Du signal au concept : réseaux de neurones profonds appliqués à la compréhension de la parole

Antoine Caubriere

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Le Mans Université

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DU MANS

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Informatique

Par

Antoine CAUBRIÈRE

Du signal au concept : Réseaux de neurones profonds appliqués à la compréhension de la parole

Thèse présentée et soutenue à Le Mans Université, LIUM, le 29 Janvier 2021
Unité de recherche : Laboratoire d'Informatique de l'Université du Mans (LIUM) - EA 4023
Thèse N° : 2021LEMA1001

Rapporteurs avant soutenance :

Mme. Irina ILLINA	Maître de conférence - HDR, Université de Lorraine - LORIA / INRIA
M. Benoit FAVRE	Maître de conférence - HDR, Aix-Marseille Université - LIS

Composition du Jury :

Examinateur :	**M. François PORTET**	Professeur, Université Grenoble Alpes - LIG
Dir. de thèse :	**M. Yannick ESTÈVE**	Professeur, Avignon Université - LIA
Co-dir. de thèse :	**M. Emmanuel MORIN**	Professeur, Université de Nantes - LS2N
Co-enc. de thèse :	**M. Antoine LAURENT**	Maître de conférence, Le Mans Université - LIUM

Invitée :

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

Remerciements

Je tiens tout d'abord à remercier l'ensemble des membres de mon jury de thèse pour avoir accepté de rapporter et d'examiner les travaux présentés dans ce manuscrit. Merci à tous pour l'intérêt que vous portez à mes travaux et pour votre temps.

Je remercie très chaleureusement mon directeur de thèse, Yannick Estève, pour sa grande disponibilité, son soutien et ses nombreux conseils avisés. L'ensemble de nos discussions m'ont énormément apportées, tant sur le plan scientifique que personnel. Elles ont constitué un élément primordial à l'accomplissement de cette thèse. Mes remerciements les plus sincères vont également à mon co-directeur de thèse, Emmanuel Morin, pour sa clairvoyance, ses encouragements, ainsi que sa patience. Ses propositions ont toujours été intéressantes et m'ont guidé vers des directions prometteuses. Je remercie aussi mon co-encadrant, Antoine Laurent, pour son expérience qu'il m'a apportée et qui a été nécessaire à la réussite de ce travail de thèse.

Je souhaite exprimer ma gratitude à l'ensemble des personnes avec qui j'ai eu la chance de collaborer. Je pense particulièrement à Natalia Tomashenko, Natalie Camelin, Sahar Ghannay, Sophie Rosset, Edwin Simonnet et Renato De Mori. Travailler avec vous m'a beaucoup appris.

J'adresse mes plus vifs remerciements à toutes les personnes que j'ai rencontrées au LIUM, pour votre aide précieuse et tous les bons moments partagés au cours de cette thèse. Je pense notamment à, Rajoua Anane, Adrien Bardet, Amira Barhoumi, Emmanuelle Billard, Fethi Bougares, Rémi Bouvet, Pierre-Alexandre Broux, Ozan Caglayan, Gaëtan Caillaut, Pierre Champion, Nicolas Dugué, Grégor Dupuy, Anne-Cécile Erreau, Bruno Jacob, Malik Koné, Anthony Larcher, Martin Lebourdais, Daniel Luzzati, Salima Mdhaffar, Sylvain Meignier, Étienne Micoulaut, Valentin Pelloin, Simon Petitrenaud, Thibault Prouteau, Dominique Py, Marie Tahon, Thomas Thebaud, Kévin Vythelingum, Jane Wottawa. C'est une réelle chance d'avoir pu intégrer un laboratoire offrant un environnement de cette qualité, propice au développement personnel et professionnel.

Je n'oublie pas l'ensemble des personnes que j'ai rencontrées au LIA, le Laboratoire d'Informatique de l'université d'Avignon. Je tiens à vous remercier sincèrement pour votre accueil chaleureux, votre formidable énergie positive et tous les bons moments partagés. Je pense particulièrement à Sondes Abderrazek, Carlos González, Adrien Gresse, Thibault Grousset, Mayeul Mathias, Teva Merlin, Luis Moreno Jimenez, Tesnim Naceur, Paul-Gauthier Noe, Titouan Parcollet, Céline Portalier, Matthieu Riou, Cyril Sahuc, Thierry Vallet. C'est une grande chance d'avoir pu passer ces mois parmi vous.

Je remercie sincèrement ma famille et mes amis, qui ont su être présents, m'encourager et me soutenir moralement sans faille tout au long de cette thèse. Je pense spécialement à mes parents, mais aussi à Anthony, Bill, Francisque, Gaëlle, Gwladys, Ha, Halyna, Laura, Manon, Marie, Marine, Mathias, Thomas.

Enfin, je tiens à adresser mes remerciements les plus sincères envers toutes les personnes que je n'ai pu citer et qui ont fait partie de cette aventure. Je mesure la chance d'avoir pu réaliser une thèse dans ces conditions et aussi bien entouré.

À vous tous, merci !

Nous remercions le programme RFI Atlanstic 2020 pour son financement.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

ACRONYMES

AdaGrad	*Adaptative Gradient*
Adam	*Adaptative Moment estimation*
CER	Taux d'erreur sur les concepts, (*Concept Error Rate*)
CNN	Réseau de neurone convolutionel, (*Convolutional Neural Network*)
CRF	Champs aléatoire conditionnel, (*Conditionnal Random Fields*)
CTC	Classification temporelle connexionniste, (*Connectionist Temporal Classification*)
CVER	Taux d'erreur sur les concepts et leurs valeurs, (*Concept Value Error Rate*)
DNN	Réseau de neurone profond, (*Deep Neural Network*)
EN	Entités Nommées
EM	Expectation Maximization
ETER	*Entity Tree Error Rate*
FSM	Automates à états finis, (*Finite State Machine*)
GMM	Modèle à mélange de gaussiennes, (*Gaussian Mixture Model*)
GRU	*Gated Recurrent Unit*
HMM	Modèle de Markov caché, (*Hidden Markov Model*)
IA	Intelligence Artificielle
LSTM	Réseau récurrent à mémoire court et long terme, (*long-short term memory*)
MFCC	*Mel-Frequency Cepstral Coefficient*
MLP	Perceptron multicouches, (*MultiLayer Perceptron*)
MUC	*Message Understanding Conference*
NCE	Cross-Entropie Normalisée, (*Normalized Cross-Entropy*)
OOV	*Out Of Vocabulary*
RAP	Reconnaissance Automatique de la Parole
ReLU	*Rectified Linear Unit*
REN	Reconnaissance des Entités Nommées
RNN	Réseau de neurone récurrent, (*Recurent Neural Network*)
RNN-LM	*Recurrent Neural Network Language Model*
SVM	Machine à vecteur de support (*Support Vector Machine*)
t-SNE	*t-Distributed Stochastic Neighbor Embedding*
TDNN	Réseau de neurone à retardement, (*Time-Delay Neural Network*)
UCV	*Unseen Concept-Value pairs*
WER	Taux d'erreur sur les mots, (*Word Error Rate*)

INTRODUCTION

Cette thèse s'inscrit dans le contexte de la compréhension automatique de la parole. Ces dernières années, la compréhension de la parole a suscité et continue de susciter un vif intérêt, tant dans un cadre de recherche que d'applications industrielles. En effet, la recherche dans ce domaine rend possible l'automatisation de tâches jusque là réservées à l'humain, comme la segmentation et l'identification de thème de documents audios, ainsi que le résumé automatique de l'information essentielle. L'intérêt pour l'automatisation de ces tâches provient du flux toujours grandissant de documents et d'informations générés et partagés de nos jours. Les travaux de recherche en compréhension de la parole facilite également les interactions humain-machine. Nous pensons notamment aux objets connectés, tels que les assistants personnels, ainsi que des services téléphoniques automatisés, comme la réservation d'hôtel ou de place de théâtre.

Dans un cadre général, la compréhension de la parole peut être définie comme la tâche d'interprétation des signes véhiculés par un signal de parole [De Mori 2007]. Elle consiste en la projection d'informations de la dimension acoustique vers une représentation sémantique, ce qui correspond à l'extraction du sens d'un discours. En informatique, il est nécessaire de projeter le sens d'un message dans une représentation plus ou moins structurée, qui sera extraite à partir des signes disponibles dans la parole et de leurs caractéristiques [De Mori et al. 2008].

Il est compliqué de mettre en place une représentation structurée de la sémantique suffisamment générique pour réaliser la tâche de compréhension du langage ouvert. Ainsi, la représentation sémantique exploitée pour effectuer la compréhension de la parole est très souvent définie de manière ad hoc. C'est-à-dire en fonction d'un contexte applicatif précis. Cette représentation spécifique rend possible la prise en charge de certaines tâches de compréhension de la parole sous la forme de tâches de segmentation en mots support de concepts et d'étiquetage sémantique. Elles peuvent ainsi être traitées par l'intermédiaire de méthodes d'apprentissage automatique supervisé.

Effectuer une projection directe de la dimension acoustique vers une représentation sémantique spécifique a été jusqu'alors une tâche trop complexe pour être envisagée directement par un unique système. Il est ainsi question d'effectuer la mise en place d'une chaîne de traitements avec tout d'abord un composant de reconnaissance de la parole, puis à minima un composant de compréhension du langage appliqué sur les sorties du premier composant.

Au commencement de cette thèse en 2017, les technologies neuronales d'apprentissage pro-

fond sont les briques de base d'une chaîne de traitements à l'état de l'art, avec des résultats similaires à ceux obtenus avec des CRF [Simonnet, Ghannay, Camelin et Estève 2018]. Malgré l'apport de ces technologies pour la tâche finale de compréhension, un obstacle subsiste au travers du principe même de la chaîne de traitements. En effet, le composant de reconnaissance de la parole produit une représentation textuelle bruitées par des erreurs. Puis, par l'exploitation de cette représentation comme élément d'entrée du composant de compréhension de langue, ce bruit va inéluctablement impacter les performances du second composant. De plus, la représentation textuelle agit comme un entonnoir et ne représente que le discours prononcé, alors que la parole contient des informations allant nécessairement au-delà du discours. C'est-à-dire l'ensemble des informations présentes dans la parole, faisant partie du domaine de la paralinguistique, par exemple, la prosodie ou encore les disfluences.

Développer une approche permettant de surmonter cet obstacle correspond à l'objectif premier des travaux de cette thèse. Cela se traduit par la mise en œuvre d'un unique système neuronale, entièrement optimisé pour la tâche finale de compréhension de la parole. Ainsi, ce système doit être responsable de la projection directe d'informations de la dimension acoustique vers une représentation sémantique structurée.

Avant cette thèse, de premiers travaux dans le domaine de la reconnaissance de la parole ont permis la mise en œuvre de systèmes permettant la projection directe d'informations de la dimension acoustiques vers une représentation textuelle [Hannun et al. 2014 ; Amodei et al. 2016]. Ces travaux ont exploité un unique modèle neuronal, alors que cette tâche était au préalable réalisée par la combinaison de modèles acoustiques et de langues.

Dans le cadre de nos travaux, il est question d'étendre ces avancées à la compréhension de la parole. Pour réaliser cette thèse, nous nous inscrivons dans un cadre applicatif que nous trouvons dans les tâches de reconnaissance des entités nommées, ainsi que d'extraction des concepts sémantiques dans la parole. Il s'agit de deux tâches de compréhension faisant l'objet de travaux réguliers [Sundheim 1995 ; Galliano, Geoffrois, Mostefa et al. 2005 ; Hatmi 2014 ; Simonnet 2019], pour lesquelles une quantité suffisante de données est disponible pour évaluer nos contributions.

Cependant, les données manuellement annotées pour des tâches aussi complexes que la compréhension de la parole ont un coût important, ce qui induit nécessairement leur rareté. En effet, au-delà de nécessiter une transcription manuelle de la parole, un deuxième niveau d'annotation est nécessaire pour mettre en avant la sémantique selon une structure définie en fonction de la tâche. Ainsi, pour la réalisation de nos travaux, une difficulté supplémentaire réside dans notre besoin de compenser le manque de données annotées pour la tâche finale.

Afin de contourner cette difficulté, nous nous intéressons aux méthodes d'augmentation automatique de données, mais aussi au transfert d'apprentissage [Pan et Yang 2009]. Cette méthode consiste en l'exploitation d'un premier ensemble de données pour extraire des connais-

sances améliorant l'apprentissage sur un second ensemble de données.

Nous envisageons de tirer parti d'ensembles de données dédiés à la reconnaissance automatique de la parole, en raison de leur accessibilité en plus grande quantité. En exploitant notre cadre applicatif, nous envisageons aussi de bénéficier de connaissances apportées par les entités nommées pour traiter les concepts sémantiques.

Organisation du document

Nous structurons ce document en deux parties de quatre chapitres chacune. La première partie est dédiée à la présentation du contexte à travers un état de l'art des travaux connexes à cette thèse, tandis que la seconde concerne la description de nos contributions.

Au sein du premier chapitre, nous abordons les notions fondamentales de l'apprentissage neuronal profond. Il s'agit des technologies sur lesquelles reposent une grande majorité des travaux de cette thèse. Nous évoquons l'émergence de ces technologies à travers le perceptron, le premier neurone formel, et des perceptrons multicouches. Par la suite, nous décrivons les principaux algorithmes nécessaires à l'apprentissage profond, ainsi que les problématiques majeures de ce domaine. Nous abordons également les techniques développées pour répondre à ces problématiques. Dans ce premier chapitre, nous détaillons aussi les variantes les plus récentes, comme les approches neuronales récurrentes, les architectures de type encodeur-décodeur ou encore les mécanismes d'attention.

Dans le second chapitre, nous nous concentrons sur une description de la tâche de reconnaissance automatique de la parole. Elle est importante dans la mesure où elle constitue l'entrée du composant de compréhension dans le cadre d'une approche par chaîne de composants. En outre, les premiers systèmes de reconnaissance de la parole exploitant directement des observations sur le signal acoustique constituent un point d'appui essentiel aux travaux de cette thèse. Dans ce chapitre, nous décrivons les approches traditionnellement utilisées dans ce domaine, ainsi que les approches plus récentes s'appuyant sur des réseaux de neurones. Enfin, nous effectuons une description des mesures d'évaluation de la qualité des transcriptions produites par ce type de système.

Le troisième chapitre de ce manuscrit est dédié à la tâche de compréhension de la parole. Nous définissons tout d'abord cette tâche qui est au cœur de nos travaux. C'est en soi une tâche complexe qui dépend nécessairement de la tâche visée par un système et se définit donc selon plusieurs déclinaisons. Nous détaillons les approches traditionnellement utilisées correspondant à l'apprentissage machine, mais aussi les approches plus récentes s'appuyant sur les réseaux de neurones. Enfin, nous proposons une description des principales mesures d'évaluation des tâches de compréhension de la parole que nous avons exploitées pour nos contributions.

Dans le dernier chapitre de la première partie, nous décrivons les ensembles de données

à notre disposition selon notre cadre applicatif. Nous précisons les origines de ces ensembles, ainsi que leur composition et leur répartition. Lorsque ces données permettent une tâche de compréhension, nous détaillons l'annotation sémantique définissant la tâche associée. Nous fournissons également les références de l'ensemble de ces corpus, facilitant leur récupération et donc la reproductibilité de l'ensemble de nos travaux. Enfin, nous détaillons l'exploitation que nous effectuons de ces données dans le cadre de nos travaux.

Le cinquième chapitre de ce manuscrit correspond au premier chapitre de présentation de nos contributions. Nous abordons nos travaux centrés autour de la tâche de reconnaissance des entités nommées dans la parole. Dans ce chapitre, nous présentons un premier système permettant la projection directe d'informations d'une dimension acoustique vers une représentation sémantique structurée. Ce système est inspiré de ceux existants dans le domaine de la reconnaissance de la parole. Nous l'évaluons sur les données d'une campagne d'évaluation française. En complément, nous effectuons sa comparaison à une approche traditionnelle par chaîne de traitements successifs que nous avons mise à jour.

Au sein du sixième chapitre, nous proposons d'étendre notre approche sur une tâche d'extraction des concepts sémantiques. Nous souhaitons vérifier la viabilité de notre approche dans le cadre d'une représentation sémantique structurée plus précise. Nous envisageons également de tirer bénéfice de connaissances acquises avec les entités nommées, pour l'extraction des concepts sémantiques, par transfert d'apprentissage. Aussi, nous effectuons des travaux d'optimisation de notre architecture neuronale. Enfin, nous comparons à nouveau notre approche avec une chaîne de traitements successifs traditionnelle.

Dans le cadre du septième chapitre, nous réalisons une analyse des erreurs produites par nos systèmes dans le cadre de l'extraction des concepts sémantiques. Ces analyses nous apportent des éléments d'amélioration de l'approche que nous avons mis en œuvre. Nous proposons aussi une méthode d'extraction des représentations internes de la sémantique qui nous permet tout d'abord une analyse visuelle. Puis, nous étendons l'utilisation de ces représentations internes pour établir une mesure de confiance pertinente, concernant l'émission de la sémantique par notre approche.

Enfin, dans un dernier chapitre, nous concluons sur les travaux présentés dans ce manuscrit. Nous abordons à nouveau les points clefs de cette thèse, ainsi que les perspectives que nous pouvons dégager pour de futurs travaux. Nous proposons également un aparté concernant une estimation du coût environnemental de nos travaux.

Première partie

Contexte et État de l'art

Chapitre 1

Réseau neuronal profond

Sommaire

En informatique, le domaine de l'intelligence artificielle (IA) est vaste. Il concerne l'ensemble des algorithmes et méthodes visant la mise en place d'un système capable de simuler l'intelligence.

Parmi les méthodes employées, nous pouvons citer l'apprentissage machine. Cette méthode permet à un ordinateur d'apprendre à réaliser une tâche pour laquelle il n'est pas directement programmé. Elle est basée sur des approches mathématiques et l'exploitation de données.

Cette exploitation peut être effectuée différemment en fonction des informations disponibles. Lorsque pour une tâche donnée, les sorties attendues du système sont connues à l'avance, on parle d'apprentissages supervisés. Il s'agit d'une tâche de régression si les prédictions sont continues et d'une tâche de classification si les prédictions sont discrétisées. Également, Lorsque les sorties attendues ne sont pas connues, on parle d'apprentissage non-supervisé.

Il existe plusieurs méthodes d'apprentissage machine. Parmi lesquelles on retrouve notamment les réseaux de neurones artificiels. Il s'agit d'algorithmes sur lesquels repose l'apprentissage profond.

Nous représentons l'emplacement de l'apprentissage profond au sein de l'IA dans la figure suivante.

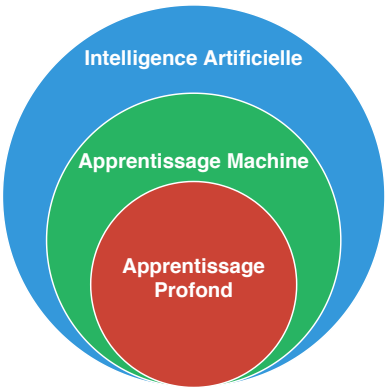

Figure 1.1 – L'apprentissage profond au sein de l'IA

L'apprentissage profond est un type d'intelligence artificielle s'appuyant sur les réseaux de neurones. Il s'agit de systèmes mettant en réseau une brique élémentaire qu'il nous est nécessaire de définir, le neurone formel.

C'est en 1943 qu'est proposé le premier modèle de neurone formel par Warren Sturgis McCulloch et Walter Pitts. Leur modélisation s'appuie sur le fonctionnement observé des neurones biologiques.

C'est ensuite en 1958 que Franck Rosenblatt met en place un algorithme d'apprentissage applicable à un neurone formel, créant ainsi le perceptron [Rosenblatt 1958]. La représentation mathématique et le fonctionnement du perceptron sont décrits dans la section suivante.

1.1 Perceptron

Le perceptron est un neurone formel binaire. Cela signifie que son unique sortie est soit 0, soit 1, correspondant ainsi à deux classes prédictibles. L'ensemble de ses entrées est connecté à sa sortie.

L'action d'un neurone sur ses entrées (x_i) correspond à une fonction d'agrégation, qui dans le cas du perceptron est une somme pondérée. Pour la calculer, des paramètres sont associés à chaque entrée (i), que nous appelons : poids (w_i). La fonction d'agrégation du perceptron s'exprime ainsi : $\sum_{i=1}^{n} w_i x_i$. Son résultat s'appelle la valeur d'agrégation, notée z.

Une fonction d'activation est alors appliquée sur z, dont le résultat est noté a. Elle permet de définir un seuil à partir duquel le neurone s'activera. La valeur de sortie du neurone est directement dépendante de son activation. Dans le cas du perceptron de Rosenblatt, pour une tâche de classification, la fonction non linéaire de Heaviside est appliquée.

Elle est définie ainsi : $f(x)\begin{cases} 0, & si \quad x <= 0 \\ 1, & si \quad x > 0 \end{cases}$

Cette fonction indique que la réponse du neurone sera 0 si le résultat est inférieur ou égal à 0 et sera 1 sinon.

Il est tout à fait possible d'utiliser d'autres fonctions d'activation, par exemple la fonction sigmoïde (σ), pour une tâche de régression, qui est définie par : $f(x) = \frac{1}{1+e^{-x}}$

Nous représentons le perceptron à l'aide de la figure 1.2.

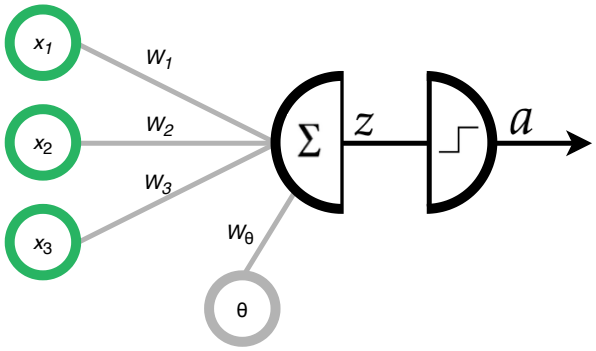

Figure 1.2 – Schéma du perceptron de Rosenblatt

Les paramètres (poids), du perceptron sont initialisés aléatoirement et sont mis à jour à l'aide d'une règle d'apprentissage définie comme : $W_i' = W_i + \alpha(Y_t - Y)X_i$, avec W_i' la nouvelle valeur du poids i, W_i la valeur actuelle du poids i, α le taux d'apprentissage, Y_t la sortie attendue, Y la sortie réelle et X_i l'entrée i.

La règle d'apprentissage permet ainsi d'optimiser les paramètres du neurone grâce à des données d'exemple, pour lesquelles nous connaissons la sortie attendue. Ces données doivent

représenter un problème de classification (ou de régression), afin de laisser le perceptron trouver automatiquement la séparation linéaire applicable.

Un dernier paramètre du perceptron correspond au biais (θ). Il s'agit du potentiel d'activation d'un neurone et est un poids pouvant être mis à jour. Il sera soustrait au résultat de la somme pondérée des entrées. De cette manière, la fonction d'activation sera appliquée sur le résultat de l'opération : $\sum_{i=1}^{n} W_i X_i - \theta$.

En s'appliquant sur le résultat de la somme pondérée juste avant l'application de la fonction d'activation, le biais influe sur la capacité d'un neurone à s'activer. Il permet ainsi de rendre une unité neuronale plus flexible.

De nos jours, le perceptron est une architecture très simpliste. Sa limite concerne sa capacité à ne résoudre que des problèmes linéairement séparables. Elle a pu être surmontée par la mise en réseau de plusieurs neurones simples.

1.2 Perceptron multicouche

Le principe du perceptron multicouche (*MultiLayer Perceptron*, *MLP*) [Rumelhart et al. 1985], est d'organiser en collaboration plusieurs perceptrons simples de manière à utiliser la sortie d'un neurone comme entrée d'un ou plusieurs autres. Plusieurs neurones peuvent être présents côte à côte, on parle ainsi de couche neuronale et ils peuvent être connectés à plusieurs neurones d'une couche suivante.

Un perceptron multicouche est nécessairement doté d'une couche neuronale d'entrée et d'une couche de sortie. Il peut également être complété par une ou plusieurs couches intermédiaires, appelées couches cachées.

Comme les sorties des unités neuronales sont connectées aux entrées des unités de la couche suivante, l'information se propagera obligatoirement de la couche d'entrée vers la couche de sortie. Il s'agit donc d'un réseau neuronal à propagation directe vers l'avant. Les entrées d'une unité neuronale correspondent à l'ensemble des sorties des unités de la couche la précédent.

Nous fournissons une représentation schématique d'un perceptron multicouche dans la figure 1.3.

L'architecture de ce type de système dépend directement de la tâche à laquelle nous l'appliquons. Chacune des couches cachées possède un nombre variable d'unités, tandis que pour les couches d'entrées et de sorties, ce nombre dépend de la tâche.

Ce système peut être considéré comme le premier réseau neuronal profond. Le nombre de couches cachées définit la profondeur du réseau. Plus un réseau est profond, plus il sera en mesure de résoudre des tâches complexes, mais plus il sera difficile à apprendre efficacement.

Après la définition de son architecture, un réseau nécessite d'apprendre à résoudre la tâche visée. Il lui est nécessaire d'exploiter des données et des méthodes d'apprentissage automatique, que nous détaillons dans la section suivante.

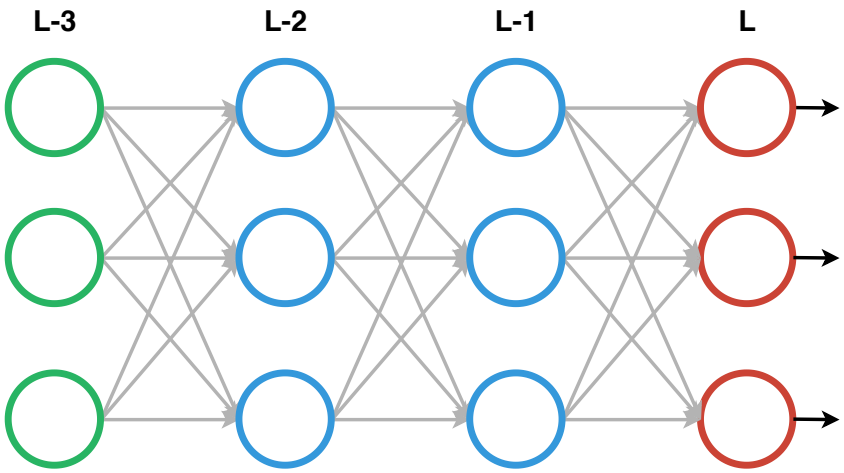

FIGURE 1.3 – Schéma d'un perceptron multicouche quelconque.

1.3 Apprentissage Automatique

L'apprentissage d'un système neuronal consiste en l'ajustement de ses paramètres pour trouver une combinaison résolvant efficacement la problématique d'une tâche. Cet ajustement s'appuie sur des données annotées selon la tâche visée. Il est important que les données utilisées représentent le plus fidèlement possible la tâche devant être résolue par le système. En présentant des données dont la réponse attendue est connue à l'avance, il est possible de modifier ses paramètres en favorisant les mises à jour maximisant le nombre de réponses correctes. L'apprentissage d'un système se base ainsi sur l'architecture d'un modèle, des données et des algorithmes.

Nous pouvons mentionner l'algorithme de descente du gradient qui permet l'apprentissage des paramètres neuronaux, mais aussi l'algorithme de rétropropagation du gradient [RUMELHART et al. 1985], qui rend possible le calcul efficace des gradients d'erreurs.

Nous expliquons davantage le fonctionnement de la descente de gradient dans la prochaine section, puis l'algorithme de rétropropagation dans la section suivante.

1.3.1 Algorithme de descente du gradient

Pour expliquer cet algorithme, il est d'abord nécessaire de définir le gradient. Il s'agit d'un vecteur composé de l'ensemble des dérivées partielles de la fonction de coût. C'est-à-dire des implications des paramètres du système à la production de l'erreur : $\frac{\partial C(w)}{\partial w}$. De plus, le gradient pointe nécessairement dans la direction du plus grand taux d'augmentation de la fonction considérée.

L'algorithme de descente du gradient est essentiel à l'apprentissage d'un système neuronal

et de par la nature du gradient, il se base sur la fonction de coût (C) appliquée.

Comme cette fonction quantifie l'écart entre une valeur émise par le système et la valeur attendue, il apparait naturel que l'objectif de la descente de gradient soit de minimiser cette fonction C. Cela permet de s'assurer que le système émet des valeurs les plus proches des valeurs attendues et donc maximise ses performances.

L'initialisation des paramètres d'un système permet de placer le point de départ de la descente de gradient. Minimiser la fonction, consiste donc à faire varier ses paramètres jusqu'à l'obtention d'un minimal. La descente de gradient correspond à la stratégie mise en œuvre pour s'assurer que la variation des paramètres se dirige vers un minimum.

Cette stratégie met en place une variation des paramètres de C par itérations successives. Nous appelons taux d'apprentissage (α) la taille du pas itératif appliqué pour le calcul des paramètres du temps t au temps $t+1$.

Nous donnons une représentation [1] d'une fonction de coût à deux paramètres, ainsi qu'une descente de gradient possible dans la figure 1.4.

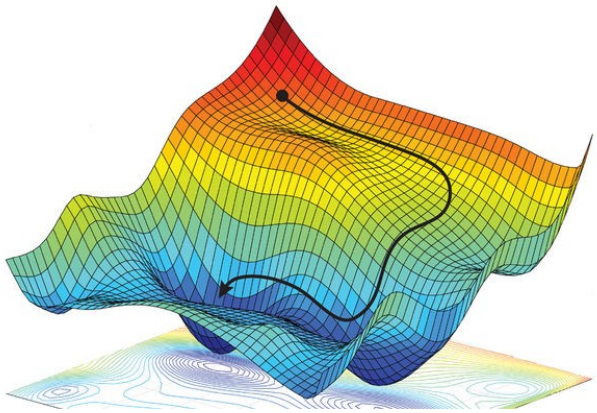

FIGURE 1.4 – Représentation de la descente de gradient sur une fonction de coût à deux paramètres.

Afin d'effectuer la descente de gradient vers un minimum, il est nécessaire de connaitre l'orientation de la pente de la fonction C. En connaissant son orientation, il est possible de mettre à jour les paramètres de façon à aller dans le sens inverse de la pente. Cette pente correspond en réalité à la dérivée de la fonction C.

Nous allons donc modifier les paramètres x en effectuant un pas itératif (α, compris entre 0 et 1) dans le sens inverse de la pente. Pour chaque paramètre d'un système, l'équation de la

1. Cette représentation est issue d'un article du magazine *Science*

descente de gradient s'exprime ainsi :

$$x_{t+1} = x_t - \alpha \frac{\partial C(x)}{\partial x} \tag{1.1}$$

Un inconvénient de la descente de gradient réside dans le risque de converger vers un mauvais minimum local. La fonction de coût considérée ne possède pas forcément un seul minimal. Il est très probable qu'elle possède des minimums locaux impliquant des performances sous-optimales par rapport à son minimum global. Notons toutefois que la descente de gradient ne garantit pas à un système de converger vers le minimum global. Il est donc important de minimiser les risques de convergence dans un mauvais minimum.

De plus, la vitesse de convergence a un rôle important dans l'apprentissage d'un réseau neuronal. Il existe plusieurs variantes de la descente de gradient permettant de converger plus où moins rapidement vers un minimum.

La première variante consiste à mettre les paramètres à jour après l'utilisation de chaque exemple d'apprentissages. Nous parlons dans ce cas d'une descente de gradient stochastique.

Une seconde variante consiste à les mettre à jour après l'utilisation de la totalité des exemples d'apprentissages. Dans ce cas, nous parlons d'une descente de gradient par lot (*batch*). Il s'agit d'exploiter la moyenne des gradients du lot, accélérant grandement la vitesse de convergence.

Enfin, une dernière variante consiste à utiliser n exemples avant de mettre à jours les paramètres. Nous parlons d'une descente de gradient par mini-lot (*mini-batch*). C'est une combinaison des deux méthodes précédentes qui est désormais la plus communément utilisée.

1.3.2 Algorithme de Rétropropagation

En fournissant un échantillon en entrée d'un système, les calculs effectués à partir de celui-ci vont se propager au fil des couches permettant ainsi d'obtenir une sortie. Cette sortie réelle est directement comparable à une sortie attendue et à l'aide d'une fonction de coût dérivable (C). L'objectif de cette fonction est de quantifier l'écart entre la sortie réelle et la sortie attendue. Ainsi, plus la valeur de cette fonction est grande, plus la sortie réelle est éloignée de la sortie attendue. Cet écart représente l'erreur en sortie du système, qui est nécessaire au calcul de l'implication d'un poids synaptique, noté $\frac{\partial C(w)}{\partial w}$.

L'algorithme de rétropropagation s'appuie sur le théorème de dérivation en chaîne des fonctions composées. Par emploi de ce théorème, il est possible de calculer efficacement l'implication des paramètres de la couche de sortie (L). Puis, suite à ce calcul, il est possible de déterminer l'erreur des unités neuronales de cette couche (δ_i^L).

À ce stade de l'algorithme, nous connaissons le gradient de la fonction de coût par rapport aux poids des unités de la couche (L), ainsi que les calculs ayant mené aux entrées de ces unités pendant la propagation avant. Il est donc possible de déterminer l'implication des paramètres

de la couche $L-1$, ainsi que le gradient de ses unités neuronales. Il sera ensuite possible d'en faire de même pour la couche $L-2$ et ainsi de suite jusqu'à la couche d'entrée. Le calcul de l'implication des paramètres et du gradient des unités neuronales s'effectue donc de la couche de sortie vers la couche d'entrée, d'où le nom d'algorithme de rétropropagation.

L'ensemble des neurones du système possède un paramètre de biais qui a lui aussi un impact sur la production de l'erreur de sortie. Ce paramètre doit être traité de la même manière que les autres poids.

En soi, les biais peuvent être considérés comme des poids associés à des vecteurs qui partent d'un seul nœud situé en dehors du réseau principal et dont l'activation est systématiquement 1. Cette valeur d'activation permet à un système de toujours avoir des neurones actifs, quelles que soient les valeurs d'entrées. Elle induit également une simplification des calculs de l'algorithme de rétropropagation pour les biais.

Cet algorithme est régi par 4 équations essentielles :

1. $\delta_i^L = C'(a_i^L) * \sigma'(z_i^L)$
2. $\delta_i^l = (\sum_k \delta_k^{l+1} * w_{ki}^{l+1}) * \sigma'(z_i^l)$
3. $\frac{\partial C}{\partial w_{ij}^l} = a_j^{l-1} * \delta_i^l$
4. $\frac{\partial C}{\partial b_i^l} = \delta_i^l$

Avec C' la dérivée de la fonction de coût, a la valeur d'activation, σ' la dérivée de la fonction d'activation et z la valeur d'agrégation. Ces équations correspondent respectivement à :

1. Le gradient de l'unité i sur la couche de sortie L.
2. Le gradient de l'unité i sur la couche cachée l.
3. L'implication du poids entre l'unité i de la couche l et l'unité j de la couche $l-1$.
4. L'implication du biais de l'unité i de la couche l.

Le calcul de l'implication des paramètres conditionne leur mise à jour lors d'une étape d'apprentissage. Ils sont mis à jour en suivant l'algorithme de descente de gradient permettant la minimisation la fonction de coût (C) appliquée.

Un système neuronal est un ensemble d'algorithmes permettant d'adapter des milliers, voir des millions, de paramètres pour trouver une configuration maximisant ses bonnes réponses. Les données exploitées pour son apprentissage doivent représenter au mieux la tâche visée, puisqu'au-delà des algorithmes, ce sont elles qui conditionnent la mise à jour des paramètres. Toutefois, des méthodes d'optimisation de l'apprentissage permettent de pousser les limites de l'apprentissage automatique. Nous décrivons les méthodes les plus communes dans la section suivante.

1.4 Optimisation de l'apprentissage

L'optimisation de l'apprentissage vise à pousser les limites, mais aussi à répondre à certaines problématiques. Comme la problématique des minimums locaux déjà évoquée. Au fil des années, différentes méthodes d'optimisation [Ruder 2016] ont été mises au point pour outrepasser cette difficulté, ainsi qu'optimiser la descente de gradient et l'apprentissage automatique. Nous décrivons certaines pratiques d'optimisation dans les sections suivantes.

1.4.1 Momentum

Cette méthode [Qian 1999] a pour objectif d'accélérer la descente de gradient et limite les risques de converger dans un mauvais minimum local de la fonction de coût. Concrètement, elle exploite la mise à jour précédente des paramètres du système pour optimiser la mise à jour suivante. Elle ajoute un coefficient de vélocité permettant d'accélérer la descente de gradient et réduire les oscillations. Ce coefficient augmentera le pas d'apprentissage si le gradient au temps t et au temps $t-1$ vont dans la même direction et le diminuera s'ils vont dans des directions opposées. La méthode du momentum peut être imagée comme l'action de la gravité sur une bille descendant le long d'une surface concave.

D'un point de vue mathématique, cette méthode transforme la mise à jour des paramètres de la manière suivante :

$$x_{t+1} = x_t + v_{t+1} \tag{1.2}$$

$$v_{t+1} = \mu v_t - \alpha \frac{\partial C(x)}{\partial x} \tag{1.3}$$

Avec v_t le coefficient de vélocité et μ le paramètre du momentum.

Ainsi, comme pour la descente de gradient traditionnelle cette méthode calcule le gradient présent puis effectue un déplacement dans la direction minimisant la fonction de coût. Elle possède toutefois un inconvénient, puisqu'elle n'est pas en mesure d'anticiper un changement de direction du gradient et donc de ralentir préventivement le coefficient de vélocité pour ne pas dépasser le minimum. Une extension de cette méthode corrige cet inconvénient.

Gradient accéléré Nesterov

Cette méthode [Nesterov 2013] est une évolution de la méthode du momentum. Elle consiste à effectuer un premier déplacement propre au terme du momentum, puis calculer le gradient présent pour effectuer un déplacement dans la direction minimisant la fonction de coût. Le déplacement effectué sur la fonction de coût s'effectue donc en deux temps. De plus, le terme du momentum ne pointe pas obligatoirement dans la direction minimisant la fonction de coût

contrairement au gradient. Ce qui signifie qu'avec un momentum à l'opposé du gradient cette méthode permet de ralentir préventivement le déplacement final.

Elle se traduit mathématiquement par les équations suivantes :

$$x_{t+1} = x_t + v_{t+1} \tag{1.4}$$

$$v_{t+1} = \mu v_t - \alpha \frac{\partial C(x + \mu v_t)}{\partial x + \mu v_t} \tag{1.5}$$

Avec v_t le coefficient de vélocité et μ le paramètre du momentum.

Les méthodes s'appuyant sur le momentum ne sont pas les seules exploitées pour optimiser la descente de gradient. Une famille d'algorithmes y est spécifiquement dédiée.

1.4.2 Algorithmes d'optimisation adaptatifs

Il s'agit d'une catégorie d'algorithmes qui vise à adapter la mise à jour de chacun des paramètres d'un système pendant l'apprentissage. Les stratégies mises en place par ces algorithmes permettent un apprentissage plus robuste par l'intermédiaire d'une descente de gradient plus efficace. Nous décrivons ci-dessous certains des algorithmes les plus communs, bien qu'il ne s'agisse pas d'un paysage exhaustif.

AdaGrad

L'algorithme AdaGrad (*Adaptative gradient*) [Duchi et al. 2011] est un algorithme dont l'intuition est d'adapter le taux d'apprentissage de chaques paramètres proportionnellement à leur historique de mise à jour. Il s'agit en soi de réaliser des mises à jour plus importantes pour les paramètres peu fréquents et des mises à jour moins importantes pour les paramètres plus fréquents. L'intérêt de cette approche réside dans le cas de données éparses, puisque le système sera capable de tirer bénéfice plus efficacement de l'information des données moins fréquentes.

Pour être mise en place, cette méthode modifie le taux d'apprentissage global en modifiant la règle de mise à jour des paramètres. La nouvelle règle mise en place est la suivante :

$$x_{t+1} = x_t - \frac{\alpha}{\sqrt{v_t^x + \epsilon}} \frac{\partial C(x)}{\partial x} \tag{1.6}$$

x représente le paramètre en cours de mise à jour, α le taux d'apprentissage, v_t^x l'historique accumulé du gradient de x et ϵ une constante très faible évitant la division par 0. Pour effectuer l'accumulation du gradient de x, v_t^x est défini ainsi :

$$v_t^x = v_{t-1}^x + \frac{\partial C(x)}{\partial x}^2 \tag{1.7}$$

Un autre avantage important de cette méthode d'optimisation est qu'elle règle automatiquement le taux d'apprentissage. Les avantages d'AdaGrad permettent un apprentissage plus efficace et plus robuste. Toutefois, cette méthode à l'inconvénient d'accumuler un gradient carré au dénominateur. Ainsi, la somme accumulée peut croitre très rapidement et rendre le taux d'apprentissage appliqué à certains paramètres extrêmement faible. Ce qui pourra rendre impossible l'apprentissage de connaissances supplémentaires pour le système. L'algorithme présenté ci-dessous est un des algorithmes visant à atténuer ce problème.

Adam

La méthode Adam (*Adaptative Moment Estimation*) [Kingma et Ba 2015] est une autre méthode ayant pour objectif d'adapter le taux d'apprentissage de chaque paramètre. Elle est complémentaire à la méthode AdaGrad, dans la mesure où elle corrige l'inconvénient de l'accumulation carré au dénominateur. Pour le corriger, elle définit une autre accumulation du gradient pour un paramètre x :

$$v_t^x = \beta_2 v_{t-1}^x + (1-\beta_2)\frac{\partial C(x)}{\partial x}^2 \tag{1.8}$$

Avec β_2 un taux de décroissance compris en 0 et 1.

En complément de l'accumulation carré des gradients (v_t^x), la méthode Adam préconise aussi la mise en place d'une accumulation simple des gradients (m_t^x) permettant un effet similaire à la méthode du momentum.

$$m_t^x = \beta_1 m_{t-1}^x + (1-\beta_1)\frac{\partial C(x)}{\partial x} \tag{1.9}$$

Avec β_1 un taux de décroissance aussi compris entre 0 et 1.

Une particularité de cette méthode concerne l'initialisation à 0 des vecteurs m_t et v_t. Il a été noté par les auteurs que ces vecteurs sont biaisés vers 0 lorsque les taux de décroissances sont faibles (soit β_1 et β_2 proche de 1). Ils ont ainsi proposé de contourner ce problème par le calcul d'une correction $m_t^{'x}$ et $v_t^{'x}$ de la manière suivante :

$$m_t^{'x} = \frac{m_t^x}{1-\beta_1^t} \tag{1.10}$$

$$v_t^{'x} = \frac{v_t^x}{1-\beta_2^t} \tag{1.11}$$

Ainsi avec cette méthode la mise à jour d'un paramètre x est définie ainsi :

$$x_t = x_{t-1} - \alpha \frac{m_t^{'x}}{\sqrt{v_t^{'x} + \epsilon}} \tag{1.12}$$

Avec α le taux d'apprentissage appliqué et ϵ une constante faible pour éviter la division par 0.

Il existe plusieurs autres variantes d'algorithmes d'optimisations, nous pouvons par exemple citer AdaDelta [Zeiler 2012], RMSprop et NAdam [Dozat 2016]. Cependant, ce sont les avantages de l'algorithme Adam qui en font actuellement un des algorithmes les plus communs. Les valeurs suggérées par défaut des taux de décroissances β_1 et β_2 sont respectivement 0.9 et 0.999.

Il est important d'optimiser la mise à jour des paramètres neuronaux pour optimiser l'apprentissage automatique. Cependant, la descente de gradient est aussi très dépendante de l'initialisation des paramètres qui en place le point de départ. La sous-section suivante nous permet de décrire certaines méthodes d'initialisation pouvant être exploitées.

1.4.3 Initialisation des paramètres neuronaux

L'initialisation des paramètres d'un modèle définit le point de départ de l'algorithme de descente du gradient. Même s'il est courant d'utiliser une initialisation aléatoire, cela ne permet pas systématiquement un point de départ favorisant une descente de gradient plus efficace. Pour s'en assurer, il est judicieux d'exploiter une stratégie d'initialisation spécifique.

Initialisation de Xavier

Nous pouvons notamment citer l'initialisation de Xavier [Glorot et Bengio 2010], du nom d'un des auteurs, qui est très largement suggérée pour l'apprentissage automatique. Dans ces travaux, les auteurs préconisent une initialisation respectant deux critères. Le premier concerne la moyenne des activations, qui doit être de 0. Le second concerne la variance des activations, qui doit rester la même pour chacune des couches neuronales.

Cette méthode d'initialisation conserve une part d'aléatoire, qui est toutefois caractérisée par les critères mentionnés. Les valeurs initiales des paramètres sont sélectionnées aléatoirement dans une distribution normale. Sa moyenne est 0 et sa variance est $\frac{1}{n^{l-1}}$ avec n le nombre de neurones de la couche $l-1$ et l la couche actuellement considérée.

Certaines méthodes n'utilisent pas la notion d'aléatoire, mais exploitent des données et le résultat d'un préapprentissage pour l'initialisation d'un réseau.

Transfert d'apprentissage

Cette méthode [Pan et Yang 2009] consiste à tirer bénéfice d'un système préentrainé pour une tâche proche, mais pas identique à la tâche qui sera apprise par le système. Le modèle préentrainé a logiquement atteint une configuration de poids stables suite à sa convergence pour

la première tâche (A). Le transfert d'apprentissage consiste en l'utilisation de cette configuration stable comme initialisation d'un système similaire. Suite à cette initialisation, le système peut être appris pour une autre tâche (B) en utilisant les données propres à celle-ci.

Les avantages de cette méthode sont multiples. Elle permet notamment d'accélérer la convergence d'un modèle. Les deux tâches exploitées étant proches, les paramètres optimisés pour la tache A modélisent nécessairement une information utile à la tâche B. Elle a également l'avantage de contrer le manque de données annotées pour une tâche finale, par l'utilisation de données tierces proches.

1.4.4 Régularisation des réseaux

La régularisation des réseaux est une optimisation globale de l'apprentissage automatique. Il s'agit de méthodes permettant de réguler l'apprentissage pour produire un système plus stable, plus robuste et plus efficace. De plus, les techniques de régularisation répondent à des problématiques qui peuvent concerner l'exploitation des caractéristiques d'entrées d'un système, ainsi que son fonctionnement interne pendant l'apprentissage.

La plupart des régularisations répondent à la problématique du sur-apprentissage [Dietterich 1995]. Ce problème consiste, pour un système en apprentissage, à trop optimiser ses paramètres pour un jeu de données précis. Cela signifie que le modèle aura capturé les informations généralisables du jeu de données, mais aussi le bruit présent. Un modèle subissant le sur-apprentissage connaît chacun de ses exemples d'apprentissages, mais est incapable de fournir des réponses satisfaisantes sur des données qui lui sont inconnues. Il perd par conséquent sa capacité de généralisation et devient inexploitable. Il est ainsi important de détecter le phénomène de sur-apprentissage et de s'en prémunir.

L'utilisation d'un ensemble de données indépendant permet de détecter un cas de sur-apprentissage. Il s'agit de l'ensemble de développement dont les exemples ne sont pas vus par le modèle pendant la phase d'apprentissage. Il est possible de comparer les résultats de la fonction de coût, sur l'ensemble d'apprentissage et de développement.

Dans un cas d'apprentissage nominal, le résultat de la fonction de coût va naturellement décroitre jusqu'à un minimum, que ce soit pour l'ensemble d'apprentissages, comme pour l'ensemble de développement. Cependant, lorsqu'il y a sur-apprentissage, le modèle s'optimise plus que nécessaire sur les exemples d'apprentissage, provoquant une réduction de sa capacité de généralisation. Il est ainsi possible d'observer une augmentation du coût sur l'ensemble de développement, tandis que le système continue de s'optimiser sur l'ensemble d'apprentissage. Nous donnons dans la figure 1.5, les courbes types du coût en fonction du nombre d'itérations des ensembles d'apprentissage et de développement, en cas de sur-apprentissage.

Il existe deux façons simples de se prémunir de ce phénomène. La première consiste à conserver la configuration de poids ayant la meilleure généralisation, c'est-à-dire le meilleur

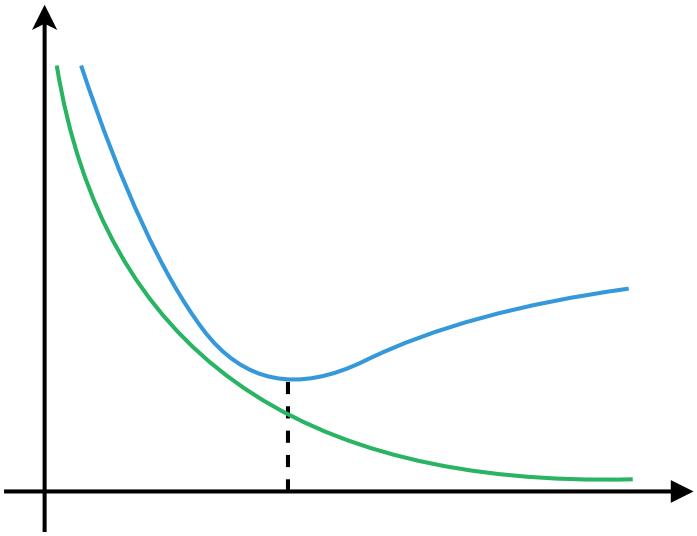

FIGURE 1.5 – Exemple de courbes du résultat d'une fonction de coût en fonction du nombre d'itérations. En bleu, l'ensemble de validation, en vert l'ensemble d'apprentissage.

coût sur l'ensemble de développement. La seconde consiste à augmenter la quantité de données d'apprentissage, permettant une plus grande variété des exemples. D'autres méthodes de régularisation permettent de s'en prémunir. Nous pouvons citer les régularisations L1 [TIBSHIRANI 1996] et L2 [HOERL et KENNARD 1970], ainsi que le dropout [SRIVASTAVA et al. 2014].

Toutefois, toutes les régularisations ne visent pas à se prémunir du sur-apprentissage. C'est notamment le cas de la récente méthode de normalisation des batchs [IOFFE et SZEGEDY 2015], qui visent principalement à optimiser l'apprentissage en agissant directement sur l'écart entre les caractéristiques d'entrées.

Nous donnons, ci-dessous, davantage de détails sur les méthodes de régularisation cités.

Régularisation L1

Cette méthode de régularisation est aussi appelée Lasso Regression [TIBSHIRANI 1996]. Il s'agit d'une pénalité appliquée durant l'apprentissage sur la fonction de coût. Son objectif est de pénaliser les connexions neuronales à forte pondération. Elles sont responsables d'une plus grande variance des sorties d'un système neuronal [GEMAN et al. 1992]. En soi, réduire la variance en sortie du système permet de rendre un modèle moins flexible, ce qui a pour effet de l'empêcher de trop bien s'optimiser aux données d'apprentissage. Cette régularisation agit directement sur le phénomène de sur-apprentissage.

Concrètement, elle ajoute à la fonction de coût la pénalité suivante :

$$C = C_0 + \lambda \sum_i |w_i| \tag{1.13}$$

C_0 représente la fonction de coût initialement utilisée, w_i les poids synaptiques du réseau et λ le coefficient associé à la pénalité. En complément, λ est un hyper paramètre. C'est-à-dire un paramètre dont la valeur est fixée humainement avant l'apprentissage d'un système. Il convient donc de mener des expérimentations pour déterminer sa valeur la plus adaptée.

Comme nous l'avons vu, l'apprentissage d'un modèle consiste à trouver la configuration de poids minimisant la fonction de coût appliquée. En ajoutant cette pénalité basée sur la valeur des poids synaptiques, nous contraignons le système à converger vers une configuration les minimisant. La pénalité va permettre de réduire les poids vers 0. Avec cette régularisation il est possible qu'un poids atteigne zéro, ce qui à pour effet de désactiver la connexion neuronale associée. Une forme similaire de régularisation réduit les poids vers 0, sans toutefois leur permettre d'être exactement égales à 0.

Régularisation L2

Également appelée Ridge Regression, cette méthode vise aussi à pénaliser les connexions neuronales à forte pondération [Hoerl et Kennard 1970]. Comme pour la régularisation L1, cette méthode exploite une pénalité qui sera ajoutée à la fonction de coût. Elle s'exprime ainsi :

$$C = C_0 + \lambda \sum_i w_i^2 \tag{1.14}$$

C_0 représente la fonction de coût initialement utilisée, w_i les poids synaptiques du réseau et λ le poids associé à la pénalité.

Cette régularisation conduira au même effet que la régularisation L1, c'est-à-dire diminuer le phénomène de sur-apprentissage.

Dropout

Le dropout [Srivastava et al. 2014] est une autre régularisation efficace contre le sur-apprentissage. Elle vise à impacter le modèle neuronal en lui même pour réduire sa spécialisation excessive.

Il s'agit de désactiver temporairement et aléatoirement des neurones composants le réseau à chaque étape d'apprentissage. Cela signifie que les unités désactivées changent à chaque étape. Les neurones désactivés ne sont pas utilisés pour propager l'exemple d'apprentissage dans le réseau et ils ne bénéficieront pas non plus de la correction des poids suite à l'application de la rétropropagation. Cette désactivation n'est appliquée que pendant la phase d'apprentissage. Pour réaliser aléatoirement la désactivation, un hyper paramètre P est défini. Il correspond à

la probabilité de désactivation d'un neurone, compris entre 0 et 1. Nous donnons un exemple d'application de dropout avec $P = 0,5$ dans la figure 1.6.

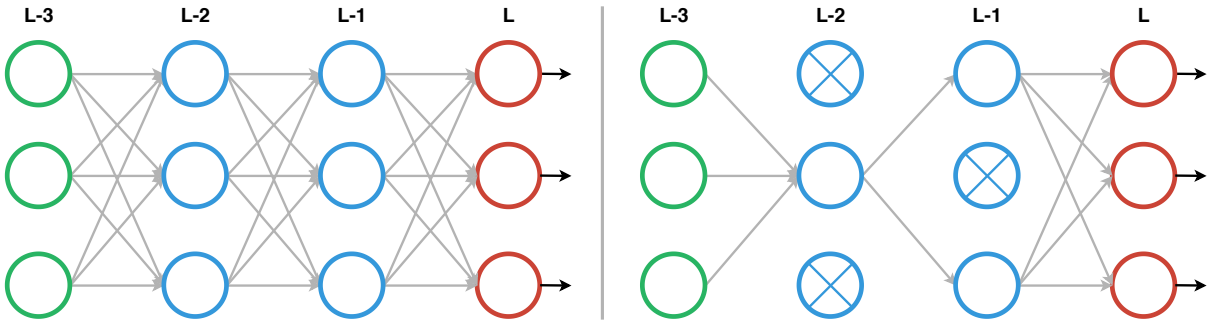

FIGURE 1.6 – Représentation du dropout avec $P = 0,5$. À gauche un système quelconque sans dropout. À droite le même système avec l'application d'un dropout.

En soi, cette méthode permet d'exploiter le système neuronal dans des configurations différentes à chaque étape d'apprentissage. Modifier ainsi sa configuration va permettre de simuler une activation éparse des unités le composant, lui permettant d'apprendre des représentations plus robuste.

Normalisation des batchs

Cette méthode vise à réduire les écarts entre les caractéristiques d'entrées de toutes les couches d'un système pour optimiser l'apprentissage de représentations efficaces [IOFFE et SZEGEDY 2015].

L'information circulant tout au long d'un réseau de neurones peut être représentée dans un espace. Un système doit trouver une représentation permettant de séparer efficacement différentes classes. Cependant, au sein d'une même classe, bien que proche, chaque exemple est différent. Ils ont donc tous une projection différente dans l'espace. Ces différences de projection correspondent au décalage des co-variables (*covariate shift*).

L'inconvénient de ce décalage est qu'il affecte négativement l'apprentissage lorsqu'il devient trop important. Il ralentira l'apprentissage d'une représentation efficace et peut conduire à l'obtention d'une représentation sous-optimale. Exploiter des batchs composés d'une sélection aléatoire des éléments d'une même classe suffirait à réduire cet inconvénient. Toutefois, cela ne s'applique que pour la couche d'entrée du système.

Au sein des couches cachées d'un système neuronal profond, les représentations intermédiaires des exemples sont en constante évolution tout au long de l'apprentissage. La mise à jour des paramètres d'une couche quelconque L, impactera nécessairement la représentation fournie à la couche L+1. Cela signifie que la distribution des entrées des unités neuronales d'une

couche est modifiée à chaque mise à jour des paramètres de la couche précédente. C'est ce qui est appelé le décalage interne des covariables (*internal covariate shift*).

La méthode de normalisation des batchs a pour objectif de réduire ce décalage interne en réalisant une normalisation appliquée individuellement à chaque unité neuronale. La normalisation est calculée avec la moyenne (μ) et la variance (σ) d'un mini-lot (B), dont les équations sont les suivantes :

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} z^i \tag{1.15}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(z^i - \mu_B)^2 \tag{1.16}$$

Où m représente le nombre d'éléments du mini-lot et z^i la valeur d'activation de l'unité neuronale en cours de normalisation suite au i^{eme} exemple du mini-lot. À partir de la moyenne et de la variance des activations d'un mini-lot, la normalisation appliquée est définie comme :

$$z_{norm}^i = \frac{z^i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{1.17}$$

La constante ϵ est ajoutée pour des raisons de stabilité numérique. Il s'agit d'une constante empêchant la division par 0. Pour finir la normalisation, un dernier calcul est à effectuer :

$$z_{out}^i = \gamma z_{norm}^i + \beta \tag{1.18}$$

La valeur d'échelle (γ) et de décalage (β) sont deux paramètres appris pendant la phase d'apprentissage du système. Ils bénéficient tous deux de l'algorithme de rétropropagation pour leur mise à jour. L'intérêt de ces deux paramètres est qu'ils permettent au système de réguler la normalisation appliquée à chaque unité pour la rendre la plus efficace possible. Ils permettent également au système d'annuler l'application de la normalisation dans le cas où $\gamma = \sqrt{\sigma_B^2 + \epsilon}$ et $\beta = \mu_B$. Ainsi, cette méthode ne peut qu'apporter une amélioration des résultats.

En soi, elle normalise les entrées de chacune des couches du système en les redimensionnant. Cela a pour effet d'accélérer l'apprentissage de représentations internes efficaces, améliorant ainsi la stabilité et les performances du système final.

Les régularisations sont des ajouts importants, qui permettent l'amélioration de l'algorithme d'optimisation utilisé. Ces ajouts participent à contrer certaines problématiques de l'apprentissage automatique. Il existe tout de même des problématiques supplémentaires liées à des spécificités de l'apprentissage neuronal. Nous proposons de couvrir ces spécificités et les problématiques engendrées dans la section suivante.

1.5 Spécificités de l'apprentissage neuronal

L'apprentissage neuronal peut engendrer un très grand nombre d'opérations, d'autant plus dans le cadre d'architectures neuronales complexes. De légères fluctuations peuvent finir par conduire à d'importantes problématiques au fil des opérations. C'est notamment le cas de celles liées au gradient.

1.5.1 Disparition / Explosion du gradient

Plus un réseau de neurones possède de couches cachées, plus il est susceptible de rencontrer un problème de gradient, que ce soit une explosion ou une disparition [Bengio, Simard et al. 1994].

Une mauvaise initialisation des paramètres peut également être à l'origine d'un problème de gradient. En effet, une initialisation trop grande de l'ensemble des paramètres conduit à un problème d'explosion de gradient. Tandis qu'une initialisation trop petite de l'ensemble des paramètres conduit à un problème de disparition de gradient.

Ces problèmes surviennent en exploitant l'algorithme de rétropropagation, puisque le calcul du gradient s'effectue par une suite de multiplication successives. Plus nous calculons le gradient d'un paramètre d'un neurone éloigné de la couche de sortie, plus le nombre de multiplications successives augmente. Ce sont elles qui sont responsables de l'explosion ou de la disparition du gradient.

Multiplier des valeurs inférieures à 1 conduira nécessairement à des valeurs extrêmement faibles. Lorsque la valeur du gradient est trop faible, le paramètre associé ne subira que des modifications minimes. Empêchant ainsi l'unité neuronale associée d'apprendre efficacement, c'est la disparition du gradient.

Multiplier des valeurs supérieures à 1 est susceptible de conduire à une augmentation forte au fil des multiplications. Lorsque le gradient est trop fort, il provoque de trop grandes modifications des paramètres à la moindre erreur du réseau. Dans ce cas, il s'agit d'une explosion de gradient.

Afin de garantir une bonne convergence des paramètres d'un réseau neuronal profond, il est important de contenir le gradient. Des méthodes ont été développées pour se prémunir de ces deux problèmes.

Gradient clipping

Nous pouvons tout d'abord citer la méthode du *gradient clipping* [Pascanu et al. 2013]. Cette méthode vise à empêcher l'explosion du gradient. Elle consiste simplement à placer un seuil haut que le gradient ne peut pas dépasser. Le gradient atteindra au maximum la valeur de ce seuil, supprimant ainsi les risques de modifications excessives d'un paramètre.

Fonction d'activation ReLU

La fonction d'activation ReLU (*Rectified Linear Unit*) est une fonction non linéaire [Nair et G. E. Hinton 2010]. Elle s'exprime ainsi : $ReLU \begin{cases} 0, & si \quad x <= 0 \\ x, & si \quad x > 0 \end{cases}$

La particularité de cette fonction est que sa dérivée est égale à 1 lorsque $x > 0$. Cela a pour effet de réduire le problème de disparition du gradient. Également, sa dérivée est égale à 0 lorsque $x >= 0$, ce qui rend l'unité neuronale concernée inactive. Cela implique que ses paramètres ne seront pas mis à jour et encourage le réseau à exclure ce qui ne lui est pas nécessaire.

Cette fonction améliore la convergence d'un réseau et empêche la saturation des neurones. Ces propriétés en font désormais une des fonctions d'activation les plus couramment utilisées. Elle possède toutefois l'inconvénient de saturer les neurones pour les valeurs négatives de x. Cela empêche certains neurones de produire d'autres valeurs que 0, ce qui peut rendre le réseau passif dans le cas d'un trop grand nombre de neurones impactés. Il s'agit du problème de *Dying ReLU*.

Plusieurs variantes de cette fonction ont été développées dans le but de réduire ce problème. Nous pouvons notamment citer les fonctions leaky ReLu [Maas et al. 2013], PReLU [K. He et al. 2015] et SeLU [Klambauer et al. 2017].

1.6 Modélisation de séquences

Une tâche de modélisation de séquence consiste à prédire un symbole n à partir des $n-1$ observations précédentes. Pour cette tâche, une séquence peut être définie comme une suite d'éléments non aléatoires et interdépendants.

Au sein de cette thèse, nous travaillons sur la parole, qui peut être considérée comme une séquence d'éléments issus d'un ensemble fini. Aussi, pour être intelligible, la parole est nécessairement une suite d'éléments qui ne sont pas produits aléatoirement. Nous pouvons ainsi considérer qu'un élément de cette séquence est dépendant des éléments le précédent.

Les réseaux à propagation avant, que nous avons décrits jusqu'ici, ne sont pas les plus adaptés à ce type de tâche. Par nature, un perceptron multicouche exploite les différentes entrées, qui lui sont présentées successivement, de manière indépendante. Il n'est donc pas en mesure de modéliser des dépendances entre des entrées successives.

Pour effectuer la modélisation d'une séquence à l'aide d'un système de ce type, il est nécessaire de fournir directement toute la séquence en entrée. Cependant, le dimensionnement de la couche d'entrée d'un perceptron multicouche est directement dépendant des entrées que nous lui fournissons. Ainsi, le nombre d'unités nécessaires est directement dépendant du nombre d'éléments présent dans la séquence à modéliser. Cela signifie aussi que les séquences en en-

trées doivent être de taille fixe. Même s'il est aussi possible d'exploiter des fenêtres glissantes sur la séquence, il ne sera toutefois pas possible de modéliser les dépendances en dehors de ces fenêtres.

Un type de réseau neuronal a été conçu pour exploiter des entrées de taille variables et modéliser des dépendances entre les caractéristiques d'entrées. Ces spécificités en font des systèmes particulièrement adaptés à la modélisation de séquences, y compris la parole. Nous les décrivons dans la sous-section suivante.

1.6.1 Réseau récurrent

Les réseaux neuronaux récurrents (*Recurrent Neural Network*, *RNN*) [Jordan 1986] sont désormais un incontournable de l'apprentissage automatique neuronal. L'intérêt de ce type de réseau réside dans leurs capacités à modéliser des séquences de tailles variables. Pour ce faire, l'intégralité de la séquence sera présentée au réseau dans l'ordre chronologique de ses éléments. En complément, leur architecture leur permet de modéliser des dépendances entre les différents éléments de la séquence, toujours selon l'ordre chronologique.

Une couche neuronale récurrente reste similaire à une couche neuronale standard. Sa différence réside dans la prise en compte de sa sortie précédente comme entrée additionnelle, créant ainsi une boucle à l'échelle des unités neuronales. L'intérêt de cette boucle réside dans l'aspect temporel qu'elle met en place. La figure 1.7 illustre le principe de boucle d'une couche neuronale récurrente, ainsi que l'aspect temporel des réseaux récurrents.

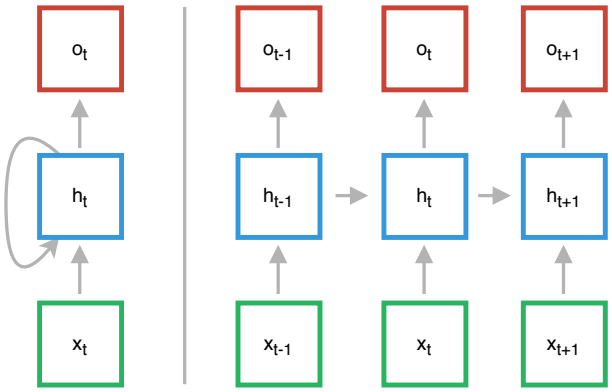

Figure 1.7 – Représentation d'un réseau neuronal récurrent. Chaque carré représente l'entièreté d'une couche. À gauche, le principe de récurrence. À droite, une représentation équivalente entre le temps t-1 et le temps t+1

Le principe de boucle permet une prise en compte de l'historique de la séquence qui favorisera la modélisation de dépendances. De plus, il rend possible l'exploitation de séquences de taille variable. Contrairement à un réseau standard qui fait correspondre une entrée à une

sortie, un réseau récurrent à la capacité d'être modulable. C'est-à-dire qu'il est possible de faire correspondre plusieurs entrées à plusieurs sorties (*many-to-many*), mais aussi plusieurs entrées à une seule sortie (*many-to-one*).

Il est à noter que ce type de réseau est particulièrement sensible aux problèmes de gradient, dans la mesure où la récurrence accroit considérablement la profondeur du réseau.

Un réseau récurrent simple possède toutefois une mémoire limitée des éléments précédents de la séquence. Même si en théorie ils sont capables de modéliser des dépendances éloignées, des études ont montré leurs limites [Bengio, Simard et al. 1994]. Il apparait difficile de modéliser des dépendances entre deux éléments éloignés dans la séquence présentée à un réseau récurrent.

Un type de réseau récurrent a été développé pour répondre à ce besoin. Il s'agit des réseaux récurrents à mémoire court et long terme, que nous explicitons dans la sous-section suivante.

Réseaux récurrents à mémoire court et long terme

Les réseaux récurrents à mémoire court et long terme (*long-short term memory, LSTM*) ont été développés en 1997 [Hochreiter et Schmidhuber 1997]. Ils ont pour but de répondre à la problématique de la modélisation de dépendances éloignées au sein d'une séquence. Les LSTM en sont capables grâce à une mémoire interne contrôlée à l'aide de portes. Elles sont au nombre de trois et sont chacune responsable d'une partie distincte du fonctionnement d'une cellule LSTM.

La porte d'oubli lui permet de réaliser la gestion de la mémoire interne. Elle permet de remettre à zéro tout ou partie du contenu de cette mémoire. Puis, la porte d'entrée effectue la gestion des ajouts à la mémoire interne de la cellule en fonction de l'élément courant de la séquence présentée. Enfin, la porte de sortie. Elle a pour but de calculer la sortie effective de la cellule en se basant sur l'élément courant de la séquence, ainsi que l'état de la mémoire interne.

Nous donnons dans la figure 1.8 le schéma d'une cellule LSTM avec chacune des portes et la mémoire interne mise en avant.

Une cellule GRU (*Gated Recurrent Unit*) ayant des propriétés similaires, avec une réduction et une réorganisation des portes a été proposé dans [K. Cho, Van Merriënboer, Bahdanau et al. 2014].

En complément, il est régulier de voir ce type de réseau avec une implémentation bidirectionnelle. Cela signifie que la séquence d'entrées sera présentée au réseau dans le sens chronologique de ses éléments, mais aussi dans le sens antéchronologique. Chaque couche bidirectionnelle est composée de deux couches neuronales, une par sens. Puis une concaténation des sorties des deux couches est effectuée pour fournir la sortie finale de la couche bidirectionnelle. Ainsi, le réseau aura la capacité de modéliser des dépendances dans les deux sens, c'est-à-dire les dépendances passées et futures.

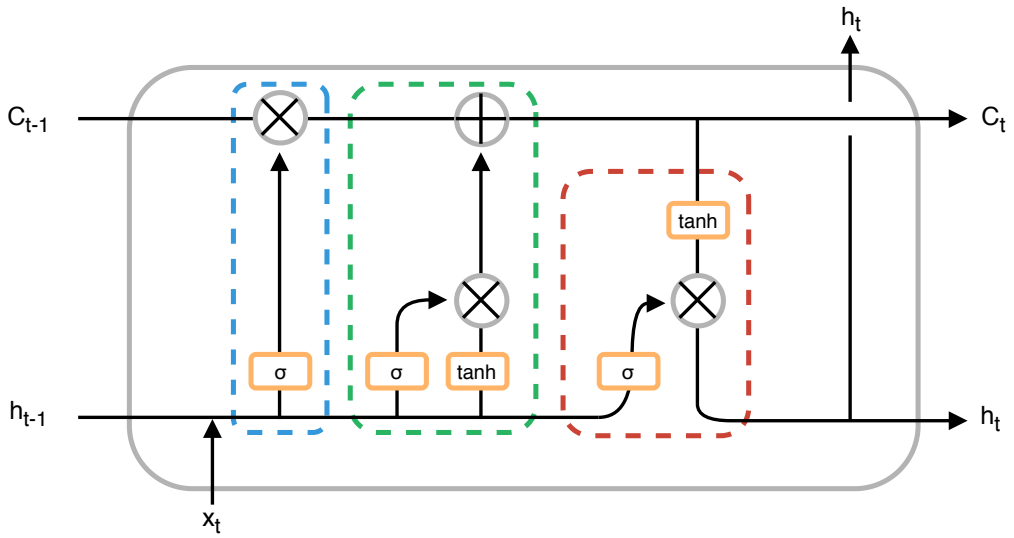

FIGURE 1.8 – Représentation d'une cellule LSTM. En bleu la porte d'oubli, en vert la porte d'entrée et en rouge la porte de sortie. La mémoire interne de la cellule est représentée par C_t.

1.6.2 Réseau neuronal convolutif

Les réseaux neuronaux convolutifs (*Convolutional Neural Network*, *CNN*) [LECUN et al. 1990] ont pour objectif de traiter efficacement des données représentées sous forme de tableau. Ils sont ainsi particulièrement efficaces pour le traitement d'image (tableau 2D) et de parole, notamment sous forme de spectrogrammes. Un réseau de ce type effectue des produits de convolution, contrairement aux réseaux précédemment détaillés qui effectuent des produits matriciels. Comme pour un perceptron multicouche, l'information au sein d'un CNN se propagera vers l'avant. Toutefois, son architecture diffère dans la mesure où toutes les unités neuronales d'une couche ne sont pas reliées à l'entièreté de la couche suivante. On parle de champ récepteur d'une unité pour mentionner les sorties neuronales, de la couche précédente, qui lui sont connectées.

Un réseau convolutif est couramment composé de couches de convolution et de couches de sous-échantillonnage entre chacune d'entre elles.

La couche de convolution permet l'extraction de caractéristiques et s'effectue à l'aide d'un filtrage par convolution. Afin d'effectuer ce filtrage, une fenêtre glissante (filtre) est appliquée sur les entrées. Elle effectue le calcul du produit de convolution entre cette fenêtre et la portion d'entrée considérée. Cela signifie que le filtre appliqué représentera la caractéristique à extraire par le réseau et ses paramètres sont estimés pendant l'apprentissage du réseau.

Les couches de sous-échantillonnages ont pour objectif la réduction de dimension des caractéristiques issues de la convolution, sans perte d'informations importante. Une méthode

commune est le *max-pooling* qui consiste à favoriser les valeurs les plus importantes des caractéristiques issues de la convolution. L'avantage de ce sous-échantillonnage réside dans sa capacité à mettre en avant les caractéristiques importantes tout en supprimant les valeurs non pertinentes.

Même si ce type de réseau a été conçu pour la modalité image, il s'est particulièrement illustré dans le cadre de la reconnaissance de la parole [Peddinti et al. 2015; Amodei et al. 2016], dont l'état de l'art sera l'objet du chapitre suivant.

1.6.3 Architecture Encodeur-Décodeur

Les architectures de ce type sont apparues récemment [K. Cho, Van Merriënboer, Gulcehre et al. 2014]. Leur principe consiste en l'utilisation conjointe de deux modules, un encodeur et un décodeur.

Un encodeur, est un réseau de neurones récurrent transformant une séquence d'entrée $X = (x_1, x_2, ..., x_n)$ en une unique représentation vectorielle de taille fixe (X').

Un second réseau récurrent est responsable de la transformation de X' en une séquence de sortie Y, il s'agit du décodeur. Pour effectuer cette transformation, il effectuera à chaque temps t une distribution de probabilités $P(Y_t|X')$ sur l'ensemble des symboles prédictibles par le système.

Nous donnons dans la figure 1.9, une représentation schématique de l'architecture encodeur-décodeur.

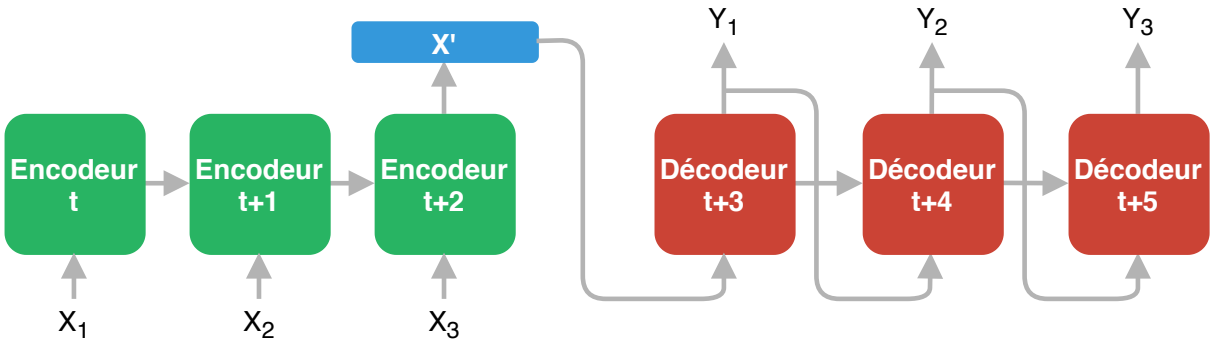

Figure 1.9 – Représentation d'une architecture encodeur-décodeur. En vert, l'encodeur à un instant t, en rouge, le décodeur à un instant t et en bleu le vecteur de contexte produit par l'encodeur.

Dans ce type d'architecture, le vecteur de contexte peut être limitant dans la mesure où il peut devenir un goulot d'étranglement. Cela a pour effet de réduire leur capacité pour effectuer la modélisation de dépendances longues distances.

Cet inconvénient à toutefois été surpassé avec les mécanismes d'attention proposés récemment dans le cadre de la traduction automatique [Bahdanau, K. Cho et al. 2015 ; Luong et al. 2015]. Ils permettent au modèle de se concentrer sur les parties pertinentes de la séquence d'entrée selon les besoins. Nous décrivons le fonctionnement de ces mécanismes ci-dessous.

Mécanisme d'attention

Apparus initialement dans le cadre de la traduction automatique, les mécanismes d'attention se sont montrés prometteurs pour la modélisation de séquences. Nous décrivons uniquement la première version des mécanismes d'attention proposée dans [Bahdanau, K. Cho et al. 2015]. Une variante a ensuite été proposée par [Luong et al. 2015], modifiant légèrement la façon de calculer les scores d'attention et le vecteur de contexte que nous allons aborder ci-dessous.

Le premier apport des mécanismes d'attentions concerne les états cachés qu'ils fournissent au décodeur. En effet, ils ne fournissent plus uniquement le dernier état caché de l'encodeur, mais l'ensemble des états cachés calculés pour chacune des entrées x_i passées dans l'encodeur. Cela permet de fournir en entrée du décodeur l'intégralité des informations disponibles pour chaque élément de la séquence d'entrée.

Le second apport de ces mécanismes concerne les décodeurs, qui doivent être en mesure de récupérer l'information pertinente au bon endroit. Cela consiste, avant de fournir le vecteur de contexte au décodeur, à attribuer un score d'attention compris entre 0 et 1 à chaque état émis par l'encodeur. Il s'agit d'une pondération, tel que la somme de ces scores est nécessairement égale à 1. Au final, le vecteur de contexte X' est construit par somme des états cachés pondérés de l'encodeur.

Un autre intérêt des mécanismes d'attention est de ne plus alimenter le décodeur avec un vecteur de contexte fixe (X'), mais plutôt de rendre ce vecteur dynamique (X'_t). Il est ainsi recalculé à chaque temps t pour mieux refléter le contexte nécessaire à l'émission y_t devant être produite par le décodeur.

1.6.4 Transformers

Les Transformers ont été proposés par [Vaswani et al. 2017]. Il s'agit d'un modèle s'appuyant sur une architecture encodeur-décodeur exploitant différemment les mécanismes d'attention. L'architecture des Transformers est aussi différente de celle qui est conventionnelle, dans la mesure où il s'agit d'un empilement de plusieurs encodeurs et de plusieurs décodeurs. Le nombre d'encodeurs et de décodeurs doit être identique.

Un encodeur est composé d'une couche d'auto-attention (*self-attention*), suivie d'une couche linéaire traditionnelle.

Le décodeur est complété par une couche d'attention, l'aidant à se concentrer sur les parties pertinentes de la séquence d'entrée.

En complément, les Transformers ont introduit les mécanismes d'attention à plusieurs têtes (*multi-head attention*) [Vaswani et al. 2017].

Ils obtiennent d'importantes performances, mais ont toutefois l'inconvénient d'être couteux à l'apprentissage. Ils exploitent un nombre important de paramètres, impliquant un temps de convergence élevé, ainsi qu'une grande quantité de données [Devlin et al. 2019 ; Brown et al. 2020].

1.7 Conclusion

Au sein de ce chapitre, nous avons évoqué les aspects primordiaux de l'apprentissage neuronal. De son origine à certaines de ses variantes les plus récentes, en passant par son fonctionnement algorithmique. Nous avons vu qu'il s'agit d'un sous-domaine de l'IA et que son avantage réside dans sa capacité à apprendre à répondre à une problématique à partir de données annotées par l'humain. Nous avons également vu qu'avec l'apprentissage neuronal certaines problématiques apparaissent, nécessitant le développement de nouvelles méthodes.

Enfin, de par la variété et la complexité de l'apprentissage neuronal, ce chapitre ne peut le représenter dans sa totalité. Pour un paysage plus complet, nous encourageons le lecteur à consulter [Goodfellow et al. 2016].

Chapitre 2

Reconnaissance de la parole

Sommaire

Dans le cadre de cette thèse, nous souhaitons effectuer des travaux de compréhension du langage parlé. Celle-ci est couramment effectuée à l'aide de systèmes successifs dont le premier est dédié à la tâche de reconnaissance automatique de la parole (RAP). Ainsi, les technologies de RAP employées ont une influence sur les performances finales de compréhension.

Dans ce chapitre, nous proposons de décrire les méthodes couramment employées pour la tâche de RAP. Nous définissons tout d'abord cette tâche, puis nous détaillons les méthodes s'appuyant sur des systèmes hybrides combinant des modèles de Markov cachés (*Hidden Markov Model*, *HMM*) [Rabiner 1989] et des modèles à mélange gaussien (*Gaussian Mixture Model*, *GMM*). Nous décrivons ensuite les approches neuronales récentes remplaçant les GMM, ainsi que celles rendant les systèmes dits de bout en bout pertinents.

2.1 Définition

Comme nous avons commencé à l'évoquer dans le chapitre précédent, la parole peut être considérée comme une suite finie d'événements non aléatoires. Effectuer la reconnaissance de la parole consiste à produire une séquence de mots à partir d'observations sur un signal acoustique de parole.

L'approche statistique est très majoritairement utilisée dans le domaine depuis des décennies [Jelinek 1976]. Le principe consiste à rechercher la séquence de mots $W' = w_1, w_2, ...w_n$ à partir d'observations acoustiques $X = x_1, x_2, ...x_t$ qui maximise :

$$W' = \arg\max P(W|X) \tag{2.1}$$

Toutefois, il est compliqué de modéliser directement cette probabilité $P(W|X)$. Par application du théorème de Bayes cette équation peut s'écrire sous la forme :

$$W' = \arg\max \frac{P(X|W)P(W)}{P(X)} \tag{2.2}$$

où $P(X)$ est une constante indépendante de W. Il est donc possible de simplifier l'équation à résoudre :

$$W' = \arg\max P(X|W)P(W) \tag{2.3}$$

La reconnaissance de la parole peut ainsi être prise en charge par l'utilisation conjointe de deux modèles plus simples à modéliser.

Le premier est le modèle acoustique. Il permet de modéliser la probabilité d'observer la séquence acoustique X lorsque les mots W sont prononcés : $P(X|W)$.

Le second est le modèle de langage, qui modélise la probabilité d'observer le mot W dans la langue reconnue : $P(W)$.

Ces deux modèles définissent des éléments essentiels à la mise en place d'un système de RAP. Ils sont toutefois complétés par plusieurs autres modules nécessaires que nous n'aborderons pas en détail.

Il s'agit tout d'abord d'un module de segmentation, dont l'objectif est de découper un signal de parole en fenêtres observables. L'intérêt étant de sélectionner les zones comportant de la parole et d'écarter celles contenant du bruit dans un signal audio.

Il s'agit ensuite d'un module d'extraction des paramètres acoustiques, qui est responsable de la conversion d'un signal audio de parole en vecteurs d'observations acoustiques. L'intérêt de ce module est, au sein d'un segment de parole, d'extraire les informations pertinentes de la parole. La méthode d'extraction la plus fréquente consiste à calculer des coefficients cepstraux (*Mel-Frequency Cepstral Coefficient*, *MFCC*) [Davis et Mermelstein 1980].

Il s'agit enfin d'un dictionnaire de prononciation qui à chaque mot du vocabulaire associe la séquence de phonèmes correspondante. Il fait le lien entre la modélisation acoustique et le modèle de langage.

Nous donnons une représentation schématique d'un système complet de reconnaissance de la parole dans la figure 2.1.

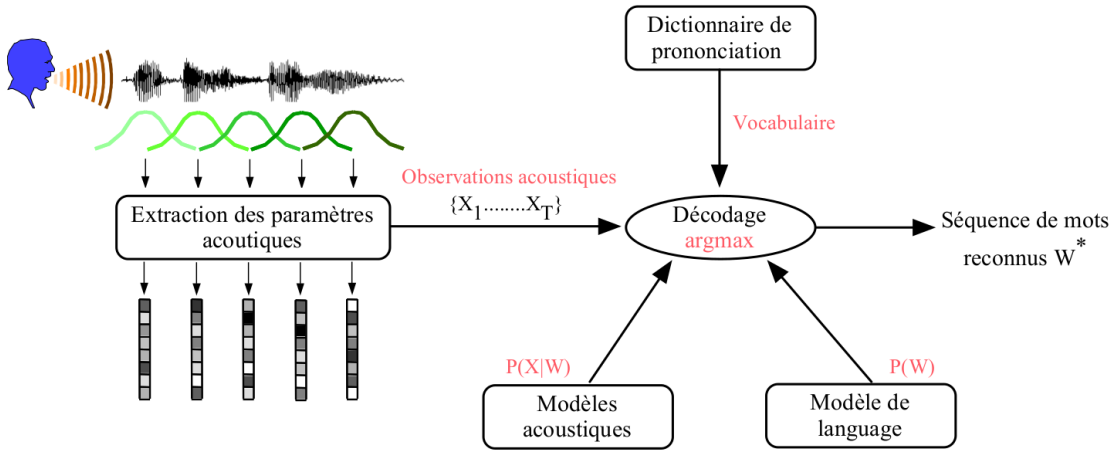

Figure 2.1 – Représentation d'un système de reconnaissance de la parole [Ghannay 2017].

2.2 Modélisation acoustique Markovienne

La modélisation acoustique consiste à estimer la probabilité $P(X|W)$. Jusqu'à très récemment, les modèles de Markov cachés (*Hidden Markov Model*, *HMM*) [Rabiner 1989] étaient utilisés. Ils ont d'abord été couplés à des modèles à mélange de gaussiennes (*Gaussian Mixture*

Modèle, *GMM*), qui ont ensuite été surpassés par l'usage de réseaux de neurones [G. Hinton et al. 2012].

Au sein des sous-sections suivantes, nous proposons de détailler ces technologies de modélisation acoustique.

2.2.1 Modèles de Markov cachés

Un modèle de Markov caché est un modèle statistique composé d'états successifs et de transitions formant une chaîne de Markov. Les transitions entre états sont unidirectionnelles et permettent de modéliser la probabilité de passer d'un état au suivant. Il existe une transition bouclant sur chaque état, permettant de conserver un état d'un temps t au temps $t+1$.

Un HMM est ainsi caractérisé par un ensemble d'états émetteurs, une matrice de transition indiquant les probabilités de transition d'un état au suivant, ou de bouclage sur un état, et des densités de probabilités pour l'émission des observations. Ces dernières sont obtenues par l'emploi de modèles GMM ou des réseaux de neurones profonds (*Deep Neural Networks*, *DNN*) que nous détaillerons plus en avant.

Dans le cadre de la modélisation acoustique, il est nécessaire de découper la parole en unités sous-lexicales associées à un mot. Le phonème est très couramment utilisé dans la mesure où il s'agit de la plus petite unité de son discriminant composant un mot. Chaque phonème est représenté par un modèle HMM distinct, puis, par composition des phonèmes successifs, il est possible de modéliser des mots ou des séquences de mots. Cela correspond à exploiter en chaîne les différents HMM des phonèmes constituant un mot ou séquence de mots. Nous fournissons l'exemple d'un modèle acoustique HMM dans la figure 2.2.

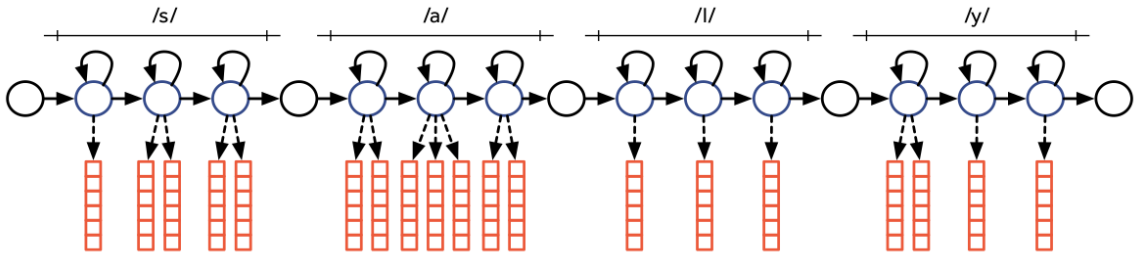

Figure 2.2 – Représentation d'un modèle acoustique exploitant des modèles de Markov cachés pour le mot *salut* [Vythelingum 2019].

Entraîner un modèle acoustique de ce type correspond à maximiser les probabilités d'observation de séquences dans un ensemble d'apprentissage. Il s'agit d'estimer les probabilités de transition d'un état au suivant de manière itérative, selon l'algorithme *Expectation-Maximization* (EM).

Enfin, un automate formé par concaténation d'HMM peut être parcouru selon différents chemins qui représentent un alignement possible entre le signal acoustique et une séquence de mots. Il s'agit de trouver le chemin le plus probable à l'aide de l'algorithme Viterbi [Forney 1973], donnant ainsi l'alignement optimal. C'est l'algorithme le plus couramment utilisé, bien qu'il en existe d'autres, comme la méthode itérative de Baulm-Welch [Baum et al. 1972].

Pour la tâche de reconnaissance de la parole, il est nécessaire d'estimer les probabilités d'observations acoustiques sur les états des HMM. Pendant des années, ce sont les GMM qui ont été une solution efficace pour cette estimation, puis, récemment les DNN les ont complétés. Nous détaillons ce processus dans les sections suivantes.

2.2.2 Modèles à mélange de gaussiennes

Le principe des GMM est d'associer à chaque état d'un HMM, une somme pondérée de densités de probabilités gaussiennes. Il s'agit ainsi d'exploiter la somme de plusieurs gaussiennes et d'estimer la variance et la moyenne de celles présentes dans le mélange. La probabilité d'observations des états est donnée par l'équation suivante :

$$b_j(x_i) = \sum_{k=1}^{K} w_{j,k} N(x_i, \mu_{j,k}, \Theta_{j,k}) \tag{2.4}$$

Avec, $N(x_i, \mu_{j,k}, \Theta_{j,k})$ une densité de probabilité gaussienne, $w_{j,k}$ les poids associés au mélange de gaussiennes et K le nombre total de gaussiennes par mélange. Le nombre optimal de gaussienne doit être déterminé empiriquement.

2.2.3 Modèles neuronaux profonds

L'exploitation de réseau de neurones comme alternative aux GMM a tout d'abord été proposée sous forme de MLP [Bourlard et Wellekens 1987], puis des réseaux plus complexes les ont remplacés [W. Ma et Van Compernolle 1990].

Toutefois, ce n'est que récemment que les DNN ont surpassé les GMM pour l'estimation des probabilités d'émission des états HMM [G. Hinton et al. 2012]. De plus, les réseaux de neurones à retardement (*Time-Delay Neural Network*, *TDNN*) sont particulièrement adaptés à la structure dynamique de la parole [Waibel et al. 1989]. Ils ont ainsi permis d'atteindre les performances à l'état de l'art avec une approche Markovienne [Peddinti et al. 2015].

Concrètement, l'objectif du système neuronal exploité est de modéliser les probabilités a posteriori des états des HMM.

Dans le cadre des HMM-DNN, l'estimation des paramètres du DNN s'effectue après l'optimisation d'un modèle HMM-GMM par remplacement de la partie GMM. Nous fournissons en figure 2.3, l'exemple d'une architecture HMM-DNN pour la reconnaissance de la parole.

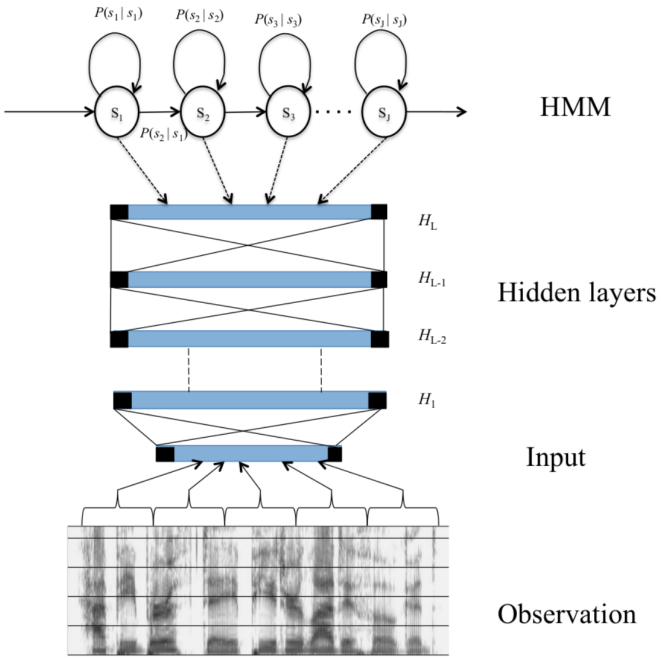

FIGURE 2.3 – Représentation d'un système HMM-DNN pour la modélisation acoustique de la parole [SAMSON JUAN 2015].

2.3 Modélisation du langage

La modélisation du langage a pour but d'estimer la probabilité $P(W)$, nécessaire à l'établissement d'un système de RAP. Il s'agit de la probabilité d'observer le mot, ou une séquence de mots W, dans la langue considérée pour le système. Le modèle de langage permet de s'assurer de la cohérence des prédictions du système puisqu'il permet d'évaluer les contraintes linguistiques. La modélisation de ces contraintes est faite selon la probabilité d'observer une séquence W de longueur k, qui s'exprime comme le produit des probabilités des mots $w_1, w_2 ... w_k$ sachant l'historique, tel que :

$$P(W) = P(w_1) \prod_{i=2}^{k} P(w_i | w_1 ... w_{i-1}) \tag{2.5}$$

2.3.1 Modèle n-grammes

L'intérêt des modèles de ce type réside dans leurs capacités à estimer cette probabilité avec un historique de $n-1$ mots. Ils possèdent toutefois des limites, puisque plus la taille de l'historique pris en compte est importante, plus il est couteux de réaliser une modélisation de ce type [S. F. CHEN et GOODMAN 1999] et plus ils font face à la problématique du manque de don-

nées. Il est rare d'observer plusieurs fois de longues séquences de mots dans un ensemble de données.

Pour estimer un modèle de langage, il est nécessaire de maximiser la vraisemblance sur un ensemble de données textuelles [Dempster et al. 1977]. Il s'agit de calculer la probabilité d'apparition d'un mot i selon l'historique h de la façon suivante :

$$P(w_i|h) = \frac{C(h, w_i)}{C(h)} \tag{2.6}$$

Avec h l'historique précédent w_i, soit $(w_{i-n+1}...w_{i-1})$ et C le nombre d'occurrences de la séquence de mots dans l'ensemble d'apprentissage.

Un manque vient du fait que les séquences de mots n'apparaissant pas dans l'ensemble d'apprentissage ne peuvent pas être modélisées et donc auront une probabilité nulle. Pour compenser ce manque, il est possible d'appliquer des méthodes de lissages, dont plusieurs sont décrites dans [S. F. Chen et Goodman 1999].

Ces techniques consistent généralement à changer la distribution de probabilité des n-grammes observés pour en attribuer aux n-grammes non observés. Parmi ces techniques, nous retrouvons notamment la méthode de repli (*back-off*) [Katz 1987].

Les modèles n-grammes peuvent être limitant, puisqu'ils deviennent rapidement couteux pour modéliser des dépendances longues distances. Les modèles de langue neuronaux, plus récents, sont moins sujets à cette limitation. Nous les décrivons dans la sous-section suivante.

2.3.2 Modèles neuronaux

Les modèles neuronaux sont une autre méthode pour modéliser le langage. Il s'agit de projeter les $n-1$ mots dans un espace continu. Ainsi, cette projection produit une représentation continue des mots (*word embeddings*). L'intérêt de cette projection est qu'elle rend possible l'exploitation de la notion de similarité entre les mots, permettant au modèle de généraliser plus facilement. Cela signifie que les modèles neuronaux ont une meilleure capacité à prendre en compte les séquences de mots n'apparaissant pas dans l'ensemble d'apprentissages (*Out Of Vocabulary, OOV*).

Les modèles de langue neuronaux ont tout d'abord été introduits par les réseaux de type perceptron multicouches (*MultiLayer Perceptron, MLP*) [Bengio, Ducharme et al. 2003 ; Schwenk 2007]. Ils ont ensuite pu bénéficier des apports des réseaux récurrents, qui facilitent la modélisation des dépendances longue distance [Tomáš Mikolov et al. 2011].

Au sein des réseaux récurrents, il est aussi possible de trouver des modèles de langage s'appuyant sur des couches bLSTM [Sundermeyer et al. 2012].

2.4 Approches neuronales de bout en bout

Ces dernières années, des modèles dits de bout en bout sont apparus [Graves, A.-r. Mohamed et al. 2013 ; Graves et Jaitly 2014 ; Hannun et al. 2014 ; Miao et al. 2015 ; Amodei et al. 2016]. L'intérêt de ce type d'approche réside dans la mise en place d'un unique modèle directement optimisé pour effectuer la transcription d'un segment de parole en une séquence de mots. Ils font partie des modèles séquence à séquence, qui effectuent la conversion d'une séquence d'une forme à une séquence d'une autre forme. Ici, de l'audio vers le texte.

Malgré leurs noms, ces systèmes ne sont pas *"tout-en-un"* puisque la majorité continue de nécessiter des données segmentées et, parfois, une extraction des paramètres acoustiques (*acoustic features*). Ils peuvent être considérés comme une forme alternative de modèle acoustique, si bien que certains travaux mettent en place des approches hybrides HMM-DNN (voir section 2.2.3) avec un système neuronal initialement de bout en bout [Yongqiang Wang et al. 2020].

Les modèles de bout en bout peuvent être complétés par des modèles de langues de deux manières. Soit externes au système par l'utilisation de l'algorithme de décodage Beam-Search et d'un modèle de langage pré appris [Amodei et al. 2016 ; Watanabe et al. 2018 ; Yiming Wang et al. 2019]. Soit interne, par l'exploitation de couches récurrentes dédiées à la modélisation du langage (RNN-LM) pouvant être appris conjointement au système [Hori, Watanabe et al. 2017 ; Hori, J. Cho et al. 2018].

Parmi les modèles de bout en bout, une première approche consiste à se passer de l'alignement a priori, entre un segment de parole et la séquence textuelle associée, grâce à l'utilisation de la fonction de coût CTC (*Connectionist Temporal Classification*) [Graves, Fernández et al. 2006].

Les premiers systèmes exploitant pleinement cette fonction, associée à des approches neuronales récurrentes, sont apparus en 2014 [Graves et Jaitly 2014]. Ils ont ensuite été enrichis de couches de convolution dédiées à la représentation de caractéristiques issue de la parole [Amodei et al. 2016]. Certains systèmes se sont même passés des couches récurrentes pour n'être composés que de couches convolutionnelles [Y. Zhang et al. 2016].

Une autre approche consiste à exploiter des architectures initialement utilisées en traduction automatique. Il s'agit des encodeurs-décodeurs bénéficiant des mécanismes d'attention [J. Chorowski et al. 2014 ; Chan et al. 2016]. Plus récemment encore, les transformers, dérivés des encodeurs-décodeurs, se sont montrés performants pour la tâche de reconnaissance de la parole [Dong et al. 2018 ; Moritz et al. 2020].

Dans les sous-sections suivantes, nous donnons des détails sur l'évolution de ces technologies dans le cadre de la reconnaissance de la parole.

2.4.1 Classification Temporelle Connectionniste

Cette fonction de coût permet à un système neuronal d'apprendre l'alignement entre une séquence d'entrée X et une séquence de sortie Y. Il s'agit de modéliser la probabilité d'observer la séquence Y sachant X : $P(Y|X)$. Un autre avantage de cette fonction concerne sa flexibilité, puisqu'elle n'impose pas d'avoir des séquences X et Y de même taille, ni de conserver un ratio.

Dans le cadre de la reconnaissance de la parole, elle peut permettre l'apprentissage d'un alignement entre un segment audio et la séquence représentant le texte associé. Nous décrivons, ici, le fonctionnement de la fonction CTC pour l'émission de caractères. Comme il s'agit de la plus petite unité composant l'écriture d'un mot, cela permet la prise en charge des mots hors vocabulaire (OOV). Toutefois, elle ne limite pas l'unité aux caractères, il est possible d'en utiliser d'autres, comme les phonèmes [Fernández et al. 2008].

Afin d'appliquer la fonction CTC, il est nécessaire de découper un segment de parole en trames fines. Une fenêtre de temps de 20ms est couramment utilisée.

Suite à ce découpage, l'extraction des paramètres acoustiques est à effectuer pour une trame considérée. Puis, ces paramètres sont propagés à travers un réseau de neurones qui sera responsable d'une distribution de probabilité sur l'ensemble du vocabulaire (les caractères) pouvant être émis. Grâce à la distribution de probabilité effectuée sur chacune des trames, il est possible de construire la séquence de caractères la plus probable associée à un segment audio.

Cependant, la parole est variable en terme de vitesse d'élocution. Ce qui signifie qu'une trame découpée arbitrairement ne correspond pas systématiquement à un caractère. Cette fonction de coût est complétée par une projection qui consiste à réduire les répétitions de caractères identiques successifs. La séquence émise [h h h e e l l l l o o o o] deviendrait [h e l o].

Il apparait que cette projection n'est pas suffisante en l'état puisqu'elle interdit l'émission de mots composés de lettres identiques successives. Pour pallier ce souci, il est nécessaire d'ajouter un symbole dans le vocabulaire (ϵ).

Ce symbole est exploité pour enrichir les séquences Y de manière à séparer deux lettres devant se suivre, par exemple [h e l ϵ l o]. La projection réalisée après construction de la séquence la plus probable s'effectuera ensuite en deux temps. Tout d'abord la réduction des caractères identiques successifs, puis la suppression des symboles ϵ. Un exemple serait : [h h h e e l l ϵ l o o o] $\rightarrow$ [h e l ϵ l o] $\rightarrow$ [h e l l o].

Nous donnons dans la figure 2.4 une représentation schématique des étapes composants la fonction CTC.

2.4.2 Algorithme de Beam Search

L'algorithme de *Beam Search*, ou algorithme de recherche en faisceau, vise à explorer un arbre de possibilités en n'explorant qu'un nombre limité (n) de branches à chaque noeud du

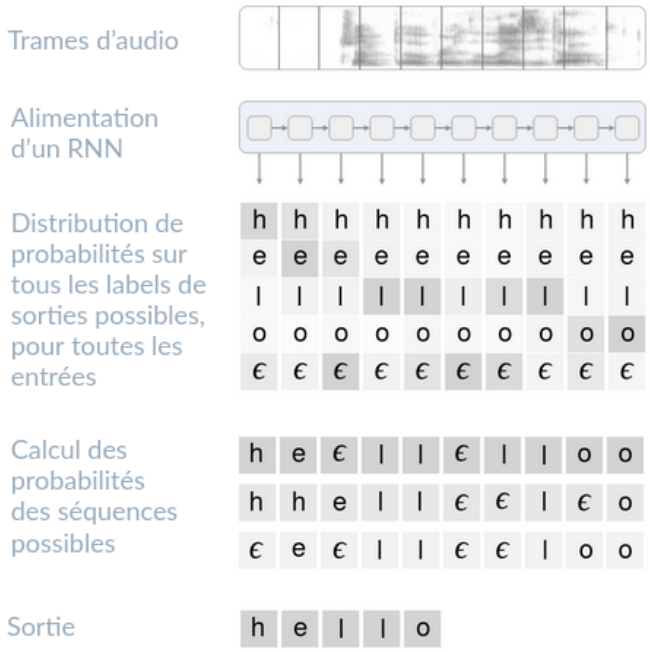

FIGURE 2.4 – Représentation du fonctionnement de la fonction de coût CTC.

graphe. Ce nombre n est aussi appelé largeur du faisceau. Avec une largeur infinie, cet algorithme est équivalent à un parcours en largeur d'un arbre. Lors du parcours de l'arbre, il s'agit de classer les successeurs du noeud courant et de sélectionner les n meilleurs suivant un score.

Cet algorithme est couramment exploité comme stratégie de recherche au sein des sorties de systèmes séquences à séquences. L'intérêt étant de construire un arbre des possibilités à l'aide des séquences de sorties immédiates d'un système et d'un modèle de langage. Le modèle de langage permet de définir le score des successeurs de chaque noeud.

Concrètement, l'emploi de cet algorithme, couplé à un modèle de langage, permet à un système d'émettre une sortie davantage vraisemblable par rapport au langage ciblé. Un exemple serait que, dans le cas d'un système de reconnaissance de la parole basé sur les caractères, cet algorithme est capable de corriger des erreurs orthographiques comme "banjour", à condition que le mot "bonjour" soit connu dans le modèle de langage utilisé.

2.4.3 Architecture encodeur-décodeur avec attention

La capacité de ce type d'architecture à modéliser des séquences et la présence des mécanismes d'attention les ont rendus prometteurs pour une tâche de reconnaissance de la parole. La première implémentation de ce type dédiée à la RAP a été publiée en fin d'année 2014 [J. CHOROWSKI et al. 2014].

Cette implémentation est très similaire à celle réalisée initialement pour la tâche de traduction automatique. Toutefois, ces travaux n'agissaient que comme preuve de concepts et n'ont pas surpassé une approche CTC pour la reconnaissance de phonèmes sur l'ensemble de données TIMIT. Plus tard, des travaux complémentaires ont permis cette fois de la surpasser légèrement [J. K. Chorowski et al. 2015].

Les travaux suivants ont permis de conforter la pertinence de ce type d'architecture, pour la RAP, en apportant des modifications améliorant leurs performances.

Certains travaux se sont concentrés sur la modification de l'architecture, par exemple un encodeur pyramidal [Chan et al. 2016].

D'autres ont modifié l'exploitation des mécanismes d'attentions, qui utilisent traditionnellement l'ensemble de la séquence d'entrée pour sélectionner les trames pertinentes. Il a été proposé de mettre en place une fenêtre d'observation limitée sur la séquence d'entrée et de regrouper l'information de trames voisines [Bahdanau, J. Chorowski et al. 2016].

Encore d'autres travaux ont permis de tirer bénéfice à la fois des mécanismes d'attention et de la fonction de coût CTC [Kim et al. 2017 ; Hori, Watanabe et al. 2017].

Enfin, ces architectures se sont positionnées comme état de l'art en reconnaissance de la parole en exploitant les mécanismes d'attention à plusieurs têtes [C.-C. Chiu et al. 2018].

Transformers

Les transformers sont une forme spécifique d'architecture encodeur-décodeur. Nous les avons décrits et avons détaillé leur fonctionnement global dans la section 1.6.4. Cette architecture neuronale a été proposée dans le cadre de la traduction automatique de la langue [Vaswani et al. 2017].

Ce n'est que très récemment que les transformers ont été appliqués à une tâche de reconnaissance de la parole [Dong et al. 2018].

Certains travaux ont permis de surpasser toutes les précédentes architectures de bout en bout pour la tâche de RAP [Pham et al. 2019]. Ils ont proposé d'augmenter considérablement la taille de l'architecture en utilisant jusqu'à 48 couches de type transformer pour l'encodeur et le décodeur, tandis que le papier originel exploite seulement 5 pour la tâche de traduction [Vaswani et al. 2017].

Encore plus récemment, des travaux ont rendu possible l'exploitation des transformers pour la reconnaissance de la parole en flux continue (*online speech recognition*) [Moritz et al. 2020]. Ils étaient jusqu'à présent limités par l'obligation de fournir une séquence d'entrée complète à l'architecture encodeur-décodeur.

Pour la reconnaissance de la parole, les transformers forment désormais un type d'architecture prometteur. Ils seront, à terme, susceptibles de totalement surpasser les encodeurs-décodeurs plus classiques.

2.5 Évaluation de la reconnaissance de la parole

La reconnaissance de la parole consiste à transcrire un signal acoustique sous forme de mots. Afin d'évaluer les systèmes mis en oeuvre, il apparait évident de comparer les mots fournis par le système aux mots effectivement prononcés. Il est donc nécessaire de comparer la réponse d'un système à une référence pour déterminer les erreurs qu'il produit. Il s'agit de la métrique du taux d'erreur sur les mots (*Word Error Rate*, *WER*).

Les erreurs prises en compte par cette métrique sont les erreurs de substitutions (S), d'insertions (I) et de suppressions (D). Les erreurs de substitution sont des mots incorrectement transcrits. Les erreurs d'insertions sont des mots ajoutés lors de la transcription et les erreurs de suppressions sont les mots omis.

L'équation de cette métrique s'exprime ainsi :

$$WER = \frac{S + D + I}{n} \tag{2.7}$$

.

Avec S, D, I respectivement le nombre d'erreurs de substitution, de suppression, d'insertion et n le nombre de mots dans la référence.

2.6 Choix technologiques pour cette thèse

L'objectif des travaux de cette thèse concerne la prise en charge de la tâche de compréhension du langage parlé, directement depuis la dimension acoustique. Ainsi, nous souhaitons mettre en place un système capable d'effectuer à la fois la tâche de reconnaissance de la parole et la tâche de compréhension du langage. Nous nous sommes donc naturellement tournés vers les technologies de reconnaissance de la parole, parmi lesquelles figuraient des approches de bout en bout.

Cette thèse a débuté en septembre 2017, alors que les technologies de RAP de bout en bout n'étaient que très récentes. Nous pouvions choisir entre une approche récurrente exploitant la fonction de coût CTC et une approche encodeur-décodeur. Notre choix s'est porté sur l'approche récurrente au travers des travaux de [Amodei et al. 2016], de par la mise à disposition d'une implémentation [1] de ce système et de la différence de performance jusqu'alors faible avec les systèmes encodeurs-décodeurs [Zenkel et al. 2017].

1. https ://github.com/SeanNaren/deepspeech.pytorch

2.7 Conclusion

Au sein de ce chapitre, nous avons évoqué les technologies employées pour effectuer la tâche de reconnaissance automatique de la parole. Nous avons ainsi vu que la parole peut être modélisée comme une séquence finie d'évènements non aléatoires. Bien qu'initialement dominées par les HMM-GMM, les technologies de RAP ont évolué vers des approches neuronales. Cette évolution s'est effectuée en plusieurs étapes avec tout d'abord l'exploitation d'approches hybrides conservant les modèles markovien (HMM-DNN). Par la suite, des approches neuronales utilisant un unique modèle sont apparues pour effectuer la tâche de RAP. Il s'agit des approches dites de bout en bout et qui peuvent principalement s'apparenter à de la modélisation acoustique. Les approches de ce type s'appuient régulièrement sur la fonction de coût CTC ou des architectures encodeurs-décodeurs. Puis elles ont évolué, notamment par des modifications des architectures neuronales, conduisant à l'utilisation des réseaux de type transformers. Les approches de bout en bout sont un élément de base pour les travaux de cette thèse qui consistent à effectuer la compréhension du langage directement depuis les signaux de parole. Dans le prochain chapitre, nous réaliserons l'état de l'art de cette tâche de compréhension.

Chapitre 3

Compréhension de la parole

Sommaire

La tâche visée dans le cadre de cette thèse concerne la compréhension de la parole. L'intérêt pour cette tâche est apparu il y a des décennies avec des travaux déjà basés sur l'intelligence artificielle. Il s'agissait de combiner une analyse syntaxique et une représentation sémantique [Klatt 1977].

Depuis, l'intérêt pour cette tâche est important dans la mesure où elle facilite, entre autres, les communications humain-machine en rendant possible la compréhension du langage naturel dans des cadres applicatifs spécifiques.

Les domaines applicatifs de la compréhension de la parole sont variés, nous pouvons notamment mentionner les objets connectés, comme les smartphones ou les assistants personnels. Nous pouvons également mentionner des services comme la réservation automatique par téléphone ou même le routage d'appel dans le cadre de centre d'appel. Les possibilités offertes par la prise en charge de la tâche de compréhension de la parole renforcent son intérêt pour un cadre industriel et commercial.

Au sein de ce chapitre, nous proposons une description de la tâche de compréhension de la parole. Nous proposons également de détailler les tâches relatives à la compréhension du langage parlé et notamment autour des entités nommées et des concepts sémantiques. Il s'agit en effet des représentations sémantiques que nous étudierons dans cette thèse. Par la suite, nous décrirons les systèmes couramment utilisés pour réaliser cette tâche, qu'ils correspondent aux approches d'apprentissage machine traditionnelles ou aux approches neuronales plus récentes. Nous évoquerons également les principales métriques d'évaluation relatives aux représentations sémantiques que nous exploitons. Enfin, nous proposons de détailler les impacts de la méthode jusque là employée pour effectuer la tâche de compréhension dans la parole. Ces impacts sont sources d'une motivation forte pour la complétion de l'objectif de cette thèse.

3.1 Compréhension du langage appliquée à la parole

Pour mieux délimiter les contours de cette thèse, il nous faut définir ce que nous entendons par compréhension de parole, nous proposons une définition dans la section suivante.

3.1.1 Définition

La compréhension de la parole peut être définie comme une tâche d'interprétation des signes véhiculés par un signal de parole. C'est une tâche complexe dans la mesure où le *sens* est mélangé au milieu d'autres informations comme l'identité du locuteur ou même son environnement [De Mori 2007].

En d'autres termes, la compréhension du langage parlé consiste en la projection des informations de parole d'une dimension acoustique vers une représentation sémantique. Cette

projection correspond à l'extraction du *sens*. Il est ainsi nécessaire de définir ce que nous entendons par le *sens* ou sémantique.

Une définition de la sémantique est donnée dans les travaux de [Woods 1975]. Selon l'auteur, la sémantique correspond aux relations entre les signes ou symboles, et ce qu'ils désignent ou signifient. En soi, cela correspond à l'organisation de la signification.

Dans le domaine informatique, la sémantique computationnelle consiste en la projection du sens d'un message en une représentation plus ou moins structurée. Cela se caractérise par une conceptualisation du monde, par l'utilisation de processus de calculs, dans le but de mettre en place une structure de représentation du sens. Cette structure est extraite à partir des signes disponibles (symboles) et de leurs caractéristiques présentes dans les mots et les phrases [De Mori et al. 2008].

De plus, il est très complexe d'effectuer la mise en place d'une représentation sémantique suffisamment générique pour réaliser la tâche de compréhension du langage ouvert. Aussi, la plupart des travaux de recherche en informatique traitant cette tâche exploitent une représentation sémantique prédéfinie pour correspondre à un usage particulier, c'est-à-dire de façon ad hoc. C'est également le cas des représentations sémantiques utilisées pour les travaux de cette thèse.

Il est possible d'effectuer une représentation sémantique plus ou moins complexe. Nous pouvons mentionner les représentations logiques à base de règles et de faits, les cadres sémantiques [Fillmore et al. 1976], les graphes sémantiques [Xie et Passonneau 2015], mais aussi les représentations structurées.

Néanmoins, dans ces travaux nous nous plaçons dans le cadre de la segmentation en séquence de mots support de concepts et l'étiquetage sémantique. Cadre qui peut être pris en charge par des méthodes d'apprentissage automatique supervisé.

Effectuer une projection directe de la dimension acoustique vers une représentation sémantique spécifique a, jusque là, été une tâche trop complexe pour être envisagée directement par un unique système bénéficiant des méthodes d'apprentissages automatiques supervisés. Afin de prendre en compte la modalité parole, il a longtemps été question d'effectuer la mise en place d'une chaîne de traitements successifs [Raymond et Riccardi 2007 ; Mesnil, X. He et al. 2013]. Nous apportons davantage de détails sur cette chaîne dans la sous-section suivante.

3.1.2 Chaîne de traitements successifs

L'intérêt de la mise en œuvre d'une chaîne de traitements successifs réside dans sa capacité à prendre en charge une tâche finale comme un ensemble successif de sous-tâches plus simples.

Dans le cadre de la compréhension de la parole, cela consiste à exploiter une représentation intermédiaire entre la dimension acoustique et la représentation sémantique possible. Cette représentation intermédiaire consite généralement en une représentation symbolique ob-

tenue grâce à un système de reconnaissance de la parole. Ainsi, la compréhension de la parole consiste en l'extraction du sens à partir de la représentation textuelle d'un discours prononcé à l'oral [Tur et De Mori 2011].

Concrètement, la tâche de reconnaissance automatique de la parole effectue la projection des informations de la dimension acoustique vers une représentation symbolique, qui correspond aux transcriptions automatiques de la parole. Puis, la tâche de compréhension du langage effectue une projection de la représentation textuelle vers la représentation sémantique prédéfinie. Cette projection est effectuée par une tâche de segmentation et d'étiquetage sémantique réalisée par analyse des caractéristiques textuelles.

Nous parlons donc d'une chaîne de traitements, constituée de deux composants optimisés chacun pour une tâche spécifique. Nous fournissons, dans la figure 3.1, une représentation schématique d'une chaîne de traitements et des modalités exploitées pour la compréhension de la parole.

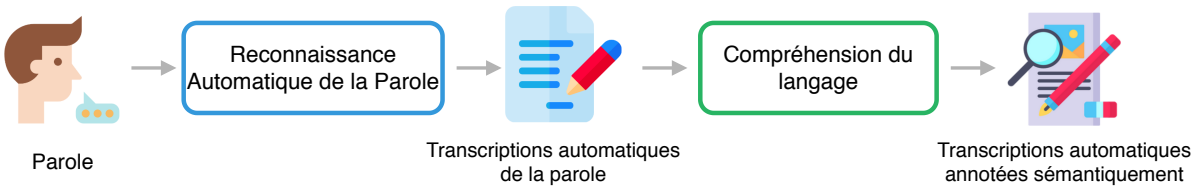

Figure 3.1 – Représentation d'une chaîne de traitements dédiés à la tâche de compréhension de la parole. L'annotation appliquée sur les transcriptions automatiques correspond à une tâche de segmentation et de classification sémantique.

Comme nous l'avons mentionné précédemment, la tâche de compréhension ciblée nécessite une définition ad hoc de la représentation sémantique qui lui est propre. Il existe ainsi plusieurs tâches de compréhension du langage, toutes définies en fonction de besoins applicatifs spécifiques. Nous donnons, dans les prochaines sous-sections, des détails concernant les tâches que nous allons cibler dans le cadre de cette thèse, mais aussi concernant les tâches majoritairement ciblées. Il ne s'agit toutefois pas d'une liste exhaustive.

3.1.3 Reconnaissance des entités nommées

L'origine de la tâche de reconnaissance des entités nommées provient des conférences MUC (*Message Understanding Conference*) initiées en 1987. L'objectif était de mettre en place des travaux dédiés à la compréhension automatique de documents [Grishman et Sundheim 1996]. La compréhension d'un document consiste à extraire des éléments informationnels pertinents qui jouent un rôle dans la description d'un événement ou d'un fait [Nouvel et al. 2015].

L'objectif de cette tâche a ainsi été défini comme une tâche d'extraction d'informations qui

se concrétise par le remplissage de champs prédéterminé d'un formulaire (*Slot Filling*), à partir de rapports textuels décrivant des événements.

La reconnaissance d'entités nommées consiste à extraire au sein d'un texte, les séquence de mots relatives aux champs du formulaire prédéfinis. Ainsi, une entité nommée est caractérisée par un type et une valeur.

La reconnaissance d'une entité consiste d'abord en la détection des séquences de mots support, puis en leur catégorisation. Cette catégorisation correspond à l'association de ces mots support à un champ du formulaire. Les entités nommées peuvent être définies comme les briques élémentaires de l'information présente dans les documents.

Au fil des années, la tâche initiale s'est enrichie par l'intermédiaire d'une gamme étendue de type et de sous-type d'entités nommées, mais aussi par sa complexification. Initialement attachées aux noms propres, les entités nommées ont ensuite été étendues à d'autres éléments comme des syntagmes nominaux.

En complément, l'imbrication des entités nommées a été proposée dans le cadre de travaux français [Galliano, Gravier et al. 2009]. Une entité nommée est imbriquée si elle est entièrement incluse dans une autre entité.

Enfin, d'autres travaux français ont proposé une définition étendue des entités nommées [Grouin et al. 2011]. Cette définition ajoute la nécessité de structurer et de décomposer les entités. Cela implique une première annotation selon les types entités nommées définis, puis une seconde consistant à typer les éléments composants les entités. Ces derniers éléments sont nommés *composants*.

Dans le cadre des travaux de cette thèse, nous prenons en compte cette définition étendue des entités nommées.

Pour davantage de détails sur la tâche de reconnaissance des entités nommées, nous encourageons le lecteur à consulter [Nouvel et al. 2015].

3.1.4 Extraction de concepts sémantiques

La tâche d'extraction des concepts sémantiques a pour objectif de prendre en charge la compréhension de la parole dans un cadre d'interaction humain-machine. Il s'agit plus particulièrement d'applications relatives à des dialogues transactionnels [Tur et De Mori 2011].

Pour des applications de ce type, il est nécessaire d'extraire dans l'énoncé d'un humain plusieurs types de concepts dépendant directement du domaine applicatif final du système. Il convient donc aussi d'exploiter une définition ad hoc de la sémantique, dédiée à ce cadre applicatif restreint.

Ainsi, comme pour la reconnaissance des entités nommées, cette tâche d'extraction peut être associée à une tâche de remplissage d'un formulaire [Hahn et al. 2010; Jabaian 2012; Mesnil, Dauphin et al. 2014].

La notion de concept sémantique peut être définie comme l'unité minimale de sens. Il s'agit donc d'un mot, ou groupe de mots rattaché à un type de concept inclus dans l'espace sémantique ciblé.

Les domaines applicatifs sont variés. Nous pouvons notamment citer les domaines de MEDIA [Bonneau-Maynard et al. 2005] et PORTMEDIA [Lefèvre et al. 2012], qui correspondent respectivement à des tâches de réservation d'hôtels et de réservation de tickets de théâtre. Il existe d'autres domaines, comme ATIS [Hemphill et al. 1990] qui concerne une tâche de réservation de vols, ou encore M2M [Shah et al. 2018] qui concerne la réservation de restaurants et de places de cinéma.

L'extraction des concepts sémantiques est une tâche très similaire à la reconnaissance des entités nommées, au niveau de sa prise en charge. La différence fondamentale entre ces deux tâches réside dans leurs objectifs. Les entités nommées s'attachent à la compréhension de documents, tandis que les concepts sémantiques sont exploités généralement dans un cadre applicatif spécifique d'interaction humain-machine.

Il convient d'exploiter des représentations sémantiques ad hoc différentes et propres à chacune des tâches. C'est pourquoi, au sein de cette thèse, nous portons notre étude également sur la tâche d'extraction des concepts sémantiques dans le cadre de MEDIA et PORTMEDIA.

3.1.5 Autres tâches de compréhension

Au sein de cette sous-section, nous effectuons une description brève d'autres tâches existantes de compréhension de la parole. Il s'agit cependant de tâches que nous ne ciblerons pas dans le cadre des travaux de cette thèse.

Détection d'intention

Pour comprendre la tâche de détection d'intention, il est tout d'abord nécessaire de définir *l'intention*. Celle-ci correspond à l'objectif d'une personne, qui peut être interprété à partir du discours qu'elle prononce [Tur et De Mori 2011].

Il s'agit d'une tâche relativement complexe dans la mesure où l'intention d'un utilisateur n'est pas nécessairement reliée à une unique formulation. Il s'agira donc de regrouper sous la même catégorie d'intention des phrases potentiellement très différentes.

L'intérêt applicatif de ce type de tâche vient de sa capacité à extraire automatiquement la requête d'une personne. Cela s'avère utile dans le cadre de centres d'appel téléphonique, notamment pour le routage d'appel [Paek et Horvitz 2004 ; Juang et Rabiner 2005].

La détection d'intention peut également être traitée comme une tâche de remplissage d'un formulaire [Xu et Sarikaya 2013 ; B. Liu et Lane 2016].

Résumé de documents

La tâche de résumé de documents peut être considérée comme la tâche visant à extraire le sens d'un document dans le but de le condenser pour produire un résumé. Il peut s'agir d'une reformulation du document d'origine [Murray et al. 2010] ou d'une sélection de segments représentatifs du sens du document [Maskey et Hirschberg 2008]. Dans les deux cas, il s'agira de détecter l'information pertinente en effectuant l'analyse de l'importance des éléments du document.

Il est également possible que cette tâche soit prise en charge à l'échelle de plusieurs documents [Maskey et Hirschberg 2008]. L'objectif étant d'extraire, de l'ensemble des documents, des informations répondant à une requête formulée par l'utilisateur du système.

L'intérêt pour cette tâche réside dans sa capacité à réduire l'effort humain pour l'indexation d'information documentaire ou son accès [Tur et De Mori 2011].

Segmentation thématique

Cette tâche de compréhension du langage s'applique au niveau des documents. Il s'agit d'effectuer une segmentation d'un document complet en sous-parties regroupées par thématiques [Tur et De Mori 2011]. Les thématiques ciblées ne sont pas nécessairement connue à priori. Il peut s'agir d'un regroupement thématique non prédéfinis, et donc appris de façon non supervisée [Seymore et Rosenfeld 1997].

L'intérêt de cette tâche vient, comme pour le résumé de documents, de sa capacité à rendre possibles l'indexation automatique de documents et l'accès à l'information.

En outre, il s'agit d'une tâche très dépendante de l'application ciblée puisque la segmentation ne peut pas être abordée de la même manière s'il s'agit de segmentation de journaux télévisés ou de débat. Dans le premier cas, il est assez simple d'effectuer une segmentation en sujet individuel [Guinaudeau 2011 ; Bouchekif 2016], tandis que dans le second cas il s'agit souvent de documents mono thématiques qui sont ainsi abordés comme une segmentation en activités ou en intentions [Passonneau et Litman 1997].

3.2 Approches historiques d'étiquetage

Comme nous l'avons mentionné, la compréhension de la parole peut être réduite à une tâche de segmentation et d'étiquetage pouvant être résolue par l'intermédiaire de méthodes d'apprentissages supervisés. Nous proposons donc, au sein de cette section, une description des principales approches historiques d'étiquetage. Par approches historiques, nous entendons les approches non neuronales.

Nous pouvons distinguer trois types d'approches principales, correspondant aux automates

à états finis, aux machines à vecteurs de support et aux champs aléatoires conditionnels. Nous effectuons une description de ces approches dans les sous-sections suivantes.

3.2.1 Automates à états finis

Dans le cadre de la compréhension de la parole, les automates à états finis (*Finite State Machine*, *FSM*) correspondent à des modèles mathématiques visant à représenter une information linguistique. Cette information linguistique est nécessairement extraite d'une grammaire issue d'une représentation finie du langage naturel.

Grammaires

Une grammaire consiste en la représentation d'un langage par l'intermédiaire d'un nombre fini de règles. Il est possible d'envisager le langage naturel comme fini et donc pouvant être décrit par un nombre fini de règles. Il peut ainsi être représenté par une grammaire dite formelle. Cette grammaire permet de définir une syntaxe et donc l'ensemble des mots admissibles par un langage [Chomsky et Lightfoot 2002].

Une définition des grammaires formelles est donnée par :

$$G = (S_T, S_N, R, A) \tag{3.1}$$

Avec, S_T l'ensemble des symboles terminaux, S_N l'ensemble des symboles non terminaux, R l'ensemble des règles de la grammaire définissant le passage d'un symbole à un autre et A le symbole initial (axiome) inclus dans S_N.

À cette définition s'ajoute une définition hiérarchique qui vise à séparer les grammaires formelles en plusieurs types s'incluant entre elles. On distingue quatre types différents de grammaires.

Tout d'abord les grammaires régulières, qui sont incluses dans les grammaires hors contexte. Ces dernières sont incluses dans les grammaires contextuelles, elles-mêmes incluses dans les grammaires générales [Chomsky et Lightfoot 2002].

En complément, les grammaires probabilistes sont apparues dans le but de retirer des ambiguïtés du langage représenté. Il s'agit de différencier les interprétations diverses d'un même mot ou d'une même expression. Le principe de ce type de grammaires repose sur la présence d'un poids associé à chaque règle la composant.

Représentation d'une grammaire

L'objectif des FSM est donc de représenter le langage correspondant à une grammaire régulière par une suite d'états finis et de transitions. Ce qui permet de déterminer l'appartenance d'une chaîne de symboles au langage représenté [Raymond 2005].

Un FSM est capable de reconnaitre une chaîne de symbols, uniquement s' il peut lire tous les symboles de la chaîne en allant de son état initial, vers son état final. Pour cela, il doit pouvoir effectuer des transitions entre chaque symbole de la chaîne. Cela implique nécessairement des systèmes en vocabulaire fermé, mais aussi qu'un automate est exploité pour la représentation de chaque règle de la grammaire.

En complément, les transducteurs à état finis sont des FSM effectuant le lien entre deux ensembles de symboles. Leur objectif est donc d'effectuer le lien entre la dimension textuelle et la dimension sémantique. Pour cela, ils réalisent, à partir du langage reconnu, une analyse grammaticale permettant la reconnaissance de formes prédéfinies correspondantes à la représentation sémantique définie.

Un exemple de transducteur à états finis est donné dans la figure 3.2.

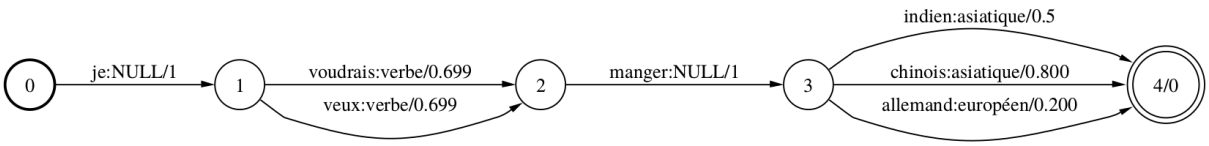

Figure 3.2 – Exemple de transducteurs à états finis extrait de [Raymond 2005].

3.2.2 Machines à vecteurs de support

Les machines à vecteurs de support (*Support Vector Machine*, *SVM*) correspondent à des séparateurs linéaires dont l'objectif est de trouver les hyperplans réalisant la séparation de groupes d'échantillons [Vapnik 2006]. Il s'agit de trouver les hyperplans possédant une marge maximale entre deux groupes.

La notion de marge correspond à la distance entre la frontière de séparation (hyperplan) et les échantillons les plus proches. Ces échantillons sont appelés vecteurs supports. La marge maximale consiste donc à trouver l'hyperplan le plus éloigné des vecteurs supports.

Ce type d'approche peut être appliquée uniquement dans le cas de problèmes linéairement séparables. Pour les exploiter sur des problèmes non linéairement séparables, il est nécessaire d'appliquer une transformation pour rendre les données exploitées linéairement séparables. Cette transformation est réalisée à l'aide d'un noyau, qui correspond à la fonction employée pour la transformation. Il peut par exemple, être linéaire, gaussien, polynomial ou laplacien.

Suite à la transformation, il suffit d'appliquer les SVM dans leur fonctionnement nominal, puisque le problème sera désormais rendu linéairement séparable.

Leur application dans le cadre de la compréhension du langage consiste à considérer le problème comme une suite de classification linéairement séparable. Pour ce faire, une classification est réalisée par élément dans la séquence [Joachims 1998 ; Hahn et al. 2010].

Afin de représenter le langage en entrée des systèmes, il est possible d'exploiter la méthode du sac de mots (*bag of words*), c'est notamment le cas dans [K. Zhang et al. 2006]. Cette méhode consiste à attribuer un index unique à chaque mot du langage afin de créer un dictionnaire. Par la suite, le segment de texte considéré sera représenté par un vecteur formé par remplacement des mots par leur index dans le dictionnaire.

3.2.3 Champs aléatoires conditionnels

Les champs aléatoires conditionnels (*Conditional Random Fields*, *CRF*) font partie des modèles conditionnels discriminants [Lafferty et al. 2001].

Contrairement aux modèles génératifs, les CRF ne modélisent que la distribution conditionnelle $P(Y|X)$, et donc ils ne modélisent pas explicitement $P(X)$. De plus, avec un CRF les étiquettes Y sont toutes conditionnées sur l'ensemble de l'observation X.

Un CRF est une forme spécifique de champs aléatoires Markovien. Nous proposons de décrire tout d'abord les champs aléatoires Markovien, avant de nous attarder sur les CRF.

Champs aléatoires Markovien

Il s'agit de modèles graphiques qui exploitent un graphe non orienté pour représenter les dépendances entre des variables aléatoires. La structure de ce graphe permet de définir les dépendances dans la mesure où un nœud correspond à une variable aléatoire et une arête représente une dépendance entre les variables qu'elle relie. Ainsi, ce type de modèle est défini par $G = (V, E)$, avec V les sommets (*Vertice*) et E les arrêtes (*Edge*).

En fonction de sa structure, le graphe représentant le modèle peut être factorisé en groupes distincts étant chacun régi par une fonction (ϕ) dont la portée se limite au sous-ensemble de variables de son groupe. Aussi, au sein d'un même groupe, toutes les variables doivent être connectées les unes aux autres.

Nous donnons une représentation d'un champ aléatoire Markovien à quatre variables dans la figure 3.3.

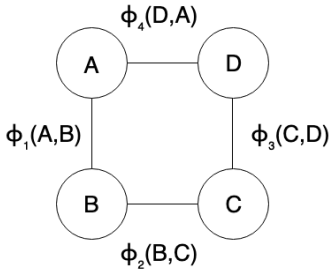

Figure 3.3 – Représentation d'un champ aléatoire Markovien à quatre variables. Représentation issue de *l'article de A. Prasad, 2019*.

Spécificités des CRF

L'objectif d'un CRF est de calculer la probabilité d'observer une séquence d'étiquettes ($Y = y_1, y_2 ... y_n$) à partir d'une séquence d'observations ($X = x_1, x_2 ... x_n$), qui s'exprime par $P(Y|X)$. Dans le cas de la compréhension du langage, il s'agit de maximiser cette probabilité pour Y, Y étant une séquence d'étiquettes sémantiques, et X une séquence de mots.

Les champs aléatoires conditionnels correspondent à un cas particulier de champs aléatoires Markovien. Lorsqu'un graphe est conditionné sur X, c'est un CRF si l'ensemble des variables aléatoires dans $Y = (Y_v)_{v \in V}$ suivent la propriété de Markov définis par :

$$P(Y_v|X, Y_w, w \neq v) = P(Y_v|X, Y_w, w \sim v) \tag{3.2}$$

Avec, $w \sim v$ signifiant que v et w sont voisins dans le graphe G. Nous donnons une représentation de la structure d'un CRF dans la figure 3.4.

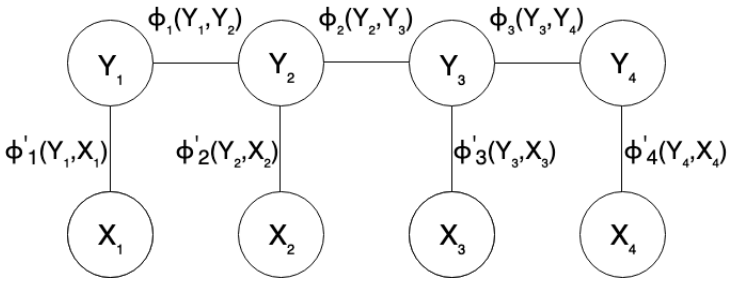

FIGURE 3.4 – Représentation de la structure d'un champ aléatoire conditionnel. Représentation issue de *l'article de A. Prasad, 2019*.

Un CRF est régi par la formule générale issue de la théorie fondamentale des champs aléatoires [HAMMERSLEY et CLIFFORD 1971]. Cette formule s'exprime par :

$$P_\theta(Y|X) \propto exp(\sum_{e \in E,k} \lambda_k t_k(e, Y|_e, X) + \sum_{v \in V,k} \mu_k s_k(v, Y|_v, X)) \tag{3.3}$$

Avec, $\lambda_k et \mu_k$ des vecteurs de poids et $t_k(e, Y|_e, X) et s_k(v, Y|_v, X)$ des vecteurs de caractéristiques supposés donnés et fixes.

L'apprentissage d'un CRF consiste donc à estimer l'ensemble des poids $\theta = (\lambda_1, \lambda_2 ... \lambda_k, \mu_1, \mu_2 ... \mu_k)$ qui maximisent le log-vraisemblance des données d'apprentissage, selon :

$$L(\theta) = \sum_{i=1}^{n} log P_\theta(y_i|x_i) \tag{3.4}$$

Il existe plusieurs méthodes de résolution proposées pour l'apprentissage des CRF, comme l'algorithme *Viterbi*, ou même *Improved Iterative Scaling*. Une comparaison de ces méthodes est

donnée dans [Malouf 2002].

Il a été montré que les CRF surpassent l'ensemble des approches historiques pour la tâche d'étiquetage en concepts sémantiques [Hahn et al. 2010]. Ces travaux ont notamment inclus une comparaison avec les machines à vecteurs de supports et les automates à états finis. Ainsi, pour les travaux de cette thèse, nous ne considérons que les CRF comme approche historique.

En complément de ces approches historiques d'apprentissage machine, des approches plus récentes ont bénéficié de l'apport des réseaux de neurones. Nous proposons dans la section suivante de réaliser une description de ces nouvelles approches neuronales.

3.3 Approches neuronales

L'essor des approches neuronales, pour la plupart des tâches d'apprentissage automatique supervisé, a conduit à leur étude dans le cadre de la compréhension parole. Nous proposons dans cette section de décrire les principales approches neuronales employées pour cette tâche.

Au début de cette thèse, ces approches été appliquées sur une représentation textuelle des mots, c'est-à-dire une transcription automatique. Dans un premier temps, nous proposons une description des méthodes employées pour représenter les mots. Ensuite, nous aborderons l'utilisation d'architecture neurales récurrentes pour le traitement des transcriptions automatique. Celles-ci peuvent être combinées à des champs aléatoires conditionnelles, voir utilisées dans des approches encodeurs-décodeurs avec mécanismes d'attention.

3.3.1 Représentation vectorielle des mots

Afin d'alimenter des modèles neuronnaux pour réaliser la tâche d'étiquetage sémantique, il est nécessaire de représenter les mots sous la forme de vecteur.

Représentation one-hot

La représentation *one-hot* correspond à la représentation vectorielle la plus simple. Elle consiste en l'utilisation d'un vecteur à n dimension dont n est la taille du vocabulaire. Chaque élément de ce vecteur est relié à un mot du vocabulaire. Ainsi, un mot est encodé par un vecteur de 0 et d'un 1 placé à l'indice lui correspondant.

L'inconvénient principal de ce type de représentation est qu'elle ne permet pas de rapprocher les mots en fonction de l'information grammaticale ou sémantique qu'elle véhicule.

De plus, elle est directement dépendante de la taille du vocabulaire représenté et peut donc s'avérer très volumineuse dans un cadre applicatif.

Une autre méthode de représentation a été proposée pour mieux correspondre aux besoins des réseaux de neurones. Il s'agit d'une représentation distribuée dans laquelle plusieurs di-

mension vont porter l'information de mot. Les plus connues sont les représentations dites de plongement de mot.

Représentation par plongements de mots

Les plongements de mots (*word embeddings*) ont été introduits par les travaux de [Bengio, Ducharme et al. 2003]. Ces travaux consistaient en la construction de modèles de langues neuronaux et ont par la suite été enrichis par d'autres travaux, comme [Schwenk et al. 2006].

Ce type de représentation est largement utilisé en association des réseaux de neurones, en raison de sa plus grande richesse informationnelle. Un système neuronal peut exploiter, via un dictionnaire, des plongements de mots appris en amont. Il est également possible de les apprendre pendant la phase d'entraînement du système, suite à une initialisation aléatoire.

Le principe des plongements est de représenter un mot par un ensemble de valeurs réelles, denses et de faible dimension, correspondant à un vecteur. Chaque dimension de ce vecteur correspond à une caractéristique latente du mot, qui peut ainsi représenter des informations syntaxiques et sémantiques [Ghannay 2017].

Depuis leur introduction, les plongements de mots ont largement été étudiés pour permettre des représentations plus efficaces. On distingue ainsi plusieurs méthodes de construction des plongements, comme les modèles de langue neuronaux [Bengio, Ducharme et al. 2003 ; Schwenk et al. 2006], ou les plongements de type *Global Vectors - GloVe* [Pennington et al. 2014]. D'autres méthodes ont été développées, comme *word2vec* (*CBOW* ou *Skip-Gram*) [Tomas Mikolov, K. Chen et al. 2013 ; Tomas Mikolov, Sutskever et al. 2013].

Enfin, les plongements de mots ont été employés avec succès dans le cadre de la compréhension de la parole [Mesnil, X. He et al. 2013 ; Yao, Peng et al. 2014 ; B. Liu et Lane 2016].

3.3.2 Réseaux de neurones récurrents

Les réseaux récurrentes correspondent aux premières approches neurales employées pour la compréhension du langage écrit [Yao, Zweig et al. 2013], ou oral [Mesnil, X. He et al. 2013]. L'intérêt de ce type de réseaux est lié à sa capacité à modéliser des séquences, ainsi que des dépendances éloignées. Ces modélisations permettent ainsi la production d'une séquence d'étiquettes Y sachant une séquence de mots X.

Plusieurs travaux ont ainsi proposé d'exploiter ces approches [Mesnil, X. He et al. 2013 ; Mesnil, Dauphin et al. 2014 ; Xu et Sarikaya 2014 ; Shi et al. 2015].

Dans [Mesnil, X. He et al. 2013], il est question d'utiliser des réseaux hybrides combinant la récurrence de Elman et de Jordan. D'un côté, la récurrence de type Elman consiste à extraire l'information d'une couche cachée d'une étape précédente et à la réinjecter au niveau de cette couche cachée à l'étape courante [Elman 1990]. D'un autre côté, la récurrence de type Jordan correspond à la réinjection de l'information issue de la couche de sortie [Jordan 1997].

Dans le cadre des travaux de [Mesnil, Dauphin et al. 2014], il est cette fois question d'évaluer les performances des réseaux récurrents en fonction de la façon dont la séquence d'entrée est exploitée. Cela correspond à comparer l'exploitation d'une séquence vers le passé ou vers le futur. Ces travaux concluent qu'il est préférable d'exploiter un réseau récurrent bidirectionnel, qui permettra de modélisation la séquence en bénéficiant du contexte dans les deux sens.

Des travaux plus récents ont proposé l'exploitation d'un troisième type de récurrence apportant un gain par rapport aux récurrences de Elman et de Jordan [Dinarelli et Tellier 2016]. Cette nouvelle récurrence consiste à injecter les informations de sorties non plus au niveau des couches cachées, mais dès les couches d'entrée du réseau.

En parallèle, des travaux ont proposé d'exploiter des réseaux de type CNN pour la compréhension du langage [Xu et Sarikaya 2013]. Il a aussi été proposé des couches de type LSTM [Yao, Peng et al. 2014], avant d'exploiter le contexte dans les deux sens (bLSTM) [Hakkani-Tür, Tür et al. 2016].

3.3.3 Combinaison aux champs aléatoires conditionnels

Plus récemment encore, il a été proposé de bénéficier des champs aléatoires conditionnels comme complément des approches neuronales. Il est ainsi question d'effectuer une approche hybride mêlant une représentation neuronale et un CRF [Mesnil, Dauphin et al. 2014 ; X. Ma et Hovy 2016 ; Kadari et al. 2018].

Le principe de ces approches hybrides est d'apprendre tout d'abord une représentation continue de la séquence de mots, le plus couramment avec des réseaux de types LSTM / bLSTM. Puis une approche CRF est appliquée pour effectuer l'étiquetage sémantique.

Dans [X. Ma et Hovy 2016], il est également question d'exploiter des couches CNN en amont des couches récurrentes bLSTM. Ces CNN permettent d'enrichir la représentation neuronale obtenue par l'ajout d'informations morphologiques extraites automatiquement au niveau caractère.

Au début de cette thèse, l'approche combinant des couches de types CNN, bLSTM, et CRF correspondait à l'état de l'art pour l'étiquetage neuronal de séquences. Une implémentation [1] de ces travaux a été rendu disponible par ses auteurs. Nos travaux se sont grandement appuyés sur cette implémentation comme référence.

3.3.4 Exploitation des mécanismes d'attention

D'autres travaux se sont tournés vers l'exploitation des approches neuronales de type encodeurs/décodeurs. Ce type d'architecture est régulièrement exploitée avec des mécanismes d'attention.

1. https ://github.com/XuezheMax/NeuroNLP2

Dans le cadre de la compréhension de la parole, les premiers travaux utilisant ce type de système ont été proposés par [Simonnet, Camelin et al. 2015].

Concrètement, ces travaux exploitent une implémentation encodeur-décodeur similaire aux approches exploitées à cette époque pour la tâche de traduction automatique [Bahdanau, K. Cho et al. 2015]. Ils effectuent la comparaison d'une approche biRNN simple et d'une approche biRNN avec attention. Jusque là, l'approche biRNN avait montré son potentiel sur une tâche de compréhension comme ATIS [Mesnil, X. He et al. 2013], mais aussi ses limites sur une tâche plus complexe telle que MEDIA. En effet, sur MEDIA, les biRNN n'étaient pas en mesure de surpasser une approche plus traditionnelle exploitant des CRF [Vukotić et al. 2015]. Les travaux de [Simonnet, Camelin et al. 2015] ont en partie permis, via une première exploitation des mécanismes d'attention, de répondre à cette difficulté.

Ces mécanismes d'attention ont par la suite été davantage utilisés, par exemple dans [B. Liu et Lane 2016 ; Zhu et K. Yu 2017 ; Serdyuk et al. 2018].

Les travaux de [Yufan Wang et al. 2018] ajoutent des couches CNN au sein de l'encodeur.

Encore plus récemment, les architectures neuronales de types Transformer ont été appliquées avec succès pour cette tâche. Dans les travaux de [L. Zhang et H. Wang 2019], il est question d'une approche hybride exploitant la représentation fournie par un encodeur Transformer pour alimenter un CRF.

3.4 Évaluation des performances d'un système de compréhension du langage

L'évaluation des performances des systèmes de compréhension du langage peut s'effectuer à l'aide de différentes métriques. Ces métriques sont le plus souvent dérivées d'un taux d'erreurs et répondent aux besoins d'une représentation sémantique. Nous décrivons ici, les métriques principalement utilisées pour l'extraction d'information. Certaines sont spécifiques aux entités nommées ou aux concepts sémantiques, qui correspondent aux représentations sémantiques de nos travaux.

3.4.1 Précision, Rappel et F-Mesure

La métrique de la F-mesure [Van Rijsbergen 1974] est composée de deux sous-éléments apportant chacun des informations complémentaires.

Tout d'abord, la précision, qui dans notre cas consiste à calculer la quantité de concepts correctement reconnus parmi ceux émis par notre système. La précision correspond au ratio

de réussite en cas d'émission d'un concept et elle est exprimée ainsi :

$$P = \frac{nC}{n} \tag{3.5}$$

Avec nC le nombre de concepts correctement émis et n le nombre total de concepts émis.

Ensuite, le rappel consiste à calculer la couverture des concepts correctement reconnus. Il s'agit d'une indication de la quantité de concepts couverts par le système et il est exprimé ainsi :

$$R = \frac{nC}{nR} \tag{3.6}$$

Avec nC le nombre de concepts corrects et nR le nombre de concepts dans la référence.

La précision et le rappel sont deux indications qui indépendamment ne peuvent être suffisantes. Un système peut tout à fait obtenir un score de précision de 0,9 avec seulement un rappel de 0,1, ou inversement. Dans la configuration mentionnée, très peu de concepts seraient couverts par le système, même s'il était très performant pour correctement les classifier.

L'intérêt de la F-mesure est d'effectuer une combinaison (moyenne harmonique) de la précision et du rappel pour proposer une mesure unique des performances d'un système.

Elle s'exprime ainsi :

$$F = 2 * \frac{P * R}{P + R} \tag{3.7}$$

Il s'agit d'une métrique couramment utilisée dans le cadre de tâches d'extraction d'information. Toutefois, en fonction des représentations sémantiques ciblées, des métriques spécifiques peuvent être employées. C'est le cas notamment pour les entités nommées, ainsi que les concepts sémantiques. Dans les sous-sections suivantes, nous détaillons les métriques associées à ces représentations.

3.4.2 Évaluation des entités nommées

Pour l'évaluation des entités nommées, nous distinguons deux métriques, le *Slot Error Rate* (SER) [Makhoul et al. 1999] et l'*Entity Tree Error Rate* (ETER) [Jannet et al. 2014].

La première métrique mentionnée (SER) s'apparente à un taux d'erreur qui nécessiterait une représentation sémantique à plat en raison de la définition d'un slot. La notion de slot peut se définir comme un regroupement d'un ou plusieurs mots caractérisés par des frontières de début et de fin, ainsi qu'un type d'entité nommée.

La seconde métrique (ETER) est quant à elle fondée sur le SER, mais en permettant la prise en compte d'une définition arborescente des entités nommées. Nous donnons ci-dessous davantage de détails concernant ces deux métriques.

Slot Error Rate

Le principe du SER est très similaire au WER puisqu'il consiste en un calcul de taux d'erreur. Il s'agit donc de prendre en compte des erreurs d'insertion, de substitution et de suppression.

Cependant, avec cette métrique on distingue trois types d'erreurs de substitution. Tout d'abord, les substitutions de frontières des slots (S_f), puis les substitutions de type d'entités nommées (S_t) et enfin, les erreurs de frontières et de types ensemble (S_{ft}).

De plus, cette métrique préconise d'affecter des coefficients à chaque catégorie d'erreurs. Ce principe permet de la rendre plus modulable en tenant compte du poids relatif des erreurs selon leurs catégories.

Son équation est la suivante :

$$SER = \frac{\alpha_1 S_t + \alpha_2 S_f + \alpha_3 S_{ft} + \beta D + \gamma I}{n} \tag{3.8}$$

Avec, S_t, S_f, S_{ft}, respectivement le nombre d'erreurs de substitution de type, de substitution de frontières, de substitution de frontières et de type. D, correspond aux erreurs de suppression et I d'insertion. Enfin, n représente le nombre d'entités de référence.

α_1, α_2, α_3, β et γ sont les coefficients affectés à chaque catégorie d'erreur. Au sein de cette thèse, les coefficients α_1 et α_2 sont définis à 0,5 et α_3, β et γ sont définis à 1.

Entity-Tree Error Rate

En soi, l'évaluation d'un système repose nécessairement sur un alignement entre l'hypothèse qu'il produit et une référence manuelle. La différence principale entre la métrique du SER et la métrique ETER se situe au niveau des alignements réalisés.

Pour le calcul du SER, il s'agit d'aligner les hypothèses et les références slot à slot. Cependant, un slot d'entité nommée ne peut représenter l'entièreté d'une entité nommée structurée. Pour ce type d'EN, il est nécessaire d'exploiter un *arbre-entité* représentant toute la structure arborescente de l'imbrication des entités. Un slot ne représente donc qu'un nœud de l'arbre-entité global. Évaluer une structure arborescente avec la métrique du SER, revient à simplifier le problème en évaluant chaque élément de cette structure indépendamment.

C'est pour répondre à cet inconvénient que la métrique ETER a été proposée [Jannet et al. 2014]. Elle consiste à réaliser un alignement entre les arbres d'hypothèses et les arbres de références pour prendre en compte tous les slots référant à un même arbre-entité.

Cette métrique se définit par l'équation suivante :

$$ETER = \frac{I + D + \sum_{(e_r, e_h)} E_{(r,h)}}{n} \tag{3.9}$$

Avec I le nombre d'insertions d'arbre-entité, D le nombre d'omissions d'arbre-entité, (e_r, e_h) les paires d'arbres-entités de référence et d'hypothèse associés suite à l'alignement, $E_{(r,h)}$ l'erreur calculée pour chaque paire et n le nombre d'arbres-entité de référence.

En complément, l'erreur pour chaque paire est donnée par l'équation :

$$E_{r,h} = (1-\alpha)E_T(e_r, e_h) + \alpha E_C(e_r, e_h) \tag{3.10}$$

Avec $E_T(e_r, e_h)$ l'erreur de détection et de classification des entités, $E_C(e_r, e_h)$ l'erreur de décomposition et α, compris entre 0 et 1, fixant le poids relatif de E_T par rapport à E_C.

Cette métrique est particulièrement adaptée aux tâches de reconnaissance des entités nommées structurée telles que QUÆRO. Nous détaillerons cette tâche plus en avant dans ce manuscrit (section 4.2.1).

Bien qu'adaptée à la tâche de reconnaissance d'entités nommées que nos travaux ciblerons (voir chapitre 5), nous n'utiliserons pas cette métrique au sein de cette thèse.

En effet, dans le but de comparer nos résultats à des travaux antérieur, nous conserverons la métrique du SER pour les entités nommées. Les travaux auxquels nous faisons mention seront l'objet du premier chapitre de contribution de ce manuscrit.

3.4.3 Évaluation des concepts sémantiques

Traditionnellement, les concepts sémantiques sont également évalués à l'aide d'un taux d'erreur. Cela correspond au taux d'erreurs sur les concepts (CER), ainsi qu'au taux d'erreurs sur les concepts et leurs valeurs (CVER), c'est-à-dire les supports de mots associés aux concepts. Il s'agit de métriques très proches du taux d'erreurs sur les mots décrits dans la section 2.5.

Le calcul du taux d'erreurs est identique, à savoir :

$$CER/CVER = \frac{S + D + I}{n} \tag{3.11}$$

Avec S, D, I respectivement le nombre d'erreurs de substitution, de suppression, d'insertion et n le nombre de références.

Pour le CER, il s'agit d'appliquer ce calcul uniquement aux concepts et donc d'évaluer la capacité d'un système à effectuer une classification correcte.

Pour le CVER, ce calcul est cette fois appliqué aux couples concepts/valeurs et un couple est correct, uniquement si l'ensemble de ce couple est correct. Cela signifie que cette métrique permet l'évaluation de la classification de concepts, mais aussi de leurs segmentations. C'est-à-dire le placement des frontières délimitant les supports de mots.

Ces deux métriques sont appliquées dans le cadre de tâche d'extraction des concepts sémantiques dans la parole. Cette modalité parole fait intervenir une contrainte supplémentaire au CVER. En effet, elle impose qu'un couple concept/valeur soit correct si la transcription

automatique des supports de mots est correcte également.

En soi, l'exploitation de la modalité parole apporte d'autres particularités que nous détaillerons au sein de la section suivante.

3.5 Impact des transcriptions automatiques

La tâche de compréhension de la parole est une tâche relativement complexe. Au commencement de cette thèse, la prise en compte de la modalité parole consistait en la mise en œuvre de différents composants. Chacun de ces composants est nécessairement optimisé pour une tâche précise. Ces optimisations sont réalisées par l'apprentissage d'un système spécifique avec des données manuellement annotées.

Concrètement, un composant parole sera optimisé avec des audios et leurs transcriptions manuelles, tandis qu'un composant de compréhension du langage exploitera des transcriptions manuelles enrichies d'une annotation sémantique manuelle.

Ce principe à l'avantage de focaliser l'apprentissage du composant de compréhension sur une tâche non bruitée. Il s'optimisera donc uniquement sur les erreurs de compréhension du langage.

Cependant, cet avantage se transforme en inconvénient lorsqu'il s'agit d'exploiter des transcriptions automatiques, puisqu'elles impliquent un déphasage entre l'apprentissage du composant et son exploitation nominale.

Ce déphasage correspond à l'un des impacts importants des transcriptions automatiques, qui agiront comme une entrée bruitée. Les erreurs produites par le composant de compréhension ne sont ainsi plus nécessairement liées au composant lui-même. Elles peuvent être provoquées par une qualité insuffisante des transcriptions automatiques.

Dans le cadre d'une chaîne de composants, les erreurs du premier composant se propageront au fil des composants et impacteront nécessairement les performances finales [Hahn et al. 2010].

Des travaux se sont concentrés sur la simulation d'erreurs de reconnaissance de la parole pour améliorer la robustesse du composant de compréhension du langage [Simonnet, Ghannay, Camelin et Estève 2018].

Malgré l'apport important réalisé par ces travaux, ils n'ont pas permis de totalement contrer la propagation des erreurs au sein du composant de compréhension du langage.

En complément, l'exploitation des transcriptions automatiques implique un second inconvénient. Il s'agit de la représentation unique sous forme textuelle de l'ensemble des informations issues de l'audio. Ce qui signifie que les transcriptions agissent comme un goulot d'étranglement.

Bien que le texte soit représentatif du sens du discours d'un locuteur, la parole comporte

de nombreuses informations pouvant être complémentaires au discours prononcé.

Enfin, la mise en œuvre d'une chaîne de composants implique une optimisation séparée de chacun des composants. Cela implique l'absence d'une optimisation jointe pour la tâche finale et peut potentiellement conduire à une optimisation sous-optimale de la chaîne de composants.

3.6 Conclusion

Au sein de ce chapitre, nous avons décrit la tâche de compréhension de la parole, ainsi que les technologies principalement employées pour l'effectuer.

Nous avons ainsi vu que la compréhension de la parole consiste en la projection des informations de paroles d'une dimension acoustique vers une représentation sémantique définie de manière ad hoc. Cette définition ad hoc permet de prendre en charge la tâche par l'intermédiaire des méthodes d'apprentissage supervisé. Ces méthodes ont jusque-là traité la tâche comme une chaîne de traitements successifs, avec tout d'abord la reconnaissance de la parole, puis la compréhension du langage appliqué sur les transcriptions automatiques.

Nous avons aussi mentionné que les tâches de compréhension de la parole correspondent souvent à une tâche de remplissage de champs. C'est notamment le cas de la reconnaissance des entités nommées et de l'extraction des concepts sémantiques, qui sont les tâches exploitées dans le cadre de cette thèse.

Nous avons vu que l'approche d'apprentissage machine traditionnelle la plus efficace correspondait aux CRF, mais aussi que les approches neuronales ont permis de pousser les limites en termes de performances.

Au début de cette thèse, des approches fondées sur une combinaison de couches neuronales CNN et bLSTM associées à un CRF constitué l'état de l'art, pour le composant de compréhension. Puis, pendant cette thèse, les technologies ont évolué vers les approches de type encodeur/décodeur, les mécanismes d'attention et les approches de bout en bout.

Enfin, nous avons évoqué les principales métriques d'évaluation concernant la compréhension de la parole, ainsi que l'impact des transcriptions automatiques pour la tâche finale dans le cadre d'une chaîne de composants.

Cet impact constitue une motivation importante pour la réalisation de l'objectif de cette thèse, qui concerne la mise en œuvre d'une approche de bout en bout entièrement optimisée pour la compréhension de la parole.

Pour terminer, davantage de détails concernant la compréhension du langage parlé sont donnés dans [Tur et De Mori 2011]. Nous encourageons le lecteur à consulter cette référence.

Chapitre 4

Ensembles de données

Sommaire

Un ensemble de données, ou corpus, est un regroupement fini de données, de même nature, dans l'optique d'une étude précise. Dans le cas de l'apprentissage automatique supervisé, ces ensembles, composés de documents (texte, parole, image...), sont analysés et annotés par l'humain selon un cadre applicatif. L'annotation experte des données est une tâche nécessitant du temps, ce qui rend l'obtention de corpus couteux. Habituellement, un corpus est séparé en trois parties distinctes ayant chacune un rôle précis :

— L'ensemble d'apprentissage est couramment composé d'environ 70 à 80 % du corpus total et permet l'entraînement d'un système. Il s'agit d'exemples permettant à un algorithme de construire un modèle.

— L'ensemble de développement est généralement composé d'environ 10 à 15 % du corpus total. Il permet d'optimiser le modèle produit lors de l'entraînement.

— L'ensemble de test, ou d'évaluation, est composé d'environ 10 à 15 % du corpus total. Il est exploité après avoir terminé l'apprentissage et l'optimisation d'un modèle. Son objectif est de permettre l'évaluation des performances du modèle dans les conditions les plus proches de son exploitation finale. Les résultats obtenus reflètent les performances du modèle, notamment concernant sa capacité de généralisation, puisqu'il s'agit de données jamais observées par le système lors de son apprentissage et de son optimisation.

Au regard des tâches applicatives de cette thèse, nous exploitons des données audio manuellement transcrites et annotées sémantiquement, que ce soit en entités nommées ou en concepts sémantiques. Nous avons choisi de développer des systèmes en français. Dans le but de limiter le biais de spécialisation des systèmes que nous entraînerons, nous avons choisi de maximiser notre quantité de données, en regroupant un ensemble de corpus cohérent. Nous avons également choisi de mettre l'accent sur la reproductibilité de nos travaux. C'est pourquoi, tous les ensembles que nous exploitons sont distribués par des organismes facilitant leurs diffusions, notamment l'European Language Resources Association (ELRA).

Dans les sections suivantes, nous présenterons les corpus s'intégrant dans le cadre applicatif de cette thèse, ainsi que leur répartition. Leur répartition est calculée sur la base des segments de parole, en excluant toutes les portions d'audio non exploitables (musiques, publicités...) et elle est exprimée en heures. Certains corpus possèdent une annotation sémantique, que nous détaillerons. Enfin, nous expliquerons les décisions prises sur ces ensembles de données permettant de former notre base de travail.

4.1 Les corpus ESTER

Deux campagnes françaises pour l'Évaluation des Systèmes de Transcriptions enrichies d'Émissions Radiophoniques (ESTER) ont permis de collecter des données entre 1998 et 2008. Les données produites correspondent à des émissions de journaux télévisés transcrites ma-

nuellement et annotées dans le but d'évaluer les performances des systèmes de traitement de la parole. Les évaluations des campagnes ESTER ont porté sur plusieurs tâches, notamment sur la transcription de la parole et l'extraction d'information. Nous décrivons plus en détail les données produites, ainsi que leurs origines dans les sous-sections suivantes.

4.1.1 ESTER 1

ESTER 1 est la première campagne d'évaluation ESTER qui s'est déroulée en deux phases, en 2003 [Gravier, Bonastre et al. 2004] et en 2005 [Galliano, Geoffrois, Mostefa et al. 2005]. Le corpus [Galliano, Geoffrois, Gravier et al. 2006] produit à l'occasion de cette campagne a été mis à disposition en 2005 par l'organisme ELRA sous la référence ELRA-S0241. L'objectif de cette campagne était d'initier des travaux sur le traitement d'émissions de journaux d'informations. Elle visait notamment la transcription orthographique, la détection et le suivi d'événement, ainsi que l'extraction d'informations (détection des entités nommées).

Ce corpus est constitué d'un ensemble de données audio transcrites manuellement et comportant 95 heures de paroles. Il est complété avec un ensemble de données audio non transcrites représentant 1700 heures. L'audio est issu de six sources francophones distinctes toutes enregistrées entre 1998 et 2004 : Radio France International, France Inter, France Info, Radio Télévision Marocaine, France Culture et Radio Classique.

Dans le cadre de cette thèse, nous exploitons uniquement les données manuellement transcrites. La répartition de ces données en fonction des ensembles d'apprentissage, de développement et de test est représentée par la figure 4.1.

Figure 4.1 – Répartition des données dans le corpus ESTER 1 exprimée en heures

Cet ensemble est annoté manuellement en entités nommées. Cependant, la tâche de Reconnaissance des Entités Nommées était prospective. Elle avait pour but de définir un formalisme, un corpus et des outils d'évaluation. Le formalisme d'annotation mis initialement en place pendant cette campagne a été corrigé et augmenté lors de la seconde campagne (ESTER 2). Ce corpus est donc annoté selon le formalisme corrigé que nous présenterons dans la sous-section 4.1.3.

4.1.2 ESTER 2

Ce corpus est un complément à l'ensemble de données ESTER 1. Il est produit dans le cadre de la campagne d'évaluation du même nom [Galliano, Gravier et al. 2009], réalisée en 2009. Cette campagne est une extension d'ESTER 1 ciblant une plus grande variété de styles d'expressions orales et d'accents. Le corpus est mis à disposition par l'ELRA depuis 2012 sous la référence ELRA-S0338 et il est constitué de 102,5 heures de paroles manuellement transcrites.

Les données audio sont, comme ESTER 1, issues de journaux d'informations de radio et de télévision française. Elles proviennent de cinq sources distinctes : Radio France International, France Inter, Radio Télévision Marocaine, Africa number one et Radio Congo. La figure 4.2 représente la répartition des données de ce corpus.

Figure 4.2 – Répartition des données du corpus ESTER 2 exprimée en heures

En complément de la transcription manuelle, l'ensemble de développement est annoté en entités nommées suivant le formalisme ESTER présenté dans la sous-section suivante.

4.1.3 Formalisme d'annotation en entités nommées ESTER

Le guide d'annotation ESTER présenté dans cette section correspond à celui mis en place pendant la campagne ESTER 2. Il s'agit de la version corrigée et augmentée des annotations mises en place initialement par la campagne ESTER 1.

Ce formalisme est composé de 7 catégories principales : *Date et heure*, *Fonction*, *Localisation*, *Organisation*, *Personne*, *Production humaine et Montant*. Chaque catégorie possède des sous-catégories pour détailler davantage les entités. Par exemple, avec ce formalisme, différentes sous-catégories permettent de distinguer les *localisations géographiques* des *localisations éducatives*. Une catégorie *Inconnu* est utilisé lorsqu'une incertitude subsiste pour une entité nommée, par exemple "le Footsie". Il s'agit d'un indice en bourse : FTSE.

En complément, la campagne ESTER 2 a utilisé le principe d'imbrication des entités nommées. Deux entités sont considérés imbriquées si l'une est totalement incluse dans l'autre. L'ensemble des catégories et sous-catégories sont présentées dans la table suivante. Nous donnons le label d'entité nommée associée, ainsi qu'un exemple de valeur possible.

Catégorie	Sous-catégories	Label	Exemple de valeur
Date et heure	Date absolue	**time.date.abs**	*en 2008*
	Date relative	**time.date.rel**	*hier*
	heure	**time.hour**	*22 heures 30*
Fonction	Administrative	**fonc.admi**	*Directeur générale*
	Aristocratique	**fonc.ari**	*Comtesse de Ségur*
	Militaire	**fonc.mil**	*Colonel*
	Politique	**fonc.pol**	*Président*
	Religieuse	**fonc.rel**	*Dalaï-lama*
Localisation	Adresse email	**loc.addr.elec**	*telsonne@radio-france.fr*
	Adresse postale	**loc.addr.post**	*rue de Lappe*
	Téléphone et fax	**loc.addr.tel**	*01 45 24 XX XX*
	Administrative	**loc.admi**	*Berlin*
	Construction humaine	**loc.fac**	*Maison de la radio*
	Géographique	**loc.geo**	*Côte basque*
	Axe de circulation	**loc.line**	*Champs-Élysées*
Organisation	Politique	**org.pol**	*Armée de terre*
	Éducative	**org.edu**	*IUT de chimie*
	Commerciale	**org.com**	*Tanger 2012*
	Non commerciale	**org.non-profit**	*Hôpital Necker*
	Divertissement	**org.div**	*Festival des alizés*
	Géo-Socio-Politique	**org.gsp**	*Royaume du Maroc*
Personne	Animal	**pers.anim**	*Félix le chat*
	Humaine	**pers.hum**	*Laure Manaudou*
Production humaine	Oeuvre artistique	**prod.art**	*Madame Bovary*
	Récompense	**prod.award**	*Prix Romy Schneider*
	Documentaire	**prod.doc**	*Le petit Robert*
	Moyen de transport	**prod.vehicule**	*Navette Endeavour*
Montant	Monétaire	**amount.cur**	*6 euros*
	Âge	**amount.phys.age**	*24 ans*
	Surface	**amount.phys.area**	*60 mètres carré*
	Durée	**amount.phys.dur**	*2 heures*
	Longueur	**amount.phys.len**	*3 centimètres*

Suite page suivante

Catégorie	Sous-catégories	Label	Exemple de valeur
	Autre	**amount.phys.other**	*5 sur l'échelle de Richter*
	Vitesse	**amount.phys.spd**	*130 kilomètres/heure*
	Température	**amount.phys.temp**	*16 degrés*
	Volume	**amount.phys.vol**	*1500 litres*
	Poids	**amount.phys.wei**	*2 kilos*
Inconnu	————————	**unk**	*Cac 40*

Table 4.1 – Description du schéma d'annotation ESTER

4.2 QUÆRO

Le projet QUÆRO, initié en 2008 et achevé en 2013, est un projet de recherche collaboratif relatif à l'analyse automatique et à l'enrichissement de contenus numériques, multimédias et multilingues. Ce projet utilise la totalité des transcriptions du corpus ESTER 1 décrit en section 4.1.1 comme ensemble d'apprentissage. Un ensemble de 6,5 heures de paroles nouvellement transcrites forme l'ensemble de test. L'ensemble QUÆRO est distribué par l'ELRA sous la référence ELRA-S0349. Nous donnons la répartition des données de cet ensemble dans la figure 4.3.

Figure 4.3 – Répartition des données du corpus QUÆRO exprimée en heures

Une particularité du projet QUÆRO correspond à la mise en place d'une définition étendue des entités nommées [Grouin et al. 2011]. Ainsi, tout ce corpus est annoté selon ce formalisme. Cela signifie que l'ensemble d'apprentissage a été réannoté selon ce formalisme. Nous le décrivons dans la sous-section suivante.

4.2.1 Formalisme d'annotation en entités nommées QUÆRO

Cette définition étendue des entités nommées reprend en partie le formalisme ESTER [4.1.3]. Nous retrouvons une définition hiérarchique suivant le même principe ainsi que les mêmes catégories principales d'entités nommées. Ce formalisme ajoute une nouvelle catégorie : *Événement*.

De plus, la définition étendue des entités nommées de ce formalisme n'est plus uniquement hiérarchique, elle est aussi compositionnelle. Cela signifie qu'une entité est décomposée

en composants et elle est donc structurée. Nous pouvons prendre en exemple une personne, *Jean-Baptiste Poquelin*. Il s'agit d'une personne (*pers.ind*) qui a un prénom (*name.first*) et un nom (*name.last*) permettant ainsi de décomposer l'entité nommée. L'aspect hiérarchique et compositionnel de cet exemple est représenté en **A** dans la figure 4.4.

Dans le formalisme QUÆRO, il est aussi possible d'imbriquer des concepts d'entités nommées. C'est-à-dire qu'une entité nommée peut être décomposée à l'aide de composants, mais aussi à l'aide d'une ou plusieurs autres entités nommées (exemple **B** de la même figure).

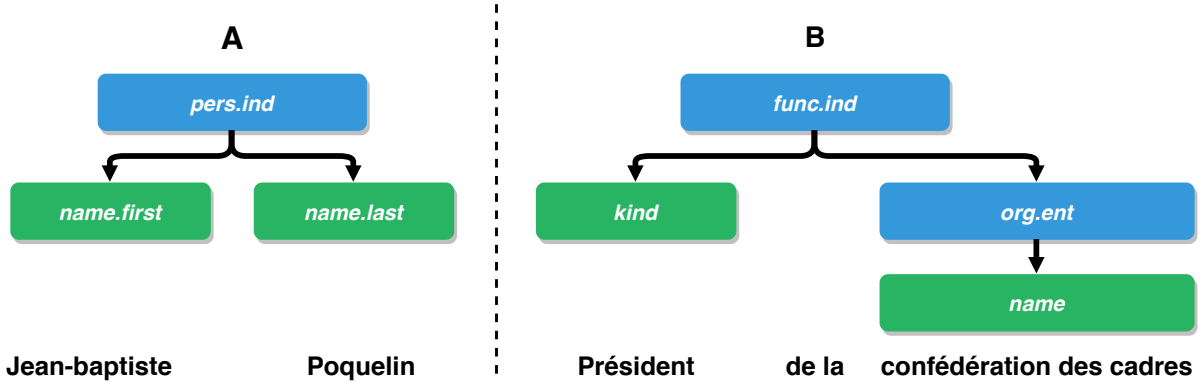

FIGURE 4.4 – Exemple de la structure QUÆRO des entités nommées. En bleu les annotations de catégories, en vert les annotations de composants

Nous fournissons la liste complète des catégories et sous-catégories du formalisme QUÆRO dans la table suivant. La liste des composants ainsi que leurs dépendances aux catégories d'entités nommées sont disponibles dans la table 4.3.

Catégorie	Sous-catégories	Label	Exemple de valeur
Durée	Date absolue	**time.date.abs**	*25 janvier 2010*
	Date relative	**time.date.rel**	*mardi prochain*
	Heure absolue	**time.hour.abs**	*une heure et demi*
	Heure relative	**time.hour.rel**	*ce matin*
Évènement	————————	**event**	*le Tour de France*
Fonction	Collectif	**func.coll**	*les pompiers*
	Individu	**func.ind**	*le maire de Paris*
Localisation	Administratif - National	**loc.adm.nat**	*le Royaume-Uni*
	Administratif - Région	**loc.adm.reg**	*au sud d'Israël*
Suite page suivante			

Catégorie	Sous-catégories	Label	Exemple de valeur
	Administratif - Supranational	**loc.adm.sup**	*la Catalogne*
	Administratif - Ville	**loc.adm.town**	*Paris*
	Adresse - Électronique	**loc.add.elec**	*98.8 MHz*
	Adresse - Physique	**loc.add.phys**	*15 rue de Vaugirard*
	Bâtiments	**loc.fac**	*la gare de Rungis*
	Physique - Aquatique	**loc.phys.hydro**	*la Seine*
	Physique - Astronomique	**loc.phys.astro**	*la Lune*
	Physique - Terrestre	**loc.phys.geo**	*le désert de Gobi*
	Oronyme	**loc.oro**	*l'autoroute A6*
Organisation	Administration	**org.adm**	*la mairie de Paris*
	Entreprise	**org.ent**	*la police française*
Personne	Collectif	**pers.coll**	*les Beatles*
	Individu	**pers.ind**	*Bertrand Delanoë*
Production	Artistique	**prod.art**	*Le Malade Imaginaire*
	Autre	**prod.other**	*le Monopoly*
	Doctrine	**prod.doctr**	*le socialisme*
	Financier	**prod.fin**	*le fonds DWS*
	Ligne de transport	**prod.serv**	*le RER B*
	Logiciel	**prod.soft**	*Twitter*
	Loi / décret	**prod.rule**	*traité de Versaille*
	Marque d'objet	**prod.object**	*navette Endeavour*
	Médiatique	**prod.media**	*Radio Bleue*
	Prix divers	**prod.award**	*Prix Nobel*
Quantité	————————	**amount**	*quelques mètres*

Table 4.2 – Description du schéma d'annotation QUÆRO : catégories

Catégorie d'Entité Nommée	Composants
Adresse (*loc.add.xxx*)	*address-number, po-box, zip-code*
Durée (*time.xxx*)	*day, week, month, year, century, millenium, reference-era, time-modifier*
Personne (*pers.xxx*)	*name.last, name.fist, name.middle*
Quantité (*amount*)	*object*
Transversale	*name, name.nickname, kind, extractor, unit, demonym.nickname, qualifier, val, range-mark*

Table 4.3 – Description du schéma d'annotation QUÆRO : composants

Nous avons analysé les données du corpus QUÆRO. Cette analyse nous a permis de comprendre la répartition des entités nommées qui sont décrites dans la table 4.4 et la figure 4.5. La table précise la composition générale du corpus en termes de mots, d'entités nommées et de composants tandis que la figure donne la répartition des catégories d'entités nommées.

Ensemble	Nombre de mots	Nombre d'Entités Nommées	Nombre de composants
Apprentissage	1 113 107	111 927	165 186
Test	81 531	5 538	8 038

Table 4.4 – Composition des données QUÆRO en nombre de mots, d'entités nommées et de composants

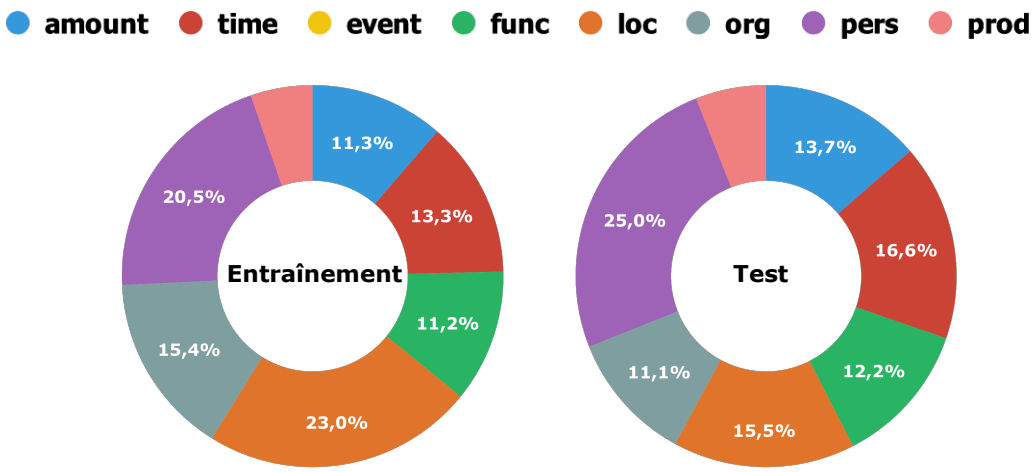

Figure 4.5 – Répartition des catégories d'entités nommées des données QUÆRO

4.3 ETAPE

La campagne d'évaluation ETAPE [Galibert, Leixa et al. 2014] correspond à la troisième campagne française pour l'évaluation du traitement d'émission de journaux d'informations et a été réalisée en 2012. Il s'agit d'une campagne faisant suite à la série des campagnes ESTER présentées précédemment. ETAPE a apporté certaines nouveautés, comme par exemple l'introduction de paroles spontanées et le chevauchement de locuteurs. Un corpus [Gravier, Adda et al. 2012], tiré de cette campagne, est distribué depuis 2017 par l'ELDA sous la référence ELRA-E0046.

Ce corpus est constitué de 32,5 heures de paroles issue d'émissions de journaux provenant de quatre sources distinctes : France Inter, BFM TV, LCP et TV8 Mont-Blanc.

Nous donnons la répartition des données de ce corpus dans la figure 4.6.

FIGURE 4.6 – Répartition des données dans le corpus ETAPE exprimée en heures

ETAPE exploite le formalisme QUÆRO des entités nommées tel que défini dans la section 4.2.1. Des statistiques sur le corpus en termes de nombre de mots, d'entités nommées et de composants, pour chaques parties du corpus, sont présentées dans la table 4.5.

Ensemble	Nombre de mots	Nombre d'Entités Nommées	Nombre de composants
Apprentissage	268 798	18 786	22 931
Développement	90 644	5 998	7 174
Test	95 314	5 366	7 690

TABLE 4.5 – Composition des données ETAPE en nombre de mots, d'entités nommées et de composants

Nous présentons la répartition des catégories des entités nommées pour les ensembles de développement, de test et d'apprentissage dans la figure 4.7.

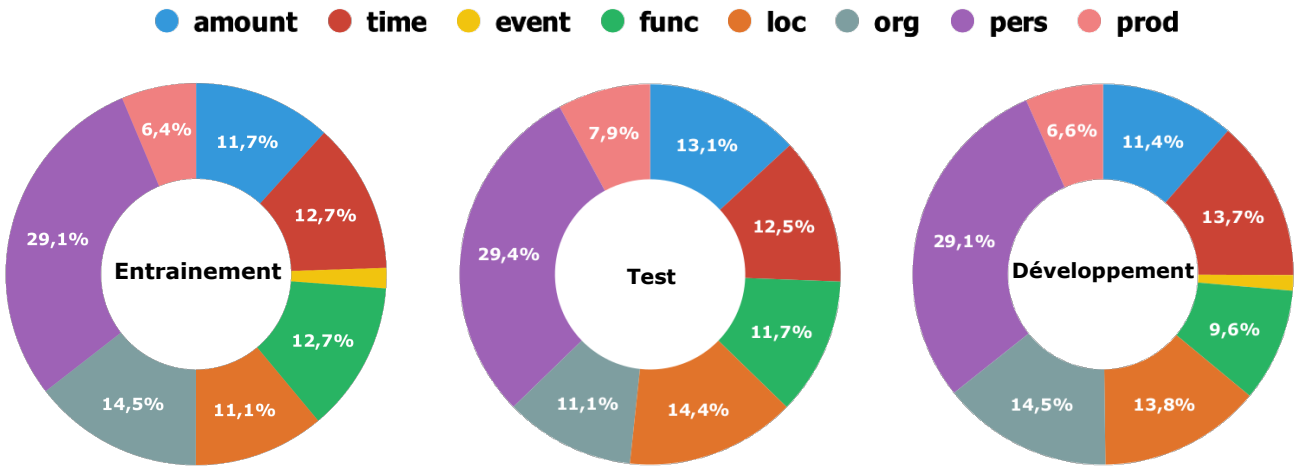

FIGURE 4.7 – Répartition des catégories d'entités nommées des données ETAPE

4.4 EPAC

Le projet d'Exploration de masse de documents audio pour l'extraction et le traitement de la PArole Conversationnelle (EPAC) s'est déroulé du 1er janvier 2007 au 31 décembre 2009. Il concerne le traitement de données audio non structurées et a pour but de proposer des méthodes d'extraction d'information et de structuration de données audio.

Pour ce faire, un corpus [Estève et al. 2010] a été construit à partir des 1 700 heures de données audio non transcrites fournies par la campagne ESTER 1. L'objectif de ce corpus est de mettre l'accent sur la parole conversationnelle. Il est composé de 90 heures de paroles manuellement transcrits et il est provient de trois sources distinctes : France Inter, France Culture et RFI. Il est mis à disposition depuis 2010 par l'ELRA sous la référence ELRA-S0305. Nous donnons la répartition de ce corpus dans la figure 4.8.

Figure 4.8 – Répartition des données dans le corpus EPAC exprimée en heures

4.5 REPERE

Le projet de REconnaissance de PERsonnes dans des Émissions audiovisuelles (REPERE) est une campagne d'évaluation [Galibert et Kahn 2013; Bernard et al. 2014] de systèmes permettant l'identification du locuteur, selon les modalités visuelle et parole. Cette campagne s'est déroulée en trois étapes entre 2012 et 2014.

Un corpus [Giraudel et al. 2012] a été construit pour mener à bien cette campagne d'évaluation. Plusieurs versions de ce corpus ont été distribuées et nous utilisons sa version finale. Le corpus est constitué d'enregistrements d'émissions de journaux télévisés provenant de deux chaînes françaises : BFM TV et LCP. REPERE est composé de 48 heures de vidéo et de paroles distribuées par l'ELRA sous la référence ELRA-E0044 depuis 2015. Nous donnons la répartition des données dans la figure 4.9.

Figure 4.9 – Répartition des données dans le corpus REPERE exprimée en heures

Pour la tâche de reconnaissance du locuteur à partir de la parole, les transcriptions manuelles du corpus sont sémantiquement annotées en personnes selon les noms et prénoms uniquement.

4.6 Les corpus MEDIA et PORTMEDIA

Nous abordons dans les sections suivantes des corpus produits dans le cadre de projet permettant l'étude de la compréhension de dialogue (MEDIA), ainsi que de la portabilité de domaine et de langue (PORTMEDIA). Les corpus issus de ces projets sont constitués de données téléphoniques. Il sont construits avec la méthode du "magicien d'Oz" qui vise à simuler un dialogue homme-machine. Pour cette méthode les réponses de la machine sont données par un humain qui simule le comportement du système. Nous détaillerons d'avantage ces corpus dans les sous-sections suivantes.

4.6.1 MEDIA

Le projet Méthodologie d'Évaluation automatique de la compréhension hors et en contexte du DIAlogue (MEDIA) est un projet débuté en 2002 et achevé en 2006. Son objectif est de définir et de tester une méthodologie de la compréhension des systèmes de dialogue.

Pour ce faire, un corpus de dialogue français [Bonneau-Maynard et al. 2005] a été créé. Il est issu d'une simulation de serveur téléphonique pour une tâche de réservation d'hôtel. Depuis 2008, ce corpus est distribué par l'ELRA sous la référence ELRA-S0272. Il est composé de 1 258 dialogues avec différents scénarios de réservation d'hôtel, allant de la réservation simple, a des réservations plus complexes intégrant des changements d'avis de la part de l'utilisateur pendant le dialogue. Ce corpus est composé de 57,5 heures de parole, dont 23,5 heures pour la partie utilisateur et 34 heures pour la partie système. Nous détaillons la répartition des heures, en fonction de la partie système et de la partie utilisateur, dans la figure 4.10.

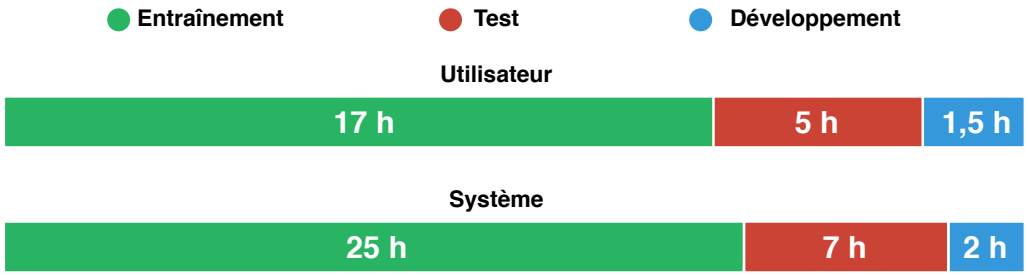

Figure 4.10 – Répartition des données dans le corpus MEDIA exprimée en heures en fonction de la partie utilisateur et de la partie système.

Afin de rendre possible la tâche de compréhension du langage, la transcription manuelle de la partie utilisateur est annotée selon une représentation sémantique propre au domaine de MEDIA. Cette représentation sémantique sera détaillée dans la sous-section 4.6.3.

4.6.2 PORTMEDIA

Le projet PORTMEDIA a pour objectif de compléter le corpus MEDIA. Les principaux axes de ce projet concerne la portabilité multilingue, multidomaines, ainsi que la représentation sémantique. Ce projet a permis la mise en place d'un corpus [Lefèvre et al. 2012] qui est un complément au corpus MEDIA. Ce corpus est séparé en deux parties distinctes : PM-Lang et PM-Dom.

La partie PM-Lang correspond au corpus MEDIA traduit en italien et annoté sémantiquement de la même manière.

La partie PM-Dom correspond à un nouveau corpus français de dialogue homme-machine suivant le paradigme et les spécifications du corpus MEDIA. Pour cette partie, le domaine est modifié, passant ainsi d'une tâche de réservation d'hôtel à une tâche de réservation de billet pour le festival d'Avignon de 2010. Ce corpus est distribué par l'ELRA sous la référence ELRA-S0371 depuis 2014.

Nous exploitons les données françaises, c'est-à-dire la partie PM-Dom. Elle est constituée de 700 dialogues manuellement transcrits représentant un total de 34 heures de parole. La partie PM-Dom est divisée en une sous partie utilisateur, représentant 12,5 heures, et en une sous partie système, représentant 21,5 heures. Nous fournissons la répartition des données de la partie PM-Dom en fonction du système et de l'utilisateur dans la figure 4.11.

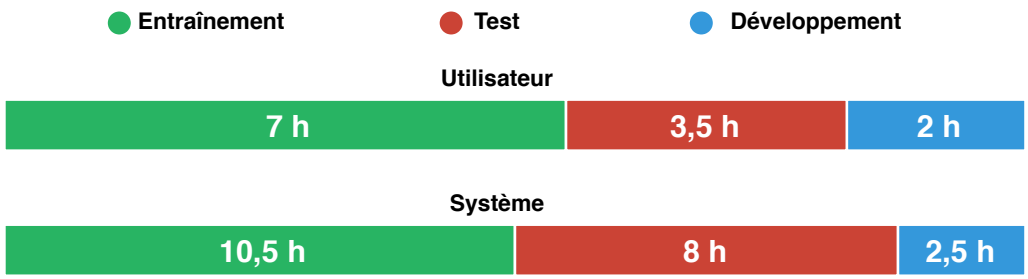

Figure 4.11 – Répartition des données dans le corpus PORTMEDIA exprimée en heures en fonction de la partie utilisateur et de la partie système.

Une annotation sémantique est appliquée sur la partie utilisateur, nous la décrivons dans la sous-section suivante.

4.6.3 Formalisme d'annotation en concepts sémantiques

Les annotations sémantiques sont très dépendantes du domaine d'application du corpus sur lesquelles elles sont appliquées. Pour MEDIA et PORTMEDIA, elles sont donc naturellement différentes, bien qu'elles partagent certains concepts sémantiques. En effet, un objectif de PORTMEDIA est d'étudier la portabilité de domaine. Il est nécessaire que son annotation sémantique se rapproche de celle de MEDIA. Nous listons les annotations sémantiques propres à MEDIA et PORTMEDIA dans la table 4.6, ainsi que leurs annotations communes.

MEDIA	chambre-equipement, chambre-fumeur, chambre-standing, chambre-type, chambre-voisine, comparatif, evenement, hotel-etat, hotel-etat-complet, hotel-etoile, hotel-marque, hotel-parking, hotel-services, localisation-arrondissement, localisation, localisation-cardinal, localisation-codePostal, localisation-departement, localisation-distanceRelative, localisation-lieuRelatif, localisation-pays, localisation-rue, localisation-quartier, localisation-region, nombre-chambre, nombre-chambre-disponible, nombre-hotel, nombreNonDigit-hotel, nombre-reservation, nombre-restaurant, nombre-temps, nom-client, nom-hotel, paiement-methodeDePaiement, paiement-montantQualitatif, personne-nomDeFamille, rang-hotel, rang-reservation, rang-temps, sejour-nbAdulte, sejour-nbCouple, sejour-nbEnfant, sejour-nbNuit, sejour-nbLitBebe, sejour-nbPersonne, temps-axeTps, temps-jourFerie, temps-plageRelative, temps-plageTps, unknown
PORTMEDIA	etat-piece-complet, nb-billets, nb-reservation, nom-lieu, nom-piece, numero-reference, piece-nom-auteur, type-artiste, type-billet, type-spectacle
Commun	command-dial, command-tache, comparatif-paiement, comparatif-temps, connectAttr, connectProp, lienRef-coDom, lienRef-coRef, lienRef-elsEns, localisation-ville, nom, nombre, nombreNonDigit, objet, objetBD, paiement-monnaie, paiement-montant-entier, rang, reponse, temps-annee, temps-date, temps-heure, temps-jour-mois, temps-jour-semaine, temps-mois, temps-unite

Table 4.6 – Concepts sémantiques MEDIA / PORTMEDIA

4.7 DECODA

Le projet de DÉpouillement automatique de COnversations provenant de centres D'Appels (DECODA) a démarré en 2009 et vise à réduire le coût de développement des systèmes d'analyse de la parole. Il s'oriente notamment sur la réduction des besoins en annotation manuelle des corpus. Son objectif est de proposer des outils robustes pour le traitement de la parole, dans le cadre des centres d'appel de la Régie Autonome des Transports Parisiens (RATP). Pour réaliser ce projet, un corpus a été collecté [Bechet et al. 2012] en conditions réelles.

Il est composé de 1 514 conversations représentant 56,5 heures de parole dont nous fournissons la répartition dans la figure 4.12.

Figure 4.12 – Répartition des données dans le corpus DECODA exprimée en heures.

Un objectif de DECODA est d'appréhender la tâche de compréhension du langage. Ses concepts sémantiques sont proches des entités nommées. En effet, nous retrouvons par exemple des catégories comme : *Personne*, *Localisation*, *Organisation*, *Production*, *Dates et Heures*. Toutefois, un nombre non négligeable de ces entités nommées sont anonymisées, ne nous permettant pas l'accès à la valeur de l'entité nommée concernée.

Ce corpus n'est pas distribué par l'ELRA, cependant il peut être récupéré par l'intermédiaire de la plateforme ORTOLANG sous la référence sldr000847 [1] et sur demande aux auteurs.

4.8 Répartition des données au sein de cette thèse

Dans cette thèse nous souhaitons exploiter tous les corpus présentés ci-dessus en respectant les répartitions officielles.

L'addition de toutes ces données pose certaines questions inhérentes à leurs natures. Bien qu'il s'agisse de paroles, nous pouvons distinguer deux types de données : Les données studio (enregistrés en 16 Khz) et les appels téléphoniques (enregistrés en 8 Khz).

Les données de journaux d'informations que nous regroupons correspondent aux corpus EPAC, ESTER 2, ETAPE, QUÆRO et REPERE. Nous excluons ESTER 1 puisqu'il correspond aux même enregistrements audio que QUÆRO. La somme des corpus retenue totalisent 372,5 heures de paroles, dont une partie est annotée en entités nommées selon les formalismes ESTER

1. http ://sldr.org/sldr000847/fr

(ESTER 1 et ESTER 2) et QUÆRO (QUÆRO et ETAPE). Ces annotations nous permettent d'envisager la tâche de reconnaissance des entités nommées. Pour cette tâche, nous exploiterons comme base de référence la campagne ETAPE qui correspond à la campagne d'évaluation la plus récente. Cette campagne utilise le formalisme QUÆRO. Ainsi, nous choisisons de conserver les annotations de ce formalisme et d'utiliser la répartition du corpus QUÆRO plutôt que celle d'ESTER 1. Ce regroupement contient donc 132,5 heures de paroles annotées en entités nommées.

Les données d'appels téléphoniques que nous regroupons correspondent aux corpus DECODA, MEDIA et PORTMEDIA. Ils représentent un total de 147,5 heures de paroles téléphoniques, composés de dialogues humain-machine et de conversation. Les 36 heures composants les parties utilisateurs des données MEDIA et PORTMEDIA sont annotés sémantiquement. Ces données rendent envisageable la tâche de compréhension de la parole par l'extraction de ces concepts sémantiques.

Les deux types de données que nous regroupons permettent la tâche de reconnaissance de la parole grâce à leur transcription. Cependant un échantillonnage différent (8 et 16 Khz) nous impose d'adapter l'audio, en sous-échantillonant, lorsque l'on ne souhaite pas considérer ces deux types indépendamment. Le rassemblement de toutes ces données nous permet d'obtenir un corpus totalisant 520 heures de paroles, dont la répartion est représentée dans la figure 4.13.

Figure 4.13 – Répartition des données dans le regroupement des corpus exprimée en heures.

Dans le cadre de cette thèse, nous avons regroupé ces corpus afin de mettre en place les bases nécessaires aux tâches de reconnaissance de la parole, de reconnaissance des entités nommées et d'extraction des concepts sémantiques, que nous étudions dans la partie suivante.

Deuxième partie

Contributions

Chapitre 5

Reconnaissance d'entités nommées

Sommaire

Ce premier chapitre de contributions concerne la mise en place d'un système de bout en bout appliqué à la compréhension de la parole.

Comme nous l'avons vu dans la partie état de l'art, les systèmes permettant la compréhension de la parole sont, jusque-là, des chaînes de traitements avec des composants successifs. Elles sont composées d'un système de reconnaissance automatique de la parole (RAP) et d'un système de compréhension du langage qui sera appliqué sur les transcriptions automatiques.

Cette thèse a pour objectif de s'émanciper de la transcription automatique intermédiaire pour ne former qu'un seul système entièrement optimisé sur la tâche finale. L'intérêt premier étant de dépasser la difficulté provoquée par l'application d'un système de compréhension sur des transcriptions automatiques imparfaites, sources de bruits et donc d'erreurs. L'intérêt second étant de disposer d'un système unique plus simple à maintenir.

Lorsque cette thèse a débuté, l'apprentissage profond constituait l'état de l'art dans plusieurs domaines, dont la reconnaissance de la parole et la compréhension du langage. Des travaux, orientant ceux de cette thèse, montrent l'intérêt d'approches de bout en bout dans le cadre de ces tâches indépendantes [Hannun et al. 2014; Amodei et al. 2016; Y. Zhang et al. 2016; X. Ma et Hovy 2016].

L'objectif de ce chapitre est de bénéficier des travaux du domaine de RAP pour mettre en place un premier système de bout en bout effectuant la reconnaissance des entités nommées (REN) directement depuis la parole.

Nous choisissons la reconnaissance des entités nommées en français comme première tâche applicative de compréhension de la parole. L'intérêt pour cette tâche réside dans les nombreux travaux réalisés par de multiples campagnes d'évaluation [Gravier, Bonastre et al. 2004; Galliano, Geoffrois, Mostefa et al. 2005; Galliano, Gravier et al. 2009; Galibert, Leixa et al. 2014]. En outre, nous avons à notre disposition l'ensemble des données de ces campagnes, renforçant notre intérêt pour cette tâche.

Toutefois, les derniers travaux réalisés correspondent à la campagne ETAPE qui s'est déroulée avant l'application avec succès des technologies neuronales pour la RAP et la REN. Ce point définit notre première contribution, qui consiste à mettre en œuvre des systèmes à l'état de l'art dans le cadre de la campagne ETAPE. C'est-à-dire, mettre en œuvre les nouvelles approches neuronales sur les ensembles de données de la campagne ETAPE. Cette mise à jour des résultats nous permettra d'avoir une vision des capacités actuelles des systèmes neuronaux employés sous forme d'une chaîne de composants.

Nous définissons ensuite notre deuxième contribution, qui concerne la mise en œuvre d'une première approche de bout en bout dédiée à la REN dans la parole. Toutefois, la campagne ETAPE fait appel à une définition structurée des entités nommées pouvant être considérée comme riche (voir sous-section 4.2.1). Pour cette contribution, nous choisissons de simplifier la tâche de REN afin de vérifier la viabilité de notre approche.

Ainsi, notre dernière contribution, pour ce chapitre, découle directement de la deuxième. Elle consiste à mettre en œuvre notre approche de bout en bout dans le cadre de la campagne ETAPE, en rendant comparable les résultats de cette dernière contribution et de la première.

Ce chapitre est donc organisé selon l'ordre de nos trois contributions. Après une présentation de notre contexte d'étude dans la première section, nous effectuerons la mise à jour des résultats de la campagne ETAPE dans la deuxième. Nous détaillerons ensuite, en troisième section, la mise en œuvre de notre système de REN de bout en bout pour la tâche simplifiée que nous définissons. Enfin, dans une dernière section, nous exploiterons notre approche de bout en bout dans le cadre de la campagne ETAPE.

5.1 Contexte des travaux : ETAPE

Nous exploitons la campagne d'évaluation ETAPE [Gravier, Adda et al. 2012] comme point de départ pour nos travaux. Il s'agit de la dernière campagne française recensée pour la reconnaissance des entités nommées dans la parole. Comme mentionnée dans la première partie de cette thèse, les entités nommées permettent la compréhension de documents. Elles peuvent être définies comme les éléments informationnels pertinents dans la description d'un événement et d'un fait [Nouvel et al. 2015].

Cette campagne exploite une définition étendue des entités nommées, mettant ainsi en place une tâche de reconnaissance d'entités nommées structurées. Il s'agit du formalisme d'annotation QUÆRO décrit dans la section 4.2.1 de ce manuscrit. Elle correspond à une tâche de remplissage de champs (*Slot filling*), dont l'objectif est de retrouver les mots composants une entité et de définir le type associé.

Pour réaliser la reconnaissance des entités nommées dans la parole lors de la campagne, il était nécessaire de mettre en place une chaîne dont les composants sont chacun optimisés pour une tâche spécifique. On retrouve ainsi l'imbrication d'un système de reconnaissance de la parole et d'un système de reconnaissance des entités nommées. Ce deuxième système est appliqué sur les transcriptions automatiques fournies par le système de RAP.

Chacun des systèmes doit être optimisé en fonction de la tâche qui lui est propre et donc être évalué avec leur propre métrique. Pour le système de reconnaissance de la parole, il s'agit de la métrique du taux d'erreur sur les mots (*Word Error Rate*, *WER*). Tandis que pour le système de reconnaissance des entités nommées, il s'agit de la métrique du taux d'erreur sur les champs (*Slot Error Rate*, *SER*) [Makhoul et al. 1999].

5.1.1 Résultats de la campagne

Nous présentons, ici, uniquement les systèmes ayant conduit aux meilleurs résultats pour la tâche de REN dans la parole. Il s'agit des couples composés des meilleurs systèmes de RAP et de REN.

En 2012, les systèmes de reconnaissance de la parole les plus avancés étaient composés de modèles de Markov cachés (*Hidden Markov Model*, *HMM*) et de modèles à mixtures de gaussiennes (*gaussian mixture model*, *GMM*) [Bougares et al. 2013]. Ce type de système a été appliqué avec succès pendant ETAPE, permettant d'obtenir un taux d'erreur (WER) de 21,8 %.

En complément, les systèmes basés sur les champs aléatoires conditionnels (*Conditional Random Field*, *CRF*) étaient les plus avancés pour une tâche de reconnaissance des entités nommées [McCallum et Li 2003 ; Sarawagi et Cohen 2004 ; Bundschus et al. 2008]. Par l'intermédiaire de 68 modèles CRF binaires, un par type et par composants d'EN, il a été possible d'atteindre un score SER de 59,3 %. Ce score est obtenu en appliquant le système décrit dans [Raymond 2013] sur les transcriptions automatiques du meilleur système de RAP de la campagne.

Lorsque cette thèse a débuté, les technologies de RAP et de REN ont bénéficié des avancées liées aux réseaux neuronaux. Ces avancées rendent désormais les résultats de la campagne incomplets, voire obsolètes. Définissant ainsi le premier objectif de cette thèse, qui consiste à mettre à jour les résultats de la campagne ETAPE avec les technologies à l'état de l'art. Ce travail de mises à jour est l'objet de la section suivante.

5.2 REN structurée par chaîne de composants

Des études ont montré l'intérêt d'architectures neuronales pour chacun des composants impliqués dans la tâche de REN dans la parole. Les travaux que nous citons sont emblématiques des tâches de RAP et de REN visées. Pour la tâche de reconnaissance de la parole, nous pouvons notamment citer [Graves, A.-r. Mohamed et al. 2013 ; Y. Zhang et al. 2016]. Nous pouvons également citer [Z. Huang et al. 2015 ; X. Ma et Hovy 2016 ; Lample et al. 2016 ; J. P. Chiu et Nichols 2016], pour la tâche de reconnaissance des entités nommées.

Dans les sections suivantes, nous détaillons les systèmes de RAP et de REN choisis pour cette étude.

5.2.1 Déploiement d'un système de RAP intégrant un modèle neuronal

Le système de RAP que nous souhaitons exploiter est un système hybride composé d'un modèle de markov caché (HMM) et d'un modèle neuronal à retardement (TDNN). Nous le choisissons en partie pour sa proximité avec les systèmes utilisés lors d'ETAPE, exploitant aussi des modèles de Markov.

De plus, ce système est déjà disponible sur les serveurs de notre laboratoire et a été mis en œuvre dans le cadre de précédents travaux. Il a montré sa capacité à atteindre des performances à l'état de l'art. Il nous est ainsi apparu judicieux d'exploiter ce système qui, en plus de correspondre aux besoins de cette étude, ne nécessitait que peu de temps humain à sa mise en œuvre.

5.2.2 Déploiement d'un système de REN intégrant un modèle neuronal

Nous avons choisi d'exploiter et d'adapter le système NeuroNLP2, en raison de sa mise à disposition par les auteurs de l'étude [X. Ma et Hovy 2016].

Ce système est un empilement de couches neuronales CNN et LSTM bidirectionnelles, suivi d'un CRF pour l'extraction d'informations. Les couches CNN sont utiles à l'extraction de représentation au niveau caractère (*Character embeddings*). Ces représentations sont additionnées à des représentations de mots (*Word embeddings*) issues d'un dictionnaire précalculé. Cette représentation double est utilisée comme entrée d'une couche neuronale LSTM bidirectionnelle qui permet le calcul d'une représentation qui sera fournie à la couche CRF finale.

L'exploitation d'un CRF implique l'utilisation du formalisme "*Begin*, *Inside*, *Outside*" (BIO) pour la représentation des informations sémantiques. Ce formalisme, bien qu'ayant démontré son efficacité, possède une limite pour la représentation d'informations structurées, comme celles de la campagne ETAPE.

5.2.3 Limite du formalisme BIO

Avant d'approfondir sa limite, pour la représentation d'informations structurées, il est nécessaire de définir le formalisme. Le formalisme BIO [Tjong Kim Sang et Veenstra 1999] est utilisé dans le cadre de tâches de labélisation de séquences, qui consistent à représenter l'information sémantique en associant un label à chaque mot présent dans un texte. Les labels sont nécessairement dépendants de la tâche ciblée et ce formalisme ajoute un préfixe explicitant, en partie, la position du mot dans un fragment (*chunk*) de texte annotable. Ces préfixes sont au nombre de trois, "Begin" (B) pour le premier mot du fragment, "Inside" (I) pour tous les mots suivants le premier et "Outside" (O) pour l'ensemble des mots en dehors de tout fragment.

L'inconvénient de l'annotation BIO est lié à sa représentation de l'information sémantique, qui associe un mot à une unique classe sémantique. Si ce format est performant dans le cadre d'une représentation à plat des entités nommées, il n'est pas en mesure de représenter efficacement les entités nommées structurées, en raison de leur structure arborescente. La définition structurée permet une imbrication des entités, qui impliquera qu'un mot puisse appartenir à plusieurs classes sémantiques en même temps. Un exemple illustrant ce principe a été proposé dans la figure 4.4.

Un moyen simple de contourner cette difficulté aurait pu être la mise en place d'une annotation intermédiaire. Il s'agirait de concaténer l'ensemble des labels BIO associé à un mot pour générer un label exploitable.

En s'appuyant sur l'ensemble d'apprentissage d'ETAPE, nous avons considéré que cette solution n'était pas envisageable. Il existe un total de 68 classes sémantiques dans la définition structurée des entités nommées. La mise en place de cette annotation intermédiaire induit

la création de 1 690 classes sémantiques distinctes. Cette nette augmentation du nombre de classes induit nécessairement une faible quantité d'exemples d'apprentissage pour chacune d'elles, avec ainsi une augmentation de la complexité pour la résolution de la tâche.

En 2012, lors de la campagne, les approches exploitaient déjà ce formalisme. Le système vainqueur a contourné la difficulté de la structure arborescente par l'exploitation d'un modèle CRF binaire attribué à chacune des classes sémantiques. La gestion de la structure arborescente s'effectuant en dehors des systèmes CRF par recomposition [Raymond 2013].

Pour les travaux de cette thèse, nous proposons d'aborder ce problème différemment. À partir de l'annotation intermédiaire mentionnée, nous proposons une séparation en trois niveaux d'annotations distincts.

Au sein de la sous-section suivante, nous donnons les détails de cette séparation. Nous fournissons également notre méthode d'implémentation des systèmes exploitant cette triple annotation.

5.2.4 Implémentation en trois niveaux

Nous utilisons, comme point de départ, l'annotation intermédiaire concaténée que nous avons mentionnée dans la sous-section précédente. Comme l'annotation en entités nommées structurées suit une structure arborescente, elle est formée par concaténation des labels dans le sens descendant de l'arborescence, c'est-à-dire de la racine vers les feuilles.

Comme décrit en section 4.2.1, l'annotation en entités nommées structurées est hiérarchique et compositionnelle. Cela signifie que les entités nommées sont décomposables en composants, mais aussi en entités nommées. Les entités obtenues après une décomposition sont à nouveau décomposables en composants et en entités nommées.

Au sein de ce formalisme, tout ce qui est décomposable doit être décomposé. Il n'y a donc pas de limite théorique de profondeur de l'arborescence. Cela dépend des données présentes dans le corpus étudié.

Ce principe de décomposition implique que la racine de l'arborescence soit une entité nommée, tandis que les feuilles en sont des composants. En complément, les annotations entre la racine et les feuilles sont un ensemble organisé de composants et d'entités nommées.

À partir de cette observation, nous mettons en place l'annotation en trois niveaux :

1. La racine de l'arborescence est nécessairement une entité nommée.
2. L'ensemble concaténé des annotations entre la racine et les feuilles est un mélange d'entités et de composants.
3. Les feuilles de l'arborescence sont nécessairement des composants.

Nous fournissons en figure 5.1 l'exemple d'un fragment de texte annoté, la représentation de son arborescence et l'annotation BIO en trois niveaux qui en découle.

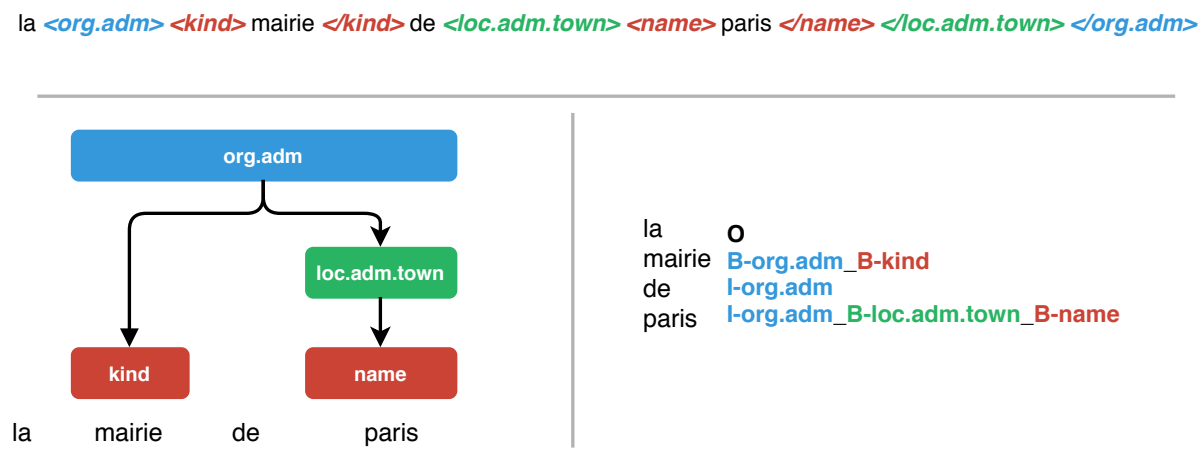

FIGURE 5.1 – Triple représentation d'une séquence, en haut la séquence de mots enrichie en entités nommées, à gauche la représentation de l'arborescence de l'annotation et à droite l'annotation BIO concaténée. En bleu, le premier niveau, en vert, le deuxième et en rouge, le dernier.

L'intérêt principal de l'annotation en trois niveaux réside dans sa capacité à transformer la tâche initiale en trois tâches de labélisation plus simples. Ainsi, pour effectuer la tâche de REN structurée avec le formalisme BIO, nous proposons d'utiliser un système pour chacun des niveaux d'annotation que nous avons définis. Cela signifie qu'un système sera spécialisé pour les types entités (le premier), un autre pour les composants (le troisième) et enfin, un dernier sera responsable de la structuration intermédiaire de l'arborescence (le deuxième). Un segment textuel doit nécessairement passer au travers des trois systèmes de REN pour être totalement annoté. La structure arborescente est reconstruite après l'utilisation de chacun des systèmes en respectant l'ordre des niveaux d'annotation.

Un composant est dépendant du type de l'entité qu'il décompose. Il existe donc un lien entre composants et entités nommées. Nous proposons de le modéliser en réutilisant les prédictions d'un système comme entrée additionnelle des systèmes des niveaux suivants. Les sorties du premier système seront des entrées additionnelles des systèmes des niveaux 2 et 3 et les sorties du deuxième système seront des entrées supplémentaires du troisième.

Nous donnons dans la figure 5.2 une représentation schématique de l'implémentation des systèmes de REN proposée.

À partir de l'annotation et de l'implémentation proposées, nous avons réalisé des expérimentations ayant pour but de répondre au premier objectif de ces travaux. La sous-section suivante est dédiée à ces expérimentations et l'analyse de leurs résultats.

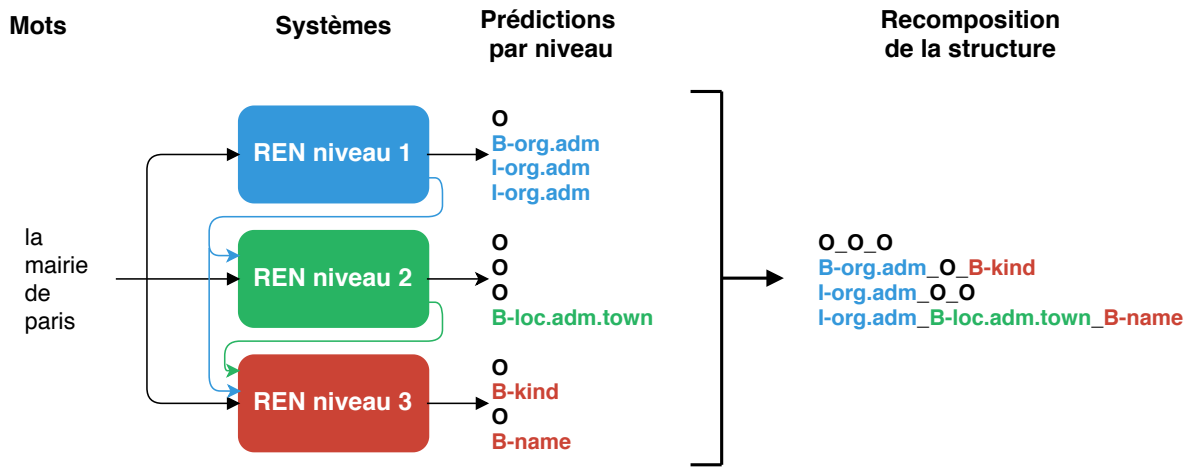

FIGURE 5.2 – Représentation schématique de l'implémentation proposée en 3 niveaux. Chaque système de REN mis en œuvre à sa charge un niveau d'annotation.

5.2.5 Expérimentations et résultats

Les expérimentations que nous menons dans cette partie visent à de mettre à jour les résultats de la campagne ETAPE. Comme la REN dans la parole s'effectue par l'intermédiaire de chaînes de composants, nous avons précédemment identifié quatre systèmes pour nos expérimentations. Le système de RAP HMM-GMM de 2012, le système de RAP HMM-TDNN de 2017, le système de REN CRF de 2012 et le système de REN CNN-bLSTM-CRF de 2017.

Pour réaliser notre étude, nous avons à disposition les scripts d'évaluation utilisés pendant la campagne. Nous pouvons donc mener l'évaluation de nos systèmes dans des conditions identiques à celles-ci.

Nous avons également à disposition les transcriptions automatiques du meilleur système de RAP de la campagne ETAPE. Nous n'effectuerons donc pas la mise en œuvre d'un système HMM-GMM.

À partir de ces 4 systèmes, des scripts d'évaluations et des données autorisées lors de la campagne, nous sommes en mesure de déterminer expérimentalement l'impact de la mise à jour de chacun des composants de RAP et de REN.

Nous détaillons ci-dessous la mise en œuvre des systèmes utilisés, ainsi que les résultats que nous avons obtenus.

Système HMM-TDNN

Pour la mise en œuvre de ce système, le modèle acoustique est appris à l'aide des ensembles de données, ESTER 1, ESTER 2, REPERE et VERA. Une description des trois premiers en-

sembles cités peut être trouvée dans le chapitre 4 de ce manuscrit. VERA [Goryainova et al. 2014] correspond à un ensemble de données recueillies dans le même cadre que projet ETAPE. Par l'addition des parties d'apprentissage de ces ensembles, le modèle acoustique du système HMM-TDNN est appris sur l'équivalent de 220 heures de paroles entièrement transcrites manuellement.

Le modèle de langage de ce système est, quant à lui, appris à l'aide des transcriptions manuelles des quatre ensembles cités. Des données issues d'articles de journaux les complètent. Ces données correspondent à 19 ans d'articles du journal *Le Monde*, les articles du journal *L'Humanité* de 1990 à 2007, ainsi que le corpus français Giga Word.

Système CRF

Nous mettons en œuvre un système CRF à l'aide de l'outil WAPITI [Lavergne et al. 2010]. Il s'agit d'un logiciel largement utilisé en raison de son efficacité à déployer des systèmes de type CRF. Les modèles que nous entraînons s'appuient sur différentes caractéristiques :

— Mots et bigrammes des mots localisés autour des mots cibles sur une fenêtre [-2,+2].
— Préfixes et suffixes localisés autour des mots cibles sur une fenêtre [-2,+2].
— Caractéristiques de types Oui / Non : la présence de chiffres dans le mot, la présence d'une majuscule comme première lettre.
— Plusieurs caractéristiques morphosyntaxiques extraites via l'outil tree-tagger [1].

La mise en œuvre de ce système de REN suit naturellement l'implémentation en trois niveaux que nous proposons. Nous utilisons donc des caractéristiques complémentaires correspondant aux prédictions des modèles CRF précédents.

En appliquant notre annotation en trois niveaux, nous dénombrons 96 labels distincts pour le premier niveau, 187 pour le deuxième niveau et 57 pour le dernier niveau. Ces chiffres sont obtenus par l'utilisation de l'ensemble de données ETAPE dont l'apprentissage est augmenté par les données QUÆRO, comme exploités durant la campagne.

Nous effectuons l'apprentissage de nos modèles CRF à l'aide de l'algorithme rprop [Riedmiller et Braun 1993] pour un maximum de 40 époques.

Système NeuroNLP2

Le système NeuroNLP2 est une combinaison d'une couche CNN, de deux couches bLSTM et d'un CRF. Pour la couche CNN, nous utilisons 30 filtres de taille 3 (*kernel size*). Les couches bLSTM possèdent 200 unités et un dropout de 0,5 est appliqué entre chaque couche. Ce dropout est également appliqué en entrée de la couche CNN. La taille des minibatchs est de 10 et le taux d'apprentissage initial est 0,001.

1. https ://cis.uni-muenchen.de/ schmid/tools/TreeTagger/

Nous entraînons le système sur 100 époques et sélectionnons le modèle final en fonction des performances sur l'ensemble de développement d'ETAPE.

NeuroNLP2 effectue l'extraction de représentation de caractères via sa couche CNN. Cette représentation est concaténée à une représentation de mots, avant les couches bLSTM. Nous apprenons notre propre dictionnaire de représentations de mots en amont de l'apprentissage de NeuroNLP2. Puis, nous utilisons systématiquement le même dictionnaire pour toutes nos expérimentations. Nous apprenons ce dictionnaire à l'aide de l'outil word2vec et d'un large ensemble de données textuelles de deux milliards de mots. Cet ensemble est composé de 3,5 millions de mots uniques.

Nous utilisons des étiquettes morphosyntaxiques (Part-Of-Speech, POS) obtenues à l'aide de l'outil Macaon [Nasr et al. 2011], comme données d'entrées complémentaires.

Pour effectuer l'apprentissage NeuroNLP2, nous utilisons les mêmes données que celles employées pour notre CRF (ETAPE et QUÆRO). Nous exploitons également notre implémentation en trois niveaux. L'injection des prédictions précédentes s'effectue par l'intermédiaire de leur représentation vectorielle directement issue de la couche CRF. Nous concaténons les représentations intermédiaires aux représentations de caractères et de mots en entrée des couches bLSTM.

Résultats

Dans le cadre de nos expérimentations, nous choisissons comme point de comparaison le meilleur résultat obtenu durant la campagne. Le système complet associé sera dénommé (*Sys 0*). Les résultats de ce système sont exprimés en SER sur l'ensemble de test d'ETAPE. Nos résultats seront aussi exprimés dans les mêmes conditions.

Nous nommons RAP_{2012} et RAP_{2017} respectivement, les systèmes de RAP lors de la campagne ETAPE et les systèmes à l'état de l'art au commencement de cette thèse. Nous faisons de même avec les systèmes de REN qui seront dénommés REN_{2012} et REN_{2017}.

La première expérimentation réalisée a pour but de vérifier la viabilité de l'implémentation en trois niveaux proposée. Pour ce faire, nous effectuons l'apprentissage de deux systèmes CRF distincts qui seront appliqués sur le système RAP_{2012}.

Le premier système CRF, dénommé (*Sys A*), est appris sans notre implémentation en trois niveaux. Afin d'effectuer la REN structurée, nous utilisons l'annotation intermédiaire évoquée en section 5.2.3. Le second système (*Sys B*), quant à lui, bénéficie de l'implémentation en trois niveaux.

Nous fournissons dans la table 5.1, les résultats des trois systèmes évoqués.

En comparant les systèmes *A* et *B*, nous observons une amélioration de 9,9 points de SER par l'utilisation de notre approche en 3 niveaux. Ces résultats nous indiquent la viabilité de notre approche en trois niveaux. De plus, nous pouvons atteindre des performances similaires

Système	**SER**
(*Sys 0*) Référence ETAPE 2012	59,3
(*Sys A*) RAP_{2012} / REN_{2012}	69,4
(*Sys B*) RAP_{2012} / REN_{2012} - 3 niveaux	59,5

TABLE 5.1 – Résultats expérimentaux de l'implémentation en trois niveaux, exprimés en SER pour l'ensemble de test d'ETAPE.

aux meilleurs systèmes de la campagne, par l'utilisation de 3 modèles CRF au lieu de 68 CRF binaires. L'ensemble des expérimentations suivantes s'effectuera donc avec notre implémentation en trois niveaux.

Elles consistent cette fois à quantifier les performances. Nous considérons donc 4 systèmes différents, le système (*Sys B*) déjà expérimenté, un système suite à la mise à jour du composant de REN (*Sys C*), un système suite à la mise à jour du composant de RAP (*Sys D*) et enfin un système suite à la mise à jour des deux composants (*Sys E*).

Système	**SER**
(*Sys B*) RAP_{2012} / REN_{2012}	59,5
(*Sys C*) RAP_{2017} / REN_{2012}	56,1
(*Sys D*) RAP_{2012} / REN_{2017}	55,0
(*Sys E*) RAP_{2017} / REN_{2017}	**51,1**

TABLE 5.2 – Résultats expérimentaux des mises à jour des chaînes de composants, exprimés en SER pour l'ensemble de test d'ETAPE.

Nos expérimentations montrent, par comparaison des systèmes *B* et *C*, que la nouvelle technologie de reconnaissance de la parole apporte un gain de 3,4 points de SER. Il est explicable par l'amélioration de la qualité des transcriptions automatiques qui sont fournies au composant de REN. En effet, le taux d'erreur sur les mots (WER) du système RAP_{2012} était de 21,8 % sur le test d'ETAPE [GALIBERT, LEIXA et al. 2014], tandis que lors de ces expérimentations, nous avons mesuré un nouveau WER de 16,5 % (RAP_{2017}).

La mise à jour du composant de REN seule nous permet de mesurer, par comparaison des systèmes *B* et *D*, un gain de 4,5 points de SER. Enfin, en comparant les systèmes *B* et *E*, nous observons que l'utilisation conjointe des deux systèmes mis à jour permet une amélioration globale significative de 8,4 points de SER.

Par la mise à jour de chacun des composants et l'utilisation de l'implémentation en trois niveaux que nous avons proposés, nous sommes en mesure d'obtenir 51,1 points de SER contre 59,3 points en 2012.

Nous avons effectué la mise à jour des résultats de la campagne ETAPE, qui constituait le premier objectif de nos travaux autour de la reconnaissance des entités nommées. L'objectif sui-

vant de nos travaux consiste en la mise œuvre d'un premier système de bout en bout dédié à la REN dans la parole. Il s'agit d'un unique système capable d'effectuer simultanément la reconnaissance de la parole et la reconnaissance des entités nommées, plutôt que par l'intermédiaire d'une chaîne de composants. Dans la section suivante, nous aborderons ce premier système mis en œuvre dans le cadre d'une tâche de REN simplifiée par rapport à la REN structurée.

5.3 REN simplifiée de bout en bout

Dans cette section, l'objectif est de vérifier la capacité d'un système neuronal à apprendre un unique modèle effectuant les tâches de RAP et de REN conjointement. Une approche de bout en bout implique l'optimisation d'un unique modèle contre deux dans le cadre d'une chaîne de composants. Son intérêt réside dans sa capacité à se passer de la transcription textuelle intermédiaire entre les composants. Supprimant ainsi le bruit, issu de la reconnaissance de la parole, qui été présent en entrée du composant de REN.

Des approches de ce type ont déjà été mis en œuvre avec succès pour la tâche de reconnaissance de la parole [Hannun et al. 2014; Amodei et al. 2016; Y. Zhang et al. 2016]. Nous souhaitons étendre la capacité des systèmes de RAP de bout en bout pour leur permettre d'effectuer la REN directement depuis la parole. Dans cette thèse, nous nous concentrerons sur le système de RAP DeepSpeech 2 [Amodei et al. 2016] dont l'implémentation est mise à disposition par les auteurs.

Nous avons jusque là réalisé la reconnaissance des entités nommées structurées, qui reste en soit une tâche complexe. Pour notre premier système de bout en bout, nous souhaitons réaliser une tâche de REN plus simple.

Nous avons à notre disposition des données annotées en entités nommées issues des ensembles ESTER 1 / QUÆRO, ESTER 2 (partie développement) et ETAPE. Rappelons que les données QUÆRO sont une réannotation en EN des transcriptions manuelles d'ESTER 1.

Nous souhaitons maximiser notre quantité de données d'apprentissage. Ainsi, nous récupérons toutes les données annotées et nous envisageons d'utiliser l'ensemble ESTER 1 couplé à ESTER 2 et ETAPE. Toutefois, leurs schémas d'annotation ne sont pas entièrement compatibles (plus de détails sont données dans les sections 4.1.3 et 4.2.1). La simplification de l'annotation en entités nommées nous permet de bénéficier de toutes nos données en les rendant suffisamment simples pour être compatibles.

Nous définissons ainsi une tâche de REN simplifiée et nous donnons les transformations effectuées dans la sous-section suivante.

5.3.1 Définition de la tâche simplifiée

Nos données sont annotées selon les formalismes ESTER et QUÆRO. Ces formalismes utilisent une typologie hiérarchique définissant un grand nombre de types d'EN. Notre première simplification consiste à supprimer cette hiérarchie pour conserver des entités plus génériques, comme c'est notamment le cas dans certaines définitions, par exemple MUC 7 [2].

Le formalisme QUÆRO nécessite la décomposition des EN, ce qui constitue la plus grande source de complexité de l'annotation. Nous proposons de supprimer l'aspect compositionnel en supprimant la notion de composants, mais aussi en supprimant l'imbrication des types EN. Pour supprimer cette imbrication, nous avons décidé de ne conserver que les types entités nommés de plus bas niveau, soit les plus proches des mots.

En effectuant ces transformations, nous définissons une tâche de REN non structurée basée sur 8 catégories d'EN : "*time*", "*func*", "*loc*", "*org*", "*pers*", "*prod*", "*amount*", "*event*". Comme les formalismes ESTER (voir section 4.1.3) et QUÆRO (voir section 4.2.1) sont proches, ils deviennent additionnables par application de nos transformations.

Nous donnons un exemple de séquence et sa version simplifiée en figure 5.3.

Séquence complète	<pers.ind> <title> Monsieur le <func.ind> <kind> ministre </kind> </func.ind> </title> </pers.ind>
Séquence simplifié	Monsieur le <func> ministre </func>

FIGURE 5.3 – Exemple de séquence enrichie en entités nommées. Une séquence complète au-dessus, sa version après application de nos transformations en dessous.

Après avoir défini notre tâche simplifiée, nous présentons dans la sous-section suivante le système nous servant de point de départ.

5.3.2 Système DeepSpeech 2

DeepSpeech 2 est un système de reconnaissance de la parole de bout en bout. Contrairement aux systèmes traditionnels, basés sur une combinaison de modèle acoustique et de langue, celui-ci n'apprend qu'un unique modèle entièrement dédié à la reconnaissance de la parole.

Ce système est un empilement de deux couches CNN, cinq bLSTM, d'une couche linéaire et d'une couche softmax. Nous proposons une représentation de ce système dans la figure 5.4.

L'aspect que nous souhaitons exploiter de ce système est sa fonction de coût. Il s'agit de la fonction de classification temporelle connectionniste (*Connectionist Temporal Classification*,

2. https ://www-nlpir.nist.gov/related_project/muc/proceedings/ne_task.html

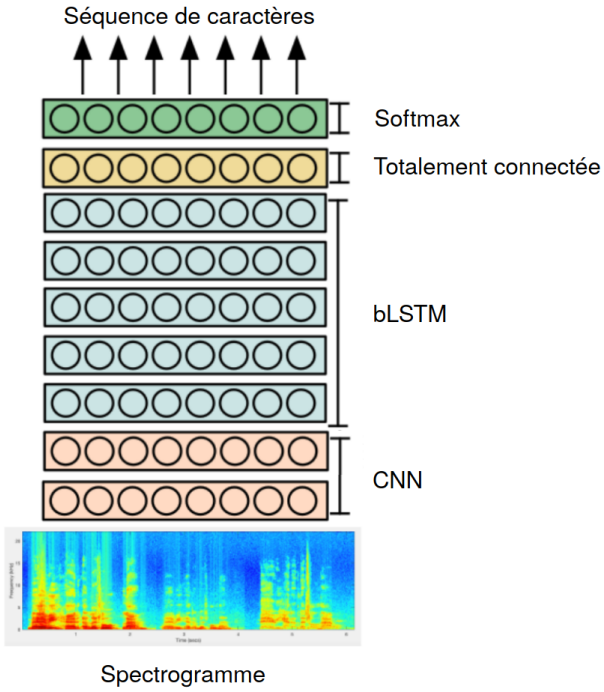

FIGURE 5.4 – Représentation du système neuronale DeepSpeech 2.

CTC) [GRAVES, FERNÁNDEZ et al. 2006].

L'intérêt de cette fonction réside dans sa capacité à permettre l'apprentissage de l'alignement entre un spectrogramme de parole et une transcription manuelle associée.

Nous décrivons dans la sous-section suivante comment nous proposons d'exploiter les propriétés de cette fonction de coût pour réaliser la tâche de REN dans la parole.

5.3.3 Alignement de parole et de transcriptions enrichies

Afin de réaliser la tâche de REN dans la parole de bout en bout, nous proposons d'enrichir les transcriptions manuelles de la parole utilisées. Nous y ajoutons l'information des entités nommées, permettant ainsi à un système d'apprendre l'alignement entre un spectrogramme de parole et sa transcription manuelle enrichie. Nous proposons de représenter les EN par leurs frontières de début et de fin encadrant les mots les composants.

Toutefois, la fonction de coût CTC fonctionne à l'échelle des caractères. Nous proposons de ne pas utiliser un label complet comme "<*func*>", mais plutôt d'exploiter des caractères uniques entièrement dédiés à la représentation des EN. Celà correspond à un caractère unique pour chacun des labels d'entrées et nous réduisons chacun des labels de sorties à un unique caractère de fin. Nous ajoutons des caractères prédictibles dédiés aux frontières des EN, dont l'alignement doit être appris par le système et la fonction CTC.

En complément, la fonction CTC donne la même importance à chaque caractère émis. Nous proposons ainsi une modification des transcriptions pour contraindre cette fonction à concentrer l'apprentissage sur les frontières d'EN et leurs valeurs. Cette modification consiste à remplacer l'ensemble du contexte (les mots à l'extérieur d'une entité), par une simple étoile. Nous appelons cette modification "mode étoile". En remplaçant par une simple étoile, nous réduisons le contexte à un unique caractère là où il était représenté par plusieurs. De cette manière, les types entités nommés et leurs valeurs représentent désormais la principale source de caractères devant être émis.

Nous donnons ci-dessous, un exemple avec chacune des annotations mentionnées dans cette sous-section.

Séquence de mots	le sculpteur césar est mort hier à paris à l'âge de soixante dix sept ans
Séquence de mots enrichie	le sculpteur <pers césar > est mort <time hier > à <loc paris > à l'âge de <amount soixante dix sept ans >
Séquence de mots enrichie encodée	le sculpteur [césar > est mort # hier > à $ paris > à l'âge de % soixante dix sept ans >
Mode étoile	* [césar > * # hier > * $ paris > * % soixante dix sept ans >

Figure 5.5 – Exemple d'enrichissement en entités nommées d'une séquence.

5.3.4 Expérimentations et résultats

À partir de DeepSpeech 2 et notre proposition de modification des séquences à produire, nous réalisons des expérimentations pour nous assurer de la viabilité de notre approche. Comme nous réalisons ces expérimentations dans le cadre d'une tâche simplifiée, nous ne pouvons utiliser l'évaluation de la campagne ETAPE. Nous effectuons donc notre propre évaluation avec la métrique F-mesure [Van Rijsbergen 1974], que nous avons présentée précédemment (voir section 3.4.1). Également, comme il s'agit de notre première implémentation de bout en bout, nous exploitons nos données de manière à maximiser leurs quantités. Nous donnons davantage de détails sur l'évaluation du système et l'exploitation des données ci-dessous. Nous détaillons ensuite nos expérimentations et les résultats obtenus.

Ensembles de données

Dans le cadre de nos expérimentations, nous souhaitons disposer du système le plus performant possible. Nous souhaitons donc naturellement maximiser la quantité de données à notre

disposition. Pour l'aspect entités nommées, nous exploitons le maximum de données manuellement annotées qui ne représentent que près de 140 heures de parole.

Nous avons la possibilité d'augmenter notre quantité de données pour l'aspect parole. Même si elles ne sont pas annotées manuellement en entités nommées, nous récupérons les données des ensembles EPAC, REPERE et ESTER 2 (apprentissage / test) qui sont manuellement transcrites.

Nous avons donc à disposition ESTER 1 / QUÆRO, ESTER 2 (développement) et ETAPE manuellement annotée pour la tâche de REN et manuellement transcrit. Nous avons aussi, ESTER 2 (apprentissage / test), REPERE et EPAC comme données manuellement transcrites.

Lorsque nous effectuons l'utilisation conjointe de différents ensembles de données, nous utilisons systématiquement la répartition d'origine des ensembles. Pour nos expériences, nous constituons donc un ensemble de près de 290 heures d'apprentissage, de 40 heures de test et de 40 heures de développement. Parmi ces données, l'annotation manuelle en EN est réalisée sur 90 heures d'apprentissage (ESTER 1, ETAPE), 16 heures de test (ESTER 1, ETAPE) et 28 heures de développement (ESTER 1, ESTER 2, ETAPE).

Toujours dans l'optique de maximiser notre quantité de données, nous proposons pour ces travaux d'effectuer une augmentation automatique des données d'EN. Pour ce faire, nous proposons d'apprendre le système NeuroNLP2 pour effectuer la tâche de REN simplifiée dans les transcriptions de la parole. Nous effectuons sa mise en œuvre de la même façon que présentée en section 5.2.

Au vu des simplifications appliquées, nous n'utilisons pas notre approche en trois niveaux. Nous effectuons l'apprentissage d'un modèle NeuroNLP2 à partir des annotations manuelles d'EN. Nous réalisons ensuite l'annotation automatique des transcriptions manuelles des données de paroles à notre disposition.

Cette augmentation automatique est portée sur les ensembles d'apprentissages uniquement, permettant ainsi d'atteindre 290 heures de paroles annotées en entités nommées automatiquement et manuellement.

À partir du système DeepSpeech 2, de notre proposition d'enrichissement des transcriptions, de nos données et de la métrique d'évaluation choisis, nous menons les expérimentations visant à vérifier la viabilité de notre approche.

Expériences et résultats

Dans le cadre de nos expérimentations, nous paramétrons l'architecture de DeepSpeech 2 avec 2 couches CNN, 5 couches bLSTM avec normalisation des batchs, 1 couche linéaire et enfin la couche de sortie softmax. Pour les couches CNN, nous utilisons 32 filtres et nous paramétrons la taille des couches bLSTM à 800 unités. Afin de tirer bénéfice de toutes nos données, et construire un modèle robuste, nous proposons d'effectuer l'entraînement de notre

système par transfert d'apprentissage.

Nous réalisons d'abord l'apprentissage d'un modèle de transcription de la parole, puis un second dédié à la reconnaissance des entités nommées. Pour ce faire, nous conservons le modèle obtenu lors du premier apprentissage, hormis la couche de sortie softmax qui sera réinitialisée. Cette réinitialisation est nécessaire puisque pour la REN, nous ajoutons des caractères prédictibles correspondant aux types des entités nommées. Ainsi, la couche de sorties réapprend entièrement la représentation de chacun des caractères. Nous symbolisons ce principe par une flèche : $\rightarrow$.

Pour le premier apprentissage, nous exploitons uniquement les données de parole, tandis que pour le second, nous utilisons les données enrichies en EN.

Nous effectuons l'apprentissage des deux systèmes de REN dans la parole. Le premier sans l'emploi du mode étoile ($RAP \rightarrow REN$) et le second avec l'emploi de ce mode ($RAP \rightarrow REN^*$).

L'évaluation des sorties est réalisée grâce à un alignement des références et des sorties de notre système. Nous exploitons l'outil sclite [3], couramment utilisé en reconnaissance de la parole. Avant l'alignement, nous effectuons le filtrage des sorties pour ne conserver que les informations que nous souhaitons évaluer.

Nous donnons en table 5.3 les résultats correspondant à l'évaluation de ces deux systèmes pour la reconnaissance de type entité uniquement.

Système	**Ensemble**	**Précision**	**Rappel**	**F-mesure**
$RAP \rightarrow REN$	développement	**0,85**	0,57	0,68
$RAP \rightarrow REN$	test	**0,83**	0,52	0,64
$RAP \rightarrow REN^*$	développement	0,75	**0,65**	**0,71**
$RAP \rightarrow REN^*$	test	0,82	**0,57**	**0,67**

TABLE 5.3 – Résultats exprimés en précision, rappel et F-mesure pour la détection de type entités nommées.

Ces résultats tendent à montrer la viabilité de notre approche pour la REN dans la parole de bout en bout. En effet, nous obtenons des scores de F-mesure supérieurs à 0,6 pour la classification de 8 catégories d'EN. Notre système est donc en mesure de modéliser une information pertinente.

De plus, l'évaluation sur l'ensemble de développement définit les performances atteignables lorsque nous optimisons un modèle. La comparaison des performances sur les ensembles de développement et de test montre une perte raisonnable, renforçant la pertinence de notre approche. Plus cette perte est minime, plus le système mis en œuvre aura modélisé une information généralisable.

Nous pouvons aussi voir l'intérêt du mode étoile proposé qui donne une amélioration glo-

3. http ://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm

bale des F-scores. Nous remarquons qu'elle est effective grâce à une amélioration notable du rappel. Il est ainsi possible de détecter davantage de types EN en concentrant l'apprentissage de la fonction de coût sur les types et leurs valeurs.

Toutefois, la précision chute en cas d'utilisation de ce mode. Cela peut s'expliquer par la perte du contexte qui semble être un élément nécessaire à la classification correcte d'une frontière détectée.

En complément, nous effectuons l'évaluation des deux systèmes pour la reconnaissance des types EN et de leur valeur. Pour ce faire, nous effectuons à nouveau un alignement avec l'outil sclite. Nous donnons en table 5.4 les résultats exprimés en Précision, Rappel et F-mesure pour la détection conjointe de type EN et de leurs valeurs.

Système	Ensemble	Précision	Rappel	F-mesure
$RAP \rightarrow REN$	développement	**0,64**	0,45	**0,53**
$RAP \rightarrow REN$	test	**0,55**	0,36	**0,44**
$RAP \rightarrow REN*$	développement	0,57	**0,47**	0,52
$RAP \rightarrow REN*$	test	0,47	**0,38**	0,42

Table 5.4 – Résultats exprimés en Précision, Rappel et F-mesure pour la détection de type entités nommées et leurs valeurs.

Sur ces résultats, nous observons aussi un gain en rappel et une perte en précision par l'utilisation du mode étoile. Cependant, le gain en rappel ne permet pas de compenser la perte en précision, rendant le mode étoile moins intéressant.

L'analyse de ces résultats semble indiquer que l'alignement réalisé pour l'évaluation implique une certaine rigidité. Une réponse du système est considérée correcte si le type d'EN et la valeur correspondent parfaitement à la référence. Des erreurs minimes de frontières impliquent une réponse fausse, par exemple "<loc> à Paris </loc>" est faux si la référence est "à <loc> Paris </loc>". Il en est de même pour des erreurs mineures de transcription ne modifiant pas fondamentalement le sens du segment.

Ainsi, ces résultats, bien que très en dessous de ceux de la reconnaissance de type EN seul, sont un nouvel indice de la viabilité de notre approche de bout en bout. Dans la suite des expérimentations, nous évaluerons les performances de notre système uniquement sur l'ensemble de test.

Nous proposons désormais de maximiser les performances de notre approche en exploitant la totalité de nos données annotées manuellement et automatiquement en EN. Nous apprenons ainsi deux systèmes supplémentaires avec les données augmentés en suivant également l'apprentissage par transfert. Le premier système bénéficie uniquement de l'augmentation de données et est nommé $RAP \rightarrow REN+$, tandis que le second bénéficie, en plus, du mode étoile (nommé $RAP \rightarrow REN+*$). Nous reportons les résultats de nos évaluations dans la table 5.5.

L'augmentation de données imparfaite que nous réalisons permet une amélioration globale

Système	Détection	Précision	Rappel	F-mesure
$RAP \rightarrow REN+$	type	**0,82**	0,57	0,67
$RAP \rightarrow REN+^*$	type	0,76	**0,63**	**0,69**
$RAP \rightarrow REN+$	type et valeur	**0,55**	0,40	0,46
$RAP \rightarrow REN+^*$	type et valeur	0,49	**0,41**	**0,47**

Table 5.5 – Résultats exprimés en Précision, Rappel et F-mesure pour la REN sur l'ensemble de test. Comparaison de l'approche augmentée et de l'approche augmentée en mode étoile.

des résultats en comparaison avec les tables 5.3 et 5.4. Elle est aussi bénéfique pour le mode étoile qui permet d'atteindre nos meilleurs résultats pour la REN simplifiée de bout en bout.

Afin de compléter ces résultats, nous proposons d'effectuer une comparaison à une approche traditionnelle par chaîne de composants. Nous détaillons dans la section suivante la chaîne que nous mettons en place.

Comparaison à une chaîne de composants

Nous avons vu en section 5.2 qu'il est préférable d'exploiter des composants à l'état de l'art. Nous utilisons donc naturellement le système NeuroNLP2 comme composant de REN. Nous le mettons en œuvre dans la même configuration que dans la sous-section 5.2.5, y compris pour l'exploitation des informations d'étiquettes morphosyntaxiques.

Nous utilisons comme système de RAP, le système DeepSpeech 2 obtenu lors de la première étape de l'apprentissage par transfert évoqué précédemment. Comme la qualité des transcriptions automatique a une incidence au sein d'une chaîne de composants, nous souhaitons la maximiser.

Le système DeepSpeech 2 peut effectuer la transcription de l'audio de manière totalement neuronale (*greedy decoding*), c'est-à-dire en utilisant un algorithme glouton qui selectionne la sortie la plus probable à chaque temps t pour construire la séquence de sortie. Il peut également tirer bénéfice d'un modèle de langage (*Beam Search decoding*, voir section 2.4.2). Son utilisation implique l'amélioration de la qualité des transcriptions automatiques, puisque les sorties initiales seront modifiées pour être en cohérence avec le modèle de langage. DeepSpeech 2 effectue une prédiction caractère par caractère. Ainsi, le modèle de langage est en mesure de corriger des émissions de mots inconnus ou mal orthographiés.

Nous effectuons l'évaluation des transcriptions automatiques sur les ensembles de test manuellement transcrits, indépendamment de l'annotation en entités nommées. Nous utilisons, comme métrique, le taux d'erreur sur les mots (WER) et le taux d'erreur sur les caractères. Cette deuxième métrique est identique au WER, hormis son application à l'échelle des caractères au lieu des mots. Ce système obtient un score de 19,95 % de WER et 7,68 % de taux d'erreur sur les caractères, pour les ensembles de tests conjoints de 40 heures. Nous rappelons que cet en-

semble de tests conjoint correspond à la réunion des ensembles de test manuellement transcrits d'ESTER 1 et 2, d'ETAPE, d'EPAC et de REPERE.

Nous effectuons ensuite la tâche de REN simplifiée sur les transcriptions automatiques de test. Nous donnons dans la table 5.6, les résultats de la chaîne de composants pour cette tâche.

Système	Détection	Précision	Rappel	F-mesure
DS2 / NeuroNLP2	type	0,74	0,58	0,65
"*RAP* → *REN*+*"	type	**0,76**	**0,63**	**0,69**
DS2 / NeuroNLP2	type et valeur	**0,57**	**0,45**	**0,50**
"*RAP* → *REN*+*"	type et valeur	0,49	0,41	0,47

Table 5.6 – Résultats exprimés en Précision, Rappel et F-mesure pour la REN simplifiée avec une chaîne de composants composée de DeepSpeech 2 (DS2) et NeuroNLP2. Les résultats encadrés par des guillemets sont reportés de la table 5.5.

Ces résultats peuvent êtres comparés à ceux obtenus sur l'ensemble de test des tables 5.3, 5.4 et 5.5. Ils nous montrent que l'approche de bout en bout est performante pour la reconnaissance des types entités nommées, tandis que l'approche par chaîne de composants reste préférable pour la reconnaissance des types et des valeurs. Il s'agit en soit de résultats appuyant la viabilité de notre approche de bout en bout pour la tâche de REN simplifiée.

Nous souhaitons désormais confirmer sa viabilité dans le cadre de tâches plus complexes. Nous définition ainsi un nouvel objectif qui consiste à mettre en œuvre notre approche de bout en bout pour la REN structurée. Les travaux menant à la réalisation de cet objectif sont l'objet de la section suivante.

5.4 REN structurée de bout en bout

Dans cette section, nous reprenons le contexte de la campagne ETAPE. Nous souhaitons rendre comparables les expérimentations de cette section avec celles présentées en section 5.2. Nous utiliserons donc la même métrique d'évaluation, le taux d'erreur sur les champs (SER). Nous exploiterons aussi les mêmes ensembles de données, à savoir, ESTER 1 / QUÆRO, ESTER 2, EPAC, ETAPE, REPERE et VERA. L'intégralité de ces données sera utile pour l'aspect reconnaissance de la parole, tandis que seulement ETAPE et QUÆRO seront exploités pour la reconnaissance des entités nommées. Nous les utiliserons pour l'apprentissage du système DeepSpeech 2 selon une mise en œuvre de bout en bout.

5.4.1 Mise en œuvre de DeepSpeech 2

Nous exploitons ce système avec une architecture identique à celle présentée en sous-section 5.3.2. Il s'agit donc d'une architecture basée sur 2 couches CNN à 30 filtres, 5 couches

bLSTM de 800 unités, une couche linéaire et une couche softmax.

Pour l'approche de bout en bout, nous réalisons l'enrichissement des transcriptions manuelles de la même manière que présenté en sous-section 5.3.3. Cependant, nous exploitons désormais l'annotation structurée à la place de l'annotation simplifiée.

Avec cette mise en œuvre, nous exploiterons à nouveau un transfert d'apprentissage pour tirer bénéfice de toutes nos données. Nous proposons toutefois une extension de l'apprentissage par transfert que nous détaillerons dans la section suivante.

5.4.2 Extension du transfert d'apprentissage

Nous avons précédemment exploité, par transfert, l'ensemble des données de parole, pour l'apprentissage d'un système de RAP, avant l'apprentissage du système de REN visé. L'objectif était de pallier le manque de données pour l'apprentissage de notre système final.

Pour mettre en place un apprentissage de ce type, il était nécessaire de conserver l'intégralité des paramètres du système de RAP appris, puis d'effectuer l'apprentissage du système de REN par transfert. Seule la couche de sortie softmax était réinitialisée pour permettre la prise en compte des caractères représentant les types d'entités nommées.

Ici, nous proposons d'étendre l'apprentissage par transfert en séparant la tâche de REN en deux étapes. Nous rappelons que dans le cadre de l'annotation structurée, une entité nommée doit être décomposée. Il existe donc un lien entre les types entités et les composants.

Nous proposons de différencier l'apprentissage des types d'entités nommées et des composants. Il s'agit donc conserver l'apprentissage du système de RAP initial, puis d'effectuer l'apprentissage d'un système de REN dédié aux types entités et enfin l'apprentissage d'un système de REN dédié aux types et aux composants. Le transfert effectué entre les deux systèmes de REN serait identique au transfert effectué entre les tâches de RAP et de REN.

En procédant de cette manière avec la tâche de REN structurée, nous espérons faciliter son apprentissage en exploitant le lien existant entre les types EN et les composants. L'intuition étant qu'un système déjà optimisé sur les types EN devrait être en mesure d'apprendre plus facilement une représentation des composants plutôt qu'un système s'optimisant à la fois sur les types et les composants.

À partir de DeepSpeech 2 et de nos données, nous menons des expérimentations visant à confirmer la viabilité de l'approche de bout en bout et de l'extension proposée.

5.4.3 Expérimentations et résultats

La première partie de nos expérimentations vise à valider notre extension de l'apprentissage par transfert. Nous apprenons donc deux systèmes. Un premier directement dédié à une tâche de REN structurée complète (REN_{struct}). Un second dédié à une tâche optimisée en deux

temps, d'abord la reconnaissance de types (REN_{types}), puis la reconnaissance des types et des composants, correspondant à la tâche de REN structurée complète (REN_{struct}).

Dans les deux cas, nous utilisons comme point de départ un système de RAP appris sur toutes nos données de parole.

Nous donnons dans la table 5.7 les résultats de ces deux systèmes exprimés en SER sur l'ensemble de test de la campagne ETAPE.

Système	**SER**
$RAP \rightarrow REN_{struct}$	62,9
$RAP \rightarrow REN_{types} \rightarrow REN_{struct}$	**61,9**

TABLE 5.7 – Résultats exprimés en SER pour l'approche de bout en bout avec et sans utilisation de l'extension de l'apprentissage par transfert sur l'ensemble de test ETAPE.

Par une amélioration d'un point de SER, ces résultats confirment l'intérêt d'une étape intermédiaire répartissant la difficulté d'apprentissage sur deux tâches distinctes.

Pour exploiter pleinement les modèles appris, nous envisageons de tirer parti du décodage beam search proposé par DeepSpeech 2. Nous apprenons ainsi deux modèles de langue pour la tâche de REN structurée, un premier trigramme et un second quadri gramme. Pour prendre en compte les concepts d'EN au sein des modèles de langue, nous les apprenons sur les transcriptions manuelles enrichies. En termes de données, nous utilisons les ensembles d'apprentissages d'ETAPE et de QUÆRO.

Nous donnons dans la table 5.8 les résultats du décodage de nos modèles en utilisant les deux modèles de langues.

Système	**ML**	**SER**
$RAP \rightarrow REN_{struct}$	3-gramme	57,9
$RAP \rightarrow REN_{types} \rightarrow REN_{struct}$	3-gramme	57,5
$RAP \rightarrow REN_{struct}$	4-gramme	57,3
$RAP \rightarrow REN_{types} \rightarrow REN_{struct}$	4-gramme	**56,9**

TABLE 5.8 – Résultats exprimés en SER pour l'approche de bout en bout par utilisation de modèle de langage sur l'ensemble de test d'ETAPE.

Au regard du gain obtenu, nous voyons l'utilité des modèles de langue. Nous observons aussi que l'apprentissage par transfert étendu conserve son intérêt dans le cas des deux modèles de langue. Enfin, le meilleur score SER obtenu pour une approche de bout en bout avec un modèle de langage 4-gramme est de 56,9 %.

En complément de ces expérimentations, nous proposons d'effectuer une augmentation de données automatique de façon similaire aux expérimentations de la section 5.3.4. Cette précédente augmentation de données avait un impact positif sur la reconnaissance des entités

nommées. Nous proposons, ici, de bénéficier de l'approche en trois étapes proposée pour la reconnaissance d'entités nommées structurées afin d'effectuer l'annotation automatique de nos données audio ne possédant pas d'annotation structurée manuelle. Nous nommons $REN+$ les apprentissages réalisés avec l'ensemble de données regroupant les annotations manuellement et automatiques.

Nous reportons dans la table 5.9, les résultats des expérimentations réalisées à partir de l'augmentation automatique de données. Nous proposons de bénéficier également des modèles de langages déjà appris et d'exploiter à nouveau notre méthode d'apprentissage par transferts successifs en séparant la tâche de classification en type de la tâche de décomposition des EN.

Système	**ML**	**SER**
$RAP \rightarrow REN+_{struct}$	X	57,9
$RAP \rightarrow REN+_{struct}$	3-gramme	53,5
$RAP \rightarrow REN+_{struct}$	4-gramme	53,1
$RAP \rightarrow REN+_{types} \rightarrow REN+_{struct}$	X	56,4
$RAP \rightarrow REN+_{types} \rightarrow REN+_{struct}$	3-gramme	52,3
$RAP \rightarrow REN+_{types} \rightarrow REN+_{struct}$	4-gramme	**51,9**

TABLE 5.9 – Résultats exprimés en SER pour l'approche de bout en bout par utilisation de notre augmentation de données automatique sur l'ensemble de test d'ETAPE.

Par comparaison des résultats des tables 5.9 et 5.8, nous pouvons observer l'apport positif systématique de l'augmentation automatique de l'annotation sémantique que nous proposons.

Ces résultats montrent aussi que nous conservons l'intérêt de l'apprentissage par transferts successifs en séparant l'apprentissage de la typologie des entités nommées et l'apprentissage de leurs décompositions. Les modèles de langues 3-grammes et 4-grammes maintiennent également leur apport pour les performances finales du système de bout en bout.

Enfin, la combinaison de notre approche par transferts successifs associée à l'augmentation de données automatique proposée nous permet d'atteindre nos meilleures performances pour une approche de bout en bout dans le cadre de cette thèse. Ce dernier système bénéficie également du modèle de langage 4-gramme et permet d'atteindre un score SER de 51,9 %.

5.4.4 Comparaison avec l'approche en chaînes de composants

Dans cette section, nous effectuons une brève comparaison des résultats obtenus dans le cadre de la campagne ETAPE. Nous récupérons donc les résultats issus de la campagne, de nos travaux de mises à jour des chaînes de composants et enfin de notre mise en œuvre d'un système de bout en bout. Nous les reportons dans la table 5.10.

La comparaison de ces résultats tend à confirmer la viabilité de notre approche de bout en bout. Même si elle n'est pas en mesure de surpasser une approche traditionnelle actualisée avec

Système	SER
Sys 0. Référence ETAPE 2012	59,3
$RAP \rightarrow REN+_{types} \rightarrow REN+_{struct}$ (4-gramme)	51,9
Sys E. RAP_{2017} / REN_{2017}	**51,1**

TABLE 5.10 – Résultats reportés de notre référence ETAPE, meilleure chaîne de composants et meilleur système de bout en bout. Exprimés en SER sur l'ensemble de tests d'ETAPE.

les systèmes neuronaux, elle reste toutefois intéressante. En obtenant un score SER de 51,9 %, notre premier système de bout en bout atteint de meilleures performances que les résultats initiaux de la campagne ETAPE.

Il ne s'agit que d'un premier système de bout en bout qui a été appliqué à la REN structurée. Afin d'explorer sa généricité, il serait désormais intéressant de l'appliquer à une autre tâche.

L'extension de l'apprentissage par transfert a également soulevé un point intéressant. Il s'agit de la possibilité de découper l'apprentissage d'une tâche pour faciliter l'entraînement du système. Ces deux points nous servent de bases pour orienter la suite de nos travaux.

Ainsi, nous proposons de nous intéresser à la tâche d'extraction de concepts sémantiques. C'est en soit une tâche très similaire à la reconnaissance des entités nommées, qui est toutefois plus complexe par sa diversité en terme concepts à retrouver.

Le prochain chapitre fera l'objet de nos travaux autour de la tâche d'extraction, de bout en bout, des concepts sémantiques dans la parole. Basés sur l'extension de l'apprentissage par transfert de ce chapitre, nous développerons une utilisation originale des données permettant à une approche de bout en bout de surpasser les performances d'une chaîne de composants.

5.5 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur la tâche de reconnaissance des entités nommées dans la parole. Nous avons exploité le cadre de la campagne d'évaluation française ETAPE qui correspond aux travaux les plus récents pour les données à notre disposition.

Nos premiers travaux se sont concentrés sur la mise à jour des résultats d'ETAPE en raison de l'évolution des technologies depuis 2012, notamment par l'emploi des approches neuronales.

Nous avons ensuite réalisé la mise en œuvre d'un premier système répondant à la problématique de cette thèse. Nous avons été en mesure d'effectuer une tâche de reconnaissance des entités nommées dans la parole de bout en bout dans un contexte simplifié.

Enfin, nous avons étendu le champ applicatif de notre premier système au cadre de la campagne ETAPE. Les résultats de toutes nos expérimentations ont confirmé l'intérêt d'une approche neuronale de bout en bout. Nous notons qu'elle n'est pas encore suffisamment per-

formante pour surpasser une chaîne de composants à jour.

Les travaux présentés dans ce chapitre ont mené à des publications scientifiques. La mise en œuvre d'un premier système de bout en bout a conduit à la publication [Ghannay, Caubrière et al. 2018]. Les travaux réalisés dans le cadre de la campagne ETAPE ont conduit aux publications [Caubrière, Rosset et al. 2020a] et [Caubrière, Rosset et al. 2020b].

Désormais, nous souhaitons continuer d'explorer les capacités du système de bout en bout en l'appliquant sur des tâches plus spécifiques. Nous souhaitons également, étendre nos travaux sur l'extension de l'apprentissage par transfert, qui ont montré des résultats prometteurs.

Chapitre 6

Extraction de concepts sémantiques

Sommaire

Lors du chapitre précédent, nous avons appliqué une approche de bout en bout pour la reconnaissance des entités nommées dans la parole. Nous avons vu que, bien que prometteuse, notre approche n'a pas été en mesure de rivaliser avec une approche par chaîne de composants.

Dans ce chapitre, notre premier objectif consiste à étendre nos travaux à la tâche d'extraction des concepts sémantiques dans la parole. Il s'agit d'une tâche similaire à la reconnaissance des entités nommées. Pour l'extraction des concepts sémantiques, nous envisageons une application directe du système que nous avons proposé dans le chapitre précédent.

Alors que les entités nommées correspondent à des éléments sémantiques généraux, comme une *personne*, un *lieu*, les concepts sémantiques correspondent à des éléments liés à un cadre applicatif spécifique ; il s'agit d'éléments sémantiques plus précis. Même si plus précis, ce sont des éléments qui peuvent s'exprimer sous la forme d'entités nommées et donc bénéficier d'un modèle initialement dédié à la REN par transfert d'apprentissage.

Dans le cadre de notre étude, les tâches applicatives visées correspondent à une tâche de réservation d'hôtel (MEDIA), ainsi qu'à une tâche de réservation de tickets de théâtre (PORTMEDIA). Les concepts sémantiques appuyant ces applications sont par exemple, la nom d'un hôtel (*hotel-nom*) ou le nom de l'auteur d'une pièce (*piece-nom-auteur*). Davantage de détails sur ces corpus de données sont présents dans le chapitre 4.

Dans ce nouveau chapitre, nous proposons de développer davantage notre approche par transfert d'apprentissage. Nous avons noté le caractère plus générique des entités nommées par rapport aux concepts sémantiques. De plus, la quantité de données disponibles pour estimer un modèle de reconnaissance des entités nommées à notre disposition est beaucoup plus importante que la quantité disponible pour la mise en place d'un modèle destiné à l'extraction de concepts sémantiques. Il nous semble intéressant d'étudier l'apport d'un transfert d'apprentissage d'un modèle appris pour les entités nommées vers un modèle pour l'extraction des concepts sémantiques.

Enfin, un dernier objectif consiste à effectuer des expérimentations afin d'optimiser notre architecture neuronale pour obtenir de meilleures performances.

Nous organisons ce chapitre en trois sections. Chacune d'entre elles vise à détailler un des objectifs évoqués et elles sont organisées suivant l'ordre mentionné.

6.1 Application de l'approche de bout en bout à l'extraction concepts sémantiques

Afin d'étendre notre approche de bout en bout aux concepts sémantiques, nous nous sommes concentrés sur l'ensemble de données MEDIA. Comme mentionné dans la section 4.6, il s'agit d'un ensemble de données téléphoniques de 57,5 heures réparties entre une partie système et une partie utilisateur. L'annotation en concept sémantique porte uniquement sur la partie uti-

lisateur et ne représente ainsi que 23,5 heures de paroles annotées. Une difficulté sera donc de compenser le manque de données annotées pour la tâche finale de compréhension de la parole.

Le schéma d'annotation en concepts sémantiques est plus riche que celui des entités nommées, avec 76 concepts contre 57 entités nommées et composants (QUÆRO), avec une représentation à plat de ces concepts là où nous avions une représentation structurée dans les entités nommées.

En appliquant notre approche de bout en bout sur cette tâche, notre objectif est de vérifier sa pertinence dans un cadre plus contraint par la quantité des concepts et la taille réduite du corpus d'apprentissage.

Pour prendre en compte le manque de données annotées, nous exploiterons à nouveau l'apprentissage par transfert que nous avons utilisé pour la reconnaissance des entités nommées.

En complément, nous exploitons l'ensemble de données PORTMEDIA comme augmentation de données annotées, en raison de la proximité de sa tâche applicative avec celle de MEDIA.

Afin de vérifier la pertinence de notre approche de bout en bout, nous effectuons à nouveau une comparaison à une approche par chaîne de composants. Nous effectuons donc la mise en œuvre de cette approche par chaîne de composants dans la sous-section suivante.

6.1.1 Approche par chaîne de composants

La chaîne de composants que nous mettons en œuvre correspond à l'imbrication d'un système de reconnaissance de la parole neuronale de type Deep Speech 2 et d'un système d'extraction des concepts sémantiques exploitant une approche par CRF. L'objectif est de déterminer les performances à l'état de l'art de cette approche pour l'ensemble de données MEDIA.

Nous avons choisi d'utiliser Deep Speech 2 comme système de RAP puisque celui obtient des résultats similaires à un système de RAP hybride de modèle de markov caché et neuronal.

Nous proposons également d'exploiter les données de l'ensemble PORTMEDIA pour augmentater les données avant une optimisation du système de RAP sur MEDIA. Nous mettons ainsi en place une approche par transfert d'apprentissage en trois étapes successives, dans un but d'optimisation fine (*fine tuning*).

Tout d'abord nous estimons un système de RAP (noté *RAP*) en utilisant toutes nos données audio décrites en section 4.8. Cela signifie que nous avons sous-échantillonné nos données audios d'enregistrement studio en 8 Khz, que nous avons ajoutés aux enregistrements téléphoniques déjà disponibles. Ces enregistrements téléphoniques proviennent des ensembles de données MEDIA, PORTMEDIA et DECODA.

À la suite de l'entraînement du système de RAP exploitant toutes nos données, nous effectuons un *fine tuning* sur les données des tâches MEDIA et PORTMEDIA (le système obtenu est noté $PM + M$).

Enfin, nous effectuons un dernier *fine tuning* sur les données MEDIA (dont le système est noté M). Le système final correspond aux apprentissages successifs $RAP \rightarrow PM + M \rightarrow M$ et est dénommé RAP_{cc} (cc pour chaîne de composants).

Ce système permet l'obtention d'un taux d'erreur sur les mots (WER) de 9,3 % pour l'ensemble de test de MEDIA.

Pour la tâche cible, les travaux conduisant aux meilleurs résultats publiés avant cette thèse ont utilisé un composant de RAP atteignant un score WER de 23,6 % [Simonnet, Ghannay, Camelin, Estève et De Mori 2017]. Au vu de l'impact de la qualité des transcriptions automatiques pour une tâche de compréhension de la parole, la différence de performances observée nous motive à effectuer la mise à jour des résultats de l'approche par chaîne de composants.

Ainsi, nous réalisons la mise en œuvre de deux systèmes CRF dédiés à l'extraction des concepts sémantiques. Le premier système exploite uniquement la forme de surface des mots et nous le dénommons ECS_{texte}. Ce système est appris à l'aide des transcriptions et annotations sémantiques manuelles de MEDIA.

Le second système ($ECS_{texte+carac}$) est aussi appris à l'aide des transcriptions et annotations manuelles. Toutefois, il est enrichi par l'extraction automatique de caractéristiques extraites à l'aide de l'outil MACAON [Nasr et al. 2011]. Nous utilisons notamment les lemmes, les étiquettes morphosyntaxiques, les *"governor words"* ainsi que leurs relations avec le mot courant. En complément, nous extrayons des caractéristiques morphologiques correspondant aux n-grammes de la première à la troisième lettre du mot, ainsi qu'aux mêmes n-grammes des dernières lettres du mot. Nous utilisons les mêmes caractéristiques que celles décrites dans [Simonnet, Ghannay, Camelin, Estève et De Mori 2017].

L'évaluation de la tâche est effectuée avec les métriques du taux d'erreurs sur les concepts (CER) et du taux d'erreurs sur les concepts et leurs valeurs (CVER). Comme détaillé dans la section 3.4, il s'agit des métriques couramment utilisées pour l'évaluation dans le cadre de cette tâche.

Nous réalisons l'entraînement des deux systèmes CRF mentionnés et nous fournissons, dans la table 6.1, les résultats de la chaîne de composants obtenus. Les résultats sont fournis pour l'ensemble de test de MEDIA (l'optimisation étant réalisée sur le corpus de développement).

Système	**CER**	**CVER**
RAP / ECS_{texte}	20,6	24,8
RAP / $ECS_{texte+carac}$	**16,1**	**20,4**

Table 6.1 – Résultats expérimentaux d'une chaîne de composants état de l'art appliquée à l'ensemble de test de MEDIA.

Les résultats de ces deux expérimentations montrent l'impact très positif de l'exploitation

des caractéristiques additionnelles. Aussi, ils permettent de rendre compte de l'état de l'art pour MEDIA en exploitant un système de RAP plus performant. Nous sommes ici capables d'obtenir un CER de 16,1 % contrairement aux 19,3 % obtenus par [Simonnet, Ghannay, Camelin, Estève et De Mori 2017].

Suite à la mise à jour des résultats de la chaîne de composants, nous appliquons désormais notre système de bout en bout sur les mêmes données afin de rendre ces deux approches comparables.

6.1.2 Premiers résultats avec une approche de bout en bout

La mise en œuvre de notre système de bout en bout est similaire en tout point à celle décrite dans la section 5.3. Cela signifie que nous exploitons à nouveau l'implémentation de Deep Speech 2 à laquelle nous fournissons des transcriptions enrichies sous une forme similaire à celle illustrée dans la figure 5.5. Nous permettons ainsi au système d'apprendre un alignement entre les données audios de MEDIA et leurs transcriptions manuelles enrichies avec les concepts sémantiques.

Pour lutter contre le manque de données, nous proposons d'exploiter à nouveau un transfert d'apprentissage similaire à celui réalisé pour les entités nommées. Nous effectuons donc l'entraînement d'un système de reconnaissance automatique de la parole utilisant la répartition des données de la section 4.8. Il s'agit en soi du même modèle que celui appris à la première étape de la mise en place du système RAP_{cc} de la sous-section précédente.

Après la tâche de RAP, nous ciblons la tâche de compréhension du langage. Au vu de sa proximité avec MEDIA, nous exploitons l'ensemble de données PORTMEDIA pour réaliser une augmentation de données. Cela signifie que nous exploitons les parties utilisateurs de ces deux ensembles conjointement, comme corpus augmenté et annoté en concepts sémantiques.

Nous effectuons ensuite une dernière étape qui consiste en un *fine tuning* sur MEDIA.

Décodage Greedy

Nous comparons un premier système (M) appris directement sur l'ensemble MEDIA à un second système ($RAP \rightarrow M$) bénéficiant du transfert d'apprentissage à partir d'un système de RAP. Ensuite, nous comparons ces deux systèmes avec un troisième bénéficiant de l'utilisation des données PORTMEDIA ($RAP \rightarrow PM + M \rightarrow M$).

Nous reportons les résultats obtenus en termes de CER et de CVER dans la table 6.2.

Comme dans nos expériences concernant la tâche de reconnaissance des entités nommées, ces résultats montrent les bénéfices de l'apprentissage par transferts successifs. Nous pouvons noter une amélioration importante des performances grâce à l'utilisation d'un système de RAP préentrainé. De plus, l'augmentation de données par PORTMEDIA a un impact positif, que ce soit en termes de CER et de CVER.

	Développement		*Test*	
Système	**CER**	**CVER**	**CER**	**CVER**
M	40,1	53,6	39,8	52,1
$RAP \rightarrow M$	25,3	31,8	23,7	30,3
$RAP \rightarrow PM + M \rightarrow M$	**23,1**	**29,2**	**22,2**	**28,8**

TABLE 6.2 – Résultats expérimentaux de l'extraction de concepts sémantiques de bout en bout pour les ensembles de développement et de test de MEDIA.

Bien que les performances soient en deçà d'une approche classique par composants, l'approche de bout en bout semble toutefois fonctionnelle pour une tâche comme MEDIA.

Ces résultats correspondent aux sorties immédiates du système neuronal, sans l'utilisation d'un modèle de langage entraîné pour modéliser les séquences de mots et de concepts sémantiques.

Décodage Beam Search avec un modèle de langage

De la même façon que nos travaux sur les entités nommées, nous effectuons l'apprentissage de modèles de langue n-grammes à l'aide des transcriptions manuelles enrichies de nos données MEDIA. Comparant des modèles 3-gramme à 6-gramme, nous avons déterminé qu'un modèle 5-gramme est optimal pour cet ensemble de données. Nous reportons dans la table 6.3, les résultats de nos trois systèmes ayant bénéficié de l'algorithme Beam search (*Beam-Search decoding*) et du modèle de langage 5-gramme.

	Développement		*Test*	
Système	**CER**	**CVER**	**CER**	**CVER**
M	32,2	38,2	32,8	37,9
$RAP \rightarrow M$	21,3	25,1	20,1	24,0
$RAP \rightarrow PM + M \rightarrow M$	**19,7**	**23,3**	**19,0**	**22,9**

TABLE 6.3 – Résultats expérimentaux de l'extraction de concepts sémantiques de bout en bout pour les ensembles de développement et de test de MEDIA. Exploitation de l'algorithme Beam Search et d'un modèle de langage 5-gramme.

L'utilisation d'un modèle de langage et de l'algorithme de beam search nous permet d'améliorer nos résultats et de nous rapprocher des performances du système par chaîne de composants sans être encore du même ordre.

Mode étoile

Lors de nos expérimentations autour de la tâche de REN, nous avons proposé un mode étoile dont l'objectif était d'aider le système à concentrer son apprentissage sur les EN et leurs

valeurs par l'intermédiaire de la fonction de coût CTC. Nous appliquons à nouveau notre mode étoile à notre système dédié à l'extraction de concepts sémantiques. Comme nous effectuons la tâche d'extraction de concepts en deux étapes, un apprentissage avec les données MEDIA et PORTMEDIA puis un *fine tuning* avec les données MEDIA, nous proposons d'étudier l'impact du mode étoile.

Les résultats par utilisation d'un modèle de langage et du mode étoile sont donnés dans la table 6.4.

	Développement		*Test*	
Système	**CER**	**CVER**	**CER**	**CVER**
Greedy				
$M*$	49,4	67,6	47,8	63,6
$RAP \rightarrow M*$	23,5	31,4	23,1	30,8
$RAP \rightarrow PM+M \rightarrow M*$	22,2	29,4	20,9	27,9
$RAP \rightarrow PM*+M* \rightarrow M*$	21,2	27,9	20,6	27,7
Beam Search				
$M*$	39,8	50,3	39,0	47,0
$RAP \rightarrow M*$	20,0	24,4	18,9	22,5
$RAP \rightarrow PM+M \rightarrow M*$	18,5	23,2	17,0	**21,5**
$RAP \rightarrow PM*+M* \rightarrow M*$	**18,3**	**23,0**	**16,8**	**21,5**

TABLE 6.4 – Résultats expérimentaux de l'extraction de concepts sémantiques de bout en bout pour l'ensemble de test de MEDIA par l'utilisation de modèle de langage 5-gramme et du mode étoile.

Ces résultats nous permettent d'observer des similarités avec les expérimentations concernant les entités nommées. Ils nous montrent que le mode étoile couplé à un modèle de langage 5-gramme améliore l'ensemble de nos résultats.

En complément, nous pouvons déduire de ces résultats qu'il est préférable d'appliquer notre mode étoile pour l'ensemble de la tâche d'extraction des concepts sémantiques, c'est-à-dire dès l'augmentation de données réalisée avec PORTMEDIA.

Jusqu'ici, notre meilleure approche de bout en bout ne permet pas d'atteindre des performances aussi compétitives qu'une approche par chaîne de composants.

Notre approche n'utilise que le signal audio et un modèle de langage de type n-gramme. Alors que notre meilleure chaîne de composants est enrichie via des traitements de langage naturel par des caractéristiques linguistiques, notre approche n'intègre pas explicitement ce type de informations.

Comme nous l'avons vu dans la table 6.1, la meilleure approche par chaîne de composants obtient 16,1 de CER et 20,4 de CVER, que nous comparons à notre meilleure approche de bout en bout qui obtient 16,8 de CER et 21,5 de CVER sur le test de MEDIA.

Il apparait que ces informations tierces sont une source importante d'informations. Pour la

suite de nos travaux, nous proposons d'injecter de nouvelles informations lors de l'apprentissage de notre modèle.

Injection d'informations additionnelles

Certains travaux concernant l'ajout d'informations additionnelles n'ont pas été directement réalisés dans le cadre de cette thèse. Il s'agit cependant de travaux auxquels nous avons pris part et qui s'appuient sur le système de bout en bout proposés dans ce manuscrit.

Nous pouvons citer des travaux visant l'étude de l'adaptation au locuteur et l'utilisation de données issues d'une langue étrangère pour pallier le manque de données d'apprentissage [Tomashenko, Caubrière et Estève 2019; Tomashenko, Caubrière, Estève et al. 2019]. L'information concernant le locuteur est injectée dans le système par l'intermédiaire d'une représentation vectorielle (*i-Vector*) qui sera concaténée aux représentations calculées par les couches convolutionnelles sur les spectrogrammes de l'audio. Les résultats de ces travaux ont montré l'utilité d'un système de reconnaissance de la parole anglais appris avec une quantité importante de données avant l'exploitation de la tâche française de compréhension de la parole. De plus, ils ont montré l'intérêt de l'adaptation au locuteur pour l'extraction de concepts sémantiques.

Nous pouvons également citer les travaux effectuant l'étude de l'impact de l'historique de dialogue pour la reconnaissance des concepts [Tomashenko, Raymond et al. 2020]. Comme pour l'adaptation au locuteur, ces travaux exploitent une représentation vectorielle qui sera concaténée à la représentation issue des couches de convolution. Il s'agit toutefois d'une représentation de l'historique de dialogue (*h-vectors*). Ces travaux explorent plusieurs types d'*h-vectors* et montrent l'intérêt de l'exploitation de cet historique. Nous avons été impliqués dans l'ensemble de ces travaux.

Dans le cadre de cette thèse, nous avons également proposé de bénéficier des entités nommées comme information additionnelle. Par l'intermédiaire de la tâche de REN précédemment explorée, nous avons montré son exploitabilité par notre système. Au sein de la section suivante, nous détaillons nos motivations concernant la prise en compte des entités nommées, notre méthode d'utilisation, ainsi que nos expérimentations et résultats.

6.2 Transfert d'apprentissage piloté par une stratégie de curriculum

Après avoir manipulé les entités nommées et les concepts sémantiques de l'application MEDIA, nous avons pu observer des concepts sémantiques et des entités nommées appartenant à des catégories sémantiques similaires. Nous pouvons par exemple rapprocher l'entité nommée *"amount"* du concept sémantique *"paiement-montant-entier"*. Comme nous l'avons vu dans le chapitre 3.1.3, les entités nommées peuvent être considérées comme des briques élémen-

taires de l'information contenue dans les documents. Par nature, elles permettent de répondre à des questions simples et générales comme *Qui ? Quoi ? où ? Quand ?* [Nouvel et al. 2015]. Au contraire, les concepts sémantiques dans le contexte de MEDIA sont ultraspécialisés pour la tâche de réservation d'hôtel (par exemple *chambre-equipement*).

Nous émettons l'hypothèse qu'il est possible de tirer partie les données existantes concernant les entités nommées pour notre tâche d'extraction de concepts sémantiques. En apprenant dans un premier temps un modèle capable d'extraire des concepts généraux comme les entités nommées, nous pouvons ensuite le spécialiser sur des concepts sémantiques plus spécifiques. Ce processus d'apprentissage s'apparente à l'approche par curriculum [Bengio, Louradour et al. 2009].

L'idée sous-jacente est que le premier modèle devrait être capable de construire des représentations internes portant de la sémantique générale réexploitable dans un domaine spécialisé.

6.2.1 Apprentissage par curriculum

L'apprentissage par curriculum est une méthode d'entraînement s'inspirant des méthodes d'apprentissage humaines. Une personne s'appuie généralement sur les notions les plus simples qu'il a apprises afin d'apprendre des notions plus compliquées. C'est sur ce principe que s'appuie l'apprentissage par curriculum.

Concrètement, cette méthode consiste à attribuer un score de difficulté aux exemples présent dans un ensemble d'apprentissage. Ce score permet ensuite de réorganiser l'ensemble de manière à présenter les données des plus simples aux plus complexes lors de la phase d'entrainement d'un système.

Cette méthode a montré son efficacité pour améliorer la capacité de généralisation d'un système, ainsi que pour obtenir une convergence vers un meilleur minima local [Bengio, Louradour et al. 2009].

Dans le cadre de nos travaux, nous avons à notre disposition des ensembles de données différents, ciblant des tâches différentes. Nous n'envisageons donc pas de considérer nos données comme un seul corpus au sein duquel nous ordonnerions les exemples d'apprentissages.

Nous proposons une méthode s'inspirant directement de l'apprentissage par curriculum, et exploitant pleinement les principes du transfert d'apprentissage, que nous décrivons dans la prochaine section. De plus, plutôt que de considérer les données des plus simples aux plus complexes, nous proposons de les exploiter des plus générales aux plus spécifiques.

6.2.2 Association du transfert et du curriculum d'apprentissage

Notre approche consiste à associer l'approche du curriculum et du transfert d'apprentissage. Cette approche est fondée sur la gestion des transferts d'apprentissages du plus générale

aux plus spécialisés, l'ensemble de la chaîne de transfert est alors reconsidéré. Nous proposons d'apprendre un modèle d'extraction des concepts sémantiques fondé sur la séquence d'apprentissage et de transfert d'apprentissages suivantes :

1. Reconnaissance automatique de la parole (RAP)
2. Reconnaissance automatique des entités nommées (REN)
3. Extraction des concepts sémantiques selon MEDIA et PORTMEDIA ($PM + M$)
4. *Fine tuning* sur la tâche MEDIA (M)

Nous nommons la chaîne d'apprentissage obtenue $RAP \rightarrow REN \rightarrow PM + M \rightarrow M$.

Pour rappel, chacune des flèches ($\rightarrow$) de cette annotation représente un transfert d'apprentissage réalisé de façon identique aux transferts de la sous-section 6.1.2. C'est-à-dire que nous conservons l'ensemble des poids du système neuronal appris à gauche de la flèche, excepté la couche de sortie softmax. Cette dernière couche est réinitialisée en raison des différences de symboles pouvant être émis en fonction de la tâche ciblée à droite de la flèche.

Les poids ainsi conservés sont utilisés pour l'initialisation des poids du modèle qui sera appris à l'aide de la tâche située à droite de la flèche. Ce nouvel apprentissage est réalisé avec les valeurs réinitialisées des hyper paramètres (taux d'apprentissage, nombre d'époques, etc.).

Nous donnons une représentation schématique de la chaîne d'apprentissage finale dans la figure 6.1.

6.2.3 Expérimentations et résultats

À partir de la chaîne d'apprentissage décrite précédemment, nous réalisons des expérimentations visant à confirmer la viabilité de notre stratégie de transfert par curriculum. Nous effectuons donc l'ajout d'une étape de reconnaissance des entités nommées au sein des chaînes d'apprentissages réalisées précédemment en section 6.1.2.

Pour l'apprentissage du modèle visant la tâche de reconnaissance des entités nommées, nous exploitons la totalité de nos données studio annotées selon notre annotation simplifiée du formalisme QUÆRO (voir section 5.3.1). Cela signifie que nous exploitons l'annotation manuelle des ensembles ETAPE et QUÆRO afin d'effectuer l'augmentation automatique des données studio. Nous apprenons un modèle NeuroNLP2 à l'aide des annotations d'EN manuelles, puis nous portons la prédiction sur toutes nos données de parole dépourvues d'annotation en EN. Nous n'ajoutons pas de données supplémentaires et nous ne portons pas l'annotation en EN sur les données téléphoniques en raison de leurs annotations initiales en concept sémantique. Par l'intermédiaire de l'annotation automatique, nous cumulons un ensemble d'apprentissage de près de 290 heures annotées manuellement et automatiquement en EN.

Pour la réalisation de nos expérimentations, nous continuons d'exploiter notre implémentation de Deep Speech 2 de façon identique. C'est-à-dire, une architecture composée de 2 couches

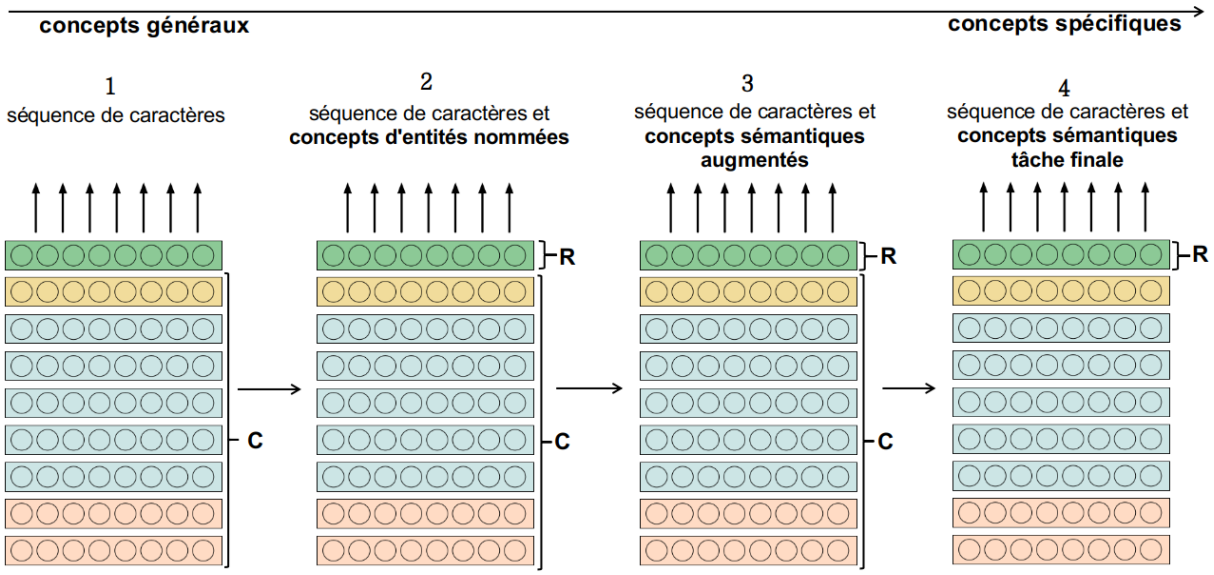

FIGURE 6.1 – Représentation schématique de la chaîne d'apprentissages successifs en quatre étapes. Les couleurs représentent un type de couche neuronale, en orange, les couches CNN, en bleu, les couches bLSTM, en jaune, la couche linéaire, et en vert, la couche softmax. "C" représente les poids conservés et "R" représente la couche réinitialisée.

CNN, 5 couches bLSTM avec normalisation des batchs, 1 couche linéaire et une couche de sorties softmax. Pour les couches CNN, nous utilisons à nouveau 32 filtres et nous conservons la taille des couches bLSTM à 800 unités.

L'évaluation des performances de nos systèmes s'effectue aussi avec les métriques du CER et du CVER calculées sur les ensembles de développement et de test du corpus MEDIA.

Nous fournissons dans la table 6.5, les résultats des expérimentations bénéficiant de l'étape de transfert d'apprentissage portant sur les entités nommées. Il s'agit des résultats pour les sorties immédiates du système neuronal (*greedy*). Ils peuvent ainsi être directement comparés aux résultats de la table 6.2. Nous effectuons donc un report de ces résultats dans la première partie de la table ci-dessous.

Les résultats de ces expérimentations montrent l'apport des entités nommées dans la chaîne d'apprentissage. Par leur utilisation, nous observons une amélioration systématique des performances, que ce soit en termes de CER et de CVER. Nous considérons que le gain provenant de cette étape vient effectivement de la tâche de reconnaissance des entités nommées elle-même, dans la mesure où les données audio associées sont déjà exploitées lors de l'étape de RAP.

Nous souhaitons confirmer que l'apport des EN provient effectivement de la stratégie de curriculum employée. Nous effectuons donc l'apprentissage d'une chaîne au sein de laquelle nous brisons le processus itératif de spécialisation, à savoir $RAP \rightarrow PM+M \rightarrow REN \rightarrow M$.

	Développement		*Test*	
Système	**CER**	**CVER**	**CER**	**CVER**
"RAP → M"	25,3	31,8	23,7	30,3
"RAP → PM + M → M"	23,1	29,2	22,2	28,8
RAP → REN → M	23,5	30,4	22,4	28,7
RAP → REN → PM + M → M	**22,0**	**28,0**	**21,6**	**27,7**

TABLE 6.5 – Résultats expérimentaux de l'extraction de concepts sémantiques pour l'ensemble de développement et de test de MEDIA avec une chaîne d'apprentissage incorporant les entités nommées. Les résultats encadrés par des guillemets sont reportés de la table 6.2.

Nous observons lors de l'apprentissage de l'étape de *REN* que le système ne converge pas, et ce malgré nos différentes tentatives de modification des paramètres, notamment le taux d'apprentissage.

Cette absence de convergence tend à confirmer l'importance et l'apport de l'ordonnancement des étapes d'apprentissage que nous avons présentées dans notre stratégie de transfert par curriculum. En inversant les étapes *REN* et $PM + M$, nous ne pouvons pas tirer bénéfice des connaissances apportées par les données étiquetées en entités nommées.

Pour compléter nos résultats, nous proposons d'appliquer l'algorithme de beam search et de notre modèle de langage 5-gramme dédié à MEDIA. Nous fournissons les résultats associés dans la table 6.6.

	Développement		*Test*	
Système	**CER**	**CVER**	**CER**	**CVER**
"RAP → M"	21,3	25,1	20,1	24,0
"RAP → PM + M → M"	19,7	23,3	19,0	22,9
RAP → REN → M	19,8	23,7	18,8	22,8
RAP → REN → PM + M → M	**19,1**	**22,9**	**18,1**	**22,1**

TABLE 6.6 – Résultats expérimentaux pour une chaîne d'apprentissage incorporant les entités nommées sur l'ensemble de développement et de test de MEDIA, par exploitation du beam search avec un modèle de langage 5-gramme. Les résultats encadrés par des guillemets sont reportés de la table 6.3.

L'emploi du beam search nous permet à nouveau d'améliorer les performances de nos systèmes. Nous observons qu'avec cet algorithme, nous conservons l'apport des entités nommées.

En complément de ces résultats, nous proposons d'appliquer à nouveau le mode étoile. Nous reportons les résultats des chaînes d'apprentissages exploitant ce mode dans la table 6.7. En raison de l'apport systématique de l'utilisation des données PORTMEDIA, nous ne considérons désormais que les chaînes d'apprentissage en bénéficiant.

Ces expérimentations nous permettent d'obtenir nos meilleures performances avec le sys-

	Développement		*Test*	
Système	**CER**	**CVER**	**CER**	**CVER**
"RAP → PM + M → M∗"	18,5	23,2	17,0	21,5
"RAP → PM ∗ +M∗ → M∗"	18,3	23,0	16,8	21,5
RAP → REN → PM + M → M∗	17,8	22,1	16,6	21,3
RAP → REN → PM ∗ +M∗ → M∗	**17,6**	**21,8**	16,4	20,9

Table 6.7 – Résultats expérimentaux pour une chaîne d'apprentissage incorporant les entités nommées sur l'ensemble de développement et de test de MEDIA, par exploitation du beam search avec un modèle de langage 5-gramme et du mode étoile. Les résultats encadrés par des guillemets sont reportés de la table 6.4.

tème actuel en ayant un taux de CER à 16,4 et un taux de CVER à 20,9. Ces résultats confirment que l'application du mode étoile à chaque étape d'extraction des concepts sémantiques ($PM + MetM$) est bénéfique.

La présence des entités nommées dans la chaîne d'apprentissage apporte une information additionnelle utile à la mise en œuvre de notre meilleur système. Nous proposons d'effectuer une analyse qualitative de l'apport des entités nommées.

6.2.4 Analyse de l'apport des Entités Nommées

Afin de mieux comprendre l'apport des entités nommées dans le cadre de notre approche par transfert d'apprentissage par curriculum, nous proposons d'analyser leur impact sur les sorties du système. Pour ce faire, nous mesurons l'évolution du nombre d'erreurs par catégories de concepts avec et sans l'utilisation des entités nommées dans notre chaîne d'apprentissage. Cette analyse est effectuée sur l'ensemble de développement de MEDIA et nous comparons les chaînes d'apprentissages $RAP \rightarrow PM + M \rightarrow M$ et $RAP \rightarrow REN \rightarrow PM + M \rightarrow M$.

Sur le corpus de développement, le score en termes de CER est de 19,7 pour la chaîne n'exploitant Notons que le nombre total de concepts sémantiques de référence de l'ensemble de développement est de 3 333. Nous fournissons dans la figure 6.2 le delta du nombre des erreurs sur les concepts sémantiques entre les deux chaînes que nous comparons.

Sur cette figure, nous pouvons observer l'impact positif des données d'entités nommées pour les 25 concepts sémantiques ayant un delta de nombre d'erreurs négatif. Nous pouvons rapprocher des entités nommées plusieurs des concepts sémantiques les plus fortement impactés. Nous pouvons par exemple relier le concept *localisation-ville* à l'entité *loc*. De même que *temps-jour-mois* se rapproche de l'entité *date*.

Il est intéressant de noter l'impact positif sur le concept sémantique *hotel-services*, qui ne semble pas directement rapprochable d'une entité nommée. En effectuant une analyse plus fine des occurrences impactées positivement, nous pouvons observer qu'il s'agit globalement de lieu, comme une piscine, ou une salle de réunion, pouvant être relié à l'entité *loc*. Nous

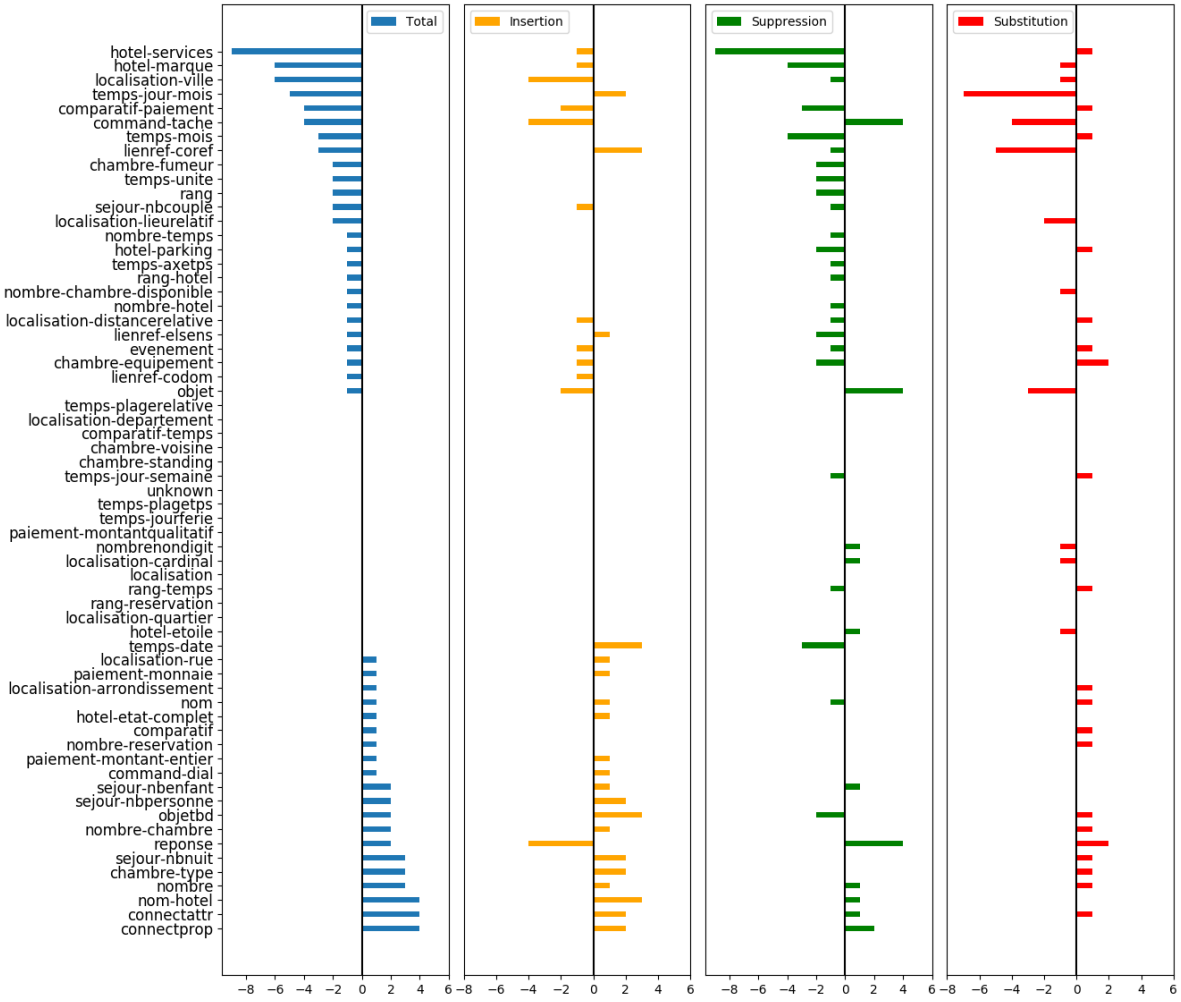

FIGURE 6.2 – Impact par concepts sur le nombre d'erreurs en fonction de l'utilisation des entités nommées dans la chaîne d'apprentissage.

notons que l'impact positif sur ce concept sémantique correspond principalement à des erreurs de suppression qui ne sont plus commises. L'information apportée par cette entité nommée semble suffisamment importante pour permettre au système d'extraire plus efficacement les concepts sémantiques pouvant s'en rapprocher dans un contexte différent.

Il est également intéressant de noter les modifications de typologie d'erreurs de certains concepts. Par exemple, le concept *temps-date* voit ses erreurs de suppression en partie corrigée, en même temps que de nouvelles insertions apparaissent. Ce phénomène pourrait être révélateur d'une plus forte sensibilité de notre système aux dates suite à l'apprentissage des entités nommées.

Il est aussi possible de noter que les entités nommées peuvent avoir un impact négatif. Par exemple, une partie importante des concepts impactés négativement peuvent se rapprocher de l'entité nommée *amount*, par exemple *nombre*, *sejour-nbNuit*, *sejour-nbEnfant*, *nombre-chambre*. Il pourrait être intéressant de mettre en place une stratégie d'apprentissage plus souple pour cette entité nommée, qui semble trop spécialisée pour les données d'EN et incapable de se placer efficacement dans le contexte de MEDIA.

Enfin, nous proposons de comparer les deux chaînes d'apprentissage pour l'émission de concepts et de leurs valeurs n'apparaissant pas dans l'ensemble d'apprentissage MEDIA. Nous nommons ces concepts les couples UCV (*Unseen Concept-Value pairs*). Nous dénombrons 467 UCV uniques pour un total de 533 occurrences sur l'ensemble de développement de MEDIA. Il s'agit de vérifier si les entités nommées ont un apport concernant la capacité de généralisation du système final.

Nous fournissons dans la table 6.8, le nombre d'UCV correctement reconnu en termes de concepts et de valeurs. Nous donnons également le nombre d'UCV pour lesquels la valeur a correctement été reconnue, correspondant uniquement à une substitution de concept.

Système	**Concept/Valeur correct**	**Valeur seule correcte**	**Concept seul correct**
$RAP \rightarrow PM + M \rightarrow M$	124	36	128
$RAP \rightarrow REN \rightarrow PM + M \rightarrow M$	132	38	123

Table 6.8 – Nombre de couples concept/Valeur, n'apparaissant pas dans l'ensemble d'apprentissages, correctement reconnus au sein de l'ensemble de développement de MEDIA.

Nous voyons que le système est capable de reconnaitre des couples concepts valeurs jamais rencontrées dans le corpus d'apprentissage, sans toutefois être capable d'une très forte généralisation, puisque seulement 25 % d'entre eux sont correctement reconnus.

Il est important de noter que la capacité de reconnaissance de ce type de concepts est de moins de 25 % pour les deux systèmes, indiquant une difficulté de généralisation.

Nous pouvons toutefois noter que l'ajout des entités nommées dans la chaîne d'appren-

tissage permet un gain relatif de 6 % pour la reconnaissance des couples UCV. Les entités nommées apportent une information améliorant la capacité de généralisation de notre système final.

Par la suite, nous essayons d'améliorer les performances de notre modèle en agissant sur sa topologie.

6.3 Impact de la profondeur du modèle

Toutes nos expériences ont été réalisées avec les mêmes paramètres du système Deep Speech 2. Il s'agissait de paramètres donnant de bonnes performances, que nous n'avons pas optimisées en raison du temps de calcul, hormis pour la largeur du modèle.

Nous avons complexifié la tâche cible du système en incorporant des concepts sémantiques ou des entités nommées. Nous supposons que la profondeur du système, c'est-à-dire le nombre de couches bLSTM cachées, ne peut être plus optimale pour la tâche ciblée.

Nous émettons aussi l'hypothèse que l'ajout d'une couche neuronale au-dessus de celles ayant bénéficié d'un préentrainement rend possible une meilleure spécialisation de cette couche pour la tâche finale.

Il est à noter le coût important en termes de temps de calcul de l'entraînement d'une chaîne d'apprentissage complète. Dans le cadre de la tâche MEDIA, nous réalisons l'entraînement de quatre systèmes successifs. Aussi, la quantité de données exploitées par système joue un rôle important pour le temps de calcul nécessaire à sa convergence, de même que le nombre de paramètres à apprendre.

La première étape (*RAP*) exploite un total de 410 heures d'apprentissage. Près de 2,5 semaines de calcul sur un GPU de type K40 sont pour apprendre ce modèle. La seconde étape (*REN*) utilise moins de données avec 290 heures d'apprentissage. L'entraînement de cette étape nécessite environ 1,5 semaine de calcul. L'avant-dernière étape ($PM + M$) exploite conjointement les parties utilisateurs de MEDIA et PORTMEDIA. Seulement trois jours sont nécessaires à la bonne convergence du système associé. Enfin, la dernière étape (M) n'exploite que les données de MEDIA, permettant ainsi une convergence en deux jours. Pour l'apprentissage complet de notre modèle, cinq semaines de calculs sont nécessaires sur un seul GPU.

Dans la table 6.9, nous fournissons les résultats de trois systèmes pour des profondeurs de cinq, six et sept couches récurrentes. Ces résultats sont obtenus grâce aux sorties neuronales immédiates (*Greedy*) ou l'exploitation de l'algorithme de *beam search* avec notre modèle de langage 5-grammes. Ils sont exprimés en CER et CVER pour l'ensemble de test de MEDIA.

Ces résultats tendent à confirmer que la profondeur de notre système n'est pas optimisée pour les tâches que nous ciblons. Nous obtenons un gain de plus de 1 point de CER / CVER pour une sortie neuronale brute (*Greedy*) et de 0,3 point pour une sortie exploitant un modèle de langage (*Beam Search*).

	Greedy		*Beam Search*	
Système	**CER**	**CVER**	**CER**	**CVER**
"$RAP_5 \rightarrow REN_5 \rightarrow PM{+}M_5 \rightarrow M_5$"	21,6	27,7	18,1	22,1
$RAP_6 \rightarrow REN_6 \rightarrow PM{+}M_6 \rightarrow M_6$	**20,2**	**26,2**	18,3	22,3
$RAP_7 \rightarrow REN_7 \rightarrow PM{+}M_7 \rightarrow M_7$	20,3	26,4	**17,8**	**21,8**

TABLE 6.9 – Résultats expérimentaux exprimés en CER et CVER sur l'ensemble de test de MEDIA, pour différentes profondeurs du système. Le résultat encadré par des guillemets est reporté des tables 6.5 et 6.6.

La différence entre 6 et 7 couches cachées n'est pas significative, toutefois nos meilleurs résultats sont obtenus avec le système à 7 couches avec le décodage par *Beam Search*.

Comme nous l'avons vu plus haut, les temps de calcul pour ce type de tâche et d'architecture sont loin d'être négligeable. Afin de tester notre hypothèse de spécialisation de la dernière couche cachée, nous avons réalisé plusieurs expériences dans lesquelles une couche est ajoutée au moment d'un transfert. Nous fournissons dans la table 6.10, les résultats de plusieurs configurations d'entraînement modifiant la profondeur durant l'apprentissage.

	Greedy		*Beam Search*	
Système	**CER**	**CVER**	**CER**	**CVER**
"$RAP_5 \rightarrow REN_5 \rightarrow PM{+}M_5 \rightarrow M_5$"	21,6	27,7	18,1	22,1
$RAP_5 \rightarrow REN_5 \rightarrow PM{+}M_5 \rightarrow M_6$	21,0	26,9	18,5	22,3
$RAP_5 \rightarrow REN_5 \rightarrow PM{+}M_6 \rightarrow M_6$	20,6	26,6	17,7	**21,8**
"$RAP_6 \rightarrow REN_6 \rightarrow PM{+}M_6 \rightarrow M_6$"	20,2	26,2	18,3	22,3
$RAP_6 \rightarrow REN_6 \rightarrow PM{+}M_6 \rightarrow M_7$	21,0	27,0	18,4	22,5
$RAP_6 \rightarrow REN_6 \rightarrow PM{+}M_7 \rightarrow M_7$	19,7	25,8	**17,6**	**21,8**
"$RAP_7 \rightarrow REN_7 \rightarrow PM{+}M_7 \rightarrow M_7$"	20,3	26,4	17,8	**21,8**
$RAP_7 \rightarrow REN_7 \rightarrow PM{+}M_7 \rightarrow M_8$	19,6	25,6	17,9	21,9
$RAP_7 \rightarrow REN_7 \rightarrow PM{+}M_8 \rightarrow M_8$	**19,3**	**25,3**	17,8	**21,8**

TABLE 6.10 – Résultats expérimentaux exprimés en CER et CVER suite à la modification du nombre de couches cachées en cours d'entraînement. Les résultats encadrés par des guillemets sont reportés de la table 6.9.

Ces résultats nous montrent qu'il est préférable d'ajouter cette couche lors du traitement conjoint des données MEDIA et PORTMEDIA, plutôt que lors du dernier transfert ne concernant que les données MEDIA. En raison de contrainte de temps, nous n'avons pas eu la possibilité d'ajouter une couche cachée de spécialisation lors de l'apprentissage du modèle pour la reconnaissance des entités nommées. Dans nos expériences, les performances sont systématiquement meilleures lorsque l'entraînement de la tâche $PM + M$ est réalisé avec la même profondeur que la tâche finale M.

Dans le but de concentrer l'apprentissage du système sur les concepts et leurs valeurs, nous proposons d'appliquer une nouvelle fois notre mode étoile. Nous fournissons les résultats des deux systèmes que nous avons mis en œuvre dans la table 6.11.

	Greedy		*Beam Search*	
Système	**CER**	**CVER**	**CER**	**CVER**
$RAP_6 \rightarrow REN_6 \rightarrow PM^*{+}M^*_7 \rightarrow M^*_7$	18,8	25,2	**15,8**	**20,3**
$RAP_7 \rightarrow REN_7 \rightarrow PM^*{+}M^*_8 \rightarrow M^*_8$	**18,1**	**25,1**	**15,8**	20,5

Table 6.11 – Résultats expérimentaux exprimés en CER et CVER pour des chaînes d'apprentissages optimisées en profondeur et l'utilisation du mode étoile.

En modifiant la profondeur de l'architecture neuronale, en bénéficiant de l'approche par transfert par curriculum, du mode étoile et de l'algorithme de *beam search* avec modèle de langage 5-gramme, nous obtenons nos meilleurs résultats. Il s'agit des meilleurs résultats pour la tâche d'extraction des concepts sémantiques MEDIA obtenus dans cette thèse. Nous proposons dans la sous-section suivante d'effectuer leur comparaison au meilleur résultat d'une chaîne de composants.

6.3.1 Comparaison de l'approche proposée avec une approche par chaîne de composants

Notre meilleure approche de bout en bout exploite pleinement toutes les contributions que nous avons apportées dans le cadre de cette thèse.

Notre meilleure approche par chaîne de composants est constituée d'un système de RAP Deep Speech 2 optimisé pour MEDIA. L'extraction des concepts sémantique de MEDIA à l'aide des transcriptions et annotations manuelles enrichies par des caractéristiques extraites automatiquement est réalisée avec un CRF.

Pour faciliter la comparaison entre ces deux systèmes, nous reportons dans le table 6.12 les résultats déjà vus dans les tables 6.1 et 6.11.

Système	**CER**	**CVER**
RAP_{cc} / $ECS_{texte+carac}$	16,1	20,4
$RAP_6 \rightarrow REN_6 \rightarrow PM^*{+}M^*_7 \rightarrow M^*_7$	**15,8**	**20,3**

Table 6.12 – Comparaison des approches à chaînes de composants et de bout en bout. Report des meilleurs résultats de chaque approche obtenus dans le cadre de cette thèse.

Cette comparaison montre que nous avons réussi à mettre en œuvre une approche neuronale de bout en bout aussi compétitive qu'une approche par chaîne de composants.

Cette approche pourrait être améliorée. Il serait par exemple possible d'utiliser une architecture neuronale plus complexe, comme les encodeurs-décodeurs avec mécanismes d'atten-

tion ou les transformers. Associés à ces nouvelles architectures neuronales, nous pourrions considérer l'utilisation d'autres types de représentation de la parole, comme wav2vec, mais aussi injecter des informations additionnelles, comme l'historique de dialogue. Nous pourrions aussi considérer l'utilisation des modèles de langage neuronaux (RNN-LM).

Dans le cadre de ces travaux de thèse, il n'était pas possible d'explorer ces propositions. En revanche, dans le but de mieux comprendre l'approche que nous avons proposée, nous avons effectué une analyse des erreurs produites par notre système et proposé une méthode de calcul de mesure de confiance fiable. Ceci sera vu dans le chapitre suivant.

6.4 Conclusion

Nous nous sommes concentrés ici sur la tâche d'extraction des concepts sémantiques dans le cadre de MEDIA. À des fins de comparaison, nous avons mis en œuvre une approche par chaîne de composants afin d'obtenir des performances à l'état de l'art. Ceci nous a permis de confirmer la viabilité de notre approche préliminaire de bout en bout.

Dans la continuité de ces travaux, nous avons proposé une stratégie d'entraînement composée d'une séquence de transfert d'apprentissage guidée par curriculum. Cette approche consiste à apprendre notre modèle tout d'abord avec des tâches plus générales, puis des tâches plus spécifiques. L'emploi de cette stratégie est essentiel à l'obtention de nos meilleurs résultats.

Nous avons par la suite tenté d'agir sur la profondeur de notre modèle neuronal. Ce modèle couplé aux méthodes proposées dans cette thèse nous permet d'obtenir nos meilleurs résultats sur la tâche MEDIA.

L'approche finale que nous mettons en œuvre nous permet d'obtenir des performances légèrement supérieures à celles d'une approche par chaîne de composants à jour, même si ces différences ne sont pas significatives. Il est intéressant de noter que notre approche n'exploite que le signal audio et un modèle de langage pour obtenir ces performances, alors que l'approche par chaîne de composants bénéficie de l'enrichissement d'informations linguistiques obtenues à l'aide d'outil externe de traitement automatique du langage naturel.

Les travaux de ce chapitre concernant notre application d'une stratégie de curriculum ont conduit aux publications scientifiques [Caubrière, Tomashenko, Laurent et al. 2019; Caubrière, Tomashenko, Estève et al. 2019].

En raison des contraintes de temps liés à la durée d'une thèse, plutôt que d'optimiser davantage notre approche qui obtient déjà des performances à l'état de l'art, nous avons choisi pour la suite de nos travaux d'effectuer une analyse des erreurs produites par le système.

Enfin, en nous inspirant de travaux similaires effectués pour la reconnaissance de la parole, nous proposons dans le chapitre suivant une mesure de confiance associée aux concepts sémantiques reconnue par notre système.

Chapitre 7

Analyse d'erreurs et exploitation de représentations internes

Sommaire

Nos deux premiers chapitres de contribution se sont concentrés sur la mise en œuvre d'un système de bout en bout dédié à la compréhension de la parole. Cette mise en œuvre a été réalisée avec succès pour les tâches de reconnaissance des entités nommées et d'extraction de concepts sémantiques.

Nous proposons maintenant de réaliser une étude de notre approche dont l'objectif est de comprendre l'origine des erreurs produites.

Ces travaux nous ont conduit à l'extraction d'une représentation interne des concepts relatifs à la compréhension du langage. Nous exploitons initialement ces représentations dans un but d'analyse. Puis, nous les envisageons comme source d'informations permettant l'établissement d'une mesure de confiance relative aux concepts émis.

Nous découpons ce chapitre en fonction des points mentionnés précédemment. Nous proposons donc d'abord de décrire le contexte de notre étude, puis les axes choisis d'analyses des erreurs de notre approche. Nous détaillons ensuite notre méthode pour l'extraction des représentations internes de la sémantique, ainsi que les visualisations associées. Enfin, nous effectuons la description de l'approche que nous proposons pour établir une mesure de confiance concernant l'émission de concepts, suivi de la présentation des résultats des expérimentations réalisées pour confirmer son intérêt.

7.1 Contexte de l'analyse

Nous portons notre étude sur le système que nous avons mis en œuvre pour la tâche d'extraction des concepts sémantiques. Nous effectuons donc notre analyse sur les concepts sémantiques émis dans le cadre de la réservation d'hôtel MEDIA.

Ces travaux d'analyse ont été conduits en parallèle des travaux d'optimisation de la profondeur du système détaillés en section 6.3. Ainsi, notre étude ne portera pas sur le système entièrement optimisé obtenant les performances les plus élevées, mais sur le meilleur à son commencement. Ce système est appris à l'aide de transferts successifs pilotés par la stratégie de curriculum décrite dans le chapitre précédent.

Nous reportons les résultats de ce système dans la table 7.1. Ces résultats sont exprimés en taux d'erreur sur les concepts (CER) et en taux d'erreur sur les concepts et leurs valeurs (CVER) pour l'ensemble de développement de MEDIA. Nous donnons les résultats pour les sorties neuronales directes du système (*greedy*) et suite à l'exploitation du modèle de langage 5-gramme (*beam search*). Il s'agit du même modèle de langage que celui exploité dans le chapitre précédent. Nous fournissons également les résultats issus de l'utilisation du mode étoile que nous avons proposé dans cette thèse.

Dans la table 7.2, nous effectuons le report des résultats des mêmes approches pour l'ensemble de tests de MEDIA.

	Greedy		*Beam Search*	
Système	**CER**	**CVER**	**CER**	**CVER**
1. "$RAP_5 \rightarrow REN_5 \rightarrow PM{+}M_5 \rightarrow M_5$"	22,0	28,0	19,1	22,9
2. "$RAP_5 \rightarrow REN_5 \rightarrow PM^{*}{+}M^{*}{}_5 \rightarrow M^{*}{}_5$"	20,5	27,1	27,1	21,8

TABLE 7.1 – Résultats de notre approche de bout en bout exploitée pour l'analyse des erreurs de sorties sur l'ensemble de développement de MEDIA. Ces résultats sont reportés des tables 6.5, 6.6 et 6.7.

	Greedy		*Beam Search*	
Système	**CER**	**CVER**	**CER**	**CVER**
1. "$RAP_5 \rightarrow REN_5 \rightarrow PM{+}M_5 \rightarrow M_5$"	21,6	27,7	18,1	22,1
2. "$RAP_5 \rightarrow REN_5 \rightarrow PM^{*}{+}M^{*}{}_5 \rightarrow M^{*}{}_5$"	20,1	26,9	16,4	20,9

TABLE 7.2 – Résultats de notre approche de bout en bout exploitée pour l'analyse des erreurs de sorties sur l'ensemble de tests de MEDIA. Ces résultats sont reportés des tables 6.5, 6.6 et 6.7.

Dans ce chapitre, nos travaux porterons sur les sorties de ces deux systèmes. Nous privilégions toutefois le système *1.*, dans la mesure où il permet l'émission de l'ensemble de la transcription cible et donc du contexte. Nous commençons par effectuer l'analyse des erreurs de sorties, que nous détaillons dans la prochaine section.

7.2 Analyse d'erreurs

Un système neuronal de compréhension de la parole de bout en bout doit nécessairement calculer une représentation interne riche d'informations. Il doit en effet émettre à la fois les mots de la transcription et les concepts sémantiques associés. Le système doit effectuer en une seule fois, une tâche de transcription de la parole, une tâche de segmentation en concepts sémantiques et une tâche de classification de ces concepts.

Pour apporter des éléments de compréhension sur le fonctionnement interne de notre système, nous proposons d'effectuer trois analyses distinctes. Nous étudions tout d'abord les capacités de classification et de segmentation de notre système en analysant la distribution des erreurs d'émission de concept. Nous considérerons ensuite une analyse qualitative de la tâche de transcription réalisée par notre système. Enfin, nous proposons une méthode visant à améliorer sa capacité de segmentation.

7.2.1 Distribution des types d'erreurs

L'objectif de cette analyse est d'étudier les erreurs commises par notre approche de bout en bout en fonction de leur type, c'est-à-dire en fonction des insertions, substitutions et suppressions. Nous réalisons cette étude sur les sorties de l'ensemble de développement de MEDIA, dans le cas du système n'exploitant pas notre mode étoile.

Approche de bout en bout

Nous reportons dans la figure 7.1, la distribution des erreurs observées pour l'approche de bout en bout. La tâche MEDIA comporte plus de 70 classes sémantiques, afin de conserver une représentation lisible, nous ne conservons dans cette figure que les 30 classes sémantiques pour lesquelles le système émet le plus d'erreurs.

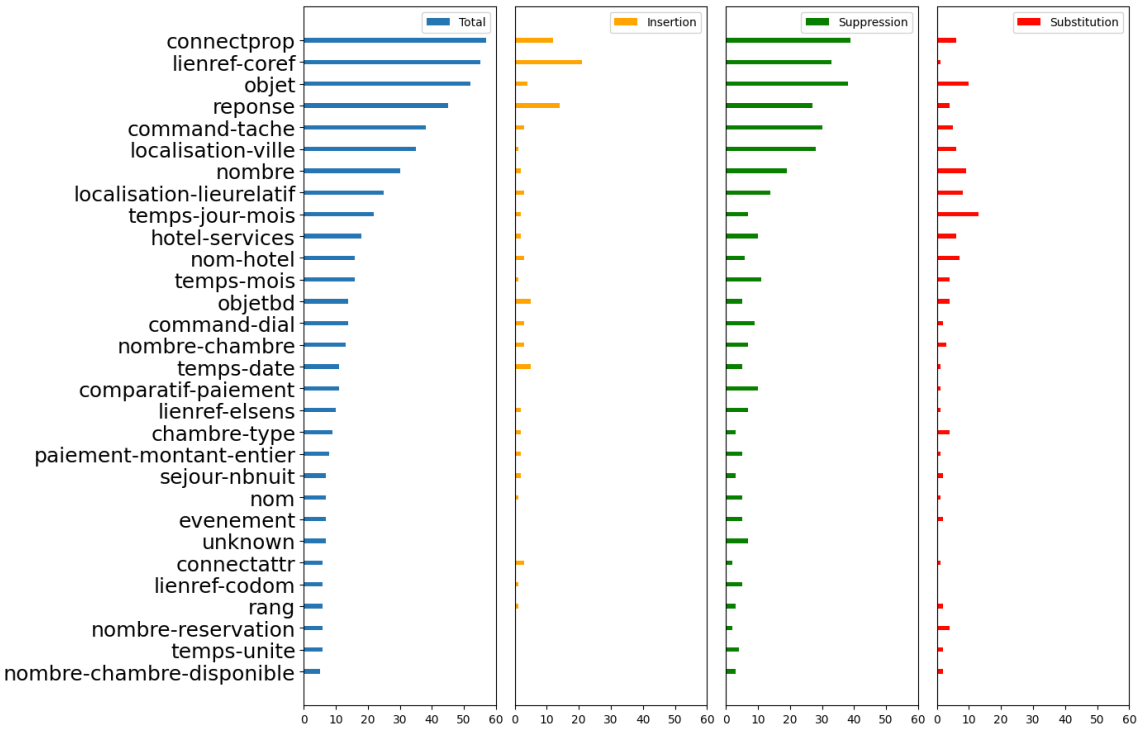

FIGURE 7.1 – Distribution des erreurs de notre approche de bout en bout pour l'ensemble de développement de MEDIA. Extraction des 30 concepts sémantiques avec le plus d'erreurs.

Sur cette figure, nous pouvons observer qu'une grande partie des erreurs produites par le système proviennent principalement de quelques classes sémantiques. Nous pouvons notamment citer les concepts *connectProp*, *lienref-coref*, *objet*, *reponse*, *command-tache*, *localisation-ville* et *nombre*, qui représentent près de 50 % des erreurs de l'ensemble de développement. Notons toutefois que ces concepts représentent 43,4 % des références de l'ensemble de développement.

Cette figure nous montre également que le type d'erreur le plus courant correspond aux suppressions des concepts sémantiques. Ce taux plus élevé de suppressions pourrait être un indicateur de l'apprentissage d'une représentation sous-optimale des concepts sémantiques. L'augmentation de la quantité et de la diversité des annotations sémantiques manuelles pour la tâche MEDIA pourrait être une piste de réduction de ces erreurs.

Approche par chaîne de composants

En complément, nous proposons d'effectuer la même représentation des erreurs de sortie pour l'approche par chaîne de composants. Cette représentation peut être comparée aux erreurs de l'approche de bout en bout afin de déterminer si les erreurs sont propres au domaine de MEDIA ou aux systèmes employés.

Nous fournissons dans la figure 7.2 la distribution des erreurs de la chaîne de composants état de l'art (cf section 6.1.1). Il est à noter que les performances de cette chaîne sur l'ensemble de développement de MEDIA sont de 18,2 points de CER et 28,0 de CVER.

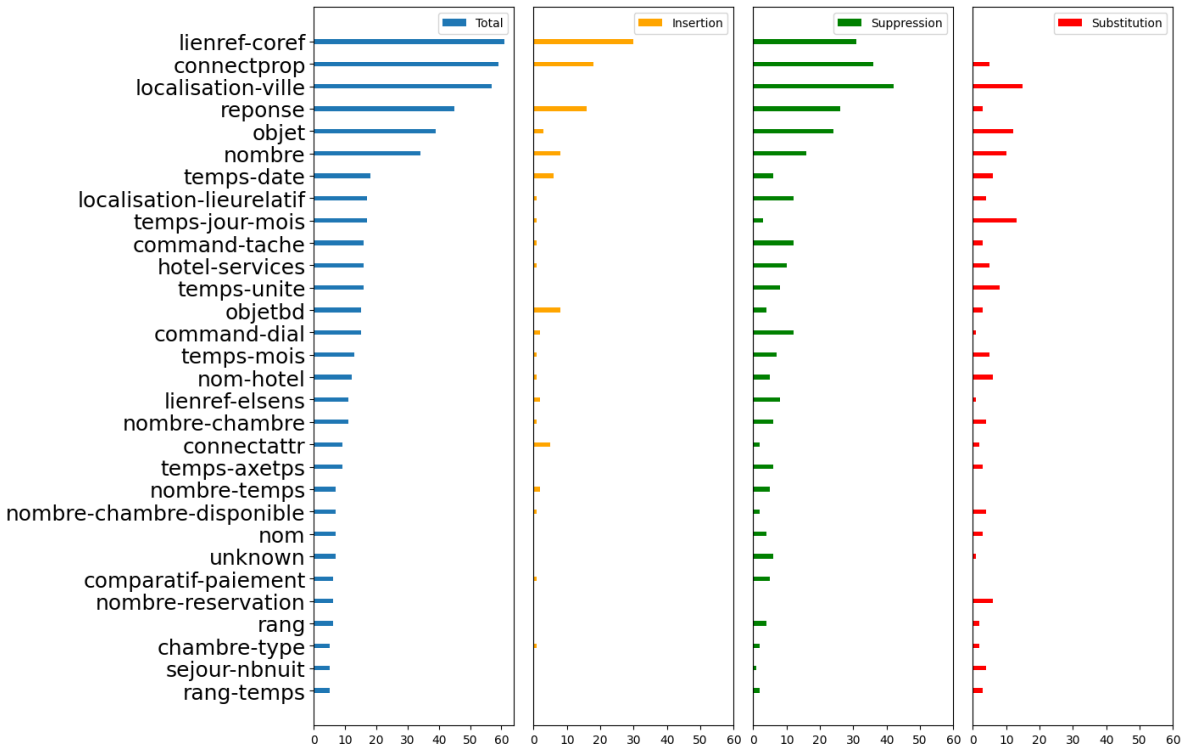

FIGURE 7.2 – Distribution des erreurs de la chaîne de composants pour l'ensemble de développement de MEDIA. Extraction des 30 concepts sémantiques avec le plus d'erreurs.

Cette figure nous permet d'observer que les deux distributions d'erreurs semblent similaires. Pour la chaîne de composants nous remarquons aussi un type d'erreur principale correspondant aux suppressions, ainsi qu'un regroupement des erreurs autour de quelques concepts.

Par comparaison des sept concepts avec le plus d'erreurs entre les deux approches, nous pouvons remarquer que six concepts sont partagés entre les deux approches. Nous retrouvons ainsi une quantité importante des erreurs sur les concepts sémantiques communs : *connectProp*, *lienref-coref*, *objet*, *reponse*, *localisation-ville* et *nombre*. Ces concepts communs semblent être

révélateurs d'erreurs fréquentes liées à la tâche MEDIA et non d'erreurs propres aux systèmes employés.

La comparaison de ces distributions montre que les deux erreurs les plus fréquentes correspondent aux concepts *connectProp* et *lienref-coref*. Ces concepts sont indépendants du domaine et correspondent à des opérateurs logiques, comme le mot *et*, ou à des références, comme le mot *il*.

Il s'agit de valeurs de concepts relativement courtes et souvent portées par un seul mot. Nous émettons l'hypothèse que ces valeurs de concepts sont mal transcrites, impliquant un taux de suppression élevé de ces concepts.

Dans la prochaine sous-section, nous détaillons l'analyse effectuée sur la capacité de notre système à reconnaitre les mots supports de concepts.

7.2.2 Problème de reconnaissance des mots

Nous proposons de porter l'analyse de reconnaissance des mots, sur notre approche, pour les deux concepts hors domaine mentionnés dans la section précédente. C'est-à-dire, *connectProp* et *lienref-coref*. Nous étendons aussi cette analyse au concept *objet*. Il s'agit en effet du troisième concept avec le plus d'erreurs dans le cas de l'approche considérée. Nous effectuons cette analyse pour les cas où il s'agit d'une erreur de suppression des concepts.

Nous nous intéressons au concept *connectProp*, car c'est une connexion logique portée très souvent par un unique phonème (celui du mot 'et'). Ce concept n'apparait que dans le cas d'une connexion de deux autres concepts relatifs au domaine, par exemple "toujours **<hotel-services** le chien **> <connectProp** et **> <comparatif-paiement** maximum **> <paiement-montant-entier** soixante cinq **> <paiement-monnaie** euros **>**".

Nous nous intéressons également au concept *lienref-coref*, puisqu'il s'agit de la référence à un co-référent. Cette co-référence peut représenter un élément présent dans l'historique de dialogue et non dans la séquence actuellement considérée. Ce concept est représenté par des valeurs courtes qui peuvent être difficiles à détecter et à segmenter, par exemple "l'**<objetBD** hôtel **>** où **<lienRef-coRef** la **> <objet** chambre **>**".

Enfin, nous considérons le concept *objet* pertinent car sa valeur peut être liée au domaine applicatif et il nécessite de lever des ambiguïtés en exploitant un contexte parfois difficile à caractériser. Par exemple, "dans **<objet** le même type **>**".

Pour effectuer l'analyse des transcriptions automatiques, nous proposons de distinguer trois cas :

1. Le système a produit une transcription automatique correcte et le concept n'est pas reconnu (à l'exclusion du troisième cas).
2. Le système n'a pas produit une transcription automatique correcte et ne peut donc pas reconnaitre le concept.

3. Le système a produit une transcription automatique correcte, mais a imbriqué le concept ciblé dans un concept adjacent.

Nous reportons dans la table 7.3, les résultats de notre analyse sur les trois concepts mentionnés.

Concept considéré	Nombre de suppression	1. RAP correcte	2. RAP incorrecte	3. RAP imbriquée
connectProp	39	**28**	6	5
lienref-coref	33	**19**	10	4
objet	38	**31**	4	3

TABLE 7.3 – Nombre d'erreurs de suppression en fonction de la transcription automatique. Résultats de l'analyse sur l'ensemble de développement de MEDIA.

Les résultats de cette table montrent que dans une grande majorité des cas, la transcription automatique des mots supports de concepts est correcte. Sa qualité n'est ainsi pas une source de suppression des concepts, ce qui indique qu'il s'agit principalement d'un problème d'étiquetage sémantique.

Lors de cette analyse, nous avons régulièrement observé la présence d'une frontière de fin de concept (">") sans la présence d'une frontière de début. Par exemple, pour le concept *connectProp* nous avons pu dénombrer 11 cas de ce type sur l'ensemble des 39 concepts supprimés, ce qui représente environ 28 % des suppressions de ce concept sémantique.

Ce phénomène est révélateur d'un problème de segmentation de la part de notre approche de bout en bout. Nous proposons donc dans la prochaine sous-section de décrire notre proposition de prise en charge de ce problème.

7.2.3 Problème de segmentation en concept

Pour réduire le problème de segmentation observé, nous proposons d'ajouter une nouvelle tâche dans notre chaîne d'apprentissages successifs. Nous envisageons de la séparer en deux étapes puisque nous observons l'absence d'une frontière de début de concept pour la tâche finale MEDIA. Ces deux étapes correspondent d'abord à une tâche de segmentation (M_{seg}), puis à une tâche de segmentation et de classification des concepts sémantiques tels que nous le faisons jusqu'à présent (M).

Pour la tâche M_{seg}, nous proposons de remplacer la frontière ouvrante des concepts par un unique symbole représentant la segmentation (<). Nous proposons donc à nouveau une modification des séquences ciblées pendant l'apprentissage. Nous donnons dans la figure 7.3 un exemple de séquence annotée pour la tâche de segmentation.

Afin de respecter la spécialisation successive des apprentissages de notre approche de bout

Séquence de mots enrichie	le sculpteur <pers césar > est mort <time hier > à <loc paris > à l'âge de <amount soixante dix sept ans >
Séquence pour la segmentation	le sculpteur < césar > est mort < hier > à < paris > à l'âge de < soixante dix sept ans >

FIGURE 7.3 – Représentation d'une séquence pour l'entraînement d'une tâche de segmentation.

en bout, nous effectuons l'entraînement de la tâche de segmentation avant celle de classification. Nous apprenons donc le système $RAP \rightarrow REN \rightarrow PM+M \rightarrow M_{seg} \rightarrow M$.

Nous donnons les résultats de l'entraînement des deux systèmes mentionnés dans la table 7.4. Ces résultats sont donnés en CER et CVER pour les ensembles de développement et de test de MEDIA, pour une sortie neuronale directe (*greedy*).

	Développement		*Test*	
chaîne d'apprentissage	CER	CVER	CER	CVER
"$RAP \rightarrow REN \rightarrow PM+M \rightarrow M$"	22,0	28,0	21,6	27,7
$RAP \rightarrow REN \rightarrow PM+M \rightarrow M_{seg} \rightarrow M$	**20,6**	**26,8**	**20,7**	**27,2**

TABLE 7.4 – Résultats de l'approche de bout en bout exploitant une tâche de segmentation pour les sorties neuronales immédiates (greedy). Les résultats encadrés par des guillemets sont reportés de la table 6.5.

Ces résultats confirment l'intérêt de la tâche de segmentation par une amélioration du CER de 0,9 point et du CVER de 0,5 point pour l'ensemble de tests. Nous pouvons également observer l'amélioration sur l'ensemble de développement avec un gain de 1,4 point de CER et de 1,2 point de CVER.

En complément, nous donnons les résultats de ces systèmes suite à l'exploitation du modèle de langage 5-gramme et de l'algorithme de beam search. Nous les fournissons dans la table 7.5.

	Développement		*Test*	
chaîne d'apprentissage	CER	CVER	CER	CVER
"$RAP \rightarrow REN \rightarrow PM+M \rightarrow M$"	19,1	22,9	18,1	22,1
$RAP \rightarrow REN \rightarrow PM+M \rightarrow M_{seg} \rightarrow M$	**18,5**	**22,5**	**17,8**	**21,8**

TABLE 7.5 – Résultats de l'approche de bout en bout exploitant une tâche de segmentation après exploitation d'un modèle de langage 5-gramme (beam search). Les résultats encadrés par des guillemets sont reportés de la table 6.6.

Ces résultats nous montrent que nous conservons l'apport de la tâche de segmentation dans le cas de l'utilisation du modèle de langage. Nous pouvons donc conclure de ces résultats qu'il

est préférable de séparer la tâche finale ciblée en deux étapes, avec tout d'abord l'apprentissage de la segmentation, puis l'apprentissage complet de la classification en concepts.

Enfin, pour compléter les analyses réalisées jusqu'ici, nous proposons d'apporter des éléments supplémentaires. Nous envisageons d'extraire de notre système une représentation interne des concepts sémantiques à des fins d'analyse. Dans la prochaine section nous détaillerons notre méthode d'extraction de ces représentations.

7.3 Analyse de représentations internes

Notre système neuronal est appris à l'aide de la fonction de coût CTC (Connectionnist Temporal Classification). Nous envisageons de bénéficier des propriétés de cette fonction pour réaliser l'extraction d'une représentation interne des concepts sémantiques.

7.3.1 Extraction des représentations

Dans le cadre de notre système, la fonction de coût CTC permet l'apprentissage d'un alignement entre l'entrée acoustique et une séquence sous forme de texte enrichi par les concepts sémantiques. Suite à l'émission d'une séquence par notre système, la fonction de mapping est appliquée pour supprimer les répétitions des caractères et produire une séquence finale de mots et de concepts (voir section 2.4.1).

À partir de la séquence d'observations d'entrée, une séquence intermédiaire est produite par les couches convolutionnelles. Pour chacun de ses éléments, une représentation interne (*embeddings*) sera propagée dans le système pour effectuer l'émission d'un caractère. Nous proposons d'exploiter ces représentations internes. Nous donnons une représentation de cette extraction pour la dernière couche récurrente du système principale dans la figure 7.4.

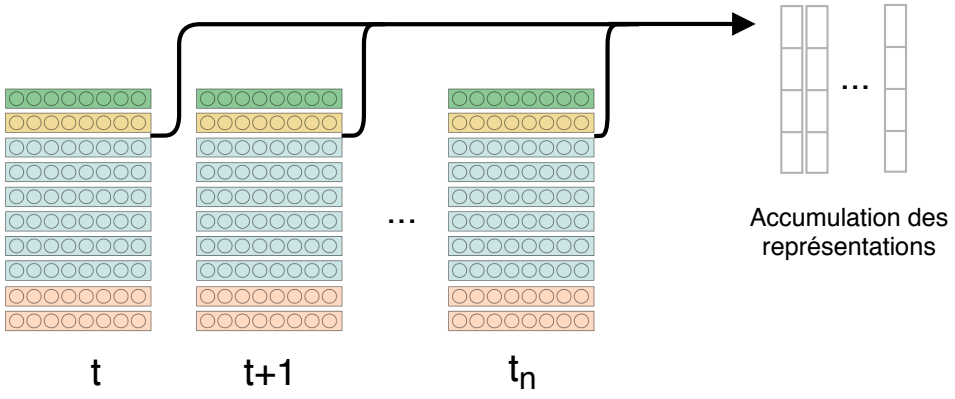

Figure 7.4 – Représentation de l'extraction des représentations de caractères à chaque temps t. Exemple pour une extraction de la dernière couche récurrente du système.

Nous proposons d'exploiter l'alignement appris par le système pour effectuer l'extraction de représentations internes de concepts. Cela se traduit par l'accumulation des représentations de chacun des caractères composants le concept sémantique ciblé, à partir d'observations sur la séquence émise par le système. Nous considérons que la valeur d'un concept correspond aux mots supports de ce concept et par conséquent nous accumulons aussi les représentations des caractères des mots supports de concept, ainsi que du marqueur de fin de concepts.

Nous présentons dans la figure 7.5, l'exemple d'une séquence de sortie du système avant réduction des répétitions et la sélection des représentations effectuées.

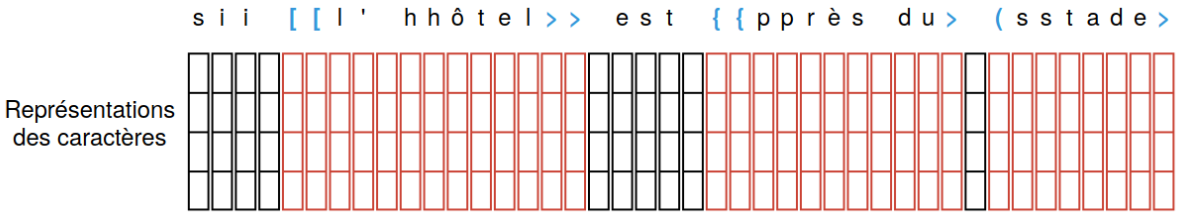

Figure 7.5 – Exemple de sorties immédiates du système pour la séquence "si [l' hôtel > { est près du > (stade >". En rouge, les représentations internes sélectionnées pour représenter les concepts associés. [correspond au concept *nom-hotel*, { correspond à *localisation-distanceRelative* et (correspond à *localisation-lieuRelatif*.

Nous exploitons les représentations des concepts sémantiques afin d'effectuer la visualisation des informations sémantiques capturées par notre système. Ainsi, nous extrayons les représentations issues des couches récurrentes de notre système neuronal. Pour rappel, notre système exploite un empilement de deux couches CNN, cinq couches bLSTM, une couche linéaire et une couche de sortie softmax. Nous souhaitons visualiser les représentations issues de la dernière couche récurrente bLSTM. Nous présentons cette visualisation dans la prochaine sous-section.

7.3.2 Visualisation des représentations

Comme nous l'avons vu, nous accumulons les représentations de chacun des caractères émis correspondants aux concepts sémantiques. Cela signifie que nous pouvons accumuler un nombre variable de représentations internes pour un concept.

Afin d'homogénéiser la taille de notre représentation des concepts, nous proposons d'effectuer une transformation moyennant les représentations accumulées. Cette moyenne nous permet d'exploiter un vecteur de taille fixe comme représentation des concepts, plutôt qu'une matrice dont la largeur dépendrait du nombre de caractères émis pour le concept.

En complément, nous souhaitons effectuer la visualisation de ces représentations dans un

espace à deux dimensions. Nous appliquons donc une autre transformation sur la représentation moyennée des concepts permettant de réduire ce vecteur à deux dimensions.

Elle consiste cette fois en une transformation *t-Distributed Stochastic Neighbor Embedding* (t-SNE). Il s'agit d'un algorithme d'apprentissage machine régulièrement utilisée pour visualiser des données de dimensions importantes dans des dimensions interprétables par l'humain [Maaten et G. Hinton 2008].

Nous proposons dans la figure 7.6, la visualisation des représentations des concepts sémantiques pour la dernière couche récurrente de notre système. Au sein de cette figure, chaque couleur représente une classe sémantique distincte et chaque point représente un concept.

En complément, nous proposons une seconde visualisation exploitant une coloration non plus en fonction de la classe sémantique des concepts, mais en fonction de la réponse du système. Nous représentons ainsi en vert les concepts correctement émis par le système et en rouge les erreurs.

Ces représentations sont obtenues pour les concepts sémantiques de l'ensemble de développement de MEDIA pour les sorties immédiates (*Greedy*) du système neuronal dénombré *1.* dans la table 7.1.

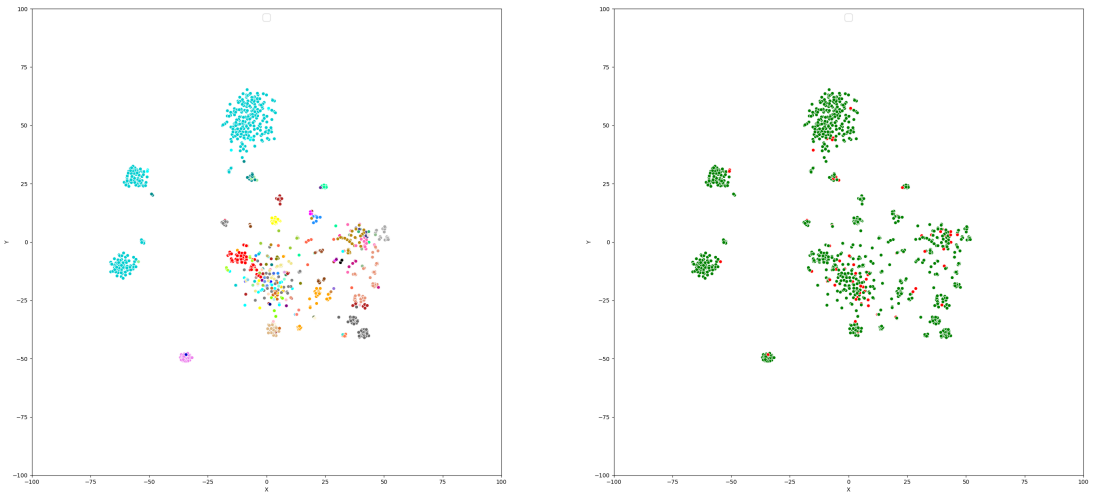

Figure 7.6 – Visualisation des représentations de concepts sémantiques par projection t-SNE pour l'ensemble de développement de MEDIA. À gauche, la coloration des points représente la classe sémantique associée à la projection. À droite, la couleur verte représente les concepts correctement émis par le système et la couleur rouge représente les erreurs.

À partir de ces visualisations nous observons un effet de regroupement en classes sémantiques dans l'espace continu. Ce regroupement par classe est un indicateur de la présence d'in-

formations sémantiques capturées par notre système. Nous observons aussi que la majorité des erreurs produites par le système sont présentes dans la zone centrale où les projections ne montrent pas des regroupements correctement définis. Ce phénomène est révélateur de l'apprentissage d'une représentation de la sémantique encore imparfaite, soulevant la problématique de l'entraînement d'un système modélisant une représentation plus pertinente.

Pour compléter l'analyse des représentations internes de notre système, nous proposons d'exploiter des classifieurs externes. Nous envisageons d'utiliser ces classifieurs pour obtenir une information complémentaire permettant l'évaluation de la qualité des représentations internes. En comparant des représentations issues de différentes couches neuronales nous pouvons localiser la couche capturant l'information sémantique la plus pertinente.

L'exploitation des représentations internes des concepts pour entraîner un classifieur externe est inspiré des travaux de [Belinkov et Glass 2017]. Ces travaux consistent à extraire les représentations de la phonétique d'un système de RAP et à apprendre un classifieur externe pour en effectuer l'analyse. Il est aussi à noter que ces travaux exploitent aussi une architecture neuronale proche de notre système de compréhension de la parole.

Dans les prochaines sous-sections, nous détaillons notre implémentation des classifieurs externes, ainsi que leurs performances de classifications pour les représentations internes des concepts extraites de différentes couches récurrentes.

7.3.3 Entraînement de classifieurs externes

L'exploitation que nous proposons d'un classifieur externe consiste en l'entraînement d'un système neuronal composé d'une couche permettant la représentation d'informations puis une couche de sortie softmax. Nous entraînons ce classifieur pour l'émission des classes sémantiques de MEDIA, à partir de l'extraction des représentations internes des concepts émis par notre système principal de bout en bout.

Pour notre classifieur, nous envisageons deux types de couches neuronales de représentation de l'information. Le premier type correspond à une couche classique totalement connectée (MLP), tandis que le second correspond à une couche récurrente de type LSTM bidirectionnelle.

Notre système de compréhension de la parole émet une représentation interne pour chaque caractère de sa séquence complète de sortie. Comme plusieurs caractères peuvent représenter un concept, nous avons jusque là exploité une représentation finale correspondant à la moyenne des représentations des caractères. Nous proposons donc l'emploi d'une couche récurrente bLSTM pour profiter de l'ensemble de la séquence de représentation des caractères plutôt que d'une représentation des concepts moyennées.

Nous schématisons l'interaction du classifieur externe proposé avec le système principal dans la figure 7.7.

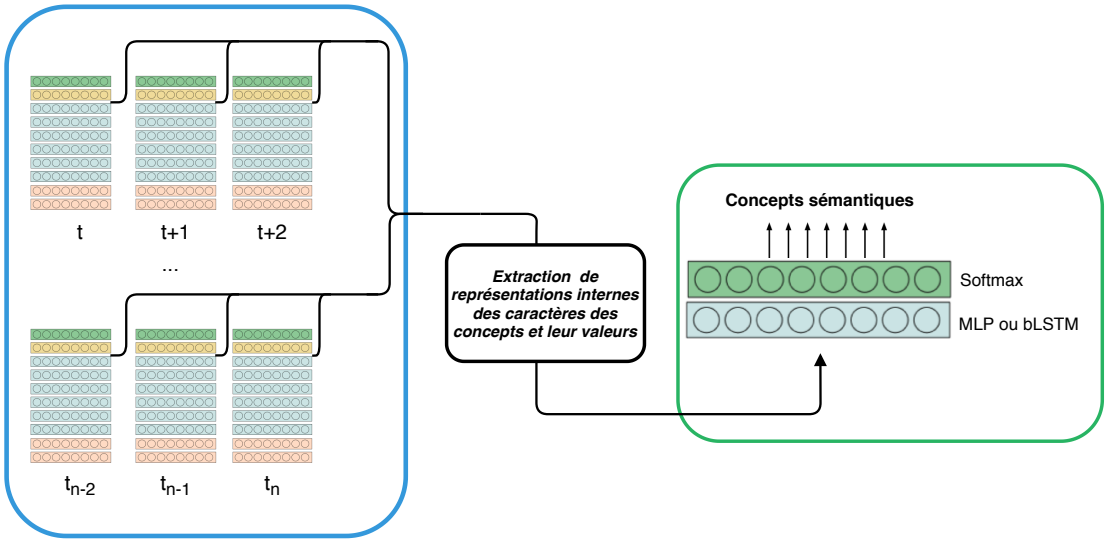

FIGURE 7.7 – Schéma de la mise en œuvre de notre classifieur externe sur les représentations internes de notre système de bout en bout. Le système principal de compréhension de la parole est encadrée en bleu et le classifieur externe encadré en vert.

Paramètres des classifieurs

Nous proposons d'exploiter deux classifieurs différents s'appuyant sur les deux couches neuronales d'extraction de l'information que nous avons mentionnée.

Le premier correspond à l'utilisation d'une couche de type perceptron multicouche. Ce classifieur exploite en entrée une représentation moyennée des caractères des concepts sémantiques de dimension 800. Nous définissons la largeur de la couche neuronale cachée à 200, puis la taille de la couche de sortie softmax est de 76. Cette taille correspond au nombre de classes sémantiques pouvant être émises.

Nous effectuons l'apprentissage de ce premier classifieur à l'aide de la fonction de coût d'entropie croisée et selon la méthode d'optimisation Adam. Nous choisissons d'exploiter les paramètres par défaut de l'implémentation de cet optimiseur, à savoir $\beta_1 = 0,9$; $\beta_2 = 0,999$ et $\epsilon = 10^{-8}$. Il est à noter que ϵ est un paramètre très faible, empêchant la division par 0. Nous utilisons un taux d'apprentissage paramétré initialement à 10^{-4}.

Nous présentons les données d'entrées du classifieur par mini-lot (*minibatchs*) de taille 20 et nous entraînons le système pour 100 époques. Nous sélectionnons le modèle optimisé en fonction de la précision obtenue sur l'ensemble de développement de MEDIA pour la catégorisation en concepts sémantiques.

Enfin, pour l'entraînement et l'optimisation d'un modèle classifieur, nous exploitons uniquement les représentations des concepts sémantiques correctement émis par le système principal de compréhension de la parole.

Le second classifieur que nous mettons en œuvre exploite une couche neuronale de type LSTM bidirectionnelle. La différence principale réside dans la prise en charge de l'ensemble de la séquence de représentation des caractères composants un concept et sa valeur, plutôt que la représentation moyennée. Cela signifie que nous exploitons en entrée du classifieur une séquence de taille n de vecteurs à 800 dimensions.

Nous effectuons son apprentissage selon la même paramétrisation que le premier classifieur décrit précédemment. C'est-à-dire, selon l'optimiseur Adam et la fonction de coût d'entropie croisée pour 100 époques avec des mini-lots (*minibatchs*) de taille 20. Nous exploitons à nouveau une couche cachée de taille 200 et une couche de sortie softmax de taille 76. Nous sélectionnons le modèle optimisé selon la précision calculée sur l'ensemble de développement.

Performances de classification

Nous effectuons la comparaison des performances des deux classifieurs proposés par expérimentations. Cette comparaison nous permet d'évaluer la qualité de la représentation moyennée par rapport à l'exploitation de la séquence complète de représentation des concepts.

Nous proposons aussi une comparaison des représentations internes extraites de chacune des couches récurrentes de notre système principal. L'objectif étant d'observer l'évolution des performances des classifieurs sur les représentations des couches cachées pour mettre en avant les capacités de captures d'informations sémantiques de notre système principal.

Nous reportons dans la table 7.6, les résultats de l'apprentissage des deux classifieurs sur les représentations internes des cinq couches cachées de notre système principal.

Couche d'extraction des représentations internes	**Précision du classifieur MLP**	**Précision du classifieur bLSTM**
1.	70,94%	90,42%
2.	82,15%	94,32%
3.	90,24%	97,14%
4.	**96,87%**	98,73%
5.	95,41%	**99,32%**

Table 7.6 – Comparaison des représentations moyennées et des séquences de représentation en entrée des classifieurs externes, en fonction de la précision sur l'ensemble de développement de MEDIA. Représentation interne extraite du système principal pour les concepts sémantiques correctement reconnus.

Ces résultats nous montrent l'importance de l'exploitation de l'ensemble de la séquence de représentation des caractères composants les concepts, plutôt que l'exploitation de la représentation moyennée. Nous observons que les performances de classification sont systématiquement supérieures pour le classifieur s'appuyant sur une couche bLSTM, avec des performances finales à 99, 32 % de précision.

Cela s'explique par la représentation d'entrée du classifier bLTM complète, contrairement à la moyenne des représentations utilisée pour le classifieur MLP qui implique une perte d'informations en terme de dynamique et de variabilité. L'architecture neuronale bLSTM permet d'exploiter les représentations selon les contextes passés et futurs, modélisant ainsi davantage de dépendances. Enfin, la précision finale élevée s'explique par l'exploitation des représentations des concepts sémantiques correctement reconnus par le système principal pour ces expérimentations.

Nous pouvons aussi observer l'amélioration progressive des représentations internes du système de bout en bout par une augmentation de la précision des classifieurs en fonction de la couche cachée d'extraction des représentations. Cette amélioration est révélatrice de la capacité du système principale à extraire une information sémantique de plus en plus riche au fil de ses couches cachées.

Enfin, ces résultats nous montrent la viabilité de l'apprentissage d'un classifieur externe à partir des représentations internes. Les visualisations des représentations nous ont montré que leur position dans l'espace continu semble fournir une information pertinente pouvant être exploitée pour effectuer une détection des incertitudes.

Nous envisageons pour la suite de nos travaux de nous concentrer sur l'exploitation de cette information en proposant une méthode d'extraction d'une mesure de confiance pour l'émission des concepts par notre système. Nous nous concentrons sur l'exploitation des classifieurs externes pour la production de cette mesure concernant l'émission de concepts sémantiques. Nous détaillons la méthode proposée d'extraction et l'évaluation de l'information apportée par cette mesure dans la prochaine section.

7.4 Mesure de confiance

Les mesures de confiances sont des mesures permettant l'évaluation de la fiabilité d'un résultat produit par un système. Cette évaluation fournit une information additionnelle importante pour la détection automatique de certaines erreurs de sortie d'un système. Ce type de mesure a été utilisé avec succès dans le domaine de la reconnaissance de la parole depuis des années [Jiang 2005 ; Ghannay, Estève et al. 2015].

Elles ont également été étudiées dans le cadre de la compréhension de la parole [Hazen et al. 2002 ; Raymond, Esteve et al. 2003 ; Hakkani-Tür, Béchet et al. 2006], y compris dans le cadre de MEDIA [Simonnet, Ghannay, Camelin, Estève et De Mori 2017]. Nous réalisons donc l'étude de la mise en œuvre d'une mesure de confiance concernant l'émission des concepts sémantiques par notre approche de bout en bout.

Nous décrivons dans les prochaines sous-sections, notre méthode d'extraction de la mesure de confiance, ainsi que les expérimentations menées pour confirmer sa viabilité.

7.4.1 Extraction de la mesure de confiance

Pour mettre en place la mesure, nous envisageons d'exploiter les sorties de la couche softmax de nos classifieurs externes. Cela consiste après l'apprentissage de l'ensemble des systèmes, à utiliser le score calculé par la fonction softmax du classifieur externe pour la classe sémantique du concept ayant été émis par le système principal. Nous représentons, dans la figure 7.8, l'extraction de la mesure de confiance du concept *nom-hotel* émis dans la séquence "si **<nom-hotel** l' hôtel **>** **<localisation-distanceRelative** est près du **>** **<localisation-lieuRelatif** stade **>**".

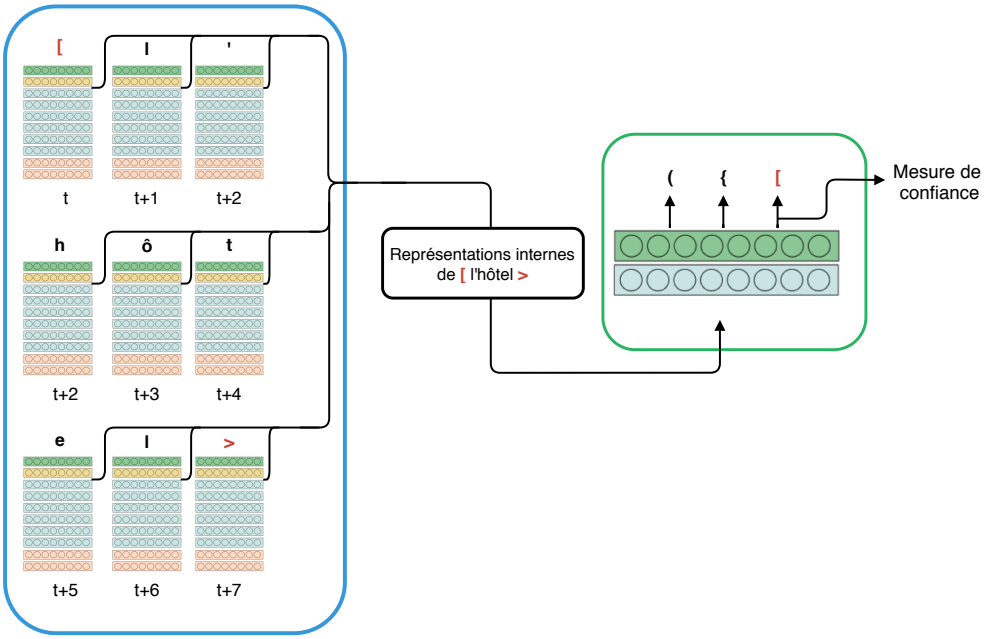

Figure 7.8 – Représentation de l'extraction de la mesure de confiance proposée. [correspond au concept *nom-hotel*, { correspond à *localisation-distanceRelative* et (correspond à *localisation-lieuRelatif*.

Nous nous inspirons des travaux sur les mesures de confiance en reconnaissance de la parole [Evermann et Woodland 2000] pour proposer la calibration des sorties des classifieurs plutôt que leur utilisation directe comme mesure de confiance. Cette calibration correspond à un ensemble de transformations linéaires appliquées par segment (*piece-wise linear mapping*) sur les sorties des classifieurs. Nous définissons la calibration par observation sur l'ensemble de développement de MEDIA, puis réalisons son application sur l'ensemble de tests.

Nous décrivons dans la prochaine sous-section les expérimentations menées autour de la mesure de confiance proposée, ainsi que les résultats obtenus. Nous proposons également d'évaluer la fiabilité de la mesure proposée, c'est-à-dire la quantité d'informations additionnelles qu'elle apporte.

7.4.2 Expérimentations et résultats

Nos expérimentations ont pour objectif de vérifier la viabilité de la mesure de confiance que nous proposons. Nous envisageons de comparer les mesures issues du classifieur MLP et du classifieur bLSTM à une base de référence.

Nous définissons notre base de référence comme la mesure pouvant être extraites des sorties softmax du système principal. Comme plusieurs caractères sont nécessaires pour représenter un concept et sa valeur, nous proposons d'utiliser la moyenne des sorties softmax de chacun des caractères composant le concept et la valeur considérés.

Enfin, les mesures de confiance rendent possible la définition d'un seuil de confiance permettant de rejeter les réponses du système qui peuvent être considérées comme non fiables. Nous proposons donc de comparer les trois mesures mentionnées par l'application d'un seuil progressif. Nous effectuons la comparaison par le calcul de la précision et du rappel pour les concepts non rejetés suite à l'application de ce seuil.

Nous reportons dans la figure 7.9, les courbes de précision obtenues en fonction du rappel par l'application d'un seuil de confiance allant de 0 à 1, par pas de 10^{-6}. Ces résultats sont calculés pour l'ensemble de tests de MEDIA.

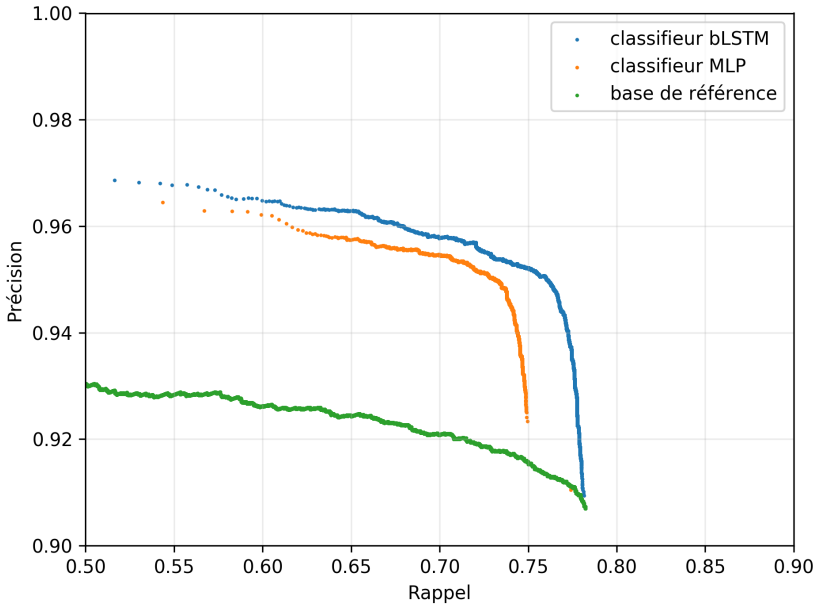

FIGURE 7.9 – Précision en fonction du rappel des concepts sémantiques après application d'un filtrage par seuil de confiance sur l'ensemble de tests de MEDIA pour les concepts émis dans le cadre du système normal (*1.* dans la table 7.1). Seuil appliqué de 0 à 1 par pas de 10^{-6}.

Ces résultats nous montrent l'apport des classifieurs externes pour la production d'une mesure de confiance. Nous pouvons observer que dans tous les cas, la précision en fonction du

rappel est meilleure par l'utilisation d'une mesure de confiance issue des classifieurs externes. Nous notons également que le classifieur bLSTM est le plus performant concernant l'émission de cette mesure, puisqu'il permet une précision systématiquement plus importante pour un rappel identique.

Nous effectuons en complément l'évaluation de la mesure de confiance pour notre système principale appris selon le mode étoile. Il s'agit du système *2.* dans la table 7.1.

Nous reportons dans la figure 7.10, les courbes de précision en fonction du rappel obtenu par le classifieur bLSTM dans le cas des représentations du système principal en mode étoile. À des fins de comparaison, nous reportons aussi la courbe obtenu par le classifieur bLSTM sans utilisation du mode étoile.

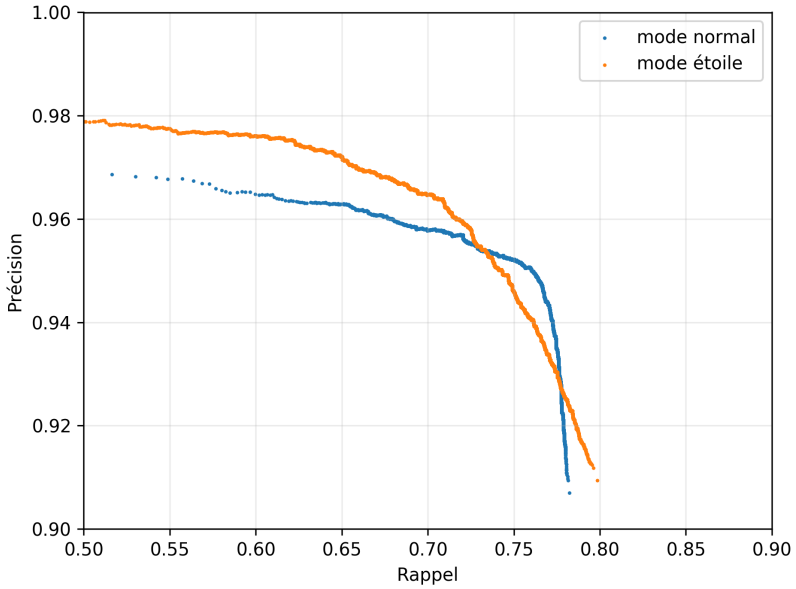

FIGURE 7.10 – Précision en fonction du rappel des concepts sémantiques après application d'un filtrage par seuil de confiance sur la mesure produite par un classifieur bLSTM, pour l'ensemble de tests de MEDIA. Seuil appliqué de 0 à 1 par pas de 10^{-6}.

Sur cette figure, il est intéressant d'observer que le mode étoile ne surpasse pas systématiquement le mode normal. Ce mode permet de meilleures performances finales en terme de CER et CVER, alors que pour un rappel entre 0,74 et 0,78, ce mode ne permet pas une précision suffisante par rapport au mode normal.

Pour compléter nos travaux de mise en place d'une mesure de confiance, nous proposons de calculer la fiabilité des meilleures mesures obtenues. Nous effectuons donc la suite de nos travaux avec les mesures émises par le classifieur bLSTM selon des représentations en mode normal et en mode étoile.

Afin d'évaluer cette fiabilité, nous exploitons la métrique d'entropie croisée normalisée (*Normalised Cross Entropy, NCE*). Il s'agit d'une métrique couramment exploitée pour les mesures de confiance en reconnaissance de la parole [Siu et al. 1997 ; Evermann et Woodland 2000].

La NCE est une mesure théorique de la quantité d'informations supplémentaires fournies par une mesure de confiance. Une mesure supérieure à 0 indique la présence d'informations additionnelles. Également, plus le score NCE est proche de 1, plus la quantité d'informations apportées est importante. Cette métrique est définie par l'équation :

$$NCE = \frac{H_{max} + \sum_{C_{cor}} \log_2(m(C)) + \sum_{C_{incor}} \log_2(1 - m(C))}{H_{max}} \tag{7.1}$$

Avec, $m(C)$ la mesure de confiance associée au concept sémantique courant, et H_{max} défini par :

$$H_{max} = -n\log_2(P) - (N-n)\log_2(1-P) \tag{7.2}$$

Avec, n le nombre de concepts correctement émis, N le nombre total de concepts émis et $P = n/N$ la probabilité moyenne qu'un concept soit correctement reconnu.

Nous effectuons le calcul de cette métrique pour les mesures de confiance produites par le classifieur bLSTM dans le cadre des représentations internes des systèmes en mode normal et en mode étoile. Nous reportons, dans la table 7.7, les scores NCE obtenus pour les mesures sur les ensembles de développement et de test de MEDIA.

Mode du système principale	**Développement**	**Test**
Normale	**0,226**	**0,288**
Étoile	0,195	0,241

Table 7.7 – Fiabilité en score NCE des mesures de confiance produites par le classifieur bLSTM pour chacun des deux modes du système de compréhension de la parole, sur les ensembles de développement et de test de MEDIA.

Ces résultats montrent la pertinence des calibrations définies à l'aide des ensembles de développement en mode normal et en mode étoile. L'application de ces calibrations sur l'ensemble de tests permet d'obtenir des scores de NCE de 0,288 pour le mode normal et 0,241 pour le mode étoile. Ces résultats montrent aussi que les mesures de confiance produites apportent une information additionnelle exploitable.

Il serait ainsi intéressant d'utiliser la mesure de confiance proposée dans le but de rejeter les concepts émis par notre système principal qui peuvent être considérés comme non fiables. Dans de futurs travaux, il peut aussi être intéressant d'explorer les possibilités offertes par la mesure de confiance pour réaliser des corrections d'erreurs.

7.5 Conclusion

Nous avons présenté ici les dernières contributions réalisées dans le cadre de cette thèse. Ces contributions se sont tout d'abord concentrées autour de l'analyse des erreurs effectuées par le système de compréhension de la parole que nous avons mis en œuvre.

Nous avons tout d'abord déterminé par analyse que les erreurs en sortie de notre système affectent principalement une poignée de concepts sémantiques. Également, le type d'erreurs le plus largement représenté concerne des erreurs de suppression, qui peuvent révéler un manque de données manuellement annotées. Il serait intéressant de réaliser une augmentation de données pour accroitre la taille de nos ensembles et vérifier ce point. En portant l'analyse sur les sorties d'un système exploitant une chaîne de composants, nous avons observé un scénario similaire. Cette similarité nous a indiqué que les erreurs produites semblent davantage liées à la tâche MEDIA qu'à notre approche de bout en bout.

Nous avons ensuite envisagé un problème de reconnaissance des mots composants les valeurs des concepts sémantiques, dégradant la qualité des transcriptions automatique de ces valeurs et impliquant l'augmentation de la quantité de suppressions de concepts. Il apparait après analyse que la qualité de la transcription des valeurs ne peut pas être mise en cause, ce qui nous signale un problème d'étiquetage sémantique. Cette analyse de la qualité de reconnaissance des mots nous a permis d'observer un problème de segmentation en concept. Pour contrer ce problème, nous avons proposé l'entraînement de la segmentation comme une tâche à part entière dans la chaîne d'apprentissage de notre système de bout en bout.

Pour compléter ces analyses, nous avons aussi proposé une méthode d'extraction de représentation interne des concepts sémantiques. Nous avons effectué une visualisation en deux dimensions de ces représentations internes des concepts sémantiques. La visualisation a mis en évidence un effet de regroupement en fonction des classes de concepts, et a montré la capacité de notre approche de bout en bout à modéliser des informations sémantiques.

Enfin, nous avons proposé d'exploiter ces représentations internes dans le but de produire une mesure de confiance évaluant la fiabilité de l'émission d'un concept sémantique. Nous avons ainsi mis en place un classifieur externe appris pour émettre une classe sémantique à partir de la représentation interne d'un concept. Nous avons proposé d'exploiter les scores de la fonction softmax en sortie de ce classifieur comme mesure de confiance et les expérimentations menées ont montré la pertinence de la mesure proposée par l'obtention d'un score NCE allant jusqu'à 0,288. L'intérêt de la mise en place d'une mesure de confiance réside dans l'information supplémentaire qu'elle apporte. Elle permet en effet d'envisager le rejet de concepts émis considérés non fiables par un seuil de confiance. Elle peut aussi rendre possible la correction de certaines erreurs émises. Ces pistes pourront être explorées dans de futurs travaux.

Enfin, les contributions décrites dans ce chapitre ont fait l'objet des publications scientifiques [Caubrière, Ghannay et al. 2020 ; Caubrière, Estève et al. 2020].

Chapitre 8

Conclusion et perspectives

Au sein de ce dernier chapitre, nous concluons sur les travaux réalisés dans le cadre de cette thèse. Nous proposons ensuite quelques perspectives pouvant orienter de futurs travaux.

8.1 Conclusion

Dans le cadre de cette thèse, nous avons travaillé à l'élaboration et l'optimisation d'un premier système neuronal entièrement dédié à une tâche de compréhension de la parole qui s'appuie directement sur des observations du signal acoustique.

L'intérêt pour la mise en œuvre d'une telle approche est qu'elle permet de lever un obstacle présent au sein des approches plus classique. Ces approches effectuent la tâche de compréhension par une chaîne de traitements via des représentations symboliques intermédiaire. Ces représentations intermédiaire induisent la propagation d'erreurs au fil de la chaîne de traitements, mais provoque également une perte d'information par la suppression de l'ensemble des informations paralinguistiques présente dans la parole.

Afin de mener à bien nos travaux, nous nous sommes situés dans le cadre applicatif de la compréhension de la parole, que nous avons trouvé dans les tâches de reconnaissance des entités nommées et d'extraction des concepts sémantiques. Il s'agit de tâches de compréhension similaires, qui peuvent être prises en charge par des méthodes d'apprentissage supervisé. La différence entre ces deux tâches provient de leur représentation de la sémantique, c'est-à-dire du sens. Les entités nommées sont définies comme des briques élémentaires de l'information générale contenue dans un document, tandis que les concepts sémantiques dans le contexte de MEDIA sont ultraspécialisés pour la tâche de réservation d'hôtel.

Dans notre cadre applicatif, nous faisons face à une problématique de rareté des données de parole manuellement transcrites et annotées sémantiquement. Ainsi, tout au long de nos travaux, nous avons mis en place des stratégies d'augmentations de données, et d'entraînements de nos systèmes répondant à cette problématique.

Dans nos contributions, nous exploitons les avancées dans le domaine de la reconnaissance de la parole comme point de départ. Dans ce domaine, de premiers systèmes ont effectués la projection d'informations de la dimension acoustique vers une représentation textuelle de la parole, à travers une architecture neuronal récurrente exploitant la fonction de coût CTC

(*Connectionnist Temporal Classification*). L'intérêt de cette fonction de coût est qu'elle permet à un système d'apprendre l'alignement entre un segment audio et la représentation textuelle du discours prononcé.

Nos travaux s'appuient sur cette fonction de coût en effectuant l'enrichissement des représentations textuelles par des marqueurs de la sémantique. Ainsi, nous entraînons nos systèmes en alignement les segments audio et leur transcriptions enrichis de ces marqueurs.

Dans un premier temps, nos travaux visent à mettre en œuvre un système de ce type dans un cadre simplifié de reconnaissance des entités nommées. En abordant ensuite la reconnaissance des entités nommées structurées, nous vérifions la viabilité de notre proposition et nous la comparons aux approches classiques par chaîne de traitements.

Nous reportons dans la table 8.1 les principaux résultats de nos expérimentations autour de la reconnaissance des entités nommées.

Approche	**Identification dans ce manuscrit**	**SER**
Résultat en 2012	*Sys 0.* Référence ETAPE 2012	59,3
Système de bout en bout	$RAP \rightarrow REN{+}_{types} \rightarrow REN{+}_{struct}$ (4-gramme)	51,9
Chaîne de composants	*Sys E.* RAP_{2017} / REN_{2017}	**51,1**

Table 8.1 – Principaux résultats de nos contributions autour des entités nommées (ces résultats sont issus de la table 5.10)

Pour cette tâche, nos résultats ont montré une amélioration des performances de nos systèmes par rapport au système de référence. Ils ont ainsi confirmé la viabilité de notre approche. Toutefois, notre approche de bout en bout ne surpasse pas une chaîne de traitements à l'état de l'art tout en s'en approchant.

La suite de nos travaux s'est orientée sur l'extraction des concepts sémantiques. Via cette tâche, nous mettons en œuvre notre approche dans un cadre applicatif lié à une définition plus précise de la sémantique. Dans ce cadre applicatif, le problème de la rareté des données est accentué. Aussi, nous mettons en place une stratégie d'entraînement visant à compenser cette rareté.

Cette stratégie s'appuie sur la méthode de transfert d'apprentissage afin de bénéficier de connaissances acquises sur des données tierces. Par exemple, nous avons pu bénéficier de l'information sémantique générale portée par les entités nommées. Nous avons vérifié la viabilité de notre approche en la comparant à une approche classique par chaîne de traitements.

Nous reportons dans la table 8.2 les principaux résultats de nos expérimentations autour de l'extraction des concepts sémantiques.

Nos résultats pour cette tâche confirment à nouveau la viabilité de l'approche proposée dans les travaux de cette thèse.

Nous avons poursuivi en nous orientant vers l'analyse des erreurs produites par notre sys-

Approche	Identification dans ce manuscrit	CER	CVER
Chaîne de composants	RAP_{cc} / $ECS_{texte+carac}$	16,1	20,4
Système de bout en bout	$RAP_6 \rightarrow REN_6 \rightarrow PM^*{+}M^*{}_7 \rightarrow M^*{}_7$	**15,8**	**20,3**

TABLE 8.2 – Principaux résultats de nos contributions autour des concepts sémantiques (ces résultats sont issus de la table 6.12).

tème d'extraction des concepts sémantiques. L'objectif de ces analyses était de mieux comprendre les raisons des erreurs commises.

Nos analyses relèvent que les erreurs produites affectent principalement quelques catégories de concepts sémantiques. De plus, il s'agit majoritairement d'erreurs de suppressions de concepts. Parmi les concepts les plus impactés, certains possèdent une valeur courte. Nous étendons nos analyses à la qualité des transcriptions automatiques de ces mots supports et déterminons que leurs transcriptions ne sont pas à mettre en cause dans les cas de suppressions. Nos analyses ont aussi permis de déceler un problème de segmentation en concepts, nous conduisant à l'apprentissage d'une tâche de segmentation par transfert d'apprentissage.

Par la suite, nous avons dirigé nos travaux vers l'extraction de représentations internes des concepts sémantiques à des fins d'analyses. En nous appuyant sur ces représentations, nous avons proposé une mesure de confiance sur les concept sémantique émis. L'évaluation de cette mesure à travers la la métrique NCE (*Normalized Cross-Entropy*) à montré sa fiabilité.

Les résultats de cette évaluation sont reportés dans la table 8.3 et ont montré la pertinence de la mesure de confiance proposée.

Mode du système principal	Développement	Test
Normale	0.226	0.288
Étoile	0.195	0.241

TABLE 8.3 – Résultats de fiabilité des mesures de confiance produites par le classifieur externe (ces résultats sont reportés de la table 7.7).

8.2 Perspectives

À partir des contributions exposées dans ce manuscrit, plusieurs perspectives de recherche peuvent être envisagées.

Optimisation de l'architecture neuronale

Tout d'abord, il est possible d'effectuer une optimisation fine de l'architecture et des hyperparamètres de notre système.

En effet, nous n'avons cherché à optimiser que sa profondeur. Une étude des paramètres des optimiseurs exploités ou du nombre d'unités neuronales par couche peut conduire à l'obtention d'un meilleur minima local. Nous n'avons pas réalisé ce type d'optimisation en raison de la quantité importante d'entraînements que cela implique, et donc du temps et de la puissance de calcul nécessaire.

Évolution de l'architecture neuronale

Il est désormais largement envisageable de remplacer l'architecture neuronale employée. Les mécanismes d'attention ont été intégrés avec succès pour la tâche de compréhension de la parole dans le cadre d'une chaîne de composants [Simonnet 2019].

De plus, des approches récentes exploitent avec succès des architectures encodeurs-décodeurs pour certaines tâches de compréhension de la parole. Nous pouvons citer [B. Liu et Lane 2016 ; Serdyuk et al. 2018] pour des tâches de détection d'intention et de domaine, mais également [Zhu et K. Yu 2017] pour la tâche d'extraction des concepts sémantiques dans le cadre du domaine de réservation de vol ATIS. Il peut donc être intéressant d'apporter les bénéfices de ce type d'architectures à nos travaux.

Enrichissement par des connaissances à priori

Dans le cadre de notre approche de bout en bout, nous entraînons un système à partir du signal acoustique et de la représentation cible des mots et de la sémantique. L'enrichissement des données d'entraînement pourrait être réalisé avec des caractéristiques additionnelles, par exemple, des lemmes ou des descripteurs morphosyntaxiques. Il s'agirait d'extraire, via des réseaux de neurones tiers, une ou plusieurs représentations vectorielles (*embeddings*) de ces informations additionnelles, puis effectuer leur injection au niveau des couches neuronales récurrentes responsables de la capture de l'information sémantique. Des travaux similaires ont été effectués pour injecter une information permettant l'adaptation au locuteur dans [Tomashenko, Caubrière et Estève 2019].

Étude détaillée de la stratégie de curriculum

Une autre perspective suite à nos travaux concernerait notre apprentissage par transferts successifs piloté par une stratégie de curriculum. Une étude complémentaire pour déterminer plus efficacement le dimensionnement des tâches impliquées pourrait permettre d'optimiser son utilisation. Nous pouvons par exemple concentrer cette étude sur la taille des ensembles de données exploités, mais aussi sur leurs degrés de spécialisation.

En complément, nous pouvons étendre nos travaux d'analyse à l'ensemble de la chaîne d'apprentissage, dans le but de comprendre les informations considérées comme pertinentes par le

système. Par analyse de l'impact des entités nommées dans cette chaîne d'apprentissage, nous avons par exemple relevé un impact négatif sur une grande partie des concepts sémantiques pouvant être considérés proche de l'entité nommée *amount*. Une analyse de ce type permettrait l'obtention d'un meilleur système final par une spécialisation plus précise du modèle.

Une autre perspective centrée sur la stratégie de curriculum correspondrait à l'enrichissement des tâches composants la chaîne d'apprentissage. Nous pouvons envisager d'étudier d'autres tâches pouvant s'intégrer pleinement dans notre stratégie d'apprentissage, comme nous l'avons fait avec la tâche de segmentation. Par exemple, nous pouvons exploitater des données multilingues, comme c'est le cas dans [Tomashenko, Caubrière et Estève 2019]. Dans cette étude, les données multilingues sont exploitées dans un but de modélisation acoustique. Il pourrait être intéressant de pouvoir tirer bénéfice de données multilingues au niveau de la sémantique, ce qui pourrait en partie répondre à la problématique du manque de données annotées manuellement. Il s'agit d'exploiter une langue étrangère suffisamment proche, afin d'extraire des informations de structure sémantique qui vont au-delà d'une langue. Dans notre cadre expérimentale, l'utilisation de la partie italienne du corpus PORTMEDIA rend possible une étude de ce type.

Détection d'incohérences dans les définitions sémantiques

Nos travaux d'analyse d'erreurs nous ont conduits à la mise en œuvre d'une projection de représentation des concepts dans un espace à deux dimensions. Ces travaux ont permis de mettre en avant la capacité d'un réseau à projeter les informations sémantiques dans cet espace. Lors de nos expérimentations, nous avons observé le regroupement des projections en fonction de la classe sémantique associée.

Il semble envisageable d'exploiter cette capacité de projection dans l'espace, afin d'effectuer la détection d'incohérences dans l'annotation sémantique. En soit, l'analyse de ces projections pourrait mettre en avant des incohérences dans l'annotation manuelle, qui pourront ensuite être corrigées par un expert. Cela revient à faciliter le débruitage de l'annotation manuelle des ensembles de données.

Il est également envisageable d'exploiter cette projection pour remettre en cause la structure sémantique définie. Par analyse du positionnement des regroupements dans l'espace, nous pouvons envisager de détecter des incohérences de définition. Par exemple, le positionnement de deux classes sémantiques différentes dans une même portion de l'espace de représentation.

Augmentation automatique des données

Dans nos travaux, nous avons exploité avec succès une augmentation de données automatique pour les entités nommées. Nous avons également vu une quantité élevée d'erreurs de suppression dans le cas des concepts sémantiques. Pour améliorer les performances de nos

systèmes et réduire la quantité de suppression, nous pouvons envisager une augmentation automatique de nos données concernant les concepts avec un fort taux de suppression.

Par exemple, dans le cadre de la tâche MEDIA, la variabilité des phrases prononcées est assez faible. Il est ainsi possible d'effectuer la génération d'une structure de phrase, par exemple "Je souhaite réserver un hôtel à *VILLE*". Au sein de cette phrase, nous pouvons ainsi remplir les champs avec des valeurs de concepts sémantiques cohérentes récupérées à partir d'un dictionnaire. Ces dictionnaires peuvent être issus d'une analyse sur l'ensemble manuellement annoté ou même bénéficier d'un enrichissement expert extérieur.

Enfin, comme nous exploitons une modalité parole en entrée de nos systèmes, nous pouvons compléter une augmentation automatique de ce type par la synthèse des phrases générées.

Compréhension de la parole à l'échelle du document

Dans nos travaux, nous avons mis en place un système neuronal de bout en bout pour la compréhension de la parole. Nous fournissons à ce système des segments audio correspondant à une phrase, impliquant une modélisation de la sémantique à l'échelle de la phrase.

L'information sémantique est généralement présente globalement au sein d'un document, elle n'est pas restreinte à l'échelle d'une phrase. Nous pouvons par exemple mentionner les reportages d'informations qui traitent d'un même sujet sur l'ensemble du document.

Il serait ainsi pertinent d'étendre les travaux de cette thèse en proposant un traitement neuronal de bout en bout effectuant une modélisation sémantique à l'échelle d'un document.

Mettre en place un système à cette échelle entraine des difficultés comme la gestion d'une plus grande quantité de données, la taille d'un document pouvant être très importante comparée à une phrase. Celà entraine aussi une modélisation sémantique plus générale.

ANNEXES

Coût environnemental de la thèse

Nous proposons ici d'effectuer un aparté pour rendre compte du coût environnemental des travaux menés dans le cadre de cette thèse.

Cette section est motivée par le coût énergétique grandissant des approches d'apprentissages profonds. Ce domaine progresse rapidement avec des méthodes se perfectionnant, mais qui nécessite un temps de calcul de plus en plus important. Certaines études s'intéressent au coût énergétique de l'apprentissage profond, notamment dans le cadre du traitement de la langue naturel [Strubell et al. 2019]. Ces études signalent l'importance de l'indication du coût énergétique afin de s'orienter à terme vers des algorithmes et du matériel de calcul plus efficaces.

Dans cette thèse, nous avons exposé nos travaux du point de vue des performances obtenues. Nous proposons au travers de cette section, une estimation factuelle du coût énergétique de l'ensemble des travaux permettant la réalisation de cette thèse. L'estimation que nous proposons ne peut être très précise en raison des variabilités matérielles lors de nos expérimentations. De plus nous n'avons pas consigné l'intégralité de nos expérimentations, notamment lorsqu'il s'agissait de développement. Ainsi, cette estimation se veut plutôt indicative d'un ordre de grandeur.

Dans le but de réaliser notre estimation, nous considérons que nous réalisons l'ensemble de nos expérimentations sur un gpu de type Nvidia Tesla K40, ayant une consommation électrique de 235 watts. Nous effectuons une estimation du temps de calcul nécessaire à un GPU de ce type pour réaliser l'apprentissage de l'ensemble des modèles exploités. Nous regroupons, dans la table 8.4, l'estimation du temps de calcul total des expérimentations que nous avons présentés dans cette thèse par chapitre de contributions.

Chapitre	**Temps de calcul GPU**	**kWh consommés**
Chapitre 5.	139 jours	784 kWh
Chapitre 6.	170 jours	959 kWh
Chapitre 7.	4 jours	23 kWh
Total	**313** jours	**1 766** kWh

Table 8.4 – Estimation du temps de calcul et de la consommation énergétique associée pour la reproduction des résultats présentés dans cette thèse.

Dans cette table, nous avons fourni qu'une estimation du temps de calcul correspondant aux expérimentations présentées dans les chapitres de contributions de ce manuscrit. Il ne peut s'agir que d'une estimation basse, puisque nous avons nécessairement effectué davantage d'expérimentations que celles présentées. En effet, nous avons réalisé plusieurs séries d'expérimentations n'ayant pas montré de résultats concluants. De plus, un nombre non négligeable d'expériences ont été nécessaires à des fins de développement.

Il est ainsi raisonnable de penser qu'avec la prise en compte de l'intégralité des expérimentations menées dans le cadre de cette thèse, nous pouvons à minima doubler l'estimation du coût en temps de calcul et énergétique de ces travaux.

Pour rendre compte de la consommation énergétique, il est à noté qu'avec $1kWh$ il est possible d'alimenter un chauffage électrique pendant 1 heure l'hiver, mais également de parcourir 2 km dans une voiture électrique smart. Pour davantage de correspondance, nous invitons le lecteur à consulter le site internet d *'EDF* [1].

1. https://www.edf.fr/groupe-edf/espaces-dedies/l-energie-de-a-a-z/tout-sur-l-energie/le-developpement-durable/que-peut-on-faire-avec-1-kwh

RÉFÉRENCES PERSONNELLES

Références personnelles liées à la thèse

Caubrière, A., Estève, Y., Laurent, A. & Morin, E., (2020), Confidence measure for speech-to-concept end-to-end spoken language understanding. In *Proceedings of the 21th Annual Conference of the International Speech Association (INTERSPEECH)*, Shanghai, China.

Caubrière, A., Ghannay, S., Tomashenko, N., De Mori, R., Laurent, A., Morin, E. & Estève, Y., (2020), Error analysis applied to end-to end spoken language understanding. In *Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain.

Caubrière, A., Rosset, S., Estève, Y., Laurent, A. & Morin, E., (2020a), Où en sommes-nous dans la reconnaissance des entités nommées structurées à partir de la parole ?, In C. Benzitoun, C. Braud, L. Huber, D. Langlois, S. Ouni, S. Pogodalla & S. Schneider (Éd.), *6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 1 : Journées d'Études sur la Parole* (p. 64-72), Nancy, France : ATALA.

Caubrière, A., Rosset, S., Estève, Y., Laurent, A. & Morin, E., (2020b), Where are we in Named Entity Recognition from Speech ? In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, Marseille, France.

Caubrière, A., Tomashenko, N., Estève, Y., Laurent, A. & Morin, E., (2019), Curriculum d'apprentissage : reconnaissance d'entités nommées pour l'extraction de concepts sémantiques. In *26e conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, Toulouse, France.

Caubrière, A., Tomashenko, N., Laurent, A., Morin, E., Camelin, N. & Estève, Y., (2019), Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability. In *Proceedings of the 20th Annual Conference of the International Speech Association (INTERSPEECH)* (p. 1198-1202), Graz, Austria, doi :10.21437/interspeech.2019-1832

Ghannay, S., Caubrière, A., Estève, Y., Camelin, N., Simonnet, E., Laurent, A. & Morin, E., (2018), End-to-end named entity and semantic concept extraction from speech. In *Proceedings of the Spoken Language Technology Workshop (SLT)*, Athens, Greece.

Tomashenko, N., Caubrière, A., Estève, Y., Laurent, A. & Morin, E., (2019), Recent Advances in End-to-End Spoken Language Understanding. In *Proceedings of the 7th International Conference on Statistical Language and Speech Processing (SLSP)*, Ljubljana, Slovenia.

Références personnelles non directement liées à la thèse

Tomashenko, N., Caubrière, A. & Estève, Y., (2019), Investigating Adaptation and Transfer Learning for End-to-End Spoken Language Understanding from Speech. In *Proceedings of the 20th Annual Conference of the International Speech Association (INTERSPEECH)* (p. 824-828), Graz, Austria : ISCA, doi :10.21437/Interspeech.2019-2158

Tomashenko, N., Raymond, C., Caubrière, A., De Mori, R. & Estève, Y., (2020), Dialogue History Integration into End-to-End Signal-to-Concept Spoken Language Understanding Systems. In *Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 5), Barcelona, Spain, doi :10.1109/ICASSP40776.2020.9053247

RÉFÉRENCES

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., ... Chen, G. et al., (2016), Deep speech 2 : End-to-end speech recognition in english and mandarin. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)* (p. 173-182), New York City, NY, USA.

Bahdanau, D., Cho, K. & Bengio, Y., (2015), Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P. & Bengio, Y., (2016), End-to-end attention-based large vocabulary speech recognition. In *Proceedings of the 41st International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 4945-4949), IEEE, Shanghai, China.

Baum, L. E. et al., (1972), An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, *Inequalities*, *31*, 1-8.

Bechet, F., Maza, B., Bigouroux, N., Bazillon, T., El-Beze, M., De Mori, R. & Arbillot, E., (2012), DECODA : a call-centre human-human spoken conversation corpus. In *Proceedings of the 8th Language Resources and Evaluation Conference (LREC)* (p. 114-118), Istanbul, Turkey : European Language Resources Association (ELRA).

Belinkov, Y. & Glass, J., (2017), Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Proceedings of the 30th Advances in Neural Information Processing Systems Conference (NIPS)* (p. 2441-2451), Long Beach, CA, USA.

Bengio, Y., Ducharme, R., Vincent, P. & Jauvin, C., (2003), A neural probabilistic language model, *Journal of machine learning research*, *3Feb*, 1137-1155.

Bengio, Y., Louradour, J., Collobert, R. & Weston, J., (2009), Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning (ICML)* (p. 41-48).

Bengio, Y., Simard, P. & Frasconi, P., (1994), Learning long-term dependencies with gradient descent is difficult, *IEEE transactions on neural networks*, *52*, 157-166.

Bernard, G., Galibert, O. & Kahn, J., (2014), The second official REPERE evaluation. In *Proceedings of the 2nd International Workshop on Speech, Language and Audio in Multimedia, SLAM* (p. 34-38), Penang, Malaysia.

Bonneau-Maynard, H., Rosset, S., Ayache, C., Kuhn, A. & Mostefa, D., (2005), Semantic annotation of the french media dialog corpus. In *Proceedings of the 9th European Conference*

on Speech Communication and Technology (EUROSPEECH) (p. 3457-3460), Lisbon, Portugal.

Bouchekif, A., (2016), *Structuration automatique de documents audio* (thèse de doct., Le Mans Université).

Bougares, F., Deléglise, P., Esteve, Y. & Rouvier, M., (2013), LIUM ASR system for Etape French evaluation campaign : experiments on system combination using open-source recognizers. In *Proceedings of the 16th International Conference on Text, Speech and Dialogue (TSD)* (p. 319-326), Pilsen, Czech Republic.

Bourlard, H. & Wellekens, C. J., (1987), Multilayer perceptrons and automatic speech recognition. In *Proceedings of the 1st International Conference on Neural Networks (ICNN)* (p. 407-416).

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Askell, A. et al., (2020), Language models are few-shot learners, *arXiv preprint arXiv :2005.14165*.

Bundschus, M., Dejori, M., Stetter, M., Tresp, V. & Kriegel, H.-P., (2008), Extraction of semantic biomedical relations from text using conditional random fields, *BMC bioinformatics*, *91*, 207, doi :10.1186/1471-2105-9-207

Caubrière, A., Estève, Y., Laurent, A. & Morin, E., (2020), Confidence measure for speech-to-concept end-to-end spoken language understanding. In *Proceedings of the 21th Annual Conference of the International Speech Association (INTERSPEECH)*, Shanghai, China.

Caubrière, A., Ghannay, S., Tomashenko, N., De Mori, R., Laurent, A., Morin, E. & Estève, Y., (2020), Error analysis applied to end-to end spoken language understanding. In *Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain.

Caubrière, A., Rosset, S., Estève, Y., Laurent, A. & Morin, E., (2020a), Où en sommes-nous dans la reconnaissance des entités nommées structurées à partir de la parole ?, In C. Benzitoun, C. Braud, L. Huber, D. Langlois, S. Ouni, S. Pogodalla & S. Schneider (Éd.), *6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 1 : Journées d'Études sur la Parole* (p. 64-72), Nancy, France : ATALA.

Caubrière, A., Rosset, S., Estève, Y., Laurent, A. & Morin, E., (2020b), Where are we in Named Entity Recognition from Speech ? In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, Marseille, France.

Caubrière, A., Tomashenko, N., Estève, Y., Laurent, A. & Morin, E., (2019), Curriculum d'apprentissage : reconnaissance d'entités nommées pour l'extraction de concepts sémantiques. In *26e conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, Toulouse, France.

Caubrière, A., Tomashenko, N., Laurent, A., Morin, E., Camelin, N. & Estève, Y., (2019), Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability. In *Proceedings of the 20th Annual Conference of the International Speech Association (INTERSPEECH)* (p. 1198-1202), Graz, Austria, doi :10.21437/interspeech.2019-1832

Chan, W., Jaitly, N., Le, Q. & Vinyals, O., (2016), Listen, attend and spell : A neural network for large vocabulary conversational speech recognition. In *Proceedings of the 41st International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 4960-4964), IEEE, Shanghai, China.

Chen, S. F. & Goodman, J., (1999), An empirical study of smoothing techniques for language modeling, *Computer Speech & Language*, *134*, 359-394.

Chiu, C.-C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., ... Gonina, E. et al., (2018), State-of-the-art speech recognition with sequence-to-sequence models. In *Proceedings of the 43th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 4774-4778), IEEE, Calgary, Alberta, Canada.

Chiu, J. P. & Nichols, E., (2016), Named entity recognition with bidirectional LSTM-CNNs, *Transactions of the Association for Computational Linguistics*, *4*, 357-370.

Cho, K., Van Merriënboer, B., Bahdanau, D. & Bengio, Y., (2014), On the properties of neural machine translation : Encoder-decoder approaches. In *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST)*, (p. 103-111), Doha, Qatar, doi :10.3115/v1/W14-4012

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y., (2014), Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing (EMNLP)* (p. 1724-1734), Doha, Qatar, doi :10.3115/v1/D14-1179

Chomsky, N. & Lightfoot, D. W., (2002), *Syntactic structures*, Walter de Gruyter.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K. & Bengio, Y., (2015), Attention-based models for speech recognition. In *Proceedings of the 28th Advances in Neural Information Processing Systems Conference (NIPS)* (p. 577-585), Montréal, Canada.

Chorowski, J., Bahdanau, D., Cho, K. & Bengio, Y., (2014), End-to-end continuous speech recognition using attention-based recurrent NN : First results. In *Proceedings of the Deep Learning and Representation Learning Workshop of the 27th Advances in Neural Information Processing Systems Conference (NIPS)*, Montréal, Canada.

Davis, S. & Mermelstein, P., (1980), Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE transactions on acoustics, speech, and signal processing*, *284*, 357-366.

De Mori, R., (2007), Spoken language understanding : a survey. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)* (p. 365-376), IEEE, Kyoto, Japan.

De Mori, R., Bechet, F., Hakkani-Tur, D., McTear, M., Riccardi, G. & Tur, G., (2008), Spoken language understanding, *IEEE Signal Processing Magazine*, *253*, 50-58.

Dempster, A. P., Laird, N. M. & Rubin, D. B., (1977), Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society : Series B (Methodological)*, *391*, 1-22.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K., (2019), Bert : Pre-training of deep bidirectional transformers for language understanding, 4171-4186, doi :10.18653/v1/N19-1423

Dietterich, T., (1995), Overfitting and undercomputing in machine learning, *ACM computing surveys (CSUR)*, *273*, 326-327.

Dinarelli, M. & Tellier, I., (2016), Improving recurrent neural networks for sequence labelling, *arXiv preprint arXiv :1606.02555*.

Dong, L., Xu, S. & Xu, B., (2018), Speech-transformer : a no-recurrence sequence-to-sequence model for speech recognition. In *Proceedings of the 43th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 5884-5888), IEEE, Calgary, Alberta, Canada.

Dozat, T., (2016), Incorporating nesterov momentum into adam.

Duchi, J., Hazan, E. & Singer, Y., (2011), Adaptive subgradient methods for online learning and stochastic optimization., *Journal of machine learning research*, *127*.

Elman, J. L., (1990), Finding structure in time, *Cognitive science*, *142*, 179-211.

Estève, Y., Bazillon, T., Antoine, J.-Y., Béchet, F. & Farinas, J., (2010), The EPAC Corpus : Manual and Automatic Annotations of Conversational Speech in French Broadcast News. In *Proceedings of the 7th international conference on Language Resources and Evaluation (LREC)* (p. 1686-1689), Malta : European Language Resources Association (ELRA).

Evermann, G. & Woodland, P., (2000), Posterior probability decoding, confidence estimation and system combination. In *Proceedings of the Speech Transcription Workshop* (T. 27, p. 78-81).

Fernández, S., Graves, A. & Schmidhuber, J., (2008), *Phoneme recognition in TIMIT with BLSTM-CTC*.

Fillmore, C. J. et al., (1976), Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences : Conference on the origin and development of language and speech* (T. 280, *1*, p. 20-32), New York.

Forney, G. D., (1973), The viterbi algorithm, *Proceedings of the IEEE*, *613*, 268-278.

Galibert, O. & Kahn, J., (2013), The first official repere evaluation. In *First Workshop on Speech, Language and Audio in Multimedia*.

Galibert, O., Leixa, J., Adda, G., Choukri, K. & Gravier, G., (2014), The ETAPE speech processing evaluation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)* (p. 3995-3999), Reykjavik, Iceland : European Language Resources Association (ELRA).

Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J.-F., Mostefa, D. & Choukri, K., (2006), Corpus description of the ESTER Evaluation Campaign for the Rich Transcription of French Broadcast News. In *Proceedings of the 5th Language Resources and Evaluation Conference (LREC)* (p. 139-142), Genoa, Italy : European Language Resources Association (ELRA).

Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K., Bonastre, J.-F. & Gravier, G., (2005), The ESTER phase II evaluation campaign for the rich transcription of French broadcast news. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH)* (p. 1149-1152), Lisbon, Portugal.

Galliano, S., Gravier, G. & Chaubard, L., (2009), The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 2543-2546), Brighton, United Kingdom.

Geman, S., Bienenstock, E. & Doursat, R., (1992), Neural networks and the bias/variance dilemma, *Neural computation*, *41*, 1-58.

Ghannay, S., (2017), *Étude sur les représentations continues de mots appliquées à la détection automatique des erreurs de reconnaissance de la parole* (thèse de doct., Le Mans Université).

Ghannay, S., Caubrière, A., Estève, Y., Camelin, N., Simonnet, E., Laurent, A. & Morin, E., (2018), End-to-end named entity and semantic concept extraction from speech. In *Proceedings of the Spoken Language Technology Workshop (SLT)*, Athens, Greece.

Ghannay, S., Estève, Y., Camelin, N., Dutrey, C., Santiago, F. & Adda-Decker, M., (2015), Combining continuous word representation and prosodic features for asr error prediction. In *Proceedings of the 3rd International Conference on Statistical Language and Speech Processing (SLSP)* (p. 84-95), Budapest, Hungary.

Giraudel, A., Carré, M., Mapelli, V., Kahn, J., Galibert, O. & Quintard, L., (2012), The REPERE Corpus : a multimodal corpus for person recognition. In *Proceedings of the 8th Language Resources and Evaluation Conference (LREC)* (p. 1102-1107), Istanbul, Turkey : European Language Resources Association (ELRA).

Glorot, X. & Bengio, Y., (2010), Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)* (p. 249-256), Sardinia, Italy.

Goodfellow, I., Bengio, Y. & Courville, A., (2016), *Deep learning*, MIT press.

Goryainova, M., Grouin, C., Rosset, S. & Vasilescu, I., (2014), Morpho-Syntactic Study of Errors from Speech Recognition System. In *Proceedings of the 9th International Conference on*

Language Resources and Evaluation (LREC) (p. 3995-3999), Reykjavik, Iceland : European Language Resources Association (ELRA).

Graves, A., Fernández, S., Gomez, F. & Schmidhuber, J., (2006), Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning (ICML)* (p. 369-376), Pittsburgh, PA, USA.

Graves, A. & Jaitly, N., (2014), Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML)* (p. 1764-1772), Beijing, China.

Graves, A., Mohamed, A.-r. & Hinton, G., (2013), Speech recognition with deep recurrent neural networks. In *Proceedings of the 38th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 6645-6649), IEEE, Vancouver, Canada.

Gravier, G., Adda, G., Paulson, N., Carré, M., Giraudel, A. & Galibert, O., (2012), The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. In *Proceedings of the 8th Language Resources and Evaluation Conference (LREC)* (p. 114-118), Istanbul, Turkey : European Language Resources Association (ELRA).

Gravier, G., Bonastre, J.-F., Geoffrois, E., Galliano, S., McTait, K. & Choukri, K., (2004), The ESTER Evaluation Campaign for the Rich Transcription of French Broadcast News. In *Proceedings of the 4th Language Resources and Evaluation Conference (LREC)* (p. 885-888), Lisbon, Portugal : European Language Resources Association (ELRA).

Grishman, R. & Sundheim, B. M., (1996), Message understanding conference-6 : A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)* (p. 466-471), Copenhagen, Denmark.

Grouin, C., Rosset, S., Zweigenbaum, P., Fort, K., Galibert, O. & Quintard, L., (2011), Proposal for an extension of traditional named entities : From guidelines to evaluation, an overview. In *Proceedings of the 5th Linguistic Annotation Workshop* (p. 92-100), Portland, OR, USA.

Guinaudeau, C., (2011), *Structuration automatique de flux télévisuels* (thèse de doct., Université de Rennes).

Hahn, S., Dinarelli, M., Raymond, C., Lefevre, F., Lehnen, P., De Mori, R., ... Riccardi, G., (2010), Comparing stochastic approaches to spoken language understanding in multiple languages, *IEEE Transactions on Audio, Speech, and Language Processing*, *196*, 1569-1583.

Hakkani-Tür, D., Béchet, F., Riccardi, G. & Tur, G., (2006), Beyond ASR 1-best : Using word confusion networks in spoken language understanding, *Computer Speech & Language*, *204*, 495-514.

Hakkani-Tür, D., Tür, G., Celikyilmaz, A., Chen, Y.-N., Gao, J., Deng, L. & Wang, Y.-Y., (2016), Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Pro-*

ceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH) (p. 715-719), San Francisco, CA, USA, doi :10.21437/Interspeech.2016-402

Hammersley, J. M. & Clifford, P., (1971), Markov fields on finite graphs and lattices, *Unpublished manuscript*, *46*.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... Coates, A. et al., (2014), Deep speech : Scaling up end-to-end speech recognition, *arXiv preprint arXiv :1412.5567*.

Hatmi, M., (2014), *Reconnaissance des entités nommées dans des documents multimodaux* (thèse de doct., Université de Nantes).

Hazen, T. J., Seneff, S. & Polifroni, J., (2002), Recognition confidence scoring and its use in speech understanding systems, *Computer Speech & Language*, *161*, 49-67.

He, K., Zhang, X., Ren, S. & Sun, J., (2015), Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. In *Proceedings of the 15th International Conference on Computer Vision (ICCV)* (p. 1026-1034), IEEE, Santiago, Chile.

Hemphill, C. T., Godfrey, J. J. & Doddington, G. R., (1990), The ATIS spoken language systems pilot corpus. In *Speech and Natural Language : Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., ... Sainath, T. N. et al., (2012), Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups, *IEEE Signal processing magazine*, *296*, 82-97.

Hochreiter, S. & Schmidhuber, J., (1997), Long short-term memory, *Neural computation*, *98*, 1735-1780.

Hoerl, A. E. & Kennard, R. W., (1970), Ridge regression : Biased estimation for nonorthogonal problems, *Technometrics*, *121*, 55-67.

Hori, T., Cho, J. & Watanabe, S., (2018), End-to-end speech recognition with word-based RNN language models. In *Proceedings of the Spoken Language Technology Workshop (SLT)* (p. 389-396), IEEE, Athens, Greece.

Hori, T., Watanabe, S., Zhang, Y. & Chan, W., (2017), Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. In *Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 949-953), Stockholm, Sweden, doi :10.21437/Interspeech.2017-1296

Huang, Z., Xu, W. & Yu, K., (2015), Bidirectional LSTM-CRF models for sequence tagging, *arXiv preprint arXiv :1508.01991*.

Ioffe, S. & Szegedy, C., (2015), Batch normalization : Accelerating deep network training by reducing internal covariate shift. (T. 37, p. 448-456), Lille, France.

Jabaian, B., (2012), *Systèmes de compréhension et de traduction de la parole : vers une approche unifiée dans le cadre de la portabilité multilingue des systèmes de dialogue* (thèse de doct., Université d'Avignon et des Pays de Vaucluse).

Jannet, M. B., Adda-Decker, M., Galibert, O., Kahn, J. & Rosset, S., (2014), Eter : a new metric for the evaluation of hierarchical named entity recognition. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)* (p. 3987-3994), Reykjavik, Iceland : European Language Resources Association (ELRA).

Jelinek, F., (1976), Continuous speech recognition by statistical methods, *Proceedings of the IEEE*, *644*, 532-556.

Jiang, H., (2005), Confidence measures for speech recognition : A survey, *Speech communication*, *454*, 455-470.

Joachims, T., (1998), Text categorization with support vector machines : Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)* (p. 137-142), Chemnitz, Germany.

Jordan, M. I., (1986), Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the 8th Annual Conference of the Cognitive Science Society (CSS)* (p. 531-546), Hillsdale, NJ, USA.

Jordan, M. I., (1997), Serial order : A parallel distributed processing approach. In *Advances in psychology* (T. 121, p. 471-495), Elsevier.

Juang, B.-H. & Rabiner, L. R., (2005), Automatic speech recognition–a brief history of the technology development, *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, *1*, 67.

Kadari, R., Zhang, Y., Zhang, W. & Liu, T., (2018), CCG supertagging via Bidirectional LSTM-CRF neural architecture, *Neurocomputing*, *283*, 31-37.

Katz, S., (1987), Estimation of probabilities from sparse data for the language model component of a speech recognizer, *IEEE transactions on acoustics, speech, and signal processing*, *353*, 400-401.

Kim, S., Hori, T. & Watanabe, S., (2017), Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Proceedings of the 42nd International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 4835-4839), IEEE, New Orleans, LA, USA.

Kingma, D. P. & Ba, J., (2015), Adam : A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Klambauer, G., Unterthiner, T., Mayr, A. & Hochreiter, S., (2017), Self-normalizing neural networks. In *Proceedings of the 30th Advances in Neural Information Processing Systems Conference (NIPS)* (p. 971-980), Long Beach, CA, USA.

Klatt, D. H., (1977), Review of the ARPA speech understanding project, *The Journal of the Acoustical Society of America*, *626*, 1345-1366.

Lafferty, J., McCallum, A. & Pereira, F. C., (2001), Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)* (p. 282-289), Williamstown, MA, USA, doi :10.5555/645530.655813

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. & Dyer, C., (2016), Neural architectures for named entity recognition. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT)* (p. 260-270), San Diego, CA, USA, doi :10.18653/v1/N16-1030

Lavergne, T., Cappé, O. & Yvon, F., (2010), Practical very large scale CRFs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)* (p. 504-513).

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E. & Jackel, L. D., (1990), Handwritten digit recognition with a back-propagation network. In *Proceedings of the 3rd Advances in Neural Information Processing Systems Conference (NIPS)* (p. 396-404), Denver, CO, USA.

Lefèvre, F., Mostefa, D., Besacier, L., Estève, Y., Quignard, M., Camelin, N., ... Rojas-Barahona, L., (2012), Robustesse et portabilités multilingue et multi-domaines des systèmes de compréhension de la parole : les corpus du projet PortMedia. In *Proceedings of the Joint Conference JEP-TALN-RECITAL* (p. 779-786), Grenoble, France.

Liu, B. & Lane, I., (2016), Attention-based recurrent neural network models for joint intent detection and slot filling, 685-689.

Luong, M.-T., Pham, H. & Manning, C. D., (2015), Effective approaches to attention-based neural machine translation. In *Proceedings of the 20th Conference on Empirical Methods in Natural Language Processing (EMNLP)* (p. 1412-1421), Lisbon, Portugal, doi :10.18653/v1/D15-1166

Ma, W. & Van Compernolle, D., (1990), TDNN Labeling for a HMM Recognizer. In *Proceedings of the 15th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 421-423), IEEE, Albuquerque, NM, USA.

Ma, X. & Hovy, E., (2016), End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)* (p. 1064-1074), Berlin, Germany, doi :10.18653/v1/P16-1101

Maas, A. L., Hannun, A. Y. & Ng, A. Y., (2013), Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, Atlanta, GA, USA.

Maaten, L. v. d. & Hinton, G., (2008), Visualizing data using t-SNE, *Journal of machine learning research*, *9Nov*, 2579-2605.

Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R. et al., (1999), Performance measures for information extraction. In *Proceedings of DARPA broadcast news workshop* (p. 249-252), Herndon, VA, USA.

Malouf, R., (2002), A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002)*, Taipei, Taiwan, doi :10.3115/1118853.1118871

Maskey, S. R. & Hirschberg, J., (2008), *Automatic broadcast news speech summarization*, Columbia University.

McCallum, A. & Li, W., (2003), Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 2nd Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT)* (T. 4, p. 188-191), Edmonton, Canada, doi :10.3115/1119176.1119206

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., ... Yu, D. et al., (2014), Using recurrent neural networks for slot filling in spoken language understanding, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *233*, 530-539.

Mesnil, G., He, X., Deng, L. & Bengio, Y., (2013), Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 3771-3775), Lyon, France.

Miao, Y., Gowayyed, M. & Metze, F., (2015), EESEN : End-to-end speech recognition using deep RNN models and WFST-based decoding. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)* (p. 167-174), IEEE, Scottsdale, AZ, USA.

Mikolov, T. [Tomas], Chen, K., Corrado, G. & Dean, J., (2013), Efficient estimation of word representations in vector space. In *Proceedings of the workshop of the 1st International Conference on Learning Representations (ICLR)*, Scottsdale, AZ, USA.

Mikolov, T. [Tomáš], Kombrink, S., Burget, L., Černockỳ, J. & Khudanpur, S., (2011), Extensions of recurrent neural network language model. In *Proceedings of the 36th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 5528-5531), IEEE, Prague, Czech Republic.

Mikolov, T. [Tomas], Sutskever, I., Chen, K., Corrado, G. S. & Dean, J., (2013), Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th Advances in Neural Information Processing Systems Conference (NIPS)* (p. 3111-3119), South Lake Tahoe, NV, USA.

Moritz, N., Hori, T. & Le, J., (2020), Streaming automatic speech recognition with the transformer model. In *Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 6074-6078), IEEE, Barcelona, Spain.

Murray, G., Carenini, G. & Ng, R., (2010), Interpretation and transformation for abstracting conversations. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT)* (p. 894-902), Los Angeles, CA, USA.

Nair, V. & Hinton, G. E., (2010), Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, Haifa, Israel.

Nasr, A., Béchet, F., Rey, J.-F., Favre, B. & Le Roux, J., (2011), Macaon : An nlp tool suite for processing word lattices. In *Proceedings of the 12th Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT) System Demonstrations* (p. 86-91), Portland, OR, USA.

Nesterov, Y., (2013), Gradient methods for minimizing composite functions, *Mathematical Programming*, *1401*, 125-161, doi :10.1007/s10107-012-0629-5

Nouvel, D., Ehrmann, M. & Rosset, S., (2015), *Les entités nommées pour le traitement automatique des langues*, ISTE Group.

Paek, T. & Horvitz, E., (2004), Optimizing Automated Call Routing by Integrating Spoken Dialog Models with Queuing Models. In *Proceedings of the 3rd Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT)* (p. 41-48), Boston, MA, USA.

Pan, S. J. & Yang, Q., (2009), A survey on transfer learning, *IEEE Transactions on knowledge and data engineering*, *2210*, 1345-1359.

Pascanu, R., Mikolov, T. & Bengio, Y., (2013), On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)* (p. 1310-1318), Atlanta, GA, USA.

Passonneau, R. J. & Litman, D., (1997), Discourse segmentation by human and automated means, *Computational Linguistics*, *231*, 103-139.

Peddinti, V., Povey, D. & Khudanpur, S., (2015), A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 3214-3218), Dresden, Germany.

Pennington, J., Socher, R. & Manning, C. D., (2014), Glove : Global vectors for word representation. In *Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing (EMNLP)* (p. 1532-1543), Doha, Qatar.

Pham, N.-Q., Nguyen, T.-S., Niehues, J., Müller, M., Stüker, S. & Waibel, A., (2019), Very deep self-attention networks for end-to-end speech recognition. In *Proceedings of the 20th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 66-70), Graz, Austria.

Qian, N., (1999), On the momentum term in gradient descent learning algorithms, *Neural networks*, *121*, 145-151.

Rabiner, L. R., (1989), A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, *772*, 257-286.

Raymond, C., (2005), *Décodage conceptuel : co-articulation des processus de transcription et compréhension dans les systèmes de dialogue* (thèse de doct., Université d'Avignon et des Pays de Vaucluse).

Raymond, C., (2013), Robust tree-structured named entities recognition from speech. In *Proceedings of the 38th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Vancouver, Canada.

Raymond, C., Esteve, Y., Béchet, F., De Mori, R. & Damnati, G., (2003), Belief confirmation in spoken dialog systems using confidence measures. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)* (p. 150-155), IEEE, US, Virgin Islands.

Raymond, C. & Riccardi, G., (2007), Generative and discriminative algorithms for spoken language understanding. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 1605-1608), Antwerp, Belgium.

Riedmiller, M. & Braun, H., (1993), A direct adaptive method for faster backpropagation learning : The RPROP algorithm. In *Proceedings of the international conference on neural networks (ICNN)* (p. 586-591), IEEE, San Francisco, CA, USA.

Rosenblatt, F., (1958), The perceptron : a probabilistic model for information storage and organization in the brain., *Psychological review*, *656*, 386.

Ruder, S., (2016), An overview of gradient descent optimization algorithms, *arXiv preprint arXiv :1609.04747*.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J., (1985), *Learning internal representations by error propagation*, California Univ San Diego La Jolla Inst for Cognitive Science.

Samson Juan, F. S., (2015), *Exploiting resources from closely-related languages for automatic speech recognition in low-resource languages from Malaysia* (thèse de doct., Université Grenoble Alpes).

Sarawagi, S. & Cohen, W. W., (2004), Semi-markov conditional random fields for information extraction. In *Proceedings of the 17th Advances in Neural Information Processing Systems Conference (NIPS)* (p. 1185-1192), Vancouver, Canada.

Schwenk, H., (2007), Continuous space language models, *Computer Speech & Language, 213*, 492-518.

Schwenk, H., Déchelotte, D. & Gauvain, J.-L., (2006), Continuous space language models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING ACL)* (p. 723-730), Sydney, Australia.

Serdyuk, D., Wang, Y., Fuegen, C., Kumar, A., Liu, B. & Bengio, Y., (2018), Towards end-to-end spoken language understanding. In *Proceedings of the 43th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 5754-5758), IEEE, Calgary, Alberta, Canada.

Seymore, K. & Rosenfeld, R., (1997), Using story topics for language model adaptation. In *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH)* (p. 1987-1990), Rhodes, Greece.

Shah, P., Hakkani-Tür, D., Tür, G., Rastogi, A., Bapna, A., Nayak, N. & Heck, L., (2018), Building a conversational agent overnight with dialogue self-play, *arXiv preprint arXiv :1801.04871*.

Shi, Y., Yao, K., Chen, H., Pan, Y.-C., Hwang, M.-Y. & Peng, B., (2015), Contextual spoken language understanding using recurrent neural networks. In *Proceedings of the 40th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 5271-5275), IEEE, Brisbane, Australia.

Simonnet, E., (2019), *Réseaux de neurones profonds appliqués à la compréhension de la parole* (thèse de doct., Le Mans Université).

Simonnet, E., Camelin, N., Deléglise, P. & Esteve, Y., (2015), Exploring the use of attention-based recurrent neural networks for spoken language understanding. In *Proceedings of the Machine Learning for Spoken Language Understanding and Interaction workshop (SLUNIPS)* (T. 11), Montreal, Canada.

Simonnet, E., Ghannay, S., Camelin, N. & Estève, Y., (2018), Simulating ASR errors for training SLU systems. In *Proceedings of the 11th international conference on Language Resources and Evaluation (LREC)*, Miyazaki, Japan : European Language Resources Association (ELRA).

Simonnet, E., Ghannay, S., Camelin, N., Estève, Y. & De Mori, R., (2017), ASR error management for improving spoken language understanding. In *Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 3329-3333), Stockholm, Sweden, doi :10.21437/Interspeech.2017-1178

Siu, M.-h., Gish, H. & Richardson, F., (1997), Improved estimation, evaluation and applications of confidence measures for speech recognition. In *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH)* (p. 831-834), Rhodes, Greece.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R., (2014), Dropout : a simple way to prevent neural networks from overfitting, *The journal of machine learning research*, *151*, 1929-1958.

Strubell, E., Ganesh, A. & McCallum, A., (2019), Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (p. 3645-3650), Florence, Italy, doi :10.18653/v1/P19-1355

Sundermeyer, M., Schlüter, R. & Ney, H., (2012), LSTM neural networks for language modeling. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 194-197), Portland, OR, USA.

Sundheim, B. M., (1995), *Overview of results of the MUC-6 evaluation*, Naval Command Control et Ocean Surveillance Center, San Diego CA, USA.

Tibshirani, R., (1996), Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society : Series B (Methodological)*, *581*, 267-288.

Tjong Kim Sang, E. F. & Veenstra, J., (1999), Representing text chunks, 173-179.

Tomashenko, N., Caubrière, A. & Estève, Y., (2019), Investigating Adaptation and Transfer Learning for End-to-End Spoken Language Understanding from Speech. In *Proceedings of the 20th Annual Conference of the International Speech Association (INTERSPEECH)* (p. 824-828), Graz, Austria : ISCA, doi :10.21437/Interspeech.2019-2158

Tomashenko, N., Caubrière, A., Estève, Y., Laurent, A. & Morin, E., (2019), Recent Advances in End-to-End Spoken Language Understanding. In *Proceedings of the 7th International Conference on Statistical Language and Speech Processing (SLSP)*, Ljubljana, Slovenia.

Tomashenko, N., Raymond, C., Caubrière, A., De Mori, R. & Estève, Y., (2020), Dialogue History Integration into End-to-End Signal-to-Concept Spoken Language Understanding Systems. In *Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 5), Barcelona, Spain, doi :10.1109/ICASSP40776.2020.9053247

Tur, G. & De Mori, R., (2011), *Spoken language understanding : Systems for extracting semantic information from speech*, John Wiley & Sons.

Van Rijsbergen, C. J., (1974), Foundation of evaluation, *Journal of documentation*.

Vapnik, V., (2006), *Estimation of dependences based on empirical data*, Springer Science & Business Media.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I., (2017), Attention is all you need. In *Proceedings of the 30th Advances in Neural Information Processing Systems Conference (NIPS)* (p. 5998-6008), Long Beach, CA, USA.

Vukotić, V., Raymond, C. & Gravier, G., (2015), Is it time to switch to word embedding and recurrent neural networks for spoken language understanding ?

Vythelingum, K., (2019), *Construction rapide, performante et mutualisée de systèmes de reconnaissance et de synthèse de la parole pour de nouvelles langues* (thèse de doct., Le Mans Université).

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. & Lang, K. J., (1989), Phoneme recognition using time-delay neural networks, *IEEE transactions on acoustics, speech, and signal processing*, *373*, 328-339.

Wang, Y. [Yiming], Chen, T., Xu, H., Ding, S., Lv, H., Shao, Y., … Khudanpur, S., (2019), Espresso : A fast end-to-end neural speech recognition toolkit. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)* (p. 136-143), IEEE, Sentosa, Singapore.

Wang, Y. [Yongqiang], Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J., … Zhang, F. et al., (2020), Transformer-based acoustic modeling for hybrid speech recognition. In *Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 6874-6878), IEEE, Barcelona, Spain.

Wang, Y. [Yufan], Tang, L. & He, T., (2018), Attention-based CNN-BLSTM networks for joint intent detection and slot filling. In *Proceedings of the 17th Chinese Computational Linguistics (CCL) and the 6th Natural Language Processing based on Naturally Annotated Big Data (NLP-NABD)* (p. 250-261), Changsha, China.

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., … Chen, N. et al., (2018), Espnet : End-to-end speech processing toolkit. In *Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 2207-2211), IEEE, Hyderabad, India, doi :10.21437/Interspeech.2018-1456

Woods, W. A., (1975), What's in a link : Foundations for semantic networks. In *Representation and understanding* (p. 35-82), Elsevier.

Xie, B. & Passonneau, R. J., (2015), Graph Structured Semantic Representation and Learning for Financial News. In *FLAIRS Conference* (p. 237-240).

Xu, P. & Sarikaya, R., (2013), Convolutional neural network based triangular crf for joint intent detection and slot filling. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)* (p. 78-83), IEEE, Olomouc, Czech Republic.

Xu, P. & Sarikaya, R., (2014), Contextual domain classification in spoken language understanding systems using recurrent neural network. In *Proceedings of the 39th International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 136-140), IEEE, Florence, Italy.

Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G. & Shi, Y., (2014), Spoken language understanding using long short-term memory neural networks. In *Proceedings of the Spoken Language Technology Workshop (SLT)* (p. 189-194), IEEE, South Lake Tahoe, NV, USA.

Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y. & Yu, D., (2013), Recurrent neural networks for language understanding. In *Proceedings of the 14th Annual Conference of the International Speech Association (INTERSPEECH)* (p. 2524-2528), Lyon, France.

Zeiler, M. D., (2012), Adadelta : an adaptive learning rate method, *arXiv preprint arXiv :1212.5701*.

Zenkel, T., Sanabria, R., Metze, F., Niehues, J., Sperber, M., Stüker, S. & Waibel, A., (2017), Comparison of decoding strategies for ctc acoustic models. In *Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 513-517), IEEE, Stockholm, Sweden, doi :10.21437/Interspeech.2017-1683

Zhang, K., Xu, H., Tang, J. & Li, J., (2006), Keyword extraction using support vector machine. In *Proceedings of the 7th International Conference on Web-Age Information Management (ICWAIM)* (p. 85-96), Hong Kong, China.

Zhang, L. & Wang, H., (2019), Using Bidirectional Transformer-CRF for Spoken Language Understanding. In *Proceedings of the 8th International Conference on Natural Language Processing and Chinese Computing (NLPCC)* (p. 130-141), Dunhuang, China.

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, C. L. Y. & Courville, A., (2016), Towards end-to-end speech recognition with deep convolutional neural networks. In *Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (p. 410-414), San Francisco, CA, USA, doi :10.21437/Interspeech.2016-1446

Zhu, S. & Yu, K., (2017), Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In *Proceedings of the 42nd International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 5675-5679), IEEE, New Orleans, LA, USA.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Le Mans Université

Titre : Du signal au concept : Réseaux de neurones profonds appliqués à la compréhension de la parole

Mot clés : Compréhension de la parole, Réseaux de neurones profonds, Du signal au concept, Reconnaissance d'entités nommées, Extraction de concepts sémantiques, Analyse d'erreurs, Mesure de confiance

Résumé : Cette thèse s'inscrit dans le cadre de l'apprentissage profond appliqué à la compréhension de la parole. Jusqu'à présent, cette tâche était réalisée par l'intermédiaire d'une chaîne de composants mettant en oeuvre, par exemple, un système de reconnaissance de la parole, puis différents traitements du langage naturel, avant d'impliquer un système de compréhension du langage sur les transcriptions automatiques enrichies. Récemment, des travaux dans le domaine de la reconnaissance de la parole ont montré qu'il était possible de produire une séquence de mots directement à partir du signal acoustique. Dans le cadre de cette thèse, il est question d'exploiter ces avancées et de les étendre pour concevoir un système composé d'un seul modèle neuronal entièrement optimisé pour la tâche de compréhension de la parole, du signal au concept. Tout d'abord, nous présentons un état de l'art décrivant les principes de l'apprentissage neuronal profond, de la reconnaissance de la parole, et de la compréhension de la parole. Nous décrivons ensuite les contributions réalisées selon trois axes principaux. Nous proposons un premier système répondant à la problématique posée et l'appliquons à une tâche de reconnaissance des entités nommées. Puis, nous proposons une stratégie de transfert d'apprentissage guidée par une approche de type curriculum learning. Cette stratégie s'appuie sur les connaissances génériques apprises afin d'améliorer les performances d'un système neuronal sur une tâche d'extraction de concepts sémantiques. Ensuite, nous effectuons une analyse des erreurs produites par notre approche, tout en étudiant le fonctionnement de l'architecture neuronale proposée. Enfin, nous mettons en place une mesure de confiance permettant d'évaluer la fiabilité d'une hypothèse produite par notre système.

Title: From signal to concept : Deep neural networks applied to spoken language understanding

Keywords: Spoken language understanding, Deep neural networks, From signal to concept, Named entity recognition, Semantic concept extraction, Errors analysis, Confidence measure

Abstract: This thesis is part of the deep learning applied to spoken language understanding. Until now, this task was performed through a pipeline of components implementing, for example, a speech recognition system, then different natural language processing, before involving a language understanding system on enriched automatic transcriptions. Recently, work in the field of speech recognition has shown that it is possible to produce a sequence of words directly from the acoustic signal. Within the framework of this thesis, the aim is to exploit these advances and extend them to design a system composed of a single neural model fully optimized for the spoken language understanding task, from signal to concept. First, we present a state of the art describing the principles of deep learning, speech recognition, and speech understanding. Then, we describe the contributions made along three main axes. We propose a first system answering the problematic posed and apply it to a task of named entities recognition. Then, we propose a transfer learning strategy guided by a curriculum learning approach. This strategy is based on the generic knowledge learned to improve the performance of a neural system on a semantic concept extraction task. Then, we perform an analysis of the errors produced by our approach, while studying the functioning of the proposed neural architecture. Finally, we set up a confidence measure to evaluate the reliability of a hypothesis produced by our system.