75 | 22,3 | 94 | 80 | 27 | 89 | 76 |
| cons. 100% | **30,9** | **95** | **71** | **35,9** | **88** | **66** | **29,3** | **96** | **72** | **34,2** | **89** | **68** |

TABLE 6.17 – Votes et consensus sur systèmes de compréhension standards et avec détection d'erreurs de reconnaissance de la parole (MEDIA automatique). On donne les résultats (taux d'erreur (%), précision (P) et rappel (R)) sur les concepts (C) et les couples concept/valeur (CV).

tèmes de compréhension décrits dans [Hahn *et al.*, 2011] (section 3.2.2) permet d'obtenir sur le corpus TEST un CER de 23,1 et un CVER de 27.

**Vote** Les résultats en vote montrent une réduction importante en CER et CVER par rapport à la meilleure architecture CRF. Le vote pondéré est meilleur que le vote égal attribuant les mêmes poids aux quatre systèmes. Les poids donnés aux systèmes CRF standards, CRF à détection d'erreurs, NN standards et NN à détection d'erreurs sont respectivement de 12, 14, 12 et 10.

**Consensus** Le consensus a pour effet de fortement dégrader les résultats en CER et CVER. Cependant, une précision de 0,96 avec un rappel de 0,72 a été observée sur l'ensemble TEST (CER). Cette précision très élevée est intéressante car, bien qu'ayant de mauvais résultats en rappel, nous pouvons être presque sûrs de l'exactitude des concepts observés. Ce consensus permet de délimiter des îlots de confiance pour lesquels le système global est quasiment sûr de lui. Les zones hors de ces îlots sont donc celles qu'il faut examiner avec précaution, voir même envisager que le système de dialogue oral demande des précisions à l'utilisateur, le fasse répéter *etc...*

## 6.3 Simulation d'erreurs de reconnaissance

### 6.3.1 Principe

Dans l'étude présentée ici nous supposons, et vérifions, que la construction des systèmes de compréhension à partir de transcriptions automatiques est une bonne solution pour les rendre plus robustes aux erreurs de transcriptions. Or, l'obtention

de transcriptions automatiques nécessite d'avoir à disposition d'une part des enregistrements audios relatifs aux annotations sémantiques et d'autre part un système de reconnaissance automatique de la parole.

Afin que ce dernier soit efficace, il nécessite lui aussi des données d'apprentissage et de validation, ces dernières étant souvent les mêmes que celles utilisées pour l'apprentissage et la validation de compréhension. Il convient donc de manipuler ces données avec prudence afin d'éviter des biais et notamment celui du sur-apprentissage.

Dans le cadre de la construction d'un système de compréhension performant, cette étude propose une approche de simulation des erreurs de reconnaissance à partir des transcriptions manuelles et d'une mesure de similarité afin d'une part de s'affranchir de la nécessité de données audios (6.1.1) et d'un système de reconnaissance lors de la phase d'apprentissage et d'autre part d'avoir néanmoins à disposition un corpus proche de celui à gérer lors du déploiement.

Partant de l'hypothèse que le système de reconnaissance confond les mots acoustiquement et linguistiquement proches (section 2.4), cette méthode s'appuie sur l'utilisation de plongements de mots acoustiques et linguistiques pour calculer une mesure de similarité entre les mots : cette mesure vise à prédire les confusions de mots faites par le système de reconnaissance. Cette approche est détaillée dans la section suivante et les résultats sont reportés dans la section 6.3.3.

### 6.3.2 Méthode de simulation

Notre approche consiste à simuler et introduire des erreurs dans les transcriptions manuelles en substituant des mots corrects par des mots similaires. Nous supposons que les mots susceptibles d'être confondus par un système de reconnaissance sont des mots acoustiquement et/ou linguistiquement proches. Pour calculer une mesure de similarité entre les mots, nous utilisons une nouvelle approche utilisant des plongements de mots acoustiques et linguistiques décrite dans la section 2.4.

Pour simuler des erreurs de reconnaissance, on utilise la mesure de similarité $confus(x, y)$ définie dans la section 2.4 afin de générer une liste de mots erronés possibles, ou liste de confusion. Il s'agit ensuite de substituer [4] dans la transcription manuelle des mots corrects par des mots erronés de la liste de confusion.

On détermine un taux d'erreur $e$ souhaité dans la transcription manuelle sans erreur. On modifie ensuite aléatoirement un pourcentage $e$ des occurrences de mots. Les substitutions sont faites après avoir défini deux seuils qui ont pour but de limiter la profondeur de la recherche d'un mot erroné possible :

— le seuil $c$ qui correspond à la valeur la plus basse de $confus(\overline{r}, h)$ qui permet de substituer le mot $\overline{r}$ par le mot $h$. Cela veut dire que l'on ne substituera le mot correct $\overline{r}$ que par un mot erroné $h$ dont le score de similarité est supérieur ou égal à $c$.

4. Il s'agira donc d'erreur de substitutions, les insertions et suppressions ne sont pas gérées ici.

— le seuil $n$ qui limite le nombre de substitutions possibles de $\overline{r}$ parmi les $n$ mots $h_i$ les plus proches (*i.e.* les mots $h_i$ tels que la valeur $confus(\overline{r}, h_i)$ est l'une des $n$ valeurs les plus hautes étant donné $\overline{r}$).

Le mot $h$ est choisi aléatoirement dans la liste des mots $h_i$ qui respectent les contraintes des seuils $n$ et $c$. Un schéma récapitulatif est présenté dans la figure 6.1.

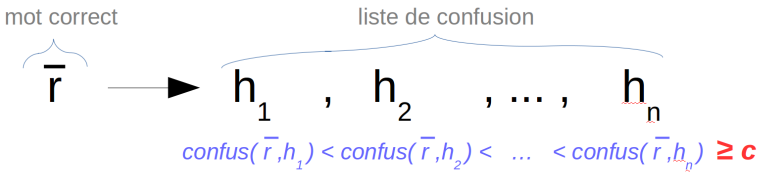



FIGURE 6.1 – Schéma récapitulatif de la substitution d'un mot correct par un mot erroné en respect des seuil $c$ et $n$.

À partir des annotations manuelles du corpus MEDIA, nous construisons différents ensembles de données. Dans ces simulations, nous avons fixé la valeur de $e$ à 20%, ce qui représente le taux de mots que nous corrompons au hasard dans les transcriptions manuelles et qui correspond environ au WER moyen obtenu par notre système ASR[5] (tableau 6.1).

Plusieurs simulations (**B** pour bruité) ont été testées, en choisissant différentes valeurs de seuil $n$ et $c$. Ceci est fait dans le but de tester l'impact d'avoir des substitutions plus ou moins éloignées :

**Corpus B.7 -** $n = 7$ et $c = 0.4$.

**Corpus B.10 -** $n = 10$ et $c = 0.5$. Ainsi ce corpus substitue par des mots globalement plus éloignés au niveau du score de similarité et du nombre de plus proches voisins.

**Corpus double B.7 -** Ici deux simulations d'erreurs de reconnaissance successives sur le corpus APP ont été utilisées afin de produire un plus gros corpus d'apprentissage.

**Corpus B.n -** Un autre ensemble de données artificiel dit naïf a été créé. Ce corpus ne prend pas en compte la mesure de similarité. Dans cet ensemble de données, le même pourcentage de mots $e = 20\%$ issus des transcriptions manuelles est substitué de manière aléatoire, en choisissant simplement un mot au hasard dans l'ensemble du vocabulaire MEDIA.

Nous rappelons que parmi les descripteurs de mot utilisés figurent deux mesures de confiance de reconnaissance automatique *pap* et *cm*. Auparavant nous utilisions la meilleure configuration d'utilisation de ces deux mesures (l'une, l'autre, ou les deux en même temps, *cf.* section 6.2.1). Quand un mot correct est remplacé par un mot confondu, nous utilisons la mesure de similarité comme mesure de confiance

5. Néanmoins, ce WER tient compte d'erreurs d'insertion et de suppression, ce que ne fait pas notre simulation.

de reconnaissance car elle présente le même comportement (probabilité que le mot soit erroné). Dans un but de cohérence expérimentale pour les résultats suivants, lorsque nous travaillons sur des sorties de reconnaissance automatique, nous donnons seulement une mesure de confiance parmi les deux disponibles (et jamais deux à la fois) afin d'avoir toujours le même nombre de mesures de confiance dans tous les cas (corpus bruité ou automatique). Cela vient du fait que nous utiliserons la mesure de similarité comme mesure de confiance artificielle et que nous ne sommes capables de fournir qu'une seule mesure de ce type sur les données simulées.

En résumé, un seul descripteur sera utilisé pour indiquer la mesure de confiance du mot dans tous les cas :

— Transcription automatique : une mesure de confiance automatique *pap* ou *cm* (la meilleur sur le corpus DEV)
— Transcription bruitée : la mesure de similarité *confus(x,y)*
— Transcription manuelle : toujours 1 (le mot est correct)

Pour les deux systèmes de compréhension, l'apprentissage est fait sur le corpus APP et les meilleures configurations sont choisies pour optimiser le CVER sur le corpus DEV.

### 6.3.3 Apport des transcriptions bruitées et/ou augmentées

Dans nos expériences menées sur le corpus MEDIA, nous évaluons l'impact de cette approche en bruitant le corpus d'apprentissage des deux systèmes de compréhension CRF et biRNN-EDA. **M** fait référence au corpus manuel, **A** à un corpus composé de transcriptions automatiques, et **B** à un corpus bruité. Le corpus TEST est constitué uniquement de transcriptions automatiques, alors que la nature des corpus APP ou DEV varie dans nos expériences.

#### 6.3.3.1 En apprentissage

Puisque l'évaluation sur le corpus TEST est faite sur des transcriptions automatiques, nous considérons dans un premier temps qu'un corpus DEV composé de transcriptions automatiques est également disponible. Ce corpus est moins coûteux à collecter qu'un corpus d'entraînement (1.3k phrases *vs.* 17.7k).

**Importance d'apprendre sur des transcriptions automatiques ou similaires** Dans le tableau 6.18, on remarque de nouveau l'importance d'apprendre sur des données proches des données de test (avec des transcriptions automatiques ou contenant des simulations d'erreurs) : avec le corpus APP **A**, les résultats du biRNN-EDA et des CRF sont significativement meilleurs que ceux obtenus avec un corpus APP **M**.

L'entraînement d'un système de compréhension sur des transcriptions manuelles est largement insuffisant pour gérer les transcriptions automatiques. Le système doit être préparé aux erreurs de transcription.

| | | biRNN-EDA | | | | CRF | |
|---|---|---|---|---|---|---|---|
| Corpus | | DEV | | TEST | | | |
| APP | DEV& TEST | CER | CVER | CER | CVER | CER | CVER |
| A (*cf.* tab. 6.9 & 6.10) | | 22,9 | 29,7 | **21,3** | **28,1** | **20,5** | **25,7** |
| M | A | 32,7 | 37,3 | 32,8 | 36,8 | 27,5 | 31,6 |
| B.7 | | 25,4 | 30,8 | 23,8 | 29 | **22,6** | **27,7** |
| Double B.7 | | 24,9 | 30,5 | **23,2** | **28,8** | 26,3 | 31,3 |

TABLE 6.18 – Comparaison de différents corpus APP pour des corpus DEV et TEST automatiques sur MEDIA montrant l'importance d'apprendre sur des transcriptions automatiques ou similaires.

L'entraînement sur un corpus bruité (**B.7**) permet d'obtenir des résultats intéressants. En effet, on obtient une nette amélioration par rapport aux mauvais résultats obtenus sur les transcriptions manuelles seulement. L'entraînement sur **B.7** permet de se rapprocher des résultats utilisant les transcriptions automatiques pures et confirme ainsi que notre approche pour simuler des erreurs de transcription est adaptée à cette tâche. Entraîner sur un corpus bruité doublé **double B.7** permet d'améliorer un peu les résultats sur le biRNN-EDA tout en aggravant fortement ceux des CRF. En somme le corpus bruité rapproche significativement les résultats de compréhension de ceux obtenus avec un entraînement automatique par rapport à un entraînement manuel : cela montre l'importance d'un corpus d'apprentissage adapté aux sorties de transcriptions automatiques en modifiant les données pour les faire ressembler à de la reconnaissance automatique. Les systèmes neuronaux semblent mieux tirer parti d'un plus grand ensemble de données d'apprentissage que les systèmes CRF.

**Combinaison de données manuelles et bruitées** Dans le tableau 6.19, de meilleurs résultats peuvent être obtenus en combinant des corpus manuels et bruités.

En utilisant l'ensemble de données **B.7** concaténé à l'ensemble de données manuelles, les résultats sont équivalents à ceux des transcriptions automatiques pures pour le biRNN-EDA. Les CRF obtiennent les mêmes résultats que pour **B.7** seulement. Ce résultat montre qu'en disposant d'un corpus de transcriptions manuelles, on peut d'ores et déjà construire une simulation correcte d'un corpus de transcriptions automatiques pour l'apprentissage d'un modèle biRNN-EDA.

**Différents types de bruitages** Nous pouvons également comparer les différents types de bruit, dans le tableau 6.20.

Le **B.7** obtient de meilleurs résultats que le **B.10**, ce qui montre qu'en substituant des mots corrects à des mots globalement moins semblables, les résultats sont moins bons. De plus, même si l'application de bruit naïf (**B.n**) obtient de meilleurs

| | | biRNN-EDA | | | | CRF | |
|---|---|---|---|---|---|---|---|
| **Corpus** | | **DEV** | | **TEST** | | | |
| **APP** | **DEV& TEST** | **CER** | **CVER** | **CER** | **CVER** | **CER** | **CVER** |
| A (*cf.* tab. 6.9 & 6.10) | | 22,9 | 29,7 | **21,3** | **28,1** | **20,5** | **25,7** |
| B.7 | A | 25,4 | 30,8 | 23,8 | 29 | **22,6** | **27,7** |
| Double B.7 | | 24,9 | 30,5 | 23,2 | 28,8 | 26,3 | 31,3 |
| M+B.7 | | 23,7 | 28,9 | **22,7** | **28,1** | **22,6** | **27,7** |

TABLE 6.19 – Comparaison de différents corpus APP pour des corpus DEV et TEST automatiques sur MEDIA : combinaison de données manuelles et bruitées (*cf.* tab. 6.18).

| | | biRNN-EDA | | | | CRF | |
|---|---|---|---|---|---|---|---|
| **Corpus** | | **DEV** | | **TEST** | | | |
| **APP** | **DEV& TEST** | **CER** | **CVER** | **CER** | **CVER** | **CER** | **CVER** |
| A (*cf.* tab. 6.9 & 6.10) | | 22,9 | 29,7 | 21,3 | 28,1 | **20,5** | **25,7** |
| M | A | 32,7 | 37,3 | 32,8 | 36,8 | 27,5 | 31,6 |
| M+B.7 | | 23,7 | 28,9 | **22,7** | **28,1** | **22,6** | **27,7** |
| M+B.10 | | 23,9 | 29 | 23,3 | 28,5 | 23,2 | 28,3 |
| M+B.n | | 25,5 | 30,6 | 23,7 | 28,8 | 25 | 30,3 |

TABLE 6.20 – Comparaison de différents corpus APP pour des corpus DEV et TEST automatiques sur MEDIA : différents types de bruitages (*cf.* tab. 6.18, 6.19).

| | | biRNN-EDA | | | | CRF | |
|---|---|---|---|---|---|---|---|
| **Corpus** | | **DEV** | | **TEST** | | | |
| **APP** | **DEV& TEST** | **CER** | **CVER** | **CER** | **CVER** | **CER** | **CVER** |
| A (*cf.* tab. 6.9 & 6.10) | | 22,9 | 29,7 | 21,3 | 28,1 | **20,5** | **25,7** |
| M+A | A | 21,2 | 26,3 | **20,3** | **25** | **20,2** | **25,3** |
| M+B.7+A | | 20,9 | 26,6 | 20 | 25,2 | 29,1 | 33,0 |

TABLE 6.21 – Comparaison de différents corpus APP pour des corpus DEV et TEST automatiques sur MEDIA : combinaison de données automatiques, manuelles et bruitées.

résultats que l'utilisation de transcriptions manuelles (**APP M**), nous obtenons les plus mauvais scores parmi les approches bruitées. Ceci montre l'importance d'un bruit généré intelligemment, et valide implicitement notre approche de simulation d'erreurs de transcription. Le fait que même un apprentissage sur un bruit naïf parvienne à de meilleurs résultats que sur des transcriptions manuelles est intéressant : cela laisse à penser que les erreurs aléatoires naïves crée un bruit dans les données d'apprentissage qui implique que le réseau de neurones apprend à mettre en place une stratégie robuste aux erreurs.

**Combinaison de données automatiques, manuelles et bruitées** Finalement, les meilleurs résultats surpassant les transcriptions automatiques pures (**A**) sont obtenus en entraînant les systèmes de compréhension sur une combinaison de sorties automatiques et manuelles (**M+A**), comme le montre le tableau 6.21.

Les deux systèmes trouvent leur meilleure performance dans cette configuration et l'écart entre CRF et biRNN-EDA a été fortement réduit par rapport aux expériences sur **A** ou **M** seulement. Les NN disposent cette fois-ci d'une légère avance sur les CRF : en effet les systèmes neuronaux sont beaucoup plus réceptifs à cet enrichissement de leur apprentissage que les CRF. L'entraînement sur une triple combinaison de corpus manuel, automatique et bruité n'augmente pas davantage ces résultats pour les NN et les dégradent fortement pour les CRF.

**Bilan** En général, les CRF surpassent significativement les biRNN-EDA lorsque ces systèmes sont entraînés sur un corpus manuel ou automatique. Mais les biRNN-EDA tirent mieux parti de la simulation d'erreurs, ou de la combinaison manuelle et automatique par rapport aux CRF. Au final, les meilleurs résultats des biRNN-EDA et CRF sont très proches, montrant un potentiel des réseaux de neurones, non partagé par les CRF, à apprendre des informations pertinentes à partir de données bruitées.

| | | | biRNN-EDA | | | |
|---|---|---|---|---|---|---|
| Corpus | | | DEV | | TEST | |
| APP | DEV | TEST | CER | CVER | CER | CVER |
| A (*cf.* tab. 6.9 & 6.10) | | | 22,9 | 29,7 | **21,3** | **28,1** |
| M | | A | 11,8 | 15,6 | 35,6 | 39,8 |
| B.7 | | | 17,7 | 32,3 | 23,5 | 28,6 |
| M+B.7 | B.7 | | 16,4 | 30,6 | **23,1** | **28,5** |

TABLE 6.22 – Comparaison obtenue sans transcriptions automatiques pour le corpus APP ou le corpus DEV avec un corpus TEST automatique sur MEDIA.

#### 6.3.3.2 En apprentissage et développement

Dans cette section, nous explorons le scénario dans lequel aucune donnée issue d'un système de reconnaissance automatique n'est disponible pour construire et optimiser le système de compréhension (corpus DEV inclus). Cela peut devenir problématique lorsque le système de compréhension doit effectuer des phases de validation durant le processus d'apprentissage, ce qui est le cas des biRNN-EDA. Les CRF pour leur part n'utilisent pas le corpus DEV pendant l'entraînement (la configuration optimale n'est pas modifiée et les scores des CRF restent inchangés). Ainsi, les résultats visibles dans la table 6.22 ne concernent que les biRNN-EDA.

En général, sauf pour le corpus APP bruité seul, de meilleurs résultats sont atteints en validant sur un corpus DEV automatique (*cf.* table 6.18 et 6.19), plus proche des données du corpus de TEST. Néanmoins, même si les résultats du tableau 6.22 sont un peu moins bons que ceux obtenus en validant sur un corpus DEV automatique, on peut remarquer qu'il est possible d'améliorer très significativement les performances des systèmes de compréhension en appliquant notre approche de simulation d'erreurs pour enrichir ou bruiter les données d'apprentissage et de développement des systèmes de compréhension ne disposant que de transcriptions manuelles.

Nous montrons que sans disposer d'aucune donnée automatique MEDIA, nous parvenons néanmoins à obtenir un score CER/CVER quasiment aussi bon que celui obtenu sur un apprentissage et un développement automatique (23,1/28,5 contre 21,3/28,1)

En conclusion, les expériences précédentes ont montré que cette approche de simulation améliore significativement les performances des systèmes de compréhension avec une réduction particulière quand le système est neuronal. Une comparaison avec une méthode de bruitage naïf montre la pertinence de l'approche de bruitage proposée.

| | DEV | | | | | | TEST | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | | | CV | | | C | | | CV | | |
| **Config.** | **%** | **P** | **R** | **%** | **P** | **R** | **%** | **P** | **R** | **%** | **P** | **R** |
| CRF std. *[A]* | 21,3 | 89 | 84 | 26,6 | 84 | 79 | 19,9 | 90 | 85 | 25,1 | 85 | 80 |
| CRF err. *[A]* | 21,7 | 90 | 83 | 26,5 | 85 | 78 | 20,6 | 91 | 84 | 25,4 | 86 | 79 |
| NN std. *[M+A]* | **21,2** | **90** | **83** | **26,3** | **85** | **78** | **20,3** | **92** | **83** | **25** | **86** | **79** |
| NN err. *[M+A]* | 22,2 | 92 | 82 | 27 | 86 | 77 | 21,5 | 92 | 82 | 26,3 | 86 | 77 |
| vote égal | 20,2 | 91 | 84 | 25,1 | 86 | 80 | 19,4 | 91 | 85 | 24,3 | 86 | 80 |
| vote pond. | **20,1** | **92** | **84** | **24,9** | **86** | **79** | **19,4** | **92** | **84** | **24** | **87** | **80** |
| cons. 75% | 22,4 | 93 | 81 | 27,2 | 88 | 76 | 21,6 | 94 | 81 | 25,9 | 89 | 77 |
| cons. 100% | **29** | **95** | **73** | **34,2** | **88** | **68** | **27,3** | **96** | **74** | **31,9** | **90** | **70** |

TABLE 6.23 – Vote et consensus sur systèmes améliorés et avec détection d'erreurs (MEDIA automatique). *Cf.* tab. 6.16 et 6.21. On donne les résultats (taux d'erreur (%), précision (P) et rappel (R)) sur les concepts (C) et les couples concept/valeur (CV).

### 6.3.4 Nouvelle combinaison multi-systèmes de compréhension

Dans cette partie, nous souhaitons procéder à une nouvelle combinaison de systèmes de compréhension (*cf.* section 6.2.3) mise à jour avec les nouveaux résultats obtenus dans cette section. Les résultats sont visibles dans le tableau 6.23.

Nous ne nous limitons pas ici à une seule mesure de confiance comme nous l'avions fait par rigueur dans la section précédente avec les transcriptions bruitées. Les meilleurs résultats sont obtenus avec un corpus automatique pour les CRF et avec un corpus manuel+automatique pour le biRNN-EDA. Des versions avec détection d'erreurs de reconnaissance sont toujours considérées.

Par rapport à la combinaison précédente en section 6.2.3, les NN donnent les meilleurs résultats individuellement. Les poids choisis automatiquement pour le vote pondéré sont de 10, 10, 10, 12 (CRF standard, CRF à détection d'erreurs, NN standard, NN à détection d'erreurs). Cette fois-ci, ce sont les systèmes neuronaux qui ont le plus de poids et non les CRF.

On obtient une légère amélioration en CER/CVER en vote pondéré, et en précision en consensus par rapport au tableau 6.17.

| Configuration | Tab. | D. | APP | CER | CVER |
|---|---|---|---|---|---|
| [**Hahn *et al.*, 2011**] | 3.2 | | manuelles | 23,8 | 27,3 |
| **CRF** | 6.10 | x | automatiques | **19,9** | **25,1** |
| **biRNN-EDA** | 6.9 | x | automatiques | 21,4 | 27,7 |
| **biRNN-EDA** | 6.21 | x | man.+auto. | **20,3** | **25** |
| **vote optimal** | 6.23 | x | auto./ma.+auto. | **19,4** | **24** |

TABLE 6.24 – Comparaison récapitulative entre CRF et biRNN-EDA (MEDIA TEST automatique). Ces systèmes emploient des descripteurs sémantiques et syntaxiques et des descripteurs de mesure de confiance de reconnaissance (D.) dans certains cas.

## 6.4 Conclusion

Deux variantes de deux architectures de compréhension respectivement construites sur les CRF et biRNN-EDA ont été considérées pour observer les conséquences du traitement des transcriptions automatiques.

En premier lieu, les architectures CRF ont surpassé les architectures biRNN-EDA avec une amélioration significative par rapport à la référence de [Hahn *et al.*, 2011]. Les architectures biRNN-EDA semblent être utiles lorsqu'elles sont combinées avec celles des CRF. Les résultats montrent que l'interaction entre les composants de reconnaissance et de compréhension est bénéfique par l'intégration de mesures de confiance de reconnaissance ou par la détection d'erreurs de reconnaissance intégrée à la compréhension.

Combiner une approche de compréhension s'appuyant sur des CRF et une architecture biRNN-EDA permet notamment d'identifier efficacement dans nos sorties sémantiques les îlots de confiance et les segments incertains utiles pour décider des actions appropriées de traitement des erreurs par le gestionnaire de dialogue.

De plus nos expériences sur la simulation d'erreurs et sur l'enrichissement de corpus ont entièrement réduit l'écart de performance entre CRF et système neuronal comme le montre le tableau 6.24.

Les ensembles APP, DEV et TEST sont construits à partir de transcriptions automatiques. Notre étude a montré que les biRNN-EDA obtenaient de meilleurs résultats quand ils sont entraînées sur des transcriptions manuelles et automatiques. Les systèmes neuronaux ont besoin d'une quantité de données d'apprentissage conséquente pour obtenir de bonnes performances. Ils sont également capables de mieux apprendre de données bruitées. En conséquence, l'architecture proposée biRNN-EDA atteint maintenant des performances similaires à celles du CRF.

Une piste supplémentaire consisterait à augmenter d'avantage le corpus APP de nos NN avec des données similaires à MEDIA comme par exemple celles du corpus PORTMEDIA (section 1.2).

Sachant que la reconnaissance de la parole et la compréhension de la parole sont deux tâches gérées par des réseaux neuronaux, une autre piste est de réfléchir à

la connexion de ces réseaux pour former un réseau de neurones complet gérant la totalité du traitement [Serdyuk *et al.*, 2018, Ghannay *et al.*, 2018].

Une simulation d'erreurs de transcription basée sur une mesure de similarité construite à partir de plongements de mots acoustiques et linguistiques a été proposée et utilisée pour bruiter un corpus manuel annoté. Les expériences montrent que ce bruitage est pertinent pour enrichir et préparer un corpus d'entraînement de compréhension. Si aucun corpus automatique n'est disponible pour préparer ces données, notre proposition offre une amélioration très significative des performances des systèmes de compréhension.

CHAPITRE 7

# MÉTA-ÉTIQUETTES HIÉRARCHISÉES ET SYSTÈME MULTI-PASSES

**Sommaire**

Dans le premier chapitre de contributions, nous avons présenté un système neuronal (biRNN-EDA) capable d'effectuer efficacement une tâche de compréhension de la parole (étiquetage en concepts sémantiques).

Dans le chapitre suivant, nous avons abordé la transition difficile mais nécessaire de transcriptions manuelles à automatiques. Pour faciliter cette transition, nous avons introduit plusieurs techniques aidant le système de compréhension à mieux appréhender les erreurs de reconnaissance de la parole : des mesures de confiance, une détection d'erreurs de reconnaissance intégrée au système de compréhension, un enrichissement du corpus d'apprentissage et enfin une combinaison de plusieurs systèmes de compréhension.

Ce troisième chapitre de contribution se concentre sur une hiérarchisation des étiquettes sémantiques. Nous cherchons à réduire la confusion entre les étiquettes sémantiques du corpus MEDIA en réduisant notre jeu d'étiquettes original. Il s'agit de regrouper les étiquettes en classes sémantiques hiérarchiques ou *méta-étiquettes*. Ces regroupements peuvent s'appuyer sur différents critères : ils permettent de réduire le nombre d'étiquettes en fusionnant certaines d'entre elles, en particulier les plus confuses, dans la même méta-étiquette. Réduire le nombre d'étiquettes et donc la complexité de la tâche, facilite la tâche de compréhension dans un premier temps. La définition d'ensembles de méta-étiquettes moins fines est faite dans le but d'obtenir des systèmes de compréhension plus performants à un niveau sémantique plus général. En réduisant la complexité de la tâche, nous nous attendons à ce que le premier passage du système de compréhension soit plus efficace à un niveau conceptuel plus général pour identifier ces méta-étiquettes sans être perturbé par certaines confusions difficiles à désambiguïser.

Nous présentons ensuite une stratégie multi-passes pour améliorer les résultats dans la tâche de compréhension classique en intégrant les sorties de nos différents systèmes de compréhension généralisés. Plusieurs techniques sont explorées pour tirer parti de cette information, comme l'intégration dans les descripteurs de mots ou le vote multi-système. Il est également envisagé que la stratégie appliquée ici puisse être répétée plusieurs fois afin d'obtenir de nouvelles améliorations.

## 7.1 Hiérarchisation des étiquettes : les méta-étiquettes

### 7.1.1 Motivations

La compréhension de la parole, même réduite à une tâche d'étiquetage en concepts sémantiques, et dans un domaine sémantique fixe, reste une tâche très complexe (section 1.1.3.1). Comme vu dans la section 1.1.4, des erreurs de compréhension se produisent lors de la détection d'un mauvais concept mais aussi lorsque l'on détecte le bon concept avec les mauvaises bornes, impactant ainsi l'extraction de la valeur. De plus, comme vu dans les sections 2.2 et 6.1, le passage aux transcriptions automatiques elles-mêmes erronées rajoute un bruit au processus de compréhension.

Nous nous sommes concentrés sur l'amélioration de l'architecture neuronale de

compréhension avec plusieurs techniques appliquées avec succès pour réduire les erreurs : récurrence avant, arrière, bidirectionnelle (section 5.2), mécanisme d'attention (section 5.3).

Mais si ces dernières améliorations se focalisent sur une complexification du système, d'autres améliorations peuvent être apportées en se concentrant sur la tâche de compréhension, en essayant de la simplifier. Dans [Hinton *et al.*, 2015] par exemple, un ensemble de modèles spécialisés apprennent à distinguer des classes affinées que les modèles complets confondent. Avec une approche similaire, nous cherchons dans cette étude à désambiguïser la tâche d'étiquetage afin d'obtenir de meilleurs résultats.

Nous proposons de regrouper l'ensemble initial de concepts sémantiques en concepts plus génériques afin de concentrer la tâche d'étiquetage conceptuel sur des concepts moins fins. De cette façon, nous visons dans un premier temps à faciliter la tâche d'étiquetage. Nous pourrons dissocier la détection difficile d'un concept spécifique d'une détection sémantique plus générale ou encore d'une détection binaire (par exemple *est-ce que cette séquence de mots contient un concept ou non ?*)

### 7.1.2 Définitions des ensembles de méta-étiquettes

L'idée de la détection d'un concept plus généralisé est apparue après l'analyse d'erreurs des paires de confusion les plus fréquentes faites sur le corpus DEV par notre meilleur système de compréhension neuronal (biRNN-EDA, tableau 6.24). Nous observons que des substitutions ont lieu entre des concepts partageant le même champ sémantique. Par exemple, nous avons compté 13 substitutions entre le concept *temps-jour-mois* et le concept *temps-date*, ou encore 5 substitutions entre le concept *nombre-reservation* et le concept *nombre*.

À partir de cette observation, nous cherchons à produire une classification hiérarchique aboutissant à différents regroupements sur la forme de l'étiquette comme décrit dans la figure 7.1.

Nous proposons des ensembles d'étiquettes conceptuelles, nommés *méta-étiquettes*, visant à remplir ce but de regroupement et de simplification des étiquettes de base sur l'idée d'une classification hiérarchique. Ces classes de méta-étiquettes concernent des objectifs particuliers allant de la détection de la segmentation seulement (*concept ou pas de concept ?*) à la détection du concept final en passant par différents niveaux de généralisation.

Nous traduisons cette classification par la définition de ces cinq ensembles de *méta-étiquettes*, un pour chaque niveau de représentation proposé de l'étiquette initiale. Chaque niveau suit son propre objectif qui définit un nouvel ensemble de méta-étiquettes :

1. **Le mot est-il porteur d'un concept ?** Ce niveau est le plus élevé, en ne considérant que 2 méta-étiquettes. Chaque mot faisant partie d'un support de concept est associé à la méta-étiquette *concept*, *null* sinon.
2. **Où est situé le mot par rapport au concept ?** Ce niveau spécifie un peu plus la méta-étiquette *concept* du niveau 1. Si le mot introduit le concept, il

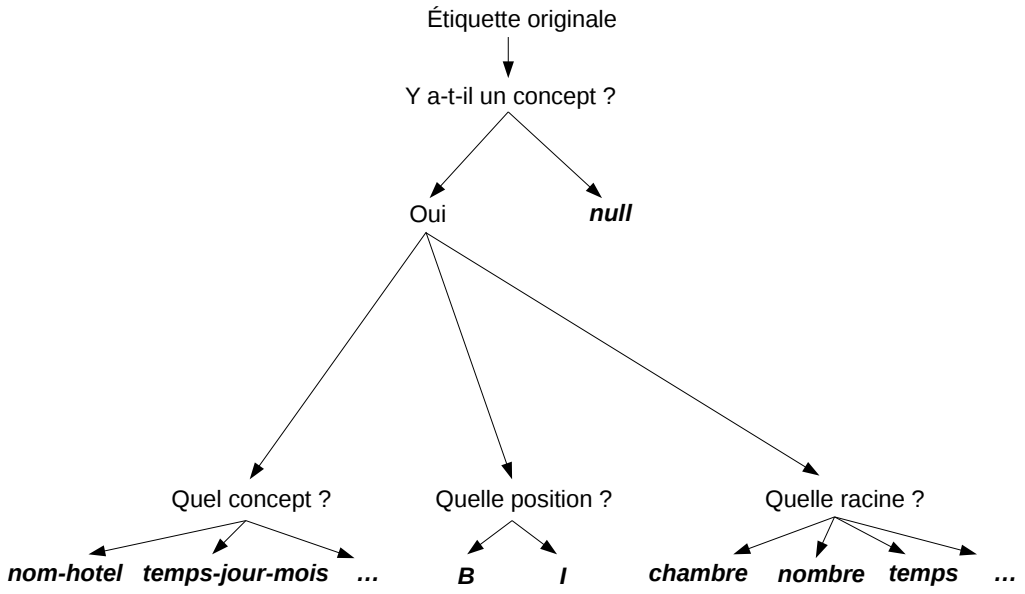



FIGURE 7.1 – Idée de classification hiérarchique des étiquettes sémantiques.

est associé à la méta-étiquette *concept_B* ; sinon, s'il appartient au concept de support, il est associé à *concept_I* ; *null* sinon. Ainsi 3 méta-étiquettes constituent le niveau 2.

3. **Quel concept précis le mot traduit-il ?** Seule l'étiquette du concept est prise en compte, en supprimant le suffixe. Ce troisième niveau contient un jeu de 77 méta-étiquettes.
4. **Quel concept générique transmet le mot ?** Ici, nous avons considéré seulement la racine du label comme méta-étiquette. Par exemple : *chambre-equipement, chambre-fumeur, chambre-type* et *chambre-voisin* sont rassemblés dans la même méta-étiquette *chambre*. Le quatrième niveau contient 22 méta-étiquettes.
5. **Où est situé le mot par rapport au concept générique ?** Ce niveau ajoute le suffixe $B$ et $I$ à la méta-étiquette précédente (sauf pour *null*), il contient 43 méta-étiquettes.

Le tableau 7.1 donne un exemple avec l'étiquette sémantique **nom-hotel** avec le suffixe **I**.

Nous aurions eu la possibilité d'effectuer des regroupements d'étiquettes en fonction des confusions les plus fréquentes faites par le système (si $A$ est souvent confondu avec $B$ alors on considère la méta-étiquette $C$ qui regroupe $A$ et $B$). Grâce à une analyse des paires de confusion, on voit que ces regroupements sont pour la majorité déjà inclus dans l'ensemble de méta-étiquettes n° 4. Certaines confusions d'étiquettes peuvent échapper à la classification sémantique que nous avons défini

| **Étiquette initiale** | | *nom-hotel-I* |
|---|---|---|
| **Ensemble de méta-étiquette** | **1 : présence/absence de concept** | *concept* |
| | **2 : position dans le concept** | *concept_I* |
| | **3 : concept** | *nom-hotel* |
| | **4 : racine du concept** | *nom* |
| | **5 : racine et position** | *nom-I* |

TABLE 7.1 – Exemples de méta-étiquettes.

mais avec un faible nombre d'occurrences et sur des concepts ne présentant pas de similarité visible.

### 7.1.3 Détection des méta-étiquettes

Dans cette section, nous abordons la manière dont nous allons pouvoir détecter les méta-étiquettes précédemment définies.

Le système neuronal classique servant de base à nos expériences est le meilleur système que nous avons obtenu dans le chapitre précédent (tableau 6.24). Ce système emploie tous les descripteurs sémantiques et syntaxiques, des mesures de confiance de reconnaissance de la parole, et s'entraîne sur un corpus automatique enrichi d'un corpus manuel. Nous appellerons ce système : *système standard*.

À titre de comparaison, nous choisissons également de considérer parmi nos méta-étiquettes l'étiquette sémantique initiale que le système de compréhension standard a trouvé (dans notre exemple : *nom-hotel-I*), que l'on note comme "étiquette" dans les tableaux.

Avant de pouvoir utiliser des méta-étiquettes en connaissance préalable d'un système de compréhension, notre première problématique est de détecter ces méta-étiquettes.

Nous considérons deux façons d'obtenir automatiquement des annotations de méta-étiquette. Tout d'abord, un biRNN-EDA spécifique est entraîné pour détecter chaque ensemble de méta-étiquettes telles qu'elles sont définies dans la section précédente. Nous avons défini ces systèmes en tant que *systèmes spécialistes*. D'autre part, par souci de comparaison, nous prenons aussi la sortie du système standard et la convertissons vers les différentes méta-étiquettes. Cela est représenté dans la figure 7.2.

Chacune de ces sorties (provenant de spécialistes ou de conversions) est ensuite évaluée en LER (*cf.* section 1.1.5.2) par rapport à une référence des méta-étiquettes attendues selon le groupe de méta-étiquette en question. Les résultats sont présentés dans le tableau 7.2.

Nous observons que la conversion de la sortie du système standard est majoritairement meilleure qu'une classification faite par un système spécialiste. Contrairement à ce que nous pouvions attendre, entraîner un biRNN-EDA à détecter une méta-étiquette en particulier ne donne pas de meilleurs résultats que l'entraîner à détecter toutes les étiquettes pour ensuite convertir la sortie en méta-étiquettes. Cela

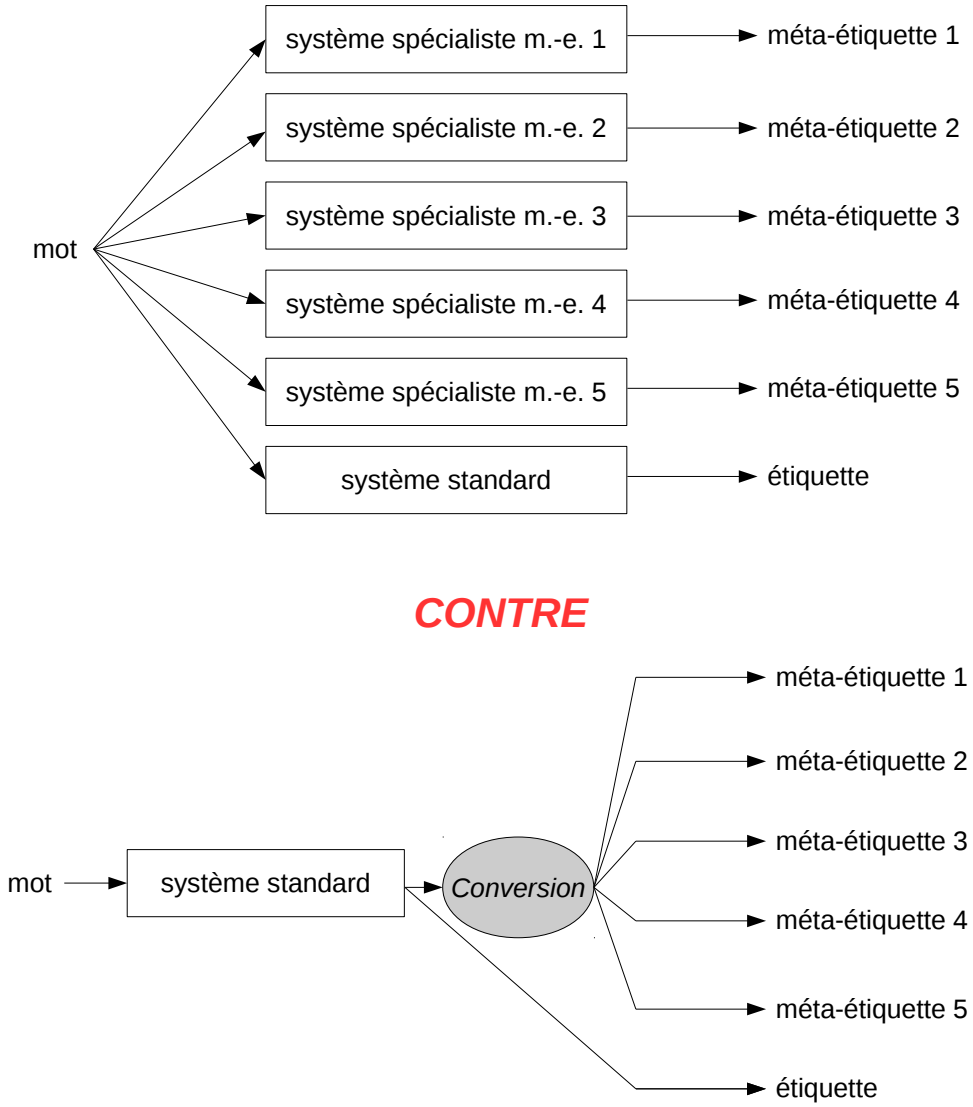



FIGURE 7.2 – Comparaison des modes de détection des méta-étiquettes.

laisse supposer que l'architecture neuronale profite davantage d'un entraînement sur un plus grand ensemble d'étiquettes sémantiques, contenant plus d'informations et de dépendances.

Pour le reste de l'étude, nous conservons par la suite les méta-étiquettes obtenues par conversion et laissons de côté les systèmes spécialistes.

On peut également observer que tous les ensembles de méta-étiquettes atteignent

| | | LER | | | |
|---|---|---|---|---|---|
| | | Conversion | | Spécialisation | |
| Méta-étiquette | #Etiquette | DEV | TEST | DEV | TEST |
| 1 | 2 | **10,7** | **11,1** | 14,1 | 14,4 |
| 2 | 3 | **13,8** | **14,3** | 14,6 | 15,3 |
| 3 | 77 | 14,4 | 14,7 | **13,9** | **14,7** |
| 4 | 22 | **13,4** | **13,9** | 13,7 | 14,2 |
| 5 | 43 | **15,5** | **16,2** | 16,5 | 17,3 |
| étiquette | 151 | **16,5** | **16,9** | **16,5** | **16,9** |

TABLE 7.2 – Comparaison en LER entre la conversion du système standard et les systèmes spécialistes par rapport à tous les ensembles d'étiquettes et méta-étiquettes (MEDIA automatique).

de meilleures performances que l'étiquette initiale en termes de LER. Cela nous conforte dans notre volonté d'utiliser l'information fournie par les méta-étiquettes comme un descripteur de connaissance sémantique préalable. Cette information apparaissant comme moins erronée et plus sémantiquement intéressante.

## 7.2 Intégration des méta-étiquettes

### 7.2.1 Potentiel de l'intégration des méta-étiquettes dans le système de compréhension final

Une des raisons principales qui nous motivent à approfondir la piste des méta-étiquettes est de regarder les résultats potentiels que nous pourrions obtenir grâce à elles.

En admettant que nous parvenions à trouver des méta-étiquettes parfaites et à les intégrer dans un système classique en connaissance préalable (via les descripteurs de mots), nous obtiendrions alors un système oracle disposant d'informations de méta-étiquettes parfaites à la fois sur les corpus APP, DEV et TEST. Un descripteur de méta-étiquettes est représenté sous forme de vecteur one-hot et a donc pour dimension le nombre de méta-étiquettes possible de sa catégorie. Les résultats sont présentés dans la table 7.3.

Nous avons essayé l'intégration d'une seule méta-étiquette (la n° 1 pour une dimension additionnelle de 2) ainsi que l'intégration de toutes les méta-étiquettes (pour une dimension de 2 + 3 + 77 + 22 + 43). L'utilisation d'une seule méta-étiquette permet déjà d'obtenir un gain intéressant avec une réduction relative de 2,7% en CVER sur le corpus TEST. Mais le gain est d'autant plus important avec le cumul des méta-étiquettes permettant d'obtenir cette fois une réduction relative de 12,8%. Les erreurs restantes malgré les informations de méta-étiquettes oracles fournies peuvent être dues à des erreurs de compréhension résiduelles ou encore à des erreurs de transcription automatique. Cela est représenté dans la figure 7.3.

| Ajout de méta-étiquette en descripteur | DEV | | TEST | |
|---|---|---|---|---|
| | **CER** | **CVER** | **CER** | **CVER** |
| non (*cf.* tab. 6.21) | 21,2 | 26,3 | 20,3 | 25 |
| 1 | **16,1** | **23,3** | **16,1** | **23** |
| 1, 2, 3, 4 & 5 | **3,9** | **16,1** | **3,8** | **15,4** |

TABLE 7.3 – Résultats potentiels obtenus avec l'ajout de méta-étiquettes sans erreurs (oracle) ajoutées en connaissance préalable dans un biRNN-EDA (MEDIA automatique).

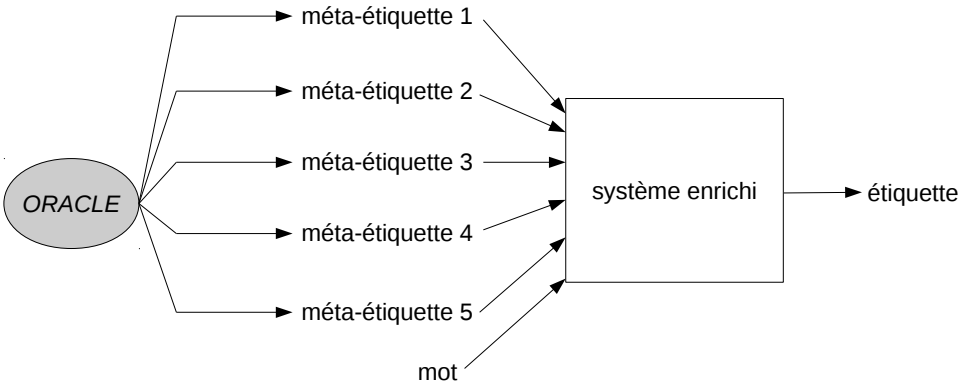



FIGURE 7.3 – Intégration de méta-étiquettes issues d'un système oracle.

Même si les méta-étiquettes sont plus simples que les étiquettes initiales et par conséquent plus faciles à trouver, nous sommes conscients que nous ne parviendrons pas à réaliser une détection des méta-étiquettes sans erreurs. Néanmoins, en prenant conscience des fortes améliorations que peuvent apporter les méta-étiquettes en connaissance préalable dans un système classique, nous sommes encouragés à poursuivre dans cette direction. Nous espérons ainsi obtenir par une application concrète des méta-étiquettes une partie de ce gain potentiel.

Les résultats obtenus en utilisant des méta-étiquettes réellement détectées sont présentés par la suite.

### 7.2.2 Représentation des méta-étiquettes

Disposant de méta-étiquettes (par conversion), nous proposons dans un deuxième temps une extension de notre système standard en y intégrant ces méta-étiquettes en connaissance préalable. Nous testons ici différentes façons de représenter les méta-étiquettes pour les intégrer dans de nouveaux systèmes. Nous n'intégrons pour l'instant qu'une seule méta-étiquette à la fois comme connaissance

préalable d'un nouveau système comme cela est montré dans la figure 7.4.

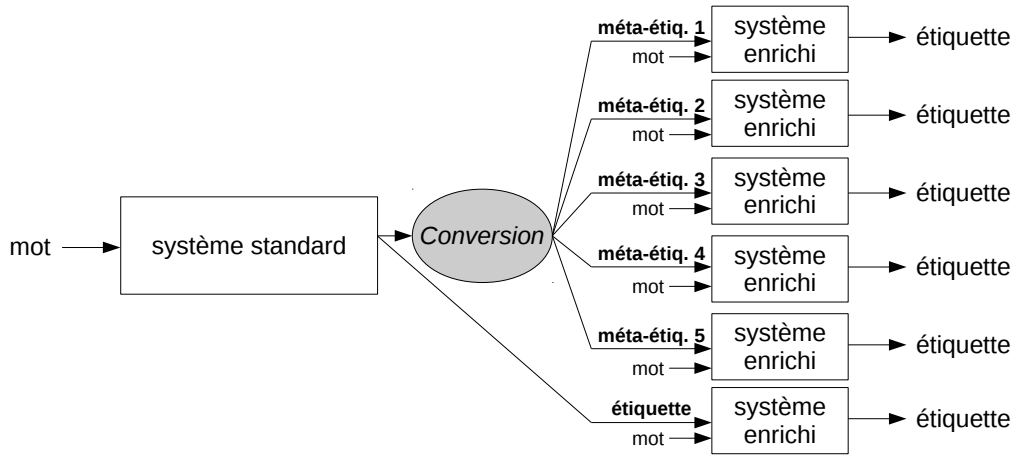



FIGURE 7.4 – Intégration de méta-étiquettes en connaissance préalable, une à la fois.

#### 7.2.2.1 Sous la forme d'un vecteur one-hot

Classiquement, nous choisissons de représenter les descripteurs méta-étiquettes sous forme one-hot comme ce fut le cas pour les descripteurs sémantiques et syntaxiques (section 5.3.3).

Un vecteur one-hot représentant la méta-étiquette est concaténé au vecteur descripteur d'entrée du système standard dans sa deuxième passe d'exécution (système enrichi). Le tableau 7.4 présente les résultats obtenus en ajoutant une seule méta-étiquette one-hot en descripteur en plus des autres descripteurs déjà décrits [1].

Comme on peut le voir, l'information apportée par les méta-étiquettes permet d'atteindre quasi-systématiquement de meilleures performances sur les ensembles DEV et TEST (excepté avec les méta-étiquettes n° 1 et n° 3 où les résultats se dégradent légèrement en TEST). De meilleurs résultats sont obtenus par rapport au système standard et également comparés à l'utilisation de l'étiquette classique (trouvée par le système standard) en descripteur.

#### 7.2.2.2 Sous la forme d'un vecteur de score

Dans cette section, nous considérons un système de compréhension enrichi avec une méta-étiquette ajoutée en descripteur sous la forme d'un vecteur de score que

1. Les dimensions de descripteurs supplémentaires apportées par les méta-étiquettes 1 à 5 sont respectivement 2, 3, 77, 22, 43 et 151 pour l'étiquette d'origine.

| Ajout de méta-étiquette en descripteur | DEV | | TEST | |
|---|---|---|---|---|
| | CER | CVER | CER | CVER |
| 1 | **20,6** | **25,6** | 20,3 | 25,4 |
| 2 | **20,6** | **25,7** | **19,6** | **24,7** |
| 3 | **20,8** | **25,9** | 20,3 | 25,1 |
| 4 | **20,8** | **25,5** | **19,9** | **24,7** |
| 5 | **20,4** | **25,5** | **20** | 25 |
| étiquette | **20,4** | **25,5** | **20** | **24,8** |
| aucune (*cf.* tab. 6.21) | 21,2 | 26,3 | 20,3 | 25 |

TABLE 7.4 – Présentation des systèmes enrichis avec des méta-étiquettes sous forme one-hot, une seule à la fois (MEDIA automatique).

nous opposons à celui précédemment obtenu sous forme one-hot.

Le descripteur de méta-étiquette peut également être mis dans une représentation de score au lieu d'une représentation one-hot. En effet, lors de la conversion d'étiquette à méta-étiquette pour une classe de méta-étiquette déterminée, en additionnant pour une méta-étiquette donnée $M$ tous les scores de sortie des étiquettes $E$ qui sont converties vers $M$, on obtient le score de la méta-étiquette :

$$score(M) = \sum_{\forall E \subset M} score(E)$$

avec :

$$\sum_{\forall M} score(M) = 1$$

Il est important de faire cette comparaison entre une représentation one-hot et sous forme de score. En effet, les réseaux de neurones peuvent accepter ces deux représentations. De plus, nous avons vu qu'une représentation continue sous forme de score peut apporter une amélioration comme ce fut le cas avec les mesures de confiance de reconnaissance (section 6.2.1).

La figure 7.5 donne un exemple sur une phrase des scores obtenus par les méta-étiquettes de type n° 1 (*concept* ou *null*).

Le score agit également comme une mesure de confiance prédisant la probabilité de la présence d'un concept. Parfois la décision est précise (un score élevé, rouge, contre un score faible, bleu) alors que dans d'autres cas on observe des zones d'incertitudes.

La représentation score est testée dans la table 7.5 sur la méta-étiquette n° 4 (la meilleure selon le corpus DEV dans le tableau 7.4).

Le résultat nous oriente néanmoins à préférer la représentation one-hot.

### 7.2.3 Apprentissage mêlant hypothèses et références

Comme vu dans la section 6.3.3, des améliorations sont obtenues dans le système de compréhension neuronal en combinant des données d'entraînement manuelles

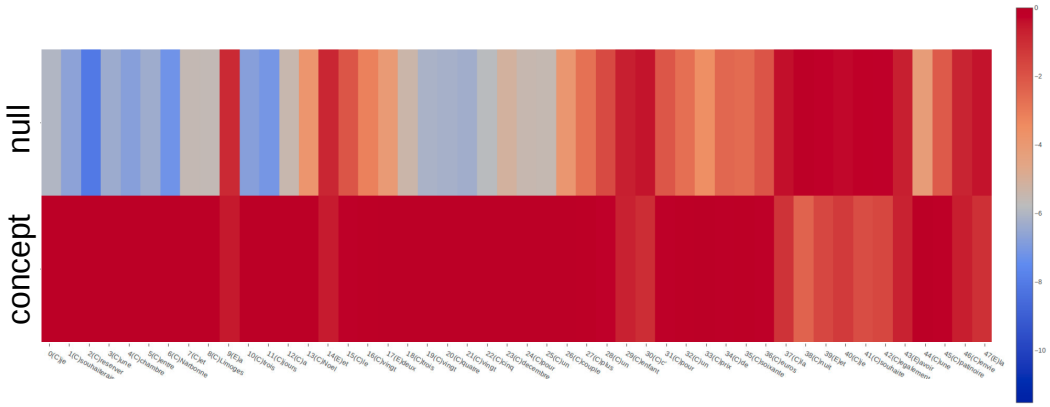



FIGURE 7.5 – Exemples de scores obtenus pour la méta-étiquette n° 1 (*null* ou *concept*). En abscisses les mots en entrée. En ordonnées les méta-étiquettes soit de haut en bas : *null* et *concept*. Plus la couleur est rouge et plus le score est élevé.

| | DEV | | TEST | |
|---|---|---|---|---|
| **Représentation** | **CER** | **CVER** | **CER** | **CVER** |
| one-hot (*cf.* tab. 7.4) | **20,8** | **25,5** | **19,9** | **24,7** |
| score | 20,2 | 25,8 | 20,2 | 25,1 |

TABLE 7.5 – Comparaison entre représentation de méta-étiquette one-hot et score sur la méta-étiquette n° 4 (MEDIA automatique).

(sans erreur) et automatiquement transcrites, augmentant la taille du corpus d'entraînement original.

Alors que les résultats présentés dans la table 7.4 utilisent des descripteurs de méta-étiquettes hypothétiques (trouvés par le système standard puis conversion), nous avons essayé de reproduire cette approche en utilisant dans le corpus d'apprentissage aussi bien des méta-étiquettes hypothèses (trouvées) que des références (sans erreur) ou encore les deux à la fois.

Ces résultats sont présentés dans la table 7.6 sur la méta-étiquette n° 4 (la meilleure selon le corpus DEV dans le tableau 7.4). Ils n'apportent cependant aucune amélioration.

| | DEV | | TEST | |
|---|---|---|---|---|
| **Descripteur méta-étiquette** | **CER** | **CVER** | **CER** | **CVER** |
| hypothèse (*cf.* tab. 7.4) | **20,8** | **25,5** | **19,9** | **24,7** |
| référence | 20,9 | 25,9 | 20,4 | 25,1 |
| hypothèse+référence | 20,9 | 25,8 | 20,3 | 25 |

TABLE 7.6 – Comparaison entre utilisation d'hypothèse et de référence sur la méta-étiquette n° 4 (MEDIA automatique).

À noter que l'apport potentiel vu en section 7.2.1 ne peut être retrouvé ici étant donné que l'on ne donne des méta-étiquettes de référence qu'au corpus APP (et non au DEV et au TEST). Nous ne conservons donc que l'utilisation de descripteurs d'hypothèses de méta-étiquettes.

### 7.2.4 Intégrer plusieurs ensembles de méta-étiquettes

Le tableau 7.4 a présenté les résultats obtenus par des systèmes enrichis avec une méta-étiquette et a montré que l'on obtenait globalement des améliorations pour chacune d'entre elles. Il est donc logique d'essayer de combiner ces améliorations en intégrant plusieurs méta-étiquettes en connaissance préalable. Deux façons de combiner les méta-étiquettes sont envisagées ici : par addition des descripteurs dans un seul système enrichi ou bien par vote de plusieurs systèmes enrichis. Cela est représenté dans la figure 7.6.

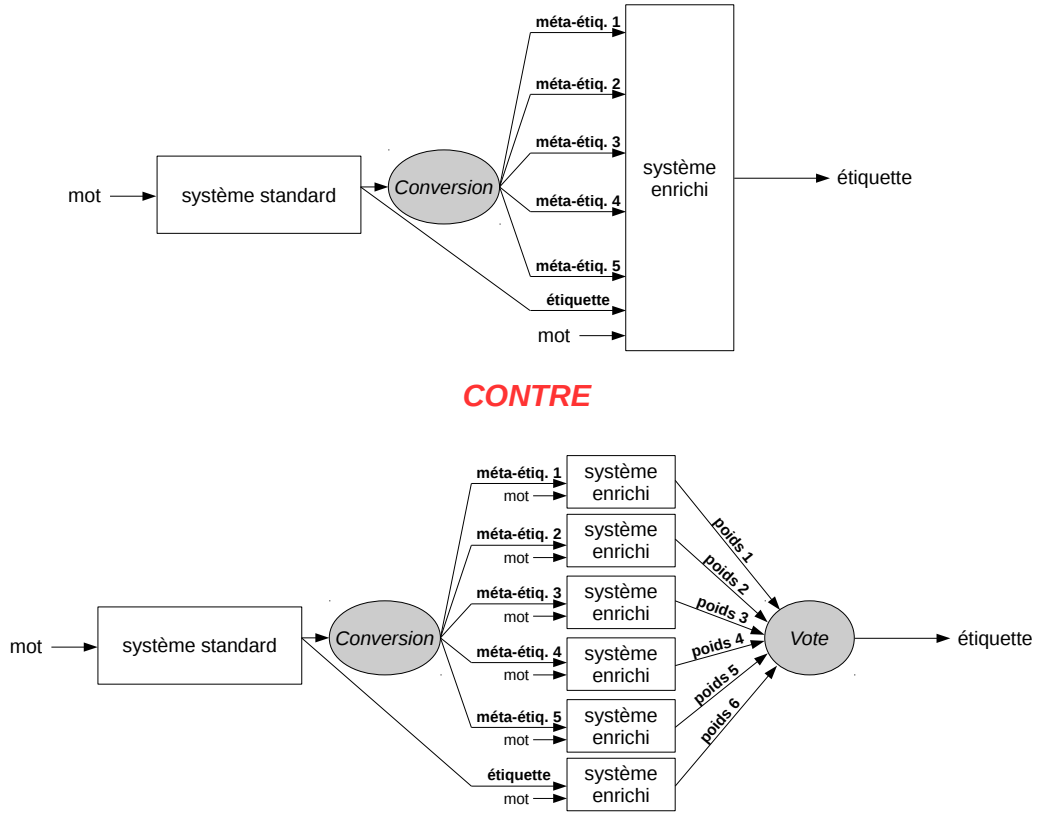



FIGURE 7.6 – Comparaison des modes de combinaison des méta-étiquettes.

| Descripteur(s) méta-étiquette(s) | DEV | | TEST | |
|---|---|---|---|---|
| | **CER** | **CVER** | **CER** | **CVER** |
| 4 (meilleur seul, *cf.* tab. 7.4) | **20,8** | **25,5** | **19,9** | **24,7** |
| tous | 20,5 | 25,6 | 20,1 | 25,1 |

TABLE 7.7 – Impact de l'utilisation simultanée de descripteurs de méta-étiquettes (MEDIA automatique).

#### 7.2.4.1 Par addition

Une première technique déjà employée pour les descripteurs sémantiques et syntaxiques est l'addition de descripteurs. Nous essayons donc ici d'ajouter plusieurs méta-étiquettes en les intégrant en même temps au vecteur de descripteurs en entrée du système.

Le tableau 7.7 montre les résultats obtenus par l'usage simultané des descripteurs de méta-étiquettes [2].

Si les descripteurs de mots sémantiques et syntaxiques peuvent être combinés avec succès (tableau 5.13), les descripteurs de méta-étiquettes semblent introduire du bruit dans le processus de compréhension une fois combinés. En effet aucune amélioration n'est observée avec la combinaison de tous les descripteurs si on se compare à l'intégration seule de la méta-étiquette permettant d'atteindre les meilleurs résultats.

#### 7.2.4.2 Par vote

Nous avons vu qu'un système biRNN-EDA enrichi par l'ajout d'un descripteur représentant une méta-étiquette apportait une amélioration intéressante par rapport au système standard mais que nous ne pouvions pas additionner leurs performances pour obtenir une amélioration plus importante en mettant plusieurs descripteurs de méta-étiquette en entrée. Nous devons donc chercher un autre moyen de combiner les méta-étiquettes pour tirer parti de leurs forces individuelles. Une autre méthode pour tirer profit de ces méta-étiquettes est de considérer plusieurs systèmes enrichis qui fonctionnent ensemble grâce à un vote.

Dans cette section, nous considérons chaque système de compréhension enrichi avec une méta-étiquette comme un expert particulier de la tâche d'étiquetage de concept. En effet, comme nous l'avons vu dans la section 7.1.2, chaque méta-étiquette remplit un objectif distinct. Nous voulons profiter de ces points de vue multiples pour résoudre cette tâche complexe afin de prendre une décision plus précise à un stade ultérieur.

Ainsi, nous avons choisi de combiner les six sorties des systèmes de compréhension enrichis présentés dans le tableau 7.4 par un vote pour prendre une décision

2. Les nombreuses combinaisons possibles nécessitent un temps de calculs important et n'ont pas été testées.

| | DEV | | TEST | |
|---|---|---|---|---|
| **Configuration** | **CER** | **CVER** | **CER** | **CVER** |
| système standard (*cf.* tab. 6.21) | 21,2 | 26,3 | 20,3 | 25 |
| meilleur système enrichi (n° 4, *cf.* tab. 7.4) | 20,8 | 25,5 | 19,9 | 24,7 |
| vote *égal* des systèmes enrichis | 20,1 | 25,2 | 19,7 | 24,6 |
| vote *pondéré* des systèmes enrichis | **19,8** | **24,8** | **19,6** | **24,5** |

TABLE 7.8 – Vote entre systèmes enrichis (MEDIA automatique).

finale sur l'étiquette classique à choisir. Deux types de votes sont effectués. Le premier est un vote égal donnant le même poids à tous les systèmes. Le second est un vote pondéré : il s'agit d'un vote plus intelligent qui ajuste le poids de chaque système sur le corpus DEV (basé sur le CVER) pour obtenir les meilleurs performances. Le tableau 7.8 montre les résultats obtenus avec ce vote.

Avec un vote égal, un premier gain est observé par rapport à chaque système enrichi seul. Le vote pondéré nous mène à une nouvelle amélioration. Nous obtenons ainsi sur le corpus TEST et en CVER un gain relatif de 0,3% par rapport au meilleur système enrichi seul et 0,7% par rapport au système standard.

### 7.2.5 Réitération de la stratégie

Dans les sections précédentes, nous avons obtenu une amélioration de nos performances grâce à une approche multi-passes de généralisation de la tâche de compréhension dont voici le bilan :

1. La première passe consiste à exécuter un système de compréhension standard et à convertir ses étiquettes en méta-étiquettes plus générales et se focalisant sur un aspect précis de la compréhension (section 7.1.3).
2. La deuxième passe consiste à intégrer ces méta-étiquettes individuellement en connaissance préalable (via les descripteurs) dans de nouveaux systèmes de compréhension. Nous avons donc autant de systèmes enrichis que de types de méta-étiquettes (section 7.2.2.1).
3. La troisième passe consiste à combiner les forces de chaque système enrichi grâce à un vote qui produit un nouvel étiquetage plus efficace que les systèmes précédents ou encore que tous les systèmes enrichis pris individuellement (section 7.2.4.2).
4. Dans cette section, nous cherchons à réitérer les étapes 1, 2 et 3 précédemment définies afin de voir si nous pouvons améliorer encore nos résultats à partir du nouvel étiquetage produit par le vote.

La figure 7.7 schématise notre système multi-passes final.
Par conséquent, la réitération de la stratégie multi-passes consiste en :

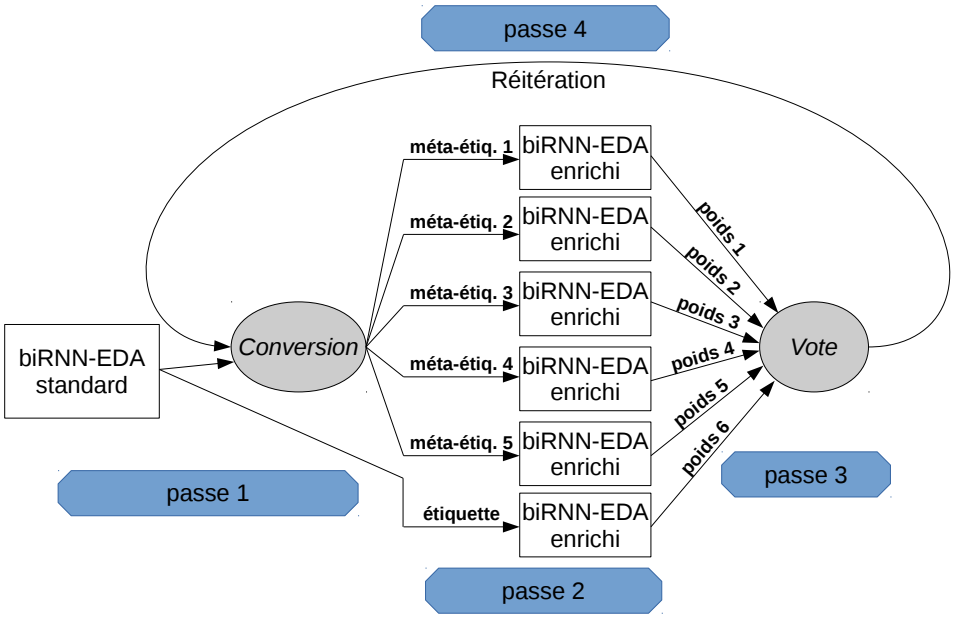



FIGURE 7.7 – Schéma récapitulatif des systèmes de compréhension enrichis par les méta-étiquettes dans une architecture multi-passes.

**Seconde itération, passe 1** Nous partons de l'étiquetage produit par le vote de la première itération. Nous procédons à la conversion en méta-étiquettes de cet étiquetage.

**Seconde itération, passe 2** Grâce aux méta-étiquettes obtenues, nous produisons nos six systèmes enrichis dont les résultats figurent dans le tableau 7.9.

Les résultats sont globalement meilleurs que ceux de la première itération (tableau 7.4). Cette fois-ci seule la méta-étiquette n° 1 présente des résultats moins bons qu'un système non enrichi (dans la première itération, c'était le cas pour les méta-étiquettes n° 1 et n° 3).

**Seconde itération, passe 3** Le second vote conduit à des améliorations sur le corpus DEV et le corpus TEST comme le montre la table 7.10.

Ici encore nous effectuons un vote égal et un vote pondéré comme défini dans la première itération (les poids optimaux peuvent donc être différents des précédents). En revanche, si l'on voit une forte amélioration sur le corpus DEV, l'amélioration obtenue sur le corpus TEST est mineure. Il semble que nous ayons atteint un plateau sur le corpus TEST. Cette amélioration prononcée du corpus DEV se stabilisant sur le corpus TEST laisse penser à un problème de sur-spécialisation sur le corpus de développement. Cela nous amène à conclure que nous ne pourrons pas tirer plus

| Ajout de méta-étiquette en descripteur | DEV | | TEST | |
|---|---|---|---|---|
| | CER | CVER | CER | CVER |
| 1 | 20,4 | 25,3 | *20,3* | *25,3* |
| 2 | 20 | 24,9 | 20 | 24,8 |
| 3 | 19,9 | 24,9 | 19,6 | 24,5 |
| 4 | 19,9 | 24,8 | 20 | 24,7 |
| 5 | 20,1 | 25 | 19,6 | 24,6 |
| étiquette | 20 | 25 | 19,8 | 24,9 |
| aucune (*cf.* tab. 6.21) | 21,2 | 26,3 | 20,3 | 25 |

TABLE 7.9 – Présentation des systèmes enrichis avec des méta-étiquettes sous forme one-hot, une seule à la fois, 2ème itération (MEDIA automatique).

| Configuration | DEV | | TEST | |
|---|---|---|---|---|
| | CER | CVER | CER | CVER |
| meilleur vote, 1ère itération (*cf.* tab. 7.8) | 19,8 | 24,8 | 19,6 | 24,5 |
| vote *égal* des systèmes enrichis | 19,7 | 24,5 | 19,6 | 24,5 |
| vote *pondéré* des systèmes enrichis | **19,3** | **24,2** | **19,5** | **24,4** |

TABLE 7.10 – Vote entre systèmes enrichis, 2ème itération (MEDIA automatique).

partie de notre approche multi-passes via une troisième itération.

## 7.3 Conclusion

Nous avons remarqué, par une analyse des paires de confusion de nos expériences précédentes, que certaines étiquettes sont souvent confondues. Nous avons supposé que ces confusions locales peuvent introduire du bruit dans le processus d'étiquetage global d'une phrase. Dans cette étude, nous présentons une approche multi-passes basée sur l'utilisation d'ensembles d'étiquettes plus génériques, donc moins confuses. Cela nous permet d'obtenir des premières sorties moins erronées, selon les méta-étiquettes utilisées à cette étape, et qui peuvent apporter des informations utiles, notamment la détection de frontières de séquences de mots supportant des concepts sémantiques, pour les prochaines itérations.

Les expériences ont montré que la meilleure approche pour obtenir des méta-étiquettes fiables à la première passe consiste à étiqueter avec les étiquettes initiales définies par la tâche (système standard), puis à convertir ces sorties en méta-étiquettes avant de fournir cette information à la passe suivante. Différents ensembles de méta-étiquettes peuvent être définis afin de former plusieurs systèmes de compréhension enrichis par une méta-étiquette.

Enfin, une stratégie de combinaison est appliquée pour tirer profit de chaque méta-étiquette. La réitération peut apporter des améliorations mineures, mais elle

est rapidement limitée par un problème de sur-apprentissage.

La piste de la désambiguïsation des étiquettes grâce aux méta-étiquettes est prometteuse. Des travaux supplémentaires pourraient être effectués dans cette voie. Nous avons ici utilisé des méta-étiquettes choisies arbitrairement à partir de notre propre analyse du corpus. Nous aurions pu en revanche essayer de laisser le système choisir lui-même ses regroupements d'étiquettes par une projection des plongements d'étiquettes (section 4.2.1) produits à la sortie de nos réseaux neuronaux.

# CHAPITRE 8

# CONCLUSION ET PERSPECTIVES

Le travail de cette thèse a consisté en une exploration de l'utilisation de réseaux de neurones pour la compréhension de la parole à travers une tâche d'étiquetage en concepts sémantiques dans le cadre d'un système de dialogue oral.

Dans le premier chapitre, nous avons présenté la tâche de compréhension de la parole. Comprendre la parole est un objectif complexe qui se décline en plusieurs conceptions et définitions différentes et donc plusieurs tâches au sens informatique. Nous choisissons, en rapport avec notre propre cadre applicatif de système de dialogue oral, d'entendre la compréhension comme une tâche d'étiquetage en concepts sémantiques dans lequel nous recherchons une représentation de la phrase en couple concept et valeur. Dans ce chapitre, nous présentons également un inventaire de différents corpus classiquement utilisés et de différentes mesures d'évaluations dont le CER/CVER que nous retenons pour évaluer nos systèmes.

Dans le deuxième chapitre, la reconnaissance automatique de la parole est décrite succinctement. Cette tâche, différente de la compréhension, est essentielle car elle en constitue l'entrée dans un cadre pratique de système de dialogue oral. Nous présentons également les mesures de confiance de reconnaissance de la parole que nous utilisons dans cette thèse.

Dans le troisième chapitre, nous avons présenté des modèles classiques d'apprentissage automatique supervisé antérieurs à ceux de l'ère neuronale et permettant d'accomplir notre tâche. Il en ressort que les systèmes à base de CRF étaient les plus performants jusqu'alors sur cette tâche et qu'ils constituent l'approche à l'état de l'art au commencement de cette thèse. En effet, cette thèse a débuté en 2015, à un moment où les réseaux de neurones avaient déjà fait leurs preuves dans beaucoup de tâches de traitement du langage naturel mais assez peu en compréhension de la parole. Les CRF sont donc choisis comme référence de comparaison pour les contributions neuronales que nous avons développées.

Dans le quatrième chapitre, nous avons présenté les modèles de classification neuronaux. Après en avoir fait une description théorique abordant leur principe de

fonctionnement, leur mode d'apprentissage et certaines de leurs architectures, nous avons présenté une utilisation concrète des modèles neuronaux en compréhension de la parole avec l'étude de Grégoire Mesnil [Mesnil *et al.*, 2013, Mesnil *et al.*, 2015] qui a servi de base au commencement de cette thèse. Nous concluons que les systèmes les plus performants étaient alors les réseaux récurrents bidirectionnels qui extraient des informations contextuelles du passé et du futur de la phrase afin de mieux traiter la tâche d'étiquetage en concepts sémantiques. Ils ont été capables de surpasser les modèles CRF sur une tâche d'étiquetage en concepts sémantiques. Néanmoins cette conclusion obtenue sur le corpus ATIS ne se renouvelle pas sur le corpus plus complexe MEDIA comme le montre l'étude de Vedran Vukotic [Vukotic *et al.*, 2015], une seconde étude importante au début de cette thèse, qui a montré que les réseaux de neurones, bien qu'étant prometteurs, requièrent une étude approfondie et de nouvelles améliorations sur la tâche de compréhension de la parole. Il est alors conclu que le corpus MEDIA est plus favorable à l'évaluation de nos systèmes car il représente un défi plus intéressant à relever, tandis qu'ATIS semble avoir atteint ses limites d'exploitation [Béchet et Raymond, 2018].

Le cinquième chapitre présente les contributions de cette thèse dont les principaux résultats sont présentés dans le tableau 8.1 Après avoir mis en place avec succès un système neuronal état de l'art de compréhension de la parole inspiré de celui de l'étude de Grégoire Mesnil, nous y avons apporté un mécanisme d'attention, une technologie issue de la traduction automatique et qui montrait alors des résultats prometteurs. Ce mécanisme d'attention nous permet d'aller plus loin en encourageant le réseau de neurones à se focaliser sur certaines parties utiles, qui représentent la phrase analysée, pour l'étiquetage du mot en cours. De plus, nous avons mis en place l'utilisation de descripteurs syntaxiques et sémantiques apportant au système de compréhension un soutien important. Cela nous a conforté à poursuivre la recherche de nouveaux descripteurs dans les études suivantes.

Dans le sixième chapitre, nous avons étudié les conséquences du traitement des transcriptions automatiques. En effet, le chapitre précédent ne traitait que des transcriptions manuelles. L'interaction entre les composants de reconnaissance et de compréhension est nécessaire pour pallier les erreurs de reconnaissance automatique : cela est possible par l'intégration de mesure de confiance de reconnaissance ou par la détection d'erreurs de reconnaissance intégrée au mécanisme de compréhension. Combiner une approche de compréhension basée sur des CRF avec une architecture neuronale a aussi permis de surpasser les deux modèles pris séparément mais aussi d'identifier efficacement les îlots de confiance et les segments sémantiques incertains utiles pour décider des actions appropriées de traitement des erreurs par le gestionnaire de dialogue. De plus nous avons mené des expériences sur la simulation d'erreurs et sur l'enrichissement de corpus qui ont permis de réduire l'écart de performance entre CRF et système neuronal.

Dans le septième et dernier chapitre, nous avons proposé une technique de désambiguïsation de la tâche de compréhension. Cela passe par une classification hiérarchique des étiquettes sémantiques recherchées aboutissant à un regroupement en méta-étiquettes moins confuses. L'utilisation des ces étiquettes plus générales peut

| Transcriptions | | Modèle | TEST | |
|---|---|---|---|---|
| **Développement & Test** | **Apprentissage** | | **CER** | **CVER** |
| Manuelles | Manuelles | CRF | 10,7 | - |
| | | Neuronal | 11,3 | - |
| Automatiques | Automatiques | CRF | 19,9 | 25,1 |
| | Automatiques +Manuelles | Neuronal | 20,3 | 25 |
| | | Neuronal généralisé | 19,5 | 24,4 |

TABLE 8.1 – Présentation des principaux résultats des contributions de cette thèse (Corpus MEDIA).

ainsi rendre les systèmes plus performants et les informations produites peuvent être utilisées à travers une approche de vote multi-passe.

## 8.1 Perspectives

Les contributions apportées par cette thèse nous ont amené à envisager de nouveaux domaines d'études ou à compléter davantage ceux ayant déjà été étudiés. Cette section vise à décrire ces pistes à envisager afin de poursuivre les recherches entamées dans cette thèse.

**Tendre vers un système de transcription/compréhension de bout en bout**
La reconnaissance de la parole et la compréhension de la parole sont deux tâches gérées par des réseaux neuronaux. Nous avons dans un premier temps exécuté le processus de reconnaissance de la parole sur les données audios puis fourni les données transcrites produites au système de compréhension de la parole. Comme nous l'avons vu, cette manière de faire a l'inconvénient de propager les erreurs de reconnaissance dans le système de compréhension et nous a imposé de chercher des stratégies de correction de ces erreurs au sein du processus de compréhension.

Une nouvelle piste est de réfléchir à la connexion de ces réseaux pour former un réseau de neurones complet gérant la totalité du traitement de la parole à la manière de [Serdyuk *et al.*, 2018] dans le domaine de la classification d'appel, une tâche néanmoins moins complexe que l'étiquetage en concepts sémantiques. Ce système neuronal complet s'affranchit de la représentation textuelle classique entre la sortie du module de reconnaissance et l'entrée de celui de compréhension. Cela permet également de procéder à une optimisation jointe des deux systèmes de reconnaissance et de compréhension au lieu de le faire séparément, et pouvant aboutir à de meilleurs résultats. Vers la fin de cette thèse, j'ai été en mesure de participer à une étude relative à la génération d'un système de compréhension de bout en bout sur des tâches d'extraction d'entités nommés et d'étiquetage sémantique présentant des résultats prometteurs ([Ghannay *et al.*, 2018], *cf.* publications 8.1).

**Enrichir l'apprentissage** Les modèles neuronaux ont montré qu'ils sont plus efficaces avec davantage de données d'apprentissage, contrairement aux CRF. Une piste supplémentaire consisterait à augmenter encore le corpus d'apprentissage de nos modèles neuronaux avec des données similaires à celles que nous utilisons dans MEDIA. Cela pourrait être dans un premier temps celles du corpus PORTMEDIA, des données d'un domaine proche mais différent. Dans [Ghannay *et al.*, 2018], les données PORTMEDIA ont été utilisées pour enrichir l'entraînement de systèmes de compréhension sur MEDIA dans une approche de bout en bout partant de sorties audios.

**Approfondir la tâche MEDIA** Le corpus MEDIA est un corpus à l'interprétation particulièrement difficile. Il serait intéressant de travailler autour des concepts les plus représentés dans les zones de non consensus des systèmes neuronaux et CRF. De plus certains concepts sont ambiguës pour les systèmes que l'on se trouve en configuration manuelle ou automatique. Une analyse des erreurs plus approfondie sur ces cas particuliers est un défi important pour les travaux futurs.

Les étiquettes recherchées dans cette thèse ont été déterminées manuellement (par l'humain). Il pourrait être intéressant de laisser le système neuronal décider lui-même des classes à rechercher, et de constater s'il rencontre toujours des concepts ambiguës et difficiles à classer. Cela pourrait être accompli par un regroupement distributionnel de nos plongements d'étiquettes produits à la sortie de nos réseaux neuronaux. L'étude [Camelin *et al.*, 2011] a montré que les étiquettes MEDIA obtenues automatiquement par un procédé automatique de *clustering* tendent à se rapprocher de celles des références manuelles.

De même, la piste de la désambiguïsation des étiquettes grâce aux méta-étiquettes est prometteuse. Des travaux supplémentaires pourraient être effectués dans cette voie. Nous avons ici utilisé des méta-étiquettes choisies arbitrairement à partir de notre propre analyse du corpus mais nous aurions pu essayer de laisser le système choisir lui-même ses regroupements d'étiquettes par une projection des plongements d'étiquettes.

**Améliorer le mécanisme d'attention** Nous pourrions concevoir un mécanisme d'attention plus spécifique à la tâche d'étiquetage en concepts sémantiques. Le mécanisme d'attention est performant mais initialement conçu pour la traduction automatique où les informations utiles pour la traduction d'un mot peuvent venir d'endroits très variés (contexte large) dans la phrase d'origine dans la limite des contraintes grammaticales de la langue. Dans l'étiquetage, plusieurs mots supportent un concept c'est-à-dire que plusieurs étiquettes représentent un concept. Parfois toute l'information nécessaire pour bien étiqueter un support de mots est dans le support et dans ce cas le contexte extérieur rajoute de l'erreur. En revanche, le contexte extérieur est parfois utile. Comme nous l'avons vu dans l'exemple de la phrase *"celui qui est à cent trente euros la chambre"* : afin d'associer le support de mots *"cent trente"* au concept *"paiement-montant-entier"*, il faut effectivement re-

garder l'information monétaire *"euros"* à l'extérieur du support. Mais pour associer le support de mot *"euros"* au concept *"paiement-monnaie"*, le support seul suffit et tout autre information pourrait ajouter de l'erreur.

De plus le mécanisme d'attention nous permet de sélectionner les informations les plus utiles dans une phrase afin d'étiqueter un mot mais il est aussi à envisager que les informations des tours de dialogue précédents contiennent également une information utile pour mieux comprendre la phrase en cours. Cela nous amène à nous interroger sur la représentation sémantique globale d'un tour de dialogue. Ainsi, les orientations suggérées pour les travaux futurs devraient envisager de nouveaux mécanismes d'attention structurés capables de sélectionner des descripteurs de contextes éloignés dans l'historique d'une conversation. L'objectif est d'identifier un ensemble suffisant de descripteurs de contexte pour désambiguïser les mentions de concepts locaux.

**Étendre la compréhension** Au cours de cette thèse, nous avons travaillé sur le corpus MEDIA nous limitant à un cadre applicatif sémantique particulier qui est la réservation de chambre d'hôtel et les informations touristiques. Une nouvelle piste serait d'apprendre de nouveaux cadres applicatifs pour tendre vers une compréhension plus ouverte ne se limitant pas à la compréhension de la parole d'une tâche spécifique mais ayant une compréhension plus générale. Cela pourrait être rendu possible grâce à des techniques connues qui consisteraient à entraîner un système par tâche différente et de disposer d'un système de routage vers le système de la tâche concernée ou au contraire d'entraîner un seul système pour toutes les tâches. On peut citer comme exemple l'étude de [Griol *et al.*, 2016] utilisant un modèle pour la prédiction de l'intention de l'utilisateur et un second modèle utilisant cette prédiction et l'historique du dialogue pour donner une réponse appropriée à l'aide d'un ensemble de classifieurs entraînés pour différentes tâches. Dans notre cas, ces différentes tâches de compréhension pourraient dans un premier temps être celles décrites dans les différents corpus du premier chapitre.

# BIBLIOGRAPHIE PERSONNELLE

**SLUNIPS 2015** Edwin Simonnet, Paul Deléglise, Nathalie Camelin, Yannick Estève. *Exploring the use of Attention-Based Recurrent Neural Networks For Spoken Language Understanding.* 29th Conference on Neural Information Processing Systems, NIPS 2015, Machine Learning for Spoken Language Understanding and Interaction workshop (SLUNIPS). Montréal, Canada. 7-12 Décembre 2015. Présentation poster.

**JEP 2016** Edwin Simonnet, Paul Deléglise, Nathalie Camelin, Yannick Estève. *Des Réseaux de Neurones avec Mécanisme d'Attention pour la Compréhension de la Parole.* 31ème Journées d'Études sur la Parole, JEP 2016. Paris, France. 4-8 Juillet 2016. Présentation orale.

**JDOC 2017** Edwin Simonnet. *Neural Networks for Spoken Language Understanding.* 17ème Journée des Doctorants de l'ED STIM, JDOC 2017. Nantes, France. 4 Mai 2017. Présentation orale.

**INTERSPEECH 2017** Edwin Simonnet, Sahar Ghannay, Nathalie Camelin, Yannick Estève, Renato De Mori. *ASR error management for improving spoken language understanding.* Interspeech 2017. Stockholm, Suède. 20-24 Août 2017. Présentation poster.

**LREC 2018** Edwin Simonnet, Sahar Ghannay, Nathalie Camelin, Yannick Estève. *Simulating ASR errors for training SLU systems.* 11th edition of the Language Resources and Evaluation Conference, LREC 2018. Miyazaki, Japon. 7-12 Mai 2018. Présentation poster.

**JEP 2018** Edwin Simonnet, Sahar Ghannay, Nathalie Camelin, Yannick Estève. *Simulation d'erreurs de reconnaissance automatique dans un cadre de compréhension de la parole.* 32ème Journées d'Études sur la Parole, JEP 2018. Aix-en-Provence, France. 4-8 Juin 2018. Présentation poster.

**SLT 2018** Sahar Ghannay, Antoine Caubrière, Yannick Estève, Nathalie Camelin, Edwin Simonnet, Antoine Laurent, Emmanuel Morin. *End-to-end named entity and semantic concept extraction from speech.* IEEE SLT 2018. Athènes, Grèce. 18-21 Décembre 2018.

# BIBLIOGRAPHIE


[Abdel-Hamid *et al.*, 2012] Abdel-Hamid, O., r. Mohamed, A., Jiang, H. et Penn, G. (2012). Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. *In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4277–4280.

[Allen *et al.*, 2007] Allen, J., Dzikovska, M., Manshadi, M. et Swift, M. (2007). Deep linguistic processing for spoken dialogue systems. *Proc. Workshop on Deep Linguistic Processing, ACL2007, Prague, 49-56.*

[Bahdanau *et al.*, 2014] Bahdanau, D., Cho, K. et Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv :1409.0473.*

[Baker *et al.*, 1998] Baker, C., Fillmore, C., et Lowe, J. (1998). The berkeley framenet project. *Proc. COLING-ACL-1998, pp. 86–99.*

[Bansal *et al.*, 2014] Bansal, M., Gimpel, K. et Livescu, K. (2014). Tailoring continuous word representations for dependency parsing. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 809–815.

[Baum, 1972] Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities 3(1), 1-8.*

[Ben Jannet, 2015] Ben Jannet, M. A. (2015). Évaluation adaptative des systèmes de transcription en contexte applicatif. *PhD.* Thèse de doctorat dirigée par Rosset, Sophie Informatique Paris Saclay 2015.

[Bengio et Heigold, 2014] Bengio, S. et Heigold, G. (2014). Word embeddings for speech recognition. *INTERSPEECH*, pages 1053–1057.

[Bengio, 2009] Bengio, Y. (2009). Learning deep architectures for ai. *Foundations*, 2:1–55.

[Bengio *et al.*, 2003] Bengio, Y., Ducharme, R., Vincent, P. et et Janvin, C. (2003). A neural probabilistic language model. *volume 3, pages 1137–1155 JMLR.org.*

[Bengio *et al.*, 2007] Bengio, Y., Lamblin, P., Popovici, D. et Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems 19*, pages 153–160.

[Bengio *et al.*, 1994] Bengio, Y., Simard, P. et Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks, 5(2) :157–166.*

[Bergstra et Bengio, 2012] Bergstra, J. et Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13(Feb) : 281–305.*

[Boite *et al.*, 1999] Boite, R., Bourlard, H., Dutoit, T., Hancq, J., et Leich, H. (1999). Traitement de la parole. *Presses polytechniques et universitaires romandes.*

[Bonneau-Maynard *et al.*, 2009] Bonneau-Maynard, H., Quignard, M. et Denis, A. (2009). Media : a semantically annotated corpus of task oriented dialogs in french. *Language Resources and Evaluation*, 43(4):329–354.

[Bonneau-Maynard *et al.*, 2005] Bonneau-Maynard, H., Rosset, S., Ayache, C., Kuhn, A. et Mostefa, D. (2005). Semantic annotation of the french media dialog corpus. *Ninth European Conference on Speech Communication and Technology.*

[Bottou, 1991] Bottou, L. (1991). Une approche théorique de l'apprentissage connexionniste : Applications à la reconnaissance de la parole. *PhD.*

[Bottou, 2010] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. *In Proceedings of COMPSTAT'2010, pages 177–186. Springer.*

[Bouchekif, 2017] Bouchekif, A. (2017). Structuration automatique de documents audio. *PhD.*

[Bougares, 2012] Bougares, F. (2012). Attelage de systèmes de transcription automatique de la parole. *PhD.*

[Brun *et al.*, 2005] Brun, A., Cerisara, C., Fohr, D., Illina, I., Langlois, D., Mella, O. et Smaïli, K. (2005). Ants : le système de transcription automatique du loria. *Workshop Évaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques (ESTER'05).*

[Béchet, 2007] Béchet, F. (2007). Modèles numériques pour la compréhension automatique de la parole. *HDR.*

[Béchet *et al.*, 2012] Béchet, F., Maza, B., Bigouroux, N., Bazillon, T., Elbèze, M., Mori, R. D. et Arbillot, E. (2012). Decoda : a call-center human-human spoken conversation corpus. *In International Conference on Language Resources and Evaluation (LREC.*

[Béchet et Raymond, 2018] Béchet, F. et Raymond, C. (2018). Is ATIS too shallow to go deeper for benchmarking Spoken Language Understanding models ? *InterSpeech 2018*, pages 1–5.

[Caglayan *et al.*, 2017] Caglayan, O., García-Martínez, M., Bardet, A., Aransa, W., Bougares, F. et Barrault, L. (2017). Nmtpy : A flexible toolkit for advanced neural machine translation systems. *arXiv preprint arXiv :1706.00457*.

[Camelin, 2007] Camelin, N. (2007). Stratégies robustes de compréhension de la parole basées sur des méthodes de classification automatique. *PhD*.

[Camelin *et al.*, 2010] Camelin, N., Béchet, F., Damnati, G. et De Mori, R. (2010). Detection and Interpretation of Opinion Expressions in Spoken Surveys. *IEEE Transactions on Audio, Speech and Language Processing*.

[Camelin *et al.*, 2011] Camelin, N., Detienne, B., Huet, S., Quadri, D. et Lefèvre, F. (2011). Unsupervised concept annotation using latent dirichlet allocation and segmental methods. *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 72–81.

[Chen et Goodman, 1999] Chen, S. F. et Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359 – 394.

[Chen *et al.*, 2016] Chen, Y.-N., Hakanni-Tür, D., Tur, G., Celikyilmaz, A., Guo, J. et Deng, L. (2016). Syntax or semantics ? knowledge-guided joint semantic frame parsing. *IEEE Workshop on Spoken Language Technology (SLT 2016)*.

[Cho *et al.*, 2014a] Cho, K., Van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H. et Bengio, Y. (2014a). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP*.

[Cho *et al.*, 2014b] Cho, K., Van Merriënboer, B., Bahdanau, D. et Bengio, Y. (2014b). On the properties of neural machine translation : Encoder-decoder approaches. *arXiv preprint arXiv :1409.1259*.

[Chomsky, 1957] Chomsky, N. (1957). Syntactic structures. *Mouton, The Hague*.

[Chomsky, 1965] Chomsky, N. (1965). Aspects of the theory of syntax. *MIT Press,Cambridge, MA*.

[Chorowski *et al.*, 2014] Chorowski, J., Bahdanau, D., Cho, K. et Bengio, Y. (2014). End-to-end continuous speech recognition using attention-based recurrent nn : First results. *arXiv preprint arXiv :1412.1602*.

[Chorowski *et al.*, 2015] Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K. et Bengio, Y. (2015). Attention-based models for speech recognition. *arXiv preprint arXiv :1506.07503*.

[Collobert et Weston, 2008] Collobert, R. et Weston, J. (2008). A unified architecture for natural language processing : Deep neural networks with multitask

learning. *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

[Collobert *et al.*, 2011] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. et Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398.

[De Mori, 1998] De Mori, R. (1998). Spoken dialogue with computers. *Academic Press, San Diego, USA*.

[De Mori, 2007] De Mori, R. (2007). Spoken language understanding : A survey. *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 365–376.

[De Mori *et al.*, 2008] De Mori, R., Béchet, F., Hakkani-Tür, D., McTear, M., Riccardi, G. et Tur, G. (2008). Spoken language understanding. *Signal Processing Magazine, IEEE*, 25(3):50–58.

[Dediu *et al.*, 2013] Dediu, A., Martín-Vide, C., Mitkov, R. et Truthe, B. (2013). *Statistical Language and Speech Processing : First International Conference, SLSP 2013, Tarragona, Spain, July 29-31, 2013, Proceedings*. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

[Devillers *et al.*, 2004] Devillers, L., Maynard, H., Rosset, S., Paroubek, P., McTait, K., Mostefa, D., Choukri, K., Charnay, L., Bousquet, C., Vigouroux, N., Béchet, F., Romary, L., Antoine, J., Villaneau, J., Vergnes, M. et Goulian, J. (2004). The french media/evalda project : the evaluation of the understanding capability of spoken language dialogue systems. *LREC*.

[Dinarelli et Tellier, 2016] Dinarelli, M. et Tellier, I. (2016). Improving recurrent neural networks for sequence labelling. *arXiv preprint arXiv :1606.02555*.

[Dinarelli *et al.*, 2017] Dinarelli, M., Vukotić, V. et Raymond, C. (2017). Label-dependency coding in Simple Recurrent Networks for Spoken Language Understanding. *Interspeech*.

[Dubuisson Duplessis *et al.*, 2015] Dubuisson Duplessis, G., Béchade, L., Sehili, M., Delaborde, A., Letard, V., Ligozat, A.-L., Deléglise, P., Estève, Y., Rosset, S. et Devillers, L. (2015). Nao is doing humour in the CHIST-ERA JOKER project. *16th Interspeech*, pages 1072–1073.

[Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive science, 14(2) :179–211*.

[Fillmore, 1976] Fillmore, C. (1976). Frame semantics and the nature of language. *Annals of the New York Academy of Sciences 280(1),20-32*.

[Fiscus, 1997a] Fiscus, J. G. (1997a). A post-processing system to yield reduced word error rates : Recognizer output voting error reduction (rover). *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347–354.

[Fiscus, 1997b] Fiscus, J. G. (1997b). A post-processing system to yield reduced word error rates : Recognizer output voting error reduction (rover). *Automatic*

*Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 347–354.

[Forney, 1973] Forney, G. D. J. (1973). The viterbi algorithm. *IEEE Transactions on Speech and Audio Processing 61(3), 268-278.*

[Fosler-Lussier *et al.*, 2002] Fosler-Lussier, E., Amdal, I. et Kuo, H.-K. J. (2002). On the road to improved lexical confusability metrics. *ISCA Tutorial and Research Workshop (ITRW) on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology.*

[Galliano *et al.*, 2006] Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J. f., Mostefa, D. et Choukri, K. (2006). Corpus description of the Ester evaluation campaign for the rich transcription of French broadcast news. *5th international Conference on Language Resources and Evaluation (LREC)*, pages 315–320.

[Galliano *et al.*, 2009] Galliano, S., Gravier, G. et Chaubard, L. (2009). The Ester 2 evaluation campaign for the rich transcription of french radio broadcasts. *Interspeech.*

[Gauvain *et al.*, 1994] Gauvain, J., Lamel, L., Adda, G. et Adda-Decker, M. (1994). The limsi continuous speech dictation system : evaluation on the arpa wall street journal task. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94), Volume 1, 557–560.*

[Ghannay, 2017] Ghannay, S. (2017). Etude sur les representations continues de mots appliquees à la detection automatique des erreurs de reconnaissance de la parole. *PhD.*

[Ghannay *et al.*, 2018] Ghannay, S., Caubrière, A., Estève, Y., Camelin, N., Simonnet, E., Laurent, A. et Morin., E. (2018). End-to-end named entity and semantic concept extraction from speech. *IEEE SLT 2018.*

[Ghannay *et al.*, 2015a] Ghannay, S., Esteve, Y. et Camelin, N. (2015a). Word embeddings combination and neural networks for robustness in asr error detection. *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 1671–1675.

[Ghannay *et al.*, 2016a] Ghannay, S., Esteve, Y., Camelin, N. *et al.* (2016a). Acoustic word embeddings for asr error detection. *Interspeech 2016*, pages 1330–1334.

[Ghannay *et al.*, 2015b] Ghannay, S., Estève, Y., Camelin, N., Dutrey, C., Santiago, F. et Adda-Decker, M. (2015b). Combining continuous word representation and prosodic features for asr error prediction. *International Conference on Statistical Language and Speech Processing*, pages 84–95.

[Ghannay *et al.*, 2016b] Ghannay, S., Favre, B., Esteve, Y. et Camelin, N. (2016b). Word embedding evaluation and combination. *of the Language Resources and Evaluation Conference (LREC 2016), Portoroz (Slovenia)*, pages 23–28.

[Gong, 1995] Gong, Y.-F. (1995). Speech recognition in noisy environments : A survey. *Speech Communication. 16, 261–291.*

[Gorin *et al.*, 1997] Gorin, A. L., Riccardi, G. et Wright, J. H. (1997). How may i help you ? *Speech Commun.*, 23(1-2):113–127.

[Graves, 2013] Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv :1308.0850.*

[Graves *et al.*, 2013] Graves, A., Mohamed, A.-r. et Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *In 2013 IEEE international conference on acoustics, speech and signal processing, pages 6645–6649. IEEE.*

[Gravier *et al.*, 2012] Gravier, G., Adda, G., Paulsson, N., Carré, M., Giraudel, A. et Galibert, O. (2012). The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. *Eighth International Conference on Language Resources and Evaluation (LREC)*, pages 114–118.

[Griol *et al.*, 2016] Griol, D., Iglesias, J. A., Ledezma, A. et Sanchis, A. (2016). A two-stage combining classifier model for the development of adaptive dialog systems. *International Journal of Neural Systems*, 26(01):1650002. PMID : 26678250.

[Grishman et Sundheim, 1996] Grishman, R. et Sundheim, B. (1996). Message understanding conference 6 : A brief history. *Proc COLING*, 96:466–471.

[Guinaudeau, 2011] Guinaudeau, C. (2011). Structuration automatique de flux télévisuels. *PhD.*

[Gupta *et al.*, 2006] Gupta, N., Tur, G., Hakkani-Tur, D., Bangalore, S., Riccardi, G. et Gilbert, M. (2006). The at t spoken language understanding system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):213–222.

[Hahn *et al.*, 2011] Hahn, S., Dinarelli, M., Raymond, C., Lefevre, F., Lehnen, P., De Mori, R., Moschitti, A., Ney, H. et Riccardi, G. (2011). Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1569–1583.

[Hahn *et al.*, 2008] Hahn, S., Lehnen, P., Raymond, C. et Ney, H. (2008). A comparison of various methods for concept tagging for spoken language. -.

[Hakkani-Tür *et al.*, 2016] Hakkani-Tür, D., Tur, G., Celikyilmaz, A., Chen, Y.-N., Gao, J., Deng, L. et Wang, Y.-Y. (2016). Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. *Proceedings of The 17th Annual Meeting of the International Speech Communication Association.*

[Hammersley et Clifford, 1971] Hammersley, J. et Clifford, P. (1971). Markov fields on finite graphs and lattices. -.

[He et Young, 2003] He, Y. et Young, S. (2003). A data-driven spoken language understanding system. *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721)*, pages 583–588.

[Hemphill *et al.*, 1990] Hemphill, C. T., Godfrey, J. J., Doddington, G. R. *et al.* (1990). The atis spoken language systems pilot corpus. *Proceedings of the DARPA speech and natural language workshop*, pages 96–101.

[Hinton *et al.*, 2012] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. et al. (2012). Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups. *Signal Processing Magazine, IEEE, 29(6) :82–97.*

[Hinton *et al.*, 2015] Hinton, G., Vinyals, O. et Dean, J. (2015). Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop.*

[Hinton *et al.*, 2006] Hinton, G. E., Osindero, S. et Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation, 18(7) :1527–1554.*

[Hochreiter et Schmidhuber, 1997] Hochreiter, S. et Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9(8) :1735–1780.*

[Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences, 79(8) :2554–2558.*

[Huang *et al.*, 1995] Huang, X., Acero, A., Alleva, F., Hwang, M., Jiang, L. et et M. Mahajan (1995). Microsoft windows highly intelligent speech recognizer : Whisper. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'95), Volume 1.*

[Iyer *et al.*, 1994] Iyer, R., Ostendorf, M. et Rohlicek, J. R. (1994). Language modeling with sentence-level mixtures. *Proceedings of the Workshop on Human Language Technology*, pages 82–87.

[Jabaian *et al.*, 2010] Jabaian, B., Besacier, L. et Lefèvre, F. (2010). Investigating multiple approaches for SLU portability to a new language. *Interspeech 2010*, pages x–x.

[Jaitly *et al.*, 2012] Jaitly, N., Nguyen, P., Senior, A. et Vanhoucke, V. (2012). Application of pretrained deep neural networks to large vocabulary speech recognition. *Proceedings of Interspeech 2012.*

[Jelinek, 1976] Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proceedings of the IEEE, 64(4) :532–556.*

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines : Learning with many relevant features. *Machine Learning : ECML-98*, pages 137–142.

[Jordan, 1997] Jordan, M. I. (1997). Serial order : A parallel distributed processing approach. *Advances in psychology, 121 :471–495.*

[Juang et Rabiner, 2005] Juang, B. et Rabiner, L. (2005). Automatic speech recognition - a brief history of the technology development. -.

[Jyothi et Fosler-Lussier, 2010] Jyothi, P. et Fosler-Lussier, E. (2010). Discriminative language modeling using simulated asr errors. *Eleventh Annual Conference of the International Speech Communication Association*.

[Kleinbauer *et al.*, 2007] Kleinbauer, T., Becker, S. et Becker, T. (2007). Combining multiple information layers for the automatic generation of indicative meeting abstracts. *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 151–154.

[Lafferty *et al.*, 2001] Lafferty, J., McCallum, A., Pereira, F. *et al.* (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning (ICML-2001)*, 1:282–289.

[Lailler *et al.*, 2015] Lailler, C., Estève, Y., De Mori, R., Bouallègue, M. et Morchid, M. (2015). Utilisation d'annotations sémantiques pour la validation automatique d'hypothèses dans des conversations téléphoniques. *TALN 2015*.

[Lavergne *et al.*, 2010] Lavergne, T., Cappé, O. et Yvon, F. (2010). Practical very large scale CRFs. *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513.

[LeCun *et al.*, 1990] LeCun, B. B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. et Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. *In Advances in neural information processing systems. Citeseer*.

[LeCun, 1985] LeCun, Y. (1985). Une procedure d'apprentissage pour reseau a seuil asymmetrique (a learning scheme for asymmetric threshold networks). -.

[Lee et Kawahar, 2009] Lee, A. et Kawahar, T. (2009). Recent development of open-source speech recognition engine julius. *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.

[Lee *et al.*, 1990] Lee, C., Giachin, E., Rabiner, L., Pieraccini, R. et Rosenberg, A. (1990). Improved acoustic modeling for continuous speech recognition. *Workshop on Speech and Natural Language, 319–326*.

[Lee et Narayanan, 2005] Lee, C. M. et Narayanan, S. S. (2005). Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 13(2):293–303.

[Lefèvre *et al.*, 2012] Lefèvre, F., Mostefa, D., Besacier, L., Estève, Y., Quignard, M., Camelin, N., Favre, B., Jabaian, B. et Rojas Barahona, L. M. (2012). Leveraging study of robustness and portability of spoken language understanding systems across languages and domains : the PORTMEDIA corpora. *The International Conference on Language Resources and Evaluation*.

[Lefèvre *et al.*, 2012] Lefèvre, F., Mostefa, D., Besacier, L., Estève, Y., Quignard, M., Camelin, N., Favre, B., Jabaian, B. et Rojas-Barahona, L. (2012). Robustesse et portabilités multilingue et multi-domaines des systèmes

de compréhension de la parole : les corpus du projet portmedia. *JEP-TALN-RECITAL 2012, volume 1 : JEP, pages 779–786, Grenoble, 4 au 8 juin 2012.*

[Levy et Goldberg, 2014] Levy, O. et Goldberg, Y. (2014). Dependency based word embeddings. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2:302–308.

[Liu et Lane, 2016] Liu, B. et Lane, I. (2016). Attention-based recurrent neural network models for joint intent detection and slot filling. *CoRR*, abs/1609.01454.

[Ma *et al.*, 2015] Ma, M., Huang, L., Xiang, B. et Zhou, B. (2015). Dependency-based convolutional neural networks for sentence embedding. *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.*

[Makhoul *et al.*, 1999] Makhoul, J., Kubala, F., Schwartz, R. et Weischedel, R. (1999). Performance measures for information extraction. *In Proceedings of DARPA Broadcast News Workshop*, pages 249–252.

[Malouf, 2002] Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. *International Conference On Computational Linguistics, 1–7. Association for Computational Linguistics Morristown, NJ, USA.*

[Mangu *et al.*, 2000] Mangu, L., Brill, E. et Stolcke, A. (2000). Finding consensus in speech recognition : word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400.

[Maskey, 2008] Maskey, S. R. (2008). Automatic broadcast news speech summarization. *PhD.* AAI3333404.

[Mdhaffar *et al.*, 2018] Mdhaffar, S., Laurent, A. et Estève, Y. (2018). Le corpus PASTEL pour le traitement automatique de cours magistraux. *25e conférence sur le Traitement Automatique des Langues Naturelles (TALN 2018).*

[Medsker et Jain, 1999] Medsker, L. et Jain, L. C. (1999). Recurrent neural networks : design and applications. *CRC press.*

[Memisevic, 2011] Memisevic, R. (2011). Gradient-based learning of higher-order image features. *2011 International Conference on Computer Vision*, pages 1591–1598.

[Mesnil *et al.*, 2015] Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D. *et al.* (2015). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(3):530–539.

[Mesnil *et al.*, 2013] Mesnil, G., He, X., Deng, L. et Bengio, Y. (2013). Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. *INTERSPEECH*, pages 3771–3775.

[Mikolov *et al.*, 2013a] Mikolov, T., Chen, K., Corrado, G. et Dean, J. (2013a). Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR.*

[Mikolov *et al.*, 2010] Mikolov, T., Karafiát, M., Burget, L., Cernocky, J. et Khudanpur, S. (2010). Recurrent neural network based language model. *Interspeech, volume 2, page 3.*

[Mikolov *et al.*, 2011] Mikolov, T., Kombrink, S., Burget, L., Černocky, J. et Khudanpur, S. (2011). Extensions of recurrent neural network language model. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5528–5531. IEEE.*

[Mikolov *et al.*, 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. et Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

[Minker et Bennacef, 2004] Minker, W. et Bennacef, S. (2004). Speech and human-machine dialog. *Springer, New York, USA.*

[Minsky et Papert, 1969] Minsky, M. et Papert, S. A. (1969). Perceptrons. *MIT press.*

[Mohri et Nederhof, 2001] Mohri, M. et Nederhof, M.-J. (2001). Regular approximation of context-free grammars through transformation. *In Jean-Claude JUNQUA et Gertjan NOORD , van, éditeurs : Robustness in Language and Speech Technology, pages 153–163. Kluwer Academic Publishers, Dordrecht.*

[Moreau *et al.*, 2000] Moreau, N., Charlet, D. et Jouvet, D. (2000). Confidence measure and incremental adaptation for the rejection of incorrect data. *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on, volume 3, pages 1807–1810. IEEE.*

[Murray *et al.*, 2010] Murray, G., Carenini, G. et Ng, R. (2010). Interpretation and transformation for abstracting conversations. *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 894–902.

[Nasr *et al.*, 2010] Nasr, A., Béchet, F. et Rey, J.-F. (2010). Macaon : Une chaîne linguistique pour le traitement de graphes de mots. *Traitement Automatique des Langues Naturelles - session de démonstrations.*

[Niekrasz et Moore, 2009] Niekrasz, J. et Moore, J. (2009). Participant subjectivity and involvement as a basis for discourse segmentation. *Proceedings of the SIGDIAL 2009 Conference : The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 54–61.

[Nocera *et al.*, 2002] Nocera, P., Linares, G. et Massonié, D. (2002). Principe et performances du décodeur parole continue speeral. *Journées d'Étude de la Parole (JEP'02).*

[Ogawa et Hori, 2015] Ogawa, A. et Hori, T. (2015). Asr error detection and recognition rate estimation using deep bidirectional recurrent neural networks. *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4370–4374.

[Paek et Horvitz, 2004] Paek, T. et Horvitz, E. (2004). Optimizing automated call routing by integrating spoken dialog models with queuing models. -.

[Passonneau et Litman, 1997] Passonneau, R. J. et Litman, D. J. (1997). Discourse segmentation by human and automated means. *Comput. Linguist.*, 23(1):103–139.

[Pennington *et al.*, 2014] Pennington, J., Socher, R. et Manning, C. D. (2014). Glove : Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

[Pieraccini *et al.*, 1992] Pieraccini, R., Tzoukermann, E., Z. Gorelov, J.-L. G., Levin, E., Lee, C.-H. et Wilpon, J. (1992). A speech understanding system based on statistical representation of semantics. *ICASSP*.

[Pietquin et Beaufort, 2005] Pietquin, O. et Beaufort, R. (2005). Comparing asr modeling methods for spoken dialogue simulation and optimal strategy learning. *Ninth European Conference on Speech Communication and Technology*.

[Povey *et al.*, 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P. *et al.* (2011). The kaldi speech recognition toolkit. *IEEE 2011 workshop on automatic speech recognition and understanding*. EPFL-CONF-192584.

[Powell, 1978] Powell, M. J. D. (1978). A fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis*, pages 144–157.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77(2) :257–286*.

[Ramshaw et Marcus, 1995] Ramshaw, L. A. et Marcus, M. P. (1995). Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040.

[Raymond, 2005] Raymond, C. (2005). Décodage conceptuel : co-articulation des processus de transcription et compréhension dans les systèmes de dialogue. *PhD*.

[Raymond et Riccardi, 2007] Raymond, C. et Riccardi, G. (2007). Generative and discriminative algorithms for spoken language understanding. *Interspeech 2007*.

[Reddy *et al.*, 2016] Reddy, S., Täckström, O., Collins, M., Kwiatkowski, T., Das, D., Steedman, M. et Lapata, M. (2016). Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

[Rosenblatt, 1957] Rosenblatt, F. (1957). The perceptron, a perceiving and recognizing automaton project para. *Cornell Aeronautical Laboratory*.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review, 65(6) :386*.

[Rousseau *et al.*, 2014] Rousseau, A., Boulianne, G., Deléglise, P., Estève, Y., Gupta, V. et Meignier, S. (2014). LIUM and CRIM ASR system combination for the REPERE evaluation campaign. *International Conference on Text, Speech, and Dialogue*, pages 441–448.

[Rubinstein et Hastie, 1997] Rubinstein, Y. D. et Hastie, T. (1997). Discriminative vs informative learning. *IN PROC. THIRD INT. CONF. ON KNOWLEDGE DISCOVERY AND DATA MINING*, pages 49–53.

[Rudy et Taylor, 2014] Rudy, J. et Taylor, G. W. (2014). Generative class-conditional autoencoders. *CoRR*, abs/1412.7009.

[Rumelhart *et al.*, 1985] Rumelhart, D. E., Hinton, G. E. et Williams, R. J. (1985). Learning internal representations by error propagation. *Rapport technique, DTIC Document*.

[Rumelhart *et al.*, 1986] Rumelhart, D. E., Hinton, G. E. et Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323 :533–536*.

[Rumelhart *et al.*, 1988] Rumelhart, D. E., Hinton, G. E. et Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling, 5(3) :1*.

[Salakhutdinov et Hinton, 2009] Salakhutdinov, R. et Hinton, G. E. (2009). Deep boltzmann machines. *In AISTATS, volume 1, page 3*.

[Samson Juan, 2015] Samson Juan, S. F. (2015). Exploiting resources from closely-related languages for automatic speech recognition in low-resource languages from Malaysia. *PhD*, -(2015GREAM061).

[Sarikaya *et al.*, 2014] Sarikaya, R., Hinton, G. E. et Deoras, A. (2014). Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(4):778–784.

[Schatzmann *et al.*, 2007] Schatzmann, J., Thomson, B. et Young, S. (2007). Error simulation for training statistical dialogue systems. *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 526–531.

[Schaul *et al.*, 2013] Schaul, T., Zhang, S. et LeCun, Y. (2013). No more pesky learning rate. *ICML (3), 28 :343–351*.

[Schuster et Paliwal, 1997] Schuster, M. et Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 45(11) :2673–2681*.

[Schwenk, 2007] Schwenk, H. (2007). Continuous space language models. *Computer Speech & Language, 21(3) :492–518*.

[Schwenk, 2013] Schwenk, H. (2013). Cslm - a modular open-source continuous space language modeling toolkit. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1198–1202.

[Schwenk *et al.*, 2006] Schwenk, H., Dchelotte, D. et Gauvain, J.-L. (2006). Continuous space language models for statistical machine translation. *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, pages 723–730.

[Seneff, 1989] Seneff, S. (1989). Tina : A probabilistic syntactic parser for speech understanding systems. *Massachusetts Institute of Technology*.

[Serdyuk *et al.*, 2018] Serdyuk, D., Wang, Y., Fuegen, C., Kumar, A., Liu, B. et Bengio, Y. (2018). Towards end-to-end spoken language understanding. *arXiv preprint arXiv :1802.08395*.

[Seymore et Rosenfeld, 1997] Seymore, K. et Rosenfeld, R. (1997). Using story topics for language model adaptation. -.

[Simonnet *et al.*, 2015] Simonnet, E., Deléglise, P., Camelin, N. et Estève, Y. (2015). Exploring the use of attention-based recurrent neural networks for spoken language understanding. *29th Conference on Neural Information Processing Systems, NIPS 2015, Machine Learning for Spoken Language Understanding and Interaction workshop (SLUNIPS)*.

[Stemmer *et al.*, 2002] Stemmer, G., Steidl, S., Nöth, E., Niemann, H. et Batliner, A. (2002). Comparison and combination of confidence measures. *Text Speech and Dialogue, pages 181–188. Springer*.

[Stolcke *et al.*, 2000] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R. A., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C. V. et Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *CoRR*, cs.CL/0006023.

[Stuttle *et al.*, 2004] Stuttle, M., Williams, J. et Young, S. (2004). A framework for dialog systems data collection using a simulated asr channel. *ICSLP 2004*.

[Sutskever *et al.*, 2014] Sutskever, I., Vinyals, O. et Le, Q. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS 2014)*.

[Tompson *et al.*, 2014] Tompson, J. J., Jain, A., LeCun, Y. et Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in Neural Information Processing Systems 27*, pages 1799–1807.

[Tur et De Mori, 2011] Tur, G. et De Mori, R. (2011). *Spoken language understanding : Systems for extracting semantic information from speech*. John Wiley & Sons.

[Tur *et al.*, 2006] Tur, G., Guz, U. et Hakkani-Tur, D. (2006). Model adaptation for dialog act tagging. *2006 IEEE Spoken Language Technology Workshop*, pages 94–97.

[Tur *et al.*, 2010] Tur, G., Hakkani-Tür, D. et Heck, L. (2010). What is left to be understood in atis ? *2010 IEEE Spoken Language Technology Workshop*, pages 19–24.

[Turian *et al.*, 2010] Turian, J., Ratinov, L. et Bengio, Y. (2010). Word representations : A simple and general method for semi-supervised learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

[Vapnik, 1982] Vapnik, V. (1982). Estimation of dependences based on empirical data. *Springer-Verlag*.

[Vapnik, 1995] Vapnik, V. (1995). The nature of statistical learning theory. *Springer-Verlag New York, Inc.*

[Vincent *et al.*, 2008] Vincent, P., Larochelle, H., Bengio, Y. et Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103.

[Vincent *et al.*, 2010] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. et Manzagol, P.-A. (2010). Stacked denoising autoencoders : Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.

[Vukotic *et al.*, 2015] Vukotic, V., Raymond, C. et Gravier, G. (2015). Is it time to switch to word embedding and recurrent neural networks for spoken language understanding ? *InterSpeech*.

[Wang *et al.*, 2003] Wang, Y.-Y., Acero, A. et Chelba, C. (2003). Is word error rate a good indicator for spoken language understanding accuracy. *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721)*, pages 577–582.

[Wang *et al.*, 2006] Wang, Y.-Y., Acero, A., Mahajan, M. et Lee, J. (2006). Combining statistical and knowledge-based spoken language understanding in conditional models. *COLING/ACL06*, page 882–889.

[Wiesler *et al.*, 2014] Wiesler, S., Richard, A., Golik, P., Schlüter, R. et Ney, H. (2014). Rasr/nn : The rwth neural network toolkit for speech recognition. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3281–3285.

[Woodland *et al.*, 1998] Woodland, P., Hain, T., Johnson, S., Niesler, T., Tuerk, A., Whittaker, E. et Young, S. (1998). The 1997 htk broadcast news transcription system. *Workshop DARPA Broadcast News Transcription and Understanding, 41–48*.

[Woods, 1975] Woods, W. (1975). What's in a link ? *Representation and Understanding, D.G. Bobrow and A. Collins, Eds. New York : Academic*.

[Xu et Sarikaya, 2013] Xu, P. et Sarikaya, R. (2013). Convolutional neural network based triangular crf for joint intent detection and slot filling. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 78–83.

[Yao *et al.*, 2014] Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G. et Shi, Y. (2014). Spoken language understanding using long short-term memory neural networks. *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 189–194.

[Yu *et al.*, 2011] Yu, D., Li, J. et Deng, L. (2011). Calibration of confidence measures in speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2461–2473.

[Zbontar et LeCun, 2015] Zbontar, J. et LeCun, Y. (2015). Stereo matching by training a convolutional neural network to compare image patches. *CoRR*, abs/1510.05970.

[Zeiler, 2012] Zeiler, M. D. (2012). Adadelta : an adaptive learning rate method. *arXiv preprint arXiv :1212.5701*.

[Zhang *et al.*, 2005] Zhang, R., Al Bawad, Z., Chan, A., Chotimongkol, A., Huggins-Daines, D. et I.Rudnicky, A. (2005). Investigations on ensemble based semi-supervised acoustic model training. *Proc. of Eurospeech, Lisbon Portugal*.

[Zhao *et al.*, 2016] Zhao, T., Lee, K. et Eskénazi, M. (2016). Dialport : Connecting the spoken dialog research community to real user data. *CoRR*, abs/1606.02562.

# ANNEXE

## Acronymes

**ASR** : Reconnaissance de la parole (*Automatic Speech Recognition*)
**biRNN** : Réseaux de neurones récurrents bidirectionnel (*Bidirectional Recurrent Neural Network*)
**biRNN-EDA** : Structure biRNN Encodeur-Décodeur avec mécanisme d'Attention.
**CER** : Taux d'erreur sur les concepts (*Concept Error Rate*)
***cm*** : Mesure de confiance MS-MLP
**CNN** : Réseau de neurones convolutif (*Convolutional Neural Network*)
**CRF** : Champs aléatoires conditionnels (*Conditional Random Fields*)
**CVER** : Taux d'erreur sur les couples concepts/valeurs (*Concept-Value Error Rate*)
**DNN** : Réseaux de neurones profonds (*Deep Neural Network*)
**FSM** : Automates à états finis (*Finite State Machine*)
**GRU** : Unité récurrente à portes (*Gated Recurrent Units*)
**HMM** : Modèles de Markov cachés (*Hidden Markov Model*)
**LER** : Taux d'erreur sur les étiquettes (*Label Error Rate*)
**LSTM** : Mémoire à court et long termes (*Long Short Term Memory*)
**MS-MLP** : Perceptron multi-couches multi-flux (*Multi-Stream Multi-Layer Perceptron*)
**NCE** : Entropie-croisée normalisée (*Normalized Cross Entropy*)
**NN** : Réseaux de neurones (*Neural Networks*)
***pap*** : Probabilité de mot a posteriori
**POS** : Étiquettes morphosyntaxiques (*Part Of Speech*)
**RNN** : Réseaux de neurones récurrents (*Recurrent Neural Network*)
**SER** : Taux d'erreur sur les champs (*Slot Error Rate*)
***sim*** : Mesure de similarité
**SLU** : Compréhension de la parole (*Spoken Language Understanding*)
**SVM** : Machines à vecteur de support (*Support Vector Machines*)

**TDE** : Taux d'erreur de détection des erreurs de reconnaissance
**WER** : Taux d'erreur sur les mots (*Word Error Rate*)

## Intervalle de confiance

Nous considérons pour nos expériences sur le corpus MEDIA un intervalle de confiance à 95% selon la loi de *Student* pour évaluer la significativité statistique des résultats.

— En CER sur le corpus manuel : nous estimons un intervalle de confiance de 1,1 sur le DEV et de 0,6 sur le TEST.
— En CER sur le corpus automatique : nous estimons un intervalle de confiance de 1,3 sur le DEV et de 0,8 sur le TEST.
— En CVER sur le corpus automatique : nous estimons un intervalle de confiance de 1,4 sur le DEV et de 0,9 sur le TEST.

# Titre : Réseaux de neurones profonds appliqués à la compréhension de la parole

**Mot clés :** Compréhension de la parole, Corpus MEDIA, Étiquetage en concept sémantiques, Réseaux de neurones profonds, Mécanisme d'attention, Erreurs de reconnaissance automatique, Simulation d'erreurs de reconnaissance, Désambiguïsation de la compréhension

**Resumé :** Cette thèse s'inscrit dans le cadre de l'émergence de l'apprentissage profond et aborde la compréhension de la parole assimilée à l'extraction et à la représentation automatique du sens contenu dans les mots d'une phrase parlée. Nous étudions une tâche d'étiquetage en concepts sémantiques dans un contexte de dialogue oral évaluée sur le corpus français MEDIA. Depuis une dizaine d'années, les modèles neuronaux prennent l'ascendant dans de nombreuses tâches de traitement du langage naturel grâce à des avancées algorithmiques ou à la mise à disposition d'outils de calcul puissants comme les processeurs graphiques. De nombreux obstacles rendent la compréhension complexe, comme l'interprétation difficile des transcriptions automatiques de la parole étant donné que de nombreuses erreurs sont introduites par le processus de reconnaissance automatique en amont du module de compréhension. Nous présentons un état de l'art décrivant la compréhension de la parole puis les méthodes d'apprentissage automatique supervisé pour la résoudre en commençant par des systèmes classiques pour finir avec des techniques d'apprentissage profond. Les contributions sont ensuite exposées suivant trois axes. Premièrement, nous développons une architecture neuronale efficace consistant en un réseau récurent bidirectionnel encodeur-décodeur avec mécanisme d'attention. Puis nous abordons la gestion des erreurs de reconnaissance automatique et des solutions pour limiter leur impact sur nos performances. Enfin, nous envisageons une désambiguïsation de la tâche de compréhension permettant de rendre notre système plus performant.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Le Mans Université

# Title : Deep learning applied to spoken langage understanding

**Keywords :** Spoken language understanding, MEDIA corpus, Semantic concept tagging, Deep learning, Attention mechanism, Automatic speech recognition errors, Simulation of recognition errors, Disambiguation of understanding

**Abstract :** This thesis is a part of the emergence of deep learning and focuses on spoken language understanding assimilated to the automatic extraction and representation of the meaning supported by the words in a spoken utterance. We study a semantic concept tagging task used in a spoken dialogue system and evaluated with the French corpus MEDIA. For the past decade, neural models have emerged in many natural language processing tasks through algorithmic advances or powerful computing tools such as graphics processors. Many obstacles make the understanding task complex, such as the difficult interpretation of automatic speech transcriptions, as many errors are introduced by the automatic recognition process upstream of the comprehension module. We present a state of the art describing spoken language understanding and then supervised automatic learning methods to solve it, starting with classical systems and finishing with deep learning techniques. The contributions are then presented along three axes. First, we develop an efficient neural architecture consisting of a bidirectional recurrent network encoder-decoder with attention mechanism. Then we study the management of automatic recognition errors and solutions to limit their impact on our performances. Finally, we envisage a disambiguation of the comprehension task making the systems more efficient.