Stage de Master 2 TRIED

6 mois

Reconnaissance Des Entités Nommées À Partir De La Parole

Réalisé par **Saoussan TRIGUI**

Encadrant : Moez AJILI

Encadrant pédagogique : Nicolas THOME

Table des matières :

1. Introduction

L'intelligence artificielle a vécu de grandes évolutions durant les dernières années, notamment grâce aux progrès scientifiques et technologiques autour de l'apprentissage profond. Ces évolutions concernant plusieurs applications, notamment dans le traitement des images, de la voix, et du texte.

Pour ce qui est de la voix, les scientifiques, mais aussi les industriels, proposent des approches et des systèmes capables de reconnaître l'âge, le genre et l'émotion à partir d'un enregistrement vocal. La tâche la plus complexe avec ce type de données est la reconnaissance automatique de la parole (Automatic Speech Recognition ou ASR) qui consiste à transcrire automatiquement le texte énoncé en suite de mots.

Du côté du texte, on parle ici du traitement automatique du langage naturel (Natural Language Processing ou NLP). Plusieurs applications ont également vu le jour comme l'analyse de sentiments, la classification en thèmes, la traduction automatique et la reconnaissance d'entités nommées. Cette dernière tâche, appelée aussi NER (pour Named Entity Recognition), consiste à identifier dans le texte les mots qui représentent des catégories prédéfinies comme les noms de personnes, les dates et les lieux.

L'objectif principal de ce stage est de lier le monde du traitement de la parole à celui du traitement de texte. Plus concrètement, nous nous intéressons à une problématique relativement récente, et peu traitée par les scientifiques, qui consiste à reconnaître les entités nommées, non pas à partir d'un texte écrit, mais à partir de la parole. Le défi principal ici réside dans les erreurs éventuelles de reconnaissance de la parole et leurs impact sur la performance de reconnaissance d'entités nommées.

Ce travail est réalisé au sein de l'entreprise « Mood », fondée en 2020 à Londres. Mood est une jeune startup composée par des ingénieurs et des scientifiques. Elle offre un produit qui permet de collecter des avis des employés, sous forme d'enregistrements vocaux, sur leurs espaces de travail, les analyser avec de l'intelligence artificielle et produire une synthèse pour les gérants des immeubles de bureaux.

Au cours des derniers mois, nous avons développé plusieurs systèmes d'analyse du texte et de la parole pour la langue anglaise, comme ceux cités ci-dessus. En plus de la reconnaissance d'entités nommées à partir de la parole, j'étais personnellement responsable de la conception de systèmes de détection d'âge et de genre. En revanche, nous décrivons uniquement, dans le reste de ce document, le travail réalisé au tour de la reconnaissance d'entités nommées à partir de la parole.

Nous commençons tout d'abord par une vue sur l'état de l'art de la reconnaissance d'entités nommées à partir de texte écrit. Nous analysons ensuite les quelques travaux qui se sont intéressés à la reconnaissance d'entités nommées à partir de la parole. Enfin, nous décrivons notre approche, nous présentons le protocole expérimental et analysons les résultats de nos expériences avant de donner quelques conclusions et perspectives.

2. État de l’art

Nous découvrons en premier temps la littérature de la reconnaissance d’entités nommées à partir des données textuelles. Ensuite, nous nous intéressons particulièrement aux derniers travaux qui attaquent la problématique de NER à partir de la parole.

2.1. NER à partir du texte écrit

Nous pouvons classer les approches NER dans la littérature en 3 grandes catégories. Nous décrivons dans cet élément chacune de ces trois catégories.

2.1.1. Approaches à base de règles

Cette catégorie de travaux se base majoritairement sur des dictionnaires d'un domaine spécifique ou des motifs (patterns) lexiques et syntaxiques. [KIM et al. 2000] proposent une génération de règles basée sur la méthode de Brill. Leur système produit automatiquement des règles en partant de la sortie de l’étiquetage morpho-syntaxique.

Par ailleurs, [HANISCH et al. 2005] ont proposé ProMiner, un système de NER appliqué au domaine biomédical, qui s’appuie sur un dictionnaire de synonymes pour identifier les différents noms de protéines et les gènes potentiels. Quant à [QUIMBAYA et al. 2016], ils ont proposé une approche fondée sur un dictionnaire pour les comptes-rendu de santé. Les résultats expérimentaux montrent que l’approche améliore le rappel tout en ayant un impact limité sur la précision.

La majorité des approches à base de règles, tels que LaSIE-II [HUMPHREYS et al. 1998], Facile [BLACK et al. 1998] et SAR [AONE et al. 1998] proposent principalement des règles sémantiques et syntaxiques pour reconnaître les entités. Cette catégorie de systèmes fonctionne très bien lorsque le lexique est exhaustif. En revanche, leur inconvénient est d’avoir un faible rappel même si la précision est élevée. Cela est dû à des règles spécifiques à un domaine particulier et à des dictionnaires limités. Par conséquent, l’utilisation de ces systèmes est limitée à leurs domaines d’application et ils ne peuvent pas être exploités dans d’autre domaines.

2.1.2. Approches d’apprentissage non supervisé

La majorité de travaux de cette famille utilisent des algorithmes de regroupement [NADEAU et al. 2007]. Ces systèmes extraient les entités nommées à partir des groupes générés en se basant sur la similarité contextuelle. L’idée principale consiste à tirer profit des schémas lexicaux et des statistiques (par exemple, fréquence inverse de documents, représentations contextuelles) calculées sur de grandes quantités de données textuelles afin de détecter les entités nommées. Pour avoir une meilleure efficacité, plusieurs de ces approches sont enrichies par des heuristiques [NADEAU et al. 2006] ou des connaissances syntaxiques relativement simples [Zhang et al. 2013].

2.1.3. Approches d'apprentissage supervisé

Avec l'apprentissage supervisé, la reconnaissance des entités nommées est considérée comme une tâche d'étiquetage automatique de séquences, une sorte de classement automatique appliqué aux séquences. Ces approches commencent par une première phase, à savoir, l'extraction de caractéristiques. Ensuite, des algorithmes d'apprentissage supervisé sont entraînés sur des données annotées afin de modéliser la relation entre ces caractéristiques et les étiquettes concernées.

- Approches classiques

L'étape d'extraction de caractéristiques est d'une très grande importance pour les approches d'apprentissage supervisé. Dans la littérature, Nous pouvons trouver des travaux qui se basent sur des représentations de mots (par exemple, présence de majuscule, morphologie et étiquettes morphosyntaxique) [ZHZHOU et al. 2002, SETTLES. 2004, LIAO et al. 2009], sur des dictionnaires (Wikipedia, DBpedia...) [HOFFART et al. 2011, TORAL et al. 2006, TORISAWA et al. 2007, MIKHEEV et al. 1999], ou des représentations de documents (syntaxe locale, occurrences multiples...) [RAVIN et al.1997, ZHU et al. 2005, JI et al. 2016, KRISHNAN et al. 2006]. Les entrées textuelles ou leurs caractéristiques sont généralement représentées sous forme vectoriel où chaque unité (mot, phrase...) est convertie en une ou plusieurs valeurs numériques. En se basant sur de telles représentations, plusieurs algorithmes d'apprentissage automatique ont été utilisés dans la tâche de NER. Par exemple, [BIKEL et al. 1998, BIKEL et al. 1999] ont développé le premier système basé sur les modèles de Markov Cachés (HMM) [Eddy 1996], qui permet de reconnaitre principalement les noms, les dates, et les nombres. [SZARVAS et al. 2006] a construit un système NER basé sur des arbres de décision [Quinlan 1986] entraînées séparément et combinées selon un vote. [MCNAMEE et al. 2002] ont entraîné des classifieurs SVM [Vapnik 1999] qui s'appuient sur des caractéristiques orthographiques et linguistiques. Sur chaque unité lexicale, les différentes classifications binaires sont effectuées pour juger de son appartenance à l'une des quatre étiquettes concernées (personne, organisation, emplacement et divers).

Dans la tâche de NER, mais aussi dans plusieurs autres tâches d'étiquetage automatique, les champs aléatoires conditionnels (Conditional Random Fields ou CRFs) [Dennis et Moré 1977] ont été parmi les algorithmes les plus à succès. Ceci est grâce à leur capacité de prendre en considération le contexte de chaque unité lexicale. Par exemple, lors du classement d'un mot, ces algorithmes analysent aussi les mots voisins, [MCNAMEE et al. 2003] applique cet algorithme sur le corpus en langue anglaise, CoNLL03, avec un F-score de 84%. L'approche CRF a été adoptée dans des tâches de NER sur plusieurs domaines (par exemple, les textes biomédicaux [SETTLES. 2004, LIU et al. 2020], les tweets [RITTER el al. 2013, LIU et al. 20011] et les textes de chimie [ROCKTÄSCHEL et al. 2012]). Elle a été également appliquée ou enrichie de plusieurs manières. [KRISHNAN et al. 2006] proposent par exemple d'entraîner un premier classifieur CRF afin de produire des représentations latentes qui vont être exploitées par un deuxième classifieur CRF.

- Approches avancées

Durant les dernières années, et avec l'évolution remarquable offerte par l'apprentissage profond, la NER, comme beaucoup de tâches d'apprentissage automatique, ont eu un grand saut vers l'avant. En effet, les réseaux de neurones artificiels sont capables d'apprendre des transformations non linéaires de l'entrée. Ceci les permet de générer des représentations complexes des données d'origine, contrairement à la majorité des approches classiques d'apprentissage automatique.

Ces représentations latentes sont tellement robustes qu'elles diminuent, voire suppriment, le besoin de concevoir des méthodes sophistiquées d'extraction de caractéristiques qui nécessitent une expertise spécialisée. Par conséquent, les réseaux de neurones permettent d'apprendre des modèles de bout-en-bout. Et grâce à des algorithmes d'optimisation et une propagation de l'erreur de la sortie jusqu'à l'entrée, l'apprentissage profond offre la possibilité de concevoir des architectures de plus en plus complexes et performantes.

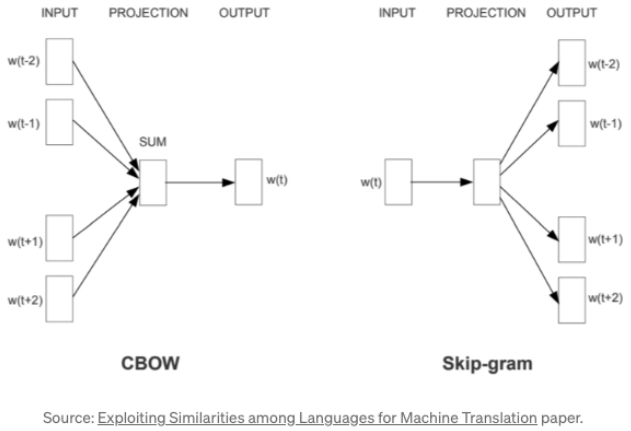
Malgré les promesses de l'apprentissage profond, la transition vers des systèmes end-to-end était très progressive. En effet, la fusion des étapes d'extraction de caractéristiques et de décision en un seul et unique modèle n'était pas évidente.

- Extraction de caractéristiques :

Les scientifiques ont commencé par chercher à concevoir des approches d'extraction de caractéristiques pouvant produire des représentations vectorielles de faible dimension où chaque dimension représente une caractéristique latente. L'avantage de ses représentations latentes réside dans leur capacité à modéliser automatiquement des propriétés syntaxiques mais aussi sémantiques.

Parmi les premières approches prometteuses qui ont vu le jour, on note les algorithmes CBOW (continuous bag of words) et skip-grams [MIKOLOV et al. 2013a] schématisés dans la Figure 1. Ces approches consistent à apprendre des projections, optimisées sur une tâche imaginaire appelée « tâche prétexte ». Pour le CBOW, cette tâche consiste à prédire le mot au milieu en utilisant les mots du contexte droite et gauche. Quant au modèle skip-gram, il consiste à prédire les mots du contexte en se basant sur le mot en cours. Plusieurs outils et modèles pré-entraînés développés par de grandes entreprises ou laboratoires de recherche ont été mis à disposition à la communauté scientifique. Parmi les plus connus on peut citer Word2vec [MIKOLOV et al. 2013a, MIKOLOV et al. 2013b] et Glove [PENNINGTON et al. 2014]. En revanche, Glove se base plutôt sur des statistiques de co-occurrences entre mots et de la factorisation de matrices.

Cette idée d'apprendre à générer des représentations latentes robustes et de faibles dimensions, par le moyen d'une tâche prétexte, et en utilisant principalement des réseaux de neurones profonds, a inspiré l'apparition de tout un nouveau paradigme d'apprentissage appelé l'apprentissage auto-supervisé. Dans le NLP de manière générale, on parle communément de modèles de langue.



Source: Exploiting Similarities among Languages for Machine Translation paper.

Figure 1 : Apprentissage de la génération des représentations CBOW et Skip-gram

- Classification :

Les réseaux de neurones récurrents (Recurrent Neural networks ou RNN) [ELMAN 1990] sont connus par leur capacité à modéliser les données séquentielles. De plus, les RNN bidirectionnels [Schuster et Paliwal 1997] offrent la possibilité d'analyser les séquences dans les deux sens et donc de capturer les dépendances des événements passés et futurs.

[ZHOU et al. 2017] utilisent un premier module LSTM bidirectionnel (BLSTM) [Graves et Schmidhuber, 2005] afin de produire des représentations locales de toute la séquence prenant en compte les dépendances lointaines. Après un module CNN [LECUN 1998] qui compile ces informations locales, un deuxième module BLSTM analyse les représentations générales avant d'effectuer la décision de la classe à l'aide d'une fonction de décision appelée « normalisation exponentielle », connue sous le nom « Softmax » [BRIDLE 1989].

[YANG et al. 2016] sont allés jusqu'à partager les poids de leur modèle pour assurer une classification de séquence multitâche et multilingue. Ils utilisent une variante des RNN appelée, Gated Reurrent Unit [CHO 2014], ou GRU, d'une manière hybride, à savoir au niveau caractère et au niveau mot avec une fonction de décision de type CRF.

En effet, l'algorithme CRF est très fréquemment utilisé grâce à sa capacité de capturer les relations entre les classes des différents événements de la séquence [HUANG et al. 2015, ZHENG et al. 2017, REI et al. 2016]. La combinaison des BLSTM et les CRF a été considérée pendant plusieurs années le socle des approches à l'état de l'art [MA et al. 2016, ZHOU et al. 2017, LIN et al. 2017, CHIU et al. 2016, NGUYEN et al. 2016].

En revanche, CRF a certains inconvénients, surtout au niveau de sa consommation de ressources de calcul. En outre, son apport est devenu moins évident comparé à la

fonction Softmax après l'apparition du concept de modélisation de langue basée sur l'apprentissage auto-supervisé [CUI 2019].

- Modélisation de la langue :

Les performances de l'état de l'art ont été dépassées récemment grâce à l'introduction des modèles de langue. L'intuition de base d'un modèle de langue et de pouvoir modéliser la probabilité d'une séquence de mots. Grâce à des réseaux de neurones d'architecture adaptées, comme les RNNs, et un pré-entraînement sur de grandes quantités de données non annotées, ces modèles ont la capacité de générer des représentations latentes riches en informations linguistiques. Ces représentations permettent également de réduire le besoin d'une grande partie du module de classification.

[REI 2017] propose un apprentissage hybride. Son modèle apprend en même temps à prédire les deux mots voisins et l'étiquette (entité nommée) du mot en cours. [PETERS et al. 2017] proposent un modèle de langue basé sur des LSTM bidirectionnels qui permet de générer des représentations pour chaque mot, prenant en compte le contexte passé et le contexte futur. Le modèle de langue est entraîné à prédire le mot courant en se basant sur les *n* mot précédents mais aussi sur les *m* mots suivants. En effet, dans ce travail, chaque direction est entraînée d'une manière indépendante. Les sorties des deux sous-modèles sont ensuite combinées pour former les représentations bidirectionnelles.

Les travaux précédents se basent sur les mots comme unité lexicale. [PETERS et al. 2018] proposent un modèle de langue bidirectionnel nommé ELMo qui analyse les phrases à l'échelle du caractère. Ce niveau de précision permet de modéliser les caractéristiques d'unités sous lexicales (lemmes, suffixes, syllabes...) et donc une meilleure compréhension des spécificités grammaticales et sémantiques. En combinant des représentations d'unités sous-lexicales, ce type de modèle peut aussi traiter des mots hors-vocabulaire.

L'introduction des réseaux de neurones basés sur le concept de « Self-attention », appelés « Transformers » [VASWANI et al. 2017] a révolutionné l'efficacité des modèles de langue. Ces architectures permettent de modéliser les dépendances entre les différents événements d'une séquence d'une manière plus efficace que les réseaux de neurones récurrents. Dotés d'une vision globale de l'entrée, et évitant la séquentialité du calcul, les Transformers sont entraînés d'une manière plus optimale avec une meilleure parallélisation des opérations.

Les Transformers sont composés de deux modules, à savoir, l'encodeur et le décodeur. L'approche GPT [BROWN et al. 2020] proposée par OpenAI entraîne le décodeur du Transformer dans la tâche basique de modélisation de langue, à savoir, la prédiction du mot suivant en utilisant uniquement les mots précédents. Plus récemment, une équipe de Google ont entraîné un encodeur de Transformer avec une contextualisation « bidirectionnelle », c'est à dire, en prenant en compte à la fois le contexte droit et le contexte gauche [DEVLIN et al. 2018]. En effet, la tâche prétexte principale, nommée

« Masked Language Model » (MLM), consiste à cacher une partie de l'entrée (15% pour BERT) choisie aléatoirement et de la prédire en se basant sur les unités sous-lexicales précédentes et suivantes. Si dans une phrase, le $i^{ème}$ événement (unité sous-lexicale) est choisi, il est remplacé par l'unité *[MASK]* pour 80% des fois. Dans le reste des cas, il est remplacé ou bien par un mot aléatoire ou par le mot lui-même (voir la Figure 2). Selon les auteurs, ceci entraînera l'encodeur à produire des représentations contextuelles génériques.

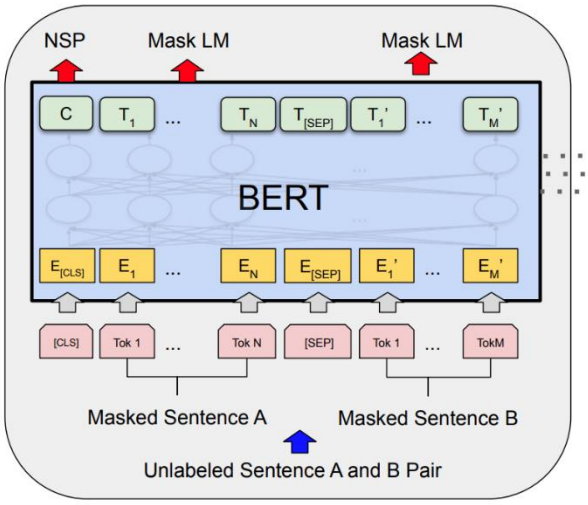


Figure 2 : Pré-entraînement du modèle BERT [DEVLIN et al. 2018]

Afin d'accentuer le pouvoir de compréhension sémantique, une deuxième tâche prétexte qui permet de modéliser la relation entre deux phrases est également utilisée lors de l'apprentissage. Cette tâche, appelée « Next Sentence Prediction » ou NSP, consiste à prédire si la phrase suivante correspond à celle du texte d'origine, ou à une phrase différente. Il s'agit donc de prédire respectivement les classes *IsNext* et *NotNext*. Les exemples relatifs à la deuxième classe sont triés aléatoirement du corpus d'apprentissage.

2.2. NER à partir de la parole

De nombreux travaux se sont intéressés à la tâche de NER à partir d'un texte écrit. Durant les dernières années, les scientifiques ont commencé à attaquer une problématique encore plus compliquée, à savoir, de reconnaître les entités nommées à partir de la parole. Deux catégories d'approches ont été proposées : les approches en cascade (Pipeline) et les approche de bout en bout (End-to-End).

Les approches en cascade consistent à résoudre cette tâche en deux étapes :

- Une reconnaissance automatique de la parole (Automatic Speech Recognition ou ASR), pour transcrire la suite de mots prononcés.
- Une reconnaissance d'entités nommées à partir du texte transcrit automatiquement.

Cette catégorie d'approches tire profit de la disponibilité des données dans chaque sous-tâche, surtout pour les langues les plus connues comme l'Anglais. En revanche, quand chaque tâche est entraînée séparément, la tâche de NER n'apprend pas les spécificités d'un texte issu de la parole (formulations spontanées, hésitations, répétitions...). En outre, si l'ASR effectue des erreurs de transcription, ces erreurs vont se propager à l'étape NER et peuvent donc l'induire à l'erreur.

Les approches de bout en bout proposent de limiter ces faiblesses en apprenant en même temps à reconnaître les entités nommées et leurs texte correspondant. Les deux étapes sont donc fusionnées en un seul modèle. Ce modèle est appris sur des données vocales annotées en étiquettes d'entités nommées. Cependant, ces données sont généralement rares et chères. L'entraînement de ces modèles est également plus couteux. Par exemple, si on veut ajuster certaines étiquettes, il faut réapprendre tout le modèle, alors que dans une approche en cascade, il suffit de réapprendre la partie NER.

Afin de reconnaître les entités nommées dans des conversations vocales médicales, [COHN et al. 2019] proposent une approche en cascade. Les auteurs se basent sur les entités nommées reconnues afin de localiser et enlever les références des patients des enregistrements audio. Le système atteint 90 % de F-mesure sur des données préparées à partir de corpus académiques anglais.

[GHANNAY et al. 2018] proposent un système en cascade et un système de bout en bout pour une NER sur la langue française. Le modèle de bout en bout est basé sur une architecture de type « DeepSpeech 2 » [AMODEI et al. 2016] avec un étiquetage au niveau « caractère » en 9 catégories d'entités nommées. Le modèle est pré-entraîné d'abord sur une tâche ASR puis ajusté sur la tâche de NER à partir de la parole. Afin d'augmenter la quantité de données d'apprentissage, les auteurs utilisent un système de NER classique afin de générer des étiquettes à partir de données audios pour les considérer comme référence. Même si le modèle de bout en bout réussit à mieux détecter l'existence d'entité nommée, l'approche en cascade réussit toujours mieux à les localiser dans la phrase.

[YADAV et al. 2020] adaptent eux aussi l'architecture DeepSpeech 2 pour reconnaître les entités nommées divisées cette fois en 3 catégories. Leur système de bout en bout surpasse un système en cascade pris comme base de référence avec respectivement 90% et 80% de F-mesure sur un corpus préparé pour les mêmes auteurs.

Bien que certains travaux récents aient remarqué que les systèmes de bout en bout peuvent avoir une meilleure performance dans certains contextes [PASAD et al. 2021]. La majorité des travaux n'y trouvent pas encore un réel avantage [SHON et al. 2022]. En effet, les scientifiques préfèrent souvent les approches en cascade pour leurs performances à l'état-de-l'art, mais aussi pour leurs avantages d'aspect pratique évoquées ci-dessus [CHEN et al. 2022, BARIL et al. 2022, NIGMATULINA et al.2022].

3. Expériences et résultats

Vu les avantages de la stratégie en cascade, nous l'adoptons dans la suite de ce travail. Si nous utilisons un moteur d'ASR propriétaire, nous développons notre propre solution de reconnaissance d'entités nommées. Nous détaillons tout d'abord le protocole expérimental. Ensuite, nous décrivons notre approche et analysons les résultats de nos expériences.

3.1. Corpus de données

VoxPopuli [WANG et al. 2021] est un corpus de données, proposé par Meta AI (anciennement appelée Facebook AI Research) qui consiste en des enregistrements audios et transcriptions d'événements au sein du parlement européen. A la base, il contient 400 000 heures de parole couvrant 23 langues, dont 1800 heures sont transcrites manuellement pour 16 langues. Parmi ces dernières 543h sont en langue anglaise.

Récemment [SHON et al. 2022] ont annoté, en 7 catégories d'entités nommées (EN), environ 25 heures de parole en langue anglaise. Nous allons appeler ce corpus « Voxpopuli-NE » dans le reste de ce rapport. Les auteurs ont pris le même partitionnement de base du corpus en prenant en compte entièrement les parties Dev et Test. En revanche, ils ont annoté uniquement 15,5 heures de de la partie Train, et l'ont appelé Fine-tune (voir Tableau 1).

Tableau 1 : Distribution du corpus Voxpopuli-NE

Partie	**Fine-tune**	**Dev**	**Test**
Durée (h)	15,5	5	4,9
Nbre de phrases	5000	1753	1,842

À notre connaissance, le corpus Voxpopuli-NE est le seul corpus d'enregistrements vocaux annotés en entités nommées, disponible en open source, pour la langue anglaise. Nous décidons donc d'effectuer nos expériences sur ces données. En revanche, les auteurs ont publié uniquement les données des parties Fine-tune et Dev et ont gardé la partie Test. Nous nous basons donc uniquement sur ces deux parties-là dans nos expériences. Voxpopuli-NE est annoté en deux niveaux d'entités nommées, décrits dans le Tableau 2. Le premier niveau concerne 7 catégories génériques. La Figure 3 présente la fréquence de ces catégories dans chacune des parties Fine-tune et Dev. Nous remarquons à partir de ce tableau une grande disparité entre les différentes classes. En effet, la classe la plus fréquente (PLACE) contient 2654 occurrences alors que le classe moins fréquentes (LAW) existe en seulement 310 occurrences. Cet écart dans la distribution des classes représente un défi supplémentaire dans cette tâche de reconnaissance d'entités nommées. En effet les futurs systèmes peuvent apprendre implicitement à privilégier les classes les plus fréquentes au moment de la prédiction.

Tableau 2 : Entités nommées du corpus Voxpopuli-NE

Étiquette générique	Description	Étiquette détaillée
PLACE	Emplacement : ville, pays, région, continent…	GPE, LOC
QUANT	Nombres	CARDINAL, MONEY, ORDINAL, PERCENT, QUANTITY
ORG	Entreprise, école, groupe politique, gouvernement…	ORG
WHEN	Date ou heure	DATE, TIME
NORP	Nationalité ou groupe religieux ou politique	NORP
PERSON	Nom de personne	PERSON
LAW	Texte légal	LAW

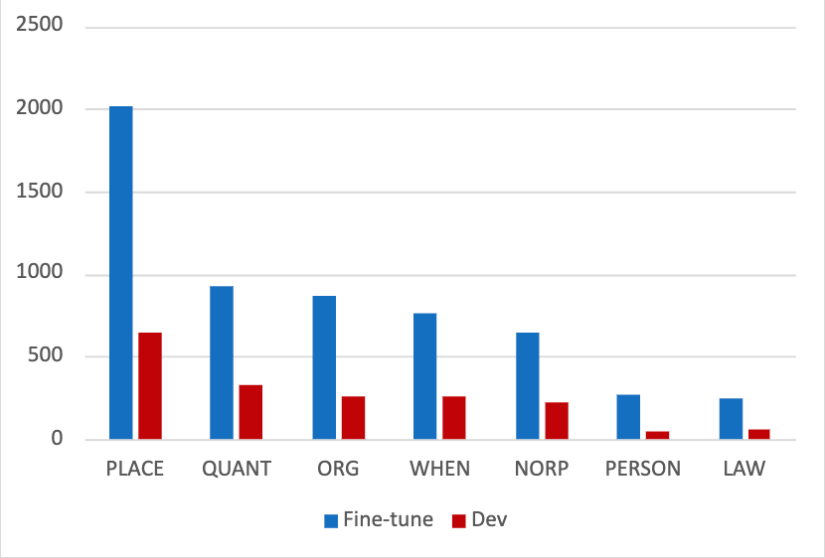


Figure 3 : Distribution des classes dans le corpus Voxpopuli-NE

Par ailleurs, ce corpus de données offre aussi un deuxième niveau d'annotation avec des définitions plus précises, surtout pour les classes les plus fréquentes. Par exemple, on distingue dans les entités de nombres entre les pourcentages, les valeurs monétaires etc. Pour nos besoins actuels, nous nous limitons à la taxonomie générique.

Comme dit précédemment, nous pouvons travailler uniquement sur les parties Fine-tune et Dev. Afin d'évaluer nos futurs systèmes nous utilisons donc la partie Dev et nous la renommons en « Eval ». Pour ce qui est de la partie Fine-tune, nous en prenons 5% pour optimiser nos futurs modèles et gardons le reste pour l'apprentissage. Pour ce faire, nous sélectionnons aléatoirement 269 phrases pour la partie dénommée « Valid » et mettons les 4731 phrases restantes dans la partie « Train » (voir Tableau 3).

Tableau 3 : Nouveau découpage du corpus Voxpopuli-NE

Partie	**Train**	**Valid**	**Eval**
Durée (h)	14,7	0,8	4,9
Nbre de phrases	4731	269	1842

En prenant en compte la préparation évoquée ci-dessus, les données Voxpopuli-NE restent tout de même disposées sous le format d'origine. En effet, deux sources d'informations sont disponibles pour chaque partie du corpus, comme illustré dans la Figure 4.

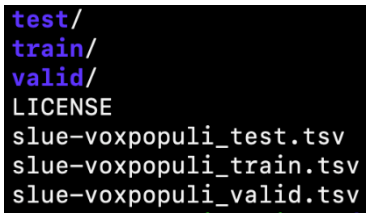


Figure 4 : Fichiers et dossiers de Voxpopuli-NE

Pour chacune des 3 parties « Train », « Valid » et « Test » nous disposons de :

- Un dossier contenant les enregistrements audio sous l'extension *.ogg*
  Example : *20150518-0900-PLENARY-15-en_20150518-18:48:27_2.ogg*
- Un fichier d'extension *.tsv* contenant les annotations manuelles sur 7 colonnes séparées par une tabulation.

Tableau 4 : Contenu du fichier tsv

Colonne	**Description**	**Exemple**
id	Identifiant de l'enregistrement	*20150518-0900-PLENARY-15-en_20150518-18:48:27_2*
raw_text	Texte d'origine	*We all agreed, at the last session in Strasbourg that development is important, but we need to remember it now when we are talking about the financial contributions.*
normalized_text	Texte normalisé en enlevant les majuscules et les ponctuations.	*we all agreed at the last session in strasbourg that development is important but we need to remember it now when we are talking about the financial contributions.*
speaker_id	Identifiant du locuteur	*119435*
split	Partie du corpus	*eval*
raw_ner	Catégorie, position et longueur des EN	*[['Place', 38, 10]]*

	dans le texte brut	
normalized_ner	Catégorie, position et longueur des EN dans le texte normalisé	*[['Place', 37, 10]]*

Les différentes colonnes ainsi qu'un exemple sont présentés dans le Tableau 4. Nous nous intéressons dans nos expériences principalement à 3 colonnes : *id*, *normalized_text*, et *normalized_ner*. La dernière colonne représente un tableau listant la catégorie et les positions des entités nommées. Dans l'exemple cité, il existe une seule entité nommée qui appartient à la catégorie « Place ». Cette EN s'étale sur le texte commençant par le 38ème caractère (on compte à partir de 0) et finissant par le 47ème caractère. Il s'agit ici du mot « strasbourg ».

Comme nous avons pu voir à travers cet exemple, le format d'origine des données est relativement compliqué et souffre d'un manque de lisibilité. Par conséquent, nous le convertissons en un format très communément utilisé dans les tâches de NER, que nous appelons « Word-NE ». Les nouveaux fichiers contiennent un mot par ligne suivi par son étiquette. Si une étiquette regroupe plusieurs mots, on ajoute le préfixe « B- » (pour *begin*) à l'étiquette du premier mot et le préfixe « I- » (pour *inside*) à celles des mots suivants. Les mots qui ne représentent pas d'entités nommées prennent l'étiquette « O » (pour *other*). L'exemple présenté dans le Tableau 4 devient alors sous la forme schématisée dans la Figure 5. Enfin, une ligne vide sépare les différentes phrases.

```
we  O
all O
agreed  O
at  O
the O
last    O
session O
in  O
strasbourg  B-PLACE
that    O
development O
is  O
important   O
but O
we  O
need    O
to  O
remember    O
it  O
now O
when    O
we  O
are O
talking O
about   O
the O
financial   O
contributions   O
```

Figure 5 : Exemple de phrase annotée au format Word-NE

3.2. Métriques d'évaluation

La F-mesure est une métrique très connue dans le domaine d'apprentissage supervisé, surtout quand il s'agit de données comportant des classes déséquilibrées, comme pour le cas de Voxpopuli-NE. S'agissant d'une moyenne harmonique du rappel et de la précision moyens, elle prend en compte les faux positifs et les faux négatifs pour chaque classe. Les rappels et précisions moyens représentent respectivement les moyennes du rappel et de la précision de chaque classe. La F-mesure considère donc toutes les classes d'une manière égale même si certaines sont beaucoup plus nombreuses que d'autres.

$$F = 2 \cdot \frac{(\text{Précision moyenne} \cdot \text{Rappel moyen})}{(\text{Précision moyenne} + \text{Rappel moyen})}$$

Pour une classe donnée, la précision représente le nombre de prédictions correctes parmi celui des prédictions de cette classe. Alors que le rappel représente le nombre de prédictions correctes parmi celui des exemples qui appartiennent réellement à cette classe.

$$\text{Précision} = \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux positifs}}$$

$$\text{Rappel} = \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux négatifs}}$$

Dans une tâche d'étiquetage automatique, comme celle de la reconnaissance d'entités nommées, chaque mot est considéré comme une occurrence d'une classe. Un mot de la classe X, peut être donc soit correctement prédit comme entité nommée X (vrai positif pour la classe X), soit affectée à tort à la classe Y (un faux négatif pour la classe X et un faux positif pour la classe Y).

Dans l'échantillon illustré dans le Tableau 5, la classe ORG a été correctement prédite 3 fois. La totalité des prédictions de cette classe est de 3 donc elle a une précision de 100% (3/3). Et elle existe en 4 occurrences dans la référence, donc elle a 60% de rappel (3/5). Pour la classe EVENT, elle a été prédite 4 fois dont 3 étaient bonnes, donc elle a une précision de 75% (3/4). Et aucune occurrence de référence n'a été ratée, donc elle a un rappel de 100% (3/3). La précision moyenne est donc de 87.5% ((100+75)/2) et le rappel moyen est de 80% ((60+100)/2). La F-mesure totale, selon la formule ci-dessus, serait donc d'environ 83.6%. Il est également possible de considérer la F-mesure par classe, qui consiste à appliquer individuellement la même formule sur les rappel et précision de chaque classe.

Jusqu'ici, le calcul de la F-mesure se base sur une prédiction effectuée sur le même texte que celui de la référence, comme évoqué dans l'exemple ci-dessus. En revanche, Dans le cadre de la tâche de NER à partir de la parole, le texte reconnu peut-être différent de la réalité, à cause des erreurs potentielles de la reconnaissance de la parole. Ce défi est présent que ce soit si on est en mode cascade ou en mode de bout en bout.

Tableau 5 : Exemple de phrase avec faux positifs et faux négatifs

Texte	Référence	Prédiction	ORG	EVENT
efsi	B-ORG	B-EVENT	Faux négatif	Faux positif
contributes				
to				
the				
cop	B-EVENT	B-EVENT		Vrai positif
twenty	I-EVENT	I-EVENT		Vrai positif
one	I-EVENT	I-EVENT		Vrai positif
objectives				
and				
reinforces				
the				
european	B-ORG	B-ORG	Vrai positif	
investment	I-ORG	I-ORG	Vrai positif	
advisory	I-ORG	I-ORG	Vrai positif	
hub	I-ORG		Faux négatif	

La solution que nous utilisons dans ce travail est proposée par [SHON et al. 2022]. Elle consiste à enlever la contrainte de position des entités nommées dans la phrase, vu que les deux phrases de référence et prédite risquent de ne pas être alignés mot par mot. Par conséquent, un vrai positif est compté s'il existe un mot *W* prédit et annoté en entité nommée *N* et que ce couple existe dans la référence aussi. Avec le même raisonnement, si ce couple *(W, N)* n'existe que dans la prédiction, il s'agit d'un faux positif, et s'il n'existe que dans la référence, il s'agit d'un faux négatif. Malgré la flexibilité de cette métrique, elle a quand même une faiblesse. En effet, si la référence contient un couple *(W, N)*, par exemple, au tout début de l'enregistrement, et que par chance, le système prédit, à tort, le même couple à la fin de l'enregistrement, la métrique considère ici qu'il s'agit tout de même d'un vrai positif, vu qu'elle ne prend pas en compte l'alignement des mots.

3.3. Environnement technique

De point de vue matériel, les expériences présentées dans le reste de ce rapport sont conduites sur un serveur ayant les spécifications suivantes :

- Une carte graphique (Graphical Processing Unit ou GPU) Nvidia de type « Tesla P40 » avec 24Go de mémoire (VRAM)
- Deux processeurs Intel Xeon Gold 5218 de 16 cœurs
- Une mémoire vive (RAM) de 256 Go

De point de vue logiciel, nous utilisons le langage python (version 3.9). Pour concevoir, entraîner et évaluer les différents modèles de réseaux de neurones, nous utilisons l'outil open-source Tensorflow (2.6.0). Afin de tirer profit de la grande capacité de la GPU à

exécuter des calculs matriciels massifs, cette librairie est basée sur la technologie CUDA (version 11.4), développée par NVIDIA.

3.4. Approche proposée

Afin de traiter la problématique de NER à partir de la parole, nous utilisons deux systèmes d'apprentissage profond inspirés de deux architectures à l'état de l'art dans cette tâche.

3.4.1. Glove-BLSTM

Notre système de base consiste en un réseau de neurones récurrent de type LSTM bidirectionnel (BLSTM), schématisé dans la Figure 6. Pour chaque mot, il prend en entrée sa représentation vectorielle, et produit en sortie un vecteur de probabilités des classes à prédire.

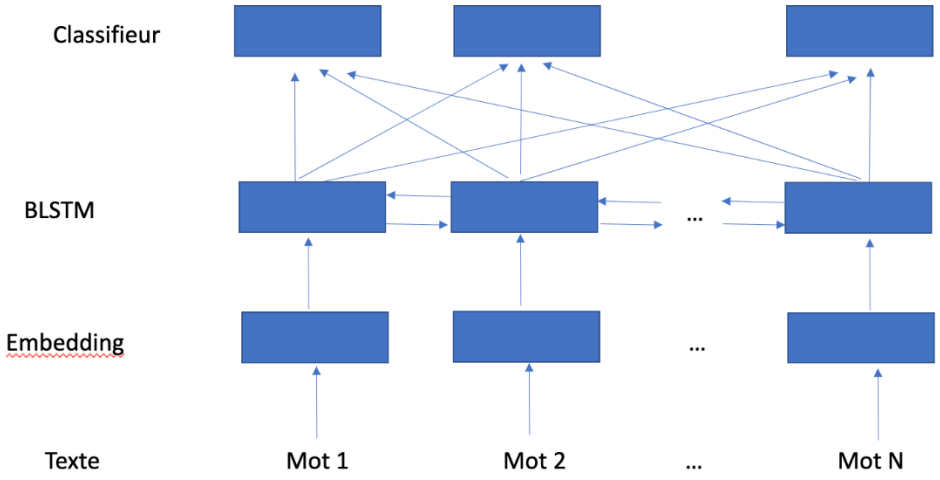


Figure 6 : Schéma de l'approche Glove-BLSTM

Pour l'extraction de caractéristiques à partir des mots d'une phrase, nous utilisons une méthode à l'état de l'art appelée Glove (Global Vectors for Word Representation) [PENNINGTON et al. 2014]. Glove est une approche qui génère des représentations vectorielles de mots à partir de statistiques calculées sur de grandes bases de données textuelles. Concrètement, l'idée consiste d'abord à produire une matrice de co-occurrence entre mots. Chaque élément de cette matrice représente le nombre d'occurrences d'un mot X dans l'entourage du mot Y, c'est-à-dire, le nombre de fois où le mot X fait partie des *n* mots précédents ou suivants du mot Y. Un exemple de ce type de matrice est illustré dans la Figure 7.

La ligne ou la colonne de chaque mot peut être considérée comme une représentation vectorielle de ce mot. En revanche, cette représentation peut être très grande si la taille du vocabulaire est grande. En plus, dans le cadre d'un vocabulaire dynamique, cette matrice doit être recalculée à chaque ajout ou suppression d'un mot. Pour réduire la dimension de ses représentations, les auteurs proposent une factorisation de matrices.

De nouvelles représentations vectorielles (ou plongements) de mots sont optimisées en minimisant, pour chaque couple de mots, la différence entre le produit scalaire de leurs nouvelles représentations respectives, et le logarithme de leur cooccurrence.

	mot1	mot2	mot3	mot4
mot1	1	5	3	2
mot2	5	1	4	2
mot3	3	4	1	7
mot4	2	2	7	1

Figure 7 : Exemple de matrice de co-occurrence entre mots[1]

L'université Stanford, qui a développé cette approche, offre aussi des dictionnaires de plongements clés-en-main pour environ 6 milliards de mots et symboles utilisés dans des documents en langue anglaise. Nous utilisons un des dictionnaires les plus populaires qui est pré-entraîné sur un corpus d'articles de presse et d'article Wikipedia totalisant 6 milliards de mots. Ce dictionnaire comporte un vocabulaire de 400 mille mots uniques avec leurs plongements. Chaque mot est représenté par un vecteur de 100 dimensions.

Pour ce qui est du composant BLSTM, schématisé dans la Figure 8, il consiste en deux couches parallèles récurrentes de type LSTM (Long Short-Term Memory). Pour le LSTM Forward (vers l'avant), chaque cellule C_i prend en entrée la représentation vectorielle du mot m_i ainsi que l'état caché h_{i-1} de la cellule précédente. Elle produit ensuite un état caché h_i et une sortie o_i. Le LSTM Backward (vers l'arrière) effectue le même traitement mais dans le sens opposé. Chaque cellule produit un vecteur de 128 nombres réels. Enfin, pour chaque étape, les deux sorties Forward et Backward, sont concaténées.

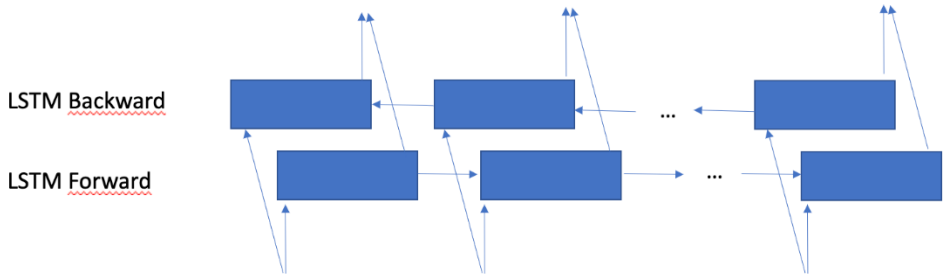


Figure 8 : Schéma d'une couche BLSTM

La décision de classification est effectuée grâce à une couche complètement connectée qui prend en entrée les vecteurs générés par le composant BLSTM. Pour chaque mot, le

[1] https://tel.archives-ouvertes.fr/tel-01902781/document

vecteur de sortie de cette couche est transformé à l'aide d'une fonction exponentielle normalisée, plus connue sous le nom « Softmax » :

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Grâce à la formule de la fonction Softmax, nous pouvons avoir un vecteur de probabilités en sortie, comprises entre 0 et 1, et dont la somme est égale à 1.

Pour l'entraînement de ce réseau de neurones, nous comparons, pour chaque mot dans un exemple d'apprentissage, les probabilités prédites avec l'encodage one-hot de la classe de référence, en utilisant l'entropie croisée comme fonction de coût :

$$\mathrm{H}(p, q) = -\sum_{x} p(x) \log q(x)$$

où *p(x)* est la probabilité réelle de la classe *x*, et *q(x)* est la probabilité prédite. Ensuite, nous propageons l'erreur calculée et mettons à jour les poids du réseau à l'aide de l'algorithme d'optimisation « Adam » qui est une extension de la descente de gradient stochastique.

3.4.2. BERT

Notre deuxième système se base sur les toutes récentes évolutions dans l'état de l'art du traitement automatiquement de la langue, à savoir, la modélisation de la langue avec les réseaux de neurones de type Transformer. Nous tirons profit spécifiquement du modèle BERT proposé par les chercheurs de Google (voir la section 2.1.3) pré-entraîné avec un paradigme d'apprentissage auto-supervisé.

BERT prend en entrée des unités sous-lexicales et non pas uniquement des mots. En effet, cette approche de tokenisation est un juste milieu entre l'utilisation des mots et l'utilisation des caractères. Elle a l'avantage de limiter la taille du vocabulaire et la probabilité de rencontrer des mots hors-vocabulaire, tout en évitant d'avoir des séquences très longues, avec des unités très élémentaires qui ne portent aucun sens. BERT utilise un algorithme de tokenisation développé dans la même entreprise, appelé « WordPiece ».

Le tableau Tableau 6 montre quelques exemples de la tokenisation WordPiece. On pourrait voir que l'algorithme a identifié une racine commune « surf » entre les mots « surf » et « surfing ». Il détecte aussi un sens commun entre « surfboard » et « snowboard » via l'unité « board ». Cette tokenisation permet à BERT de mieux comprendre les proximités et relations sémantiques qui peuvent exister entre certains mots.

Tableau 6 : Exemples de tokenisation de type WordPiece[2]

Word	Token(s)
surf	['surf']
surfing	['surf', '##ing']
surfboarding	['surf', '##board', '##ing']
surfboard	['surf', '##board']
snowboard	['snow', '##board']
snowboarding	['snow', '##board', '##ing']
snow	['snow']
snowing	['snow', '##ing']

Le modèle BERT est basé sur l'architecture Transformer qui représente maintenant l'état de l'art pour traiter les données séquentielles. Grâce aux mécanismes de « Self-attention », les Transformers prennent à la fois toute la séquence contrairement au réseau de neurones récurrents qui doivent traiter les données d'une manière séquentielle. L'architecture de base des Transformer est constituée d'un encodeur et d'un décodeur, très similaires. BERT utilise uniquement la partie « encodeur » étant donné que son but est de générer des représentations vectorielles des données en entrée.

Le cadre de fond gris dans la Figure 9 représente un bloc (ou une couche complexe) du Transformer. Il consiste principalement d'un groupe de *M* modules de Self-attention, appelés « têtes d'attention » et d'une couche complètement connectée, ainsi que des connexions résiduelles et des opérations de normalisation. Le mécanisme de Self-attention consiste en un produit matriciel pondéré où les poids d'attention sont calculés simultanément entre eux.

Google offre à la communauté plusieurs modèles pré-entraînés sur de très grandes quantités de données textuelles provenant de Wikipedia et du corpus BookCorpus[3]. Les deux modèles les plus connus sont « BERT-Base » et « BERT-Large ». Le premier consiste en un modèle de taille standard alors que le deuxième représente une tentative de construire un modèle exceptionnellement grand. Le Tableau 7 donne une idée sur la taille de ces deux modèles.

Vu les contraintes matérielles liées à notre serveur de calcul, nous conduisons la majorité de nos expériences avec le modèle pré-entraîné « BERT-Base ». Nous ajustons ce

[2] https://towardsdatascience.com/how-to-build-a-wordpiece-tokenizer-for-bert-f505d97dddbb
[3] https://yknzhu.wixsite.com/mbweb

modèle sur la tâche de reconnaissance des entités nommées en lui ajoutant une (ou plusieurs) couche complètement connectée.

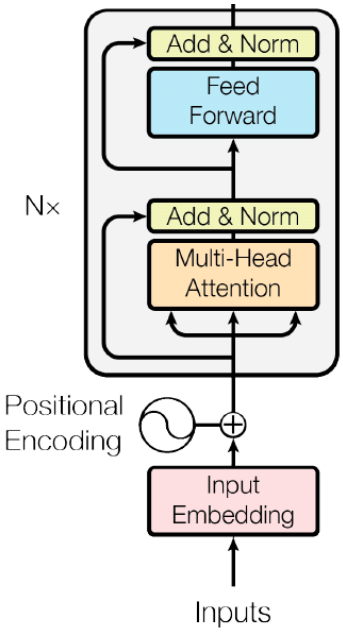


Figure 9 : Schéma d'un encodeur BERT

Tableau 7 : Caractéristiques des modèles BERT-Base et BERT-Large

	BERT-Base	**BERT-Large**
Nombre de paramètres	110M	340M
Nombre de blocs Transformer	12	24
Taille des blocs Transformer	768	1024
Nombre de têtes d'attention	12	16

Comme pour le modèle BLSTM, les sorties des dernières couches sont transformées en un vecteur de probabilités sur les classes à prédire à l'aide de la fonction Softmax. Nous entraînons également ce nouveau réseau à l'aide de l'entropie croisée et l'algorithme d'optimisation « Adam ».

3.5. NER à partir du texte de référence

Avant d'attaquer la reconnaissance des entités nommées à partir de la parole, nous entraînons et comparons les architectures présentées ci-dessus dans une tâche de NER classique. En effet nous entraînons les différents systèmes sur le texte de référence des

4731 phrases de la partie Train du corpus Voxpopuli-NE et l'évaluons sur le texte de référence des 1842 phrases de la partie Test, selon le nouveau découpage présenté dans le Tableau 3.

En premier temps, nous analysons les performances de trois systèmes :

- Un premier système basé sur le modèle BLSTM et prenant en entrée des plongements de mots de type Glove, qu'on appelle ici « **Glove-BLSTM** ». Nous le considérons comme système de base ou « baseline ».
- Un deuxième système basé sur le modèle BERT-Base avec une seule couche complètement connectée (fully connected ou FC) au-dessus. Nous nommons ce système « **BERT-FC1** ».
- Un troisième système, dénommé « **BERT-L-FC1** », basé sur le modèle BERT-Large avec aussi une seule couche complètement connectée.

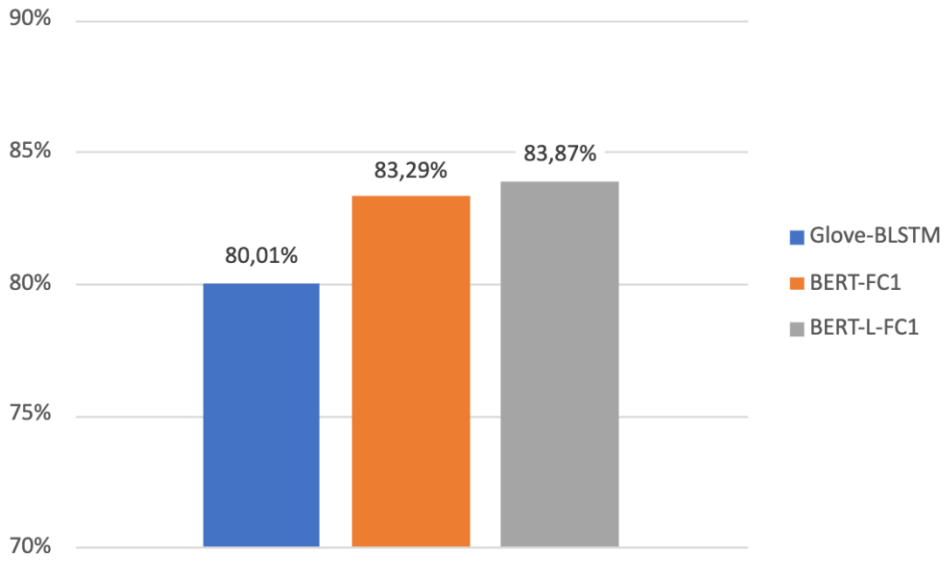


Figure 10 : F-mesure des trois approches de NER sur le texte transcrit manuellement

Les performances de ces trois systèmes, estimés avec la F-mesure, sont présentées dans la Figure 10. Premièrement, nous constatons que l'approche à base de modèle de langue (BERT) dépasse l'approche à base de plongements de mots et de modèle récurrent (LSTM). En effet, les modèles à base de BERT atteignent plus de 83% de F-mesure contre 80% pour le modèle Glove-BLSTM. Les modèles basés sur BERT réalisent donc un gain relatif de au moins 15%.

En revanche, le modèle de langue large n'apporte qu'un léger gain qui ne dépasse pas 0,6% points par rapport au modèle BERT-FC1. En effet, l'amélioration relative est inférieure à 3,5%. Il semble que, dans notre cas, la multiplication par trois de la taille du modèle n'a pas un impact aussi marquant que celui du changement vers une architecture plus performante. Comme discuté dans la section 3.4.2, le modèle BERT-L-FC1, avec ses 340M de paramètres est très lent à entraîner sur notre infrastructure matérielle. En plus, il nous a fallu faire quelque compromis sur certains hyperparamètres pour pouvoir

l'entraîner. Notamment, nous n'avons pas pu dépasser une taille de batch de 16 exemples d'apprentissage. Par conséquent, vu qu'il s'agit de la même architecture, avec une taille de modèle beaucoup plus importante, et un apport non signifiant, nous continuons le reste de nos expériences, non pas avec le modèle BERT-L-FC1, mais plutôt avec le modèle BERT-FC1.

Nous continuons l'analyse des performances des systèmes Golve-BLSTM et BERT-FC1 en observant les F-mesures sur chaque classe. Les chiffres pour ces deux systèmes sont schématisés dans la Figure 11 dans laquelle les classes sont ordonnées selon leurs nombre d'occurrences dans le corpus d'apprentissage. Nous pouvons remarquer une tendance générale dans le sens où les deux systèmes sont plus performants vis-à-vis des classes les plus fréquentes, avec une performance beaucoup plus basse sur la classe la moins fréquente, à savoir les textes légaux, que sur toutes les autres classes. Nous trouvons seulement une seule exception pour la classe NORP. En comparant entre les performances des deux systèmes, nous voyons comme attendu que le système BERT-FC1 est meilleur, ou au moins égal, que le système Glove-BLSTM sur la majorité des classes.

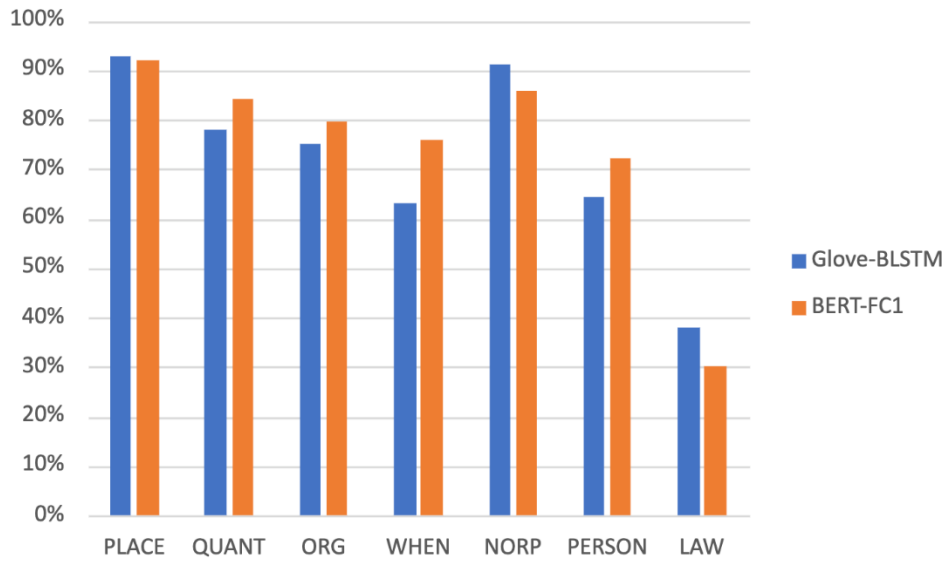


Figure 11 : F-mesure par classe des systèmes Glove-BLSTM et BERT-FC1

Nous nous concentrons maintenant sur le comportement du système BERT-FC1. Nous traçons donc la matrice de confusion entre les différentes classes prédites dans le Tableau 8. Nous ajoutons ici la classe O à titre indicatif. Les lignes représentent les classes de référence et les colonnes représente les classes prédites. Dans la matrice de confusions, les valeurs dans la diagonale représentent les vrais positifs. Les autres valeurs peuvent être lus comme dans les deux exemples suivants :

- Lecture par ligne : Les exemples de la classe O ont été prédits 69 fois comme appartenant à la classe WHEN.

- Lecture par colonne : La classe O a été prédite 42 fois à la place de la classe WHEN.

Tableau 8 : Matrice de confusions du système BERT-FC1

	O	QUANT	WHEN	PLACE	ORG	LAW	NORP	PERSON
O	4122	53	69	27	51	26	18	3
QUANT	19	440	11	0	0	0	0	0
WHEN	42	4	701	1	0	1	0	0
PLACE	12	0	0	743	7	3	8	7
ORG	40	1	1	7	358	10	5	0
LAW	35	11	4	5	14	80	0	0
NORP	10	0	0	9	8	0	200	0
PERSON	6	0	0	1	2	0	1	62

D'une manière générale, nous remarquons un taux de vrais positifs acceptable pour la majorité des classes sauf pour la classe LAW. Nous constatons aussi un certain nombre de confusions entre la classe QUANT et la classe WHEN, probablement car les deux entités contiennent des unités lexicales quantitatives (nombres), par exemple :

Phrase :	*the*	*amendment*	*two*	*hundred*	*and*	*fifty*	*three*
Référence :	*O*	*B-LAW*	*I-LAW*	*I-LAW*	*I-LAW*	*I-LAW*	*I-LAW*
Prédiction :	*O*	*O*	*B-QUANT*	*I-QUANT*	*I-QUANT*	*I-QUANT*	*I-QUANT*

D’autres confusions sont visibles également entre les entités ORG et LAW. Ceci peut être expliqué par une certaine proximité entre le nom de certaines organisations et les textes légaux, surtout quand il s’agit d’acronymes (exemple: *dcfta*, *ippc*, *cop*).

Enfin, nous remarquons que tous les exemples ratés de la classe NORP ont été prédits comme PLACE ou ORG, si on ignore la classe O. En effet, ces erreurs ont majoritairement lieu sur des exemples typiques comme :

Phrase :	the	african	*continent*
Référence :	*O*	*B-NORP*	*O*
Prédiction :	*O*	*B-Place*	*I-Place*

Phrase :	*the*	*socialist*	*and*	*democratic*	*family*
Référence :	*O*	*B-NORP*	*I-NORP*	*I-NORP*	*O*
Prédiction :	*O*	*I-ORG*	*I-ORG*	*I-ORG*	*I-ORG*

3.6. NER à partir de la parole

Jusqu'ici nous avons conçu un système de reconnaissance automatique d'entités nommées à l'état de l’art et analysé son comportement en prenant un texte de référence

comme entrée. Nous étudions maintenant la reconnaissance d'entités nommées à partir de la parole.

Comme évoqué dans la section 2.2, il existe deux stratégies pour aborder cette problématique. La première consiste en un mode cascade avec une reconnaissance automatique de la parole (Automatic Speech Recognition ou ASR) suivi d'une reconnaissance d'entité nommées. Quant à la deuxième, elle concerne les approches de bout en bout, avec un seul modèle qui prend en entrée un enregistrement vocal et produit une séquence de mots accompagnés de leurs étiquettes. Étant donné que l'approche en cascade reste toujours à l'état de l'art, offre plus de flexibilité et est moins compliquée et moins coûteuse à entretenir, nous continuons le reste de nos expériences avec cette approche.

3.6.1. Reconnaissance automatique de la parole

Pour l'ASR, nous utilisons dans ce travail l'outil propriétaire de l'entreprise « Speechmatics ». Cet outil fournit des API (Application Programming Interface ou interface de programmation applicative) d'ASR mais aussi d'autres fonctionnalités de traitement du langage et de la parole basées sur l'apprentissage automatique.

Nous donnons donc les enregistrements vocaux de la partie Test comme entrée à l'API Speechmatics et récupérons la transcription automatique. Nous l'appelons le texte obtenu « Test-auto ». Nous évaluons tout d'abord cette transcription automatique par rapport à la transcription de référence selon le taux d'erreur mots (Word Error Rate ou WER).

Le WER est estimé, après alignement des phrases de prédiction et de référence, en fonction du nombre de mots insérés à torts, le nombre de mots supprimés et le nombre de mots substitués par un autre mot. Il est calculé avec la formule

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

où S est le nombre de substitutions, I le nombre d'insertions et D et le nombre de suppressions.

En comparant les transcriptions dans leurs formats respectifs brut, nous obtenons un taux d'erreur mots d'environ 22,7% (voir Figure 12). Étant donné que, jusqu'ici, nous traitons des données textuelles en minuscules et sans ponctuation, nous transformons la transcription automatique en ce sens et l'évaluons avec la référence en format similaire et nous obtenons environ 15,7% de WER.

3.6.2. Reconnaissance d'entités nommées

Après avoir obtenu la transcription automatique (Test-auto), qui est le résultat de la première étape de l'approche en cascade, nous appliquons notre système BERT-FC1 dessus et l'évaluons avec la F-mesure. Comme montré dans la Figure 13, la performance

de la reconnaissance d'entités nommées descend sous la barre des 70%. Les erreurs de reconnaissance automatique de la parole sont donc responsables d'une chute d'environ 15 points de F-mesure.

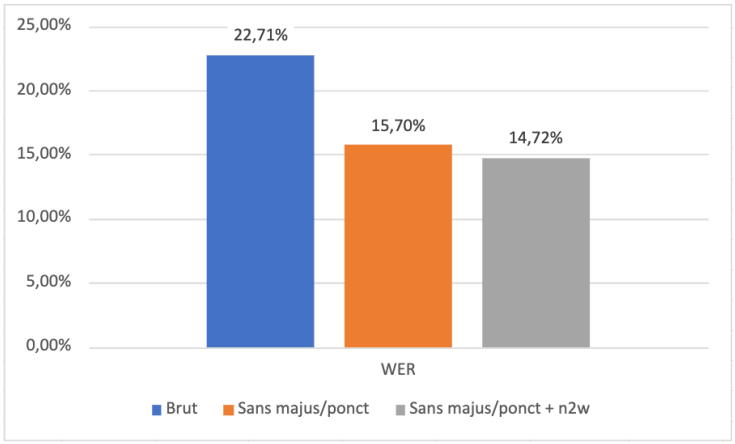


Figure 12 : WER selon le format du texte

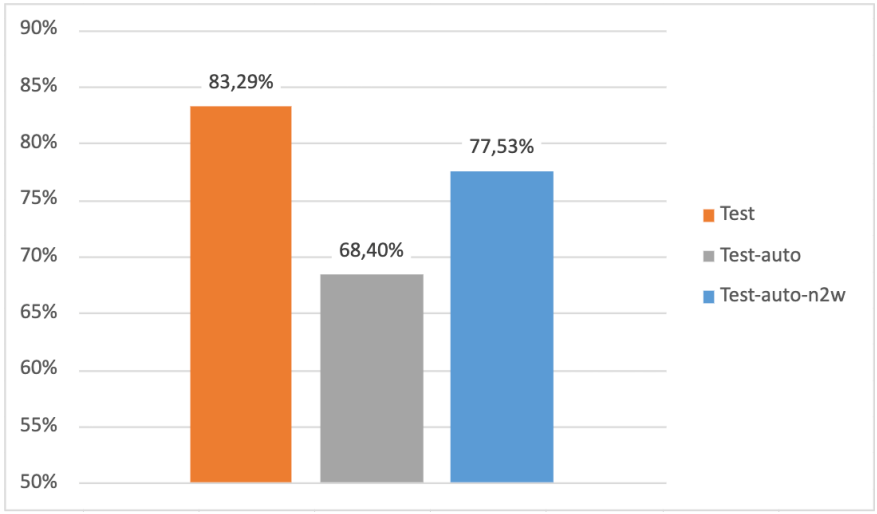


Figure 13 : F-mesure de la NER à partir de la parole

Afin de comprendre le comportement du système BERT-FC1 avec la transcription automatique, Nous regardons le comportement par classe. Nous comparons en premier temps dans la Figure 14 les scores de F-mesure par classe du système BERT-FC1 sur la transcription manuelle (Test) et la transcription automatique (Test-auto).

Nous remarquons tout d'abord, à travers cet histogramme, une baisse de la performance de BERT-FC1 sur les entités nommées qui contiennent des unités lexicales numériques, à savoir, QUANT et WHEN. En examinant davantage la transcription automatique, nous remarquons que les nombres sont transcrits en chiffres par l'outil Speechmatics.

Exemple : *We count 20.000 refugies*

Afin de remédier à ce problème, nous cherchons automatiquement les nombres écrits en chiffres (via des expressions régulières) et les transformons en des nombres en toutes

lettres. Nous effectuons cette conversion à l'aide de *num2words*[4], une librairie open-source en langage Python qui prend en compte les spécificités de plusieurs langues, dont l'anglais. Nous appelons le nouveau texte transcrit « Test-auto-n2w ».

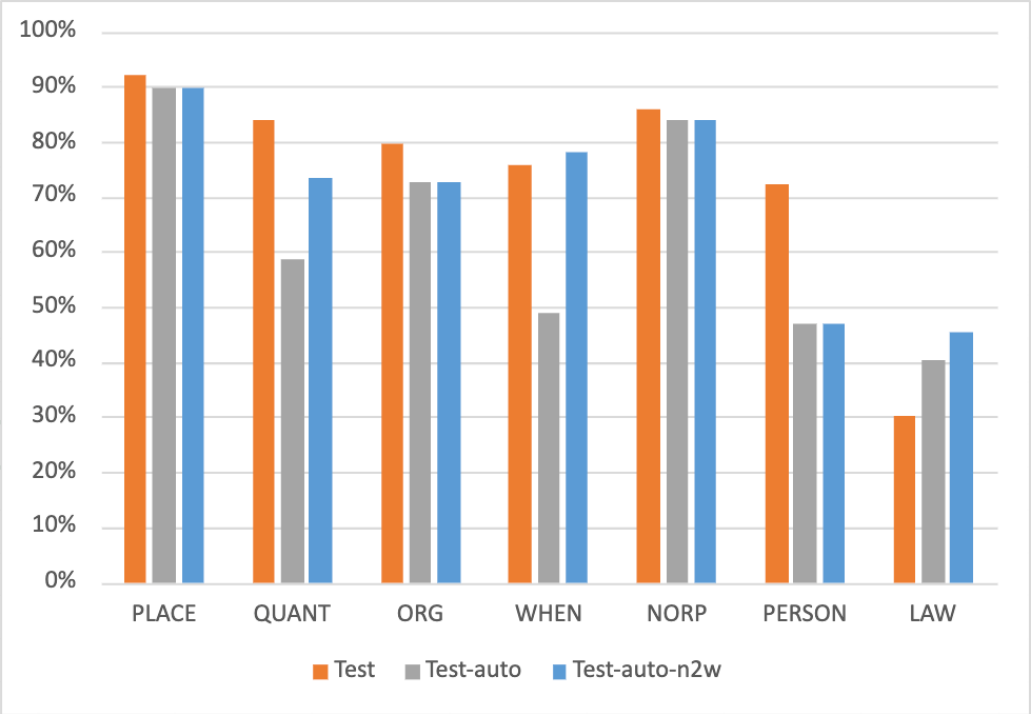


Figure 14 : F-mesure par classe de la NER à partir de la parole

Après ce traitement, nous réévaluons d'abord la qualité de la reconnaissance automatique de la parole. Comme nous pouvons voir dans la Figure 12, le WER passe de 15,7% à environ 14,7%. En ce qui concerne la reconnaissance d'entités nommées, nous pouvons voir à travers la Figure 14 que la transformation des nombres sur Test-auto-n2w a amélioré significativement la F-mesure sur les entités numériques QUANT et WHEN. Par conséquent, l'écart entre la performance de la NER sur la nouvelle transcription automatique et celle de la NER sur la transcription manuelle est significativement réduit. La F-mesure en utilisant Test-auto-n2w n'est réduite que d'environ 6 points par rapport à l'utilisation de Test comme entrée à la reconnaissance d'entités nommées pour atteindre environ 77,5%. Dans les expériences qui suivent, nous prenons toujours en entrée le nouveau texte « Test-auto-n2w » comme sortie de l'étape ASR de notre approche en cascade, et donc aussi comme entrée aux systèmes NER.

Nous constatons également à travers l'histogramme de la Figure 14 une forte diminution de la F-mesure (de 72% à 47%) pour les noms de personnes en utilisant le texte transcrit automatiquement. Pour essayer de palier un peu ce problème, nous enrichissons les données d'apprentissage par des exemples supplémentaires contenant cette catégorie d'entités nommées. Pour ce faire, nous étudions les différents corpus de données

[4] https://github.com/savoirfairelinux/num2words

annotées en entités nommées et nous trouvons trois corpus principaux en langue anglaise, listés dans le Tableau 9.

Tableau 9 : Corpus en anglais annotées en entités nommées

Corpus	Année	Instances des noms de personnes
CoNLL	2003	6600
WikiNER	2013	179199
Few-NERD	2021	75997

Nous choisissons de sélectionner des exemples provenant du corpus Few-NERD vu que ce dernier est le plus récent des trois corpus analysés. Étant donné que la classe la plus fréquente dans notre corpus Voxpopuli-NE contient environ 2000 occurrences, nous ajoutons seulement, à partir de Few-NERD, environ 1700 exemples de la classe PERSON aux 272 exemples d'apprentissage de Voxpopuli-NE. Ensuite nous entraînons notre modèle BERT-FC1 sur le nouveau corpus, que nous appelons « Voxpopuli-NE-Per ». Nous appliquons enfin le nouveau système, toujours sur la partie Test de notre corpus et analysons l'évolution de la F-mesure sur la classe PERSON à travers la Figure 15.

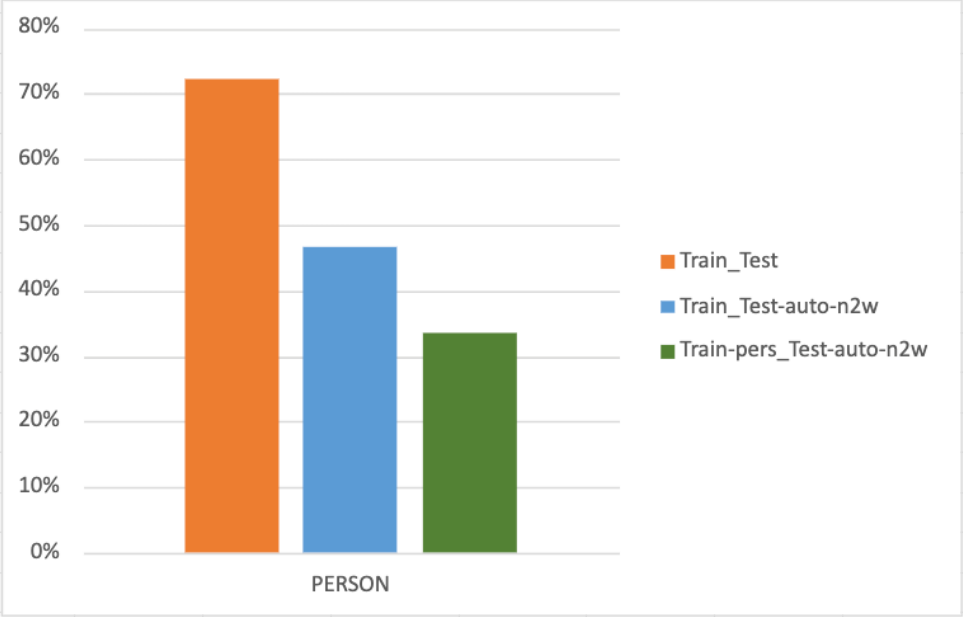


Figure 15 : F-mesures de BERT-FC1 sur la classe PERSON
(la légende est sous le format : corpus-d'apprentissage_corpus-de-test)

Nous constatons à travers cet histogramme que la performance du système BERT-FC1 sur la classe PERSON ne s'améliore pas voire se dégrade. Nous concluons donc que l'enrichissement des occurrences de la classe PERSON n'aide pas la NER à mieux reconnaître les noms de personnes. Ce n'est donc probablement pas la faute de la NER mais plutôt celle de la reconnaissance automatique de la parole.

Malheureusement, c'est très compliqué d'estimer les erreurs de l'ASR sur une catégorie particulière de mots. Nous remarquons tout de même de nombreuses erreurs de NER qui sont causés par des erreurs d'ASR comme nous pouvons voir sur l'exemple ci-dessous.

Référence :	*the*	*current*	*commissioner*	*neelie*	*kroes*
	O	*O*	*O*	*B-PERS*	*I-PERS*
Prédiction :	*the*	*current*	*commissioner*	*nearly*	*cruz*
	O	*O*	*O*	*O*	*O*

La reconnaissance de noms de personne étant un défi toujours d'actualité pour les systèmes de reconnaissance automatique de la parole [EGOROVA et al. 2018], l'ASR de Speechmatics semble commettre des erreurs fréquentes sur cette catégorie de mots. Ce problème impacte visiblement la performance de la NER sur cette catégorie. L'amélioration de l'ASR, au moins dans cet axe, est donc nécessaire pour améliorer la performance globale de la reconnaissance d'entités nommées.

4. Conclusion

Ce travail représente une des rares tentatives d'étendre la tâche de reconnaissance d'entités nommées afin de prendre en entrée, non pas des données textuelles écrites, mais plutôt de la parole. Nous avons conçu un système en cascade composé d'une ASR propriétaire et une NER développée en se basant sur les dernières avancées scientifiques dans l'apprentissage profond, à savoir la modélisation de la langue. En analysant la performance de ce système, nous avons remarqué une baisse causée par des erreurs de l'ASR, surtout dans certaines catégories où l'ASR est le moins efficace, à savoir, les noms de personnes.

Plusieurs pistes d'amélioration sont en cours d'étude. Nous comptons en premier temps nous concentrer sur les classes les plus intéressantes pour nous. Par exemple, les entités d'emplacement sont beaucoup plus importantes que les entités légales dans notre cas d'utilisation. Par ailleurs, vu la rareté des données vocales annotées en entités nommées, nous essaierons des techniques d'auto-apprentissage (self-training). Le principe consiste à prédire les entités nommées à partir d'enregistrements vocaux non annotées, en utilisant un premier système de NER à partir de la parole déjà entraîné. Nous pourrons ensuite utiliser ses prédictions comme données de référence pour enrichir notre corpus d'apprentissage.

Si cette piste vise à améliorer la performance la composante NER, elle reste toujours incapable de surmonter les difficultés dues aux erreurs de la reconnaissance automatique de la parole. Par conséquent, nous comptons par la suite, concevoir notre propre système d'ASR, entraîné sur des données adaptées à notre contexte, pour minimiser l'impact de cette étape sur la NER. Enfin, nous nous sommes intéressés dans ce travail exclusivement à l'approche en cascade. Il serait judicieux, au moins dans les mois ou les années à venir, et en fonction des progrès dans la littérature, d'étudier l'intérêt de l'approche de bout en bout.

5. Références

[AMODEI et al. 2016] AMODEI, Dario, ANANTHANARAYANAN, Sundaram, ANUBHAI, Rishita, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In: International conference on machine learning. PMLR, 2016. p. 173-182.

[AONE et al. 1998] AONE, Chinatsu, HALVERSON, Lauren, HAMPTON, Tom, et al. SRA: Description of the IE2 system used for MUC-7. In : Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998. 1998.

[BARIL et al. 2022] BARIL, Guillaume, CARDINAL, Patrick, et KOERICH, Alessandro Lameiras. Named Entity Recognition for Audio De-Identification. arXiv preprint arXiv:2204.12622, 2022.

[BIKEL et al. 1998] BIKEL, Daniel M., MILLER, Scott, SCHWARTZ, Richard, et al. Nymble: a high-performance learning name-finder. arXiv preprint cmp-lg/9803003, 1998.

[BIKEL et al. 1999] BIKEL, Daniel M., SCHWARTZ, Richard, et WEISCHEDEL, Ralph M. An algorithm that learns what's in a name. Machine learning, 1999, vol. 34, no 1, p. 211-231.

[BLACK et al. 1998] BLACK, William J., RINALDI, Fabio, et MOWATT, David. FACILE: Description of the NE system used for MUC-7. In : Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998. 1998.

[BRIDLE 1989] BRIDLE, John. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. Advances in neural information processing systems, 1989, vol. 2.

[BROWN et al. 2020] BROWN, Tom, MANN, Benjamin, RYDER, Nick, et al. Language models are few-shot learners. Advances in neural information processing systems, 2020, vol. 33, p. 1877-1901.

[CHEN et al. 2022] CHEN, Boli, XU, Guangwei, WANG, Xiaobin, et al. AISHELL-NER: Named Entity Recognition from Chinese Speech. In : ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022. p. 8352-8356.

[CHIU et al. 2016] CHIU, Jason PC et NICHOLS, Eric. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the association for computational linguistics, 2016, vol. 4, p. 357-370.

[CHO 2014] CHO, Kyunghyun, VAN MERRIËNBOER, Bart, BAHDANAU, Dzmitry, et al. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.

[COHN et al. 2019] COHN, Ido, LAISH, Itay, BERYOZKIN, Genady, et al. Audio de-identification: A new entity recognition task. arXiv preprint arXiv:1903.07037, 2019.

[CUI 2019] CUI, Leyang et ZHANG, Yue. Hierarchically-refined label attention network for sequence labeling. arXiv preprint arXiv:1908.08676, 2019.

[Dennis et Moré 1977] J. E. Dennis, Jr & J. J. Moré, 1977. Quasi-Newton methods, motivation and theory. SIAM review 19(1), 46–89.

[DEVLIN et al. 2018] DEVLIN, Jacob, CHANG, Ming-Wei, LEE, Kenton, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[Eddy 1996] S. R. Eddy, 1996. Hidden markov models. Current opinion in structural biology 6(3), 361–365.

[ELMAN 1990] ELMAN, Jeffrey L. Finding structure in time. Cognitive science, 1990, vol. 14, no 2, p. 179-211.

[EGOROVA et al. 2018] EGOROVA, Ekaterina et BURGET, Lukáš. Out-of-vocabulary word recovery using fst-based subword unit clustering in a hybrid asr system. In : 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018. p. 5919-5923.

[GHANNAY et al. 2018] GHANNAY, Sahar, CAUBRIÈRE, Antoine, ESTÈVE, Yannick, et al. End-to-end named entity and semantic concept extraction from speech. In : 2018 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2018. p. 692-699.

[Graves et Schmidhuber, 2005] A. Graves & J. Schmidhuber, 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 18(5), 602–610.

[HANISCH et al. 2005] HANISCH, Daniel, FUNDEL, Katrin, MEVISSEN, Heinz-Theodor, et al. ProMiner: rule-based protein and gene entity recognition. BMC bioinformatics, 2005, vol. 6, no 1, p. 1-9.

[HOFFART et al. 2011] HOFFART, Johannes, YOSEF, Mohamed Amir, BORDINO, Ilaria, et al. Robust disambiguation of named entities in text. In : Proceedings of the 2011 conference on empirical methods in natural language processing. 2011. p. 782-792.

[HUANG et al. 2015] HUANG, Zhiheng, XU, Wei, et YU, Kai. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991, 2015.

[HUMPHREYS et al. 1998] HUMPHREYS, Kevin, GAIZAUSKAS, Robert, AZZAM, Saliha, et al. University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In : Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998. 1998.

[JI et al. 2016] JI, Zongcheng, SUN, Aixin, CONG, Gao, et al. Joint recognition and linking of fine-grained locations from tweets. In : Proceedings of the 25th international conference on world wide web. 2016. p. 1271-1281.

[KIM et al. 2000] kim, Ji-Hwan et WOODLAND, Philip C. A rule-based named entity recognition system for speech input. In : Sixth International Conference on Spoken Language Processing. 2000.

[KRISHNAN et al. 2006] KRISHNAN, Vijay et MANNING, Christopher D. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In : Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics. 2006. p. 1121-1128.

[LECUN 1998] LECUN, Yann, BOTTOU, Léon, BENGIO, Yoshua, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, vol. 86, no 11, p. 2278-2324.

[LIAO et al. 2009] LIAO, Wenhui et VEERAMACHANENI, Sriharsha. A simple semi-supervised algorithm for named entity recognition. In : Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing. 2009. p. 58-65.

[LIN et al. 2017] LIN, Bill Yuchen, XU, Frank F., LUO, Zhiyi, et al. Multi-channel bilstm-crf model for emerging named entity recognition in social media. In : Proceedings of the 3rd Workshop on Noisy User-generated Text. 2017. p. 160-165.

[LIU et al. 2020] LIU, Shifeng, SUN, Yifang, LI, Bing, et al. HAMNER: Headword amplified multi-span distantly supervised method for domain specific named entity recognition. In : Proceedings of the AAAI Conference on Artificial Intelligence. 2020. p. 8401-8408.

[LIU et al. 20011] LIU, Xiaohua, ZHANG, Shaodian, WEI, Furu, et al. Recognizing named entities in tweets. In : Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. 2011. p. 359-367.

[MA et al. 2016] MA, Xuezhe et HOVY, Eduard. End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354, 2016.,

[MCNAMEE et al. 2002] MCNAMEE, Paul et MAYFIELD, James. Entity extraction without language-specific resources. In: COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002). 2002.

[MCNAMEE et al. 2003] MCCALLUM, Andrew et LI, Wei. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. 2003.

[MIKOLOV et al. 2013a] MIKOLOV, Tomas, CHEN, Kai, CORRADO, Greg, et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[MIKOLOV et al. 2013b] MIKOLOV, Tomas, SUTSKEVER, Ilya, CHEN, Kai, et al. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 2013, vol. 26.

[MIKHEEV et al. 1999] MIKHEEV, Andrei. A knowledge-free method for capitalized word disambiguation. In : Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. 1999. p. 159-166.

[NADEAU et al. 2007] NADEAU, David et SEKINE, Satoshi. A survey of named entity recognition and classification. Lingvisticae Investigationes, 2007, vol. 30, no 1, p. 3-26..

[NADEAU et al. 2006] NADEAU, David, TURNEY, Peter D., et MATWIN, Stan. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In : Conference of the Canadian society for computational studies of intelligence. Springer, Berlin, Heidelberg, 2006. p. 266-277.

[NIGMATULINA et al.2022] NIGMATULINA, Iuliia, ZULUAGA-GOMEZ, Juan, PRASAD, Amrutha, et al. A two-step approach to leverage contextual data: speech recognition in air-traffic communications. In :

ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022. p. 6282-6286.

[NGUYEN et al. 2016] NGUYEN, Thien Huu, SIL, Avirup, DINU, Georgiana, et al. Toward mention detection robustness with recurrent neural networks. arXiv preprint arXiv:1602.07749, 2016.

[PASAD et al. 2021] PASAD, Ankita, WU, Felix, SHON, Suwon, et al. On the use of external data for spoken named entity recognition. arXiv preprint arXiv:2112.07648, 2021.

[PENNINGTON et al. 2014] PENNINGTON, Jeffrey, SOCHER, Richard, et MANNING, Christopher D. Glove: Global vectors for word representation. In : Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. p. 1532-1543.

[PETERS et al. 2017] PETERS, Matthew E., AMMAR, Waleed, BHAGAVATULA, Chandra, et al. Semi-supervised sequence tagging with bidirectional language models. arXiv preprint arXiv:1705.00108, 2017.

[PETERS et al. 2018] PETERS, M. E., NEUMANN, M., IYYER, M., et al. Deep contextualized word representations. NAACL HLT 2018-2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies-Proceedings of the Conference. 2018.

[QUINLAN 1986] J. R. Quinlan, 1986. Induction of decision trees. Machine learning 1(1), 81–106.

[QUIMBAYA et al. 2016] QUIMBAYA, Alexandra Pomares, MÚNERA, Alejandro Sierra, RIVERA, Rafael Andrés González, et al. Named entity recognition over electronic health records through a combined dictionary-based approach. Procedia Computer Science, 2016, vol. 100, p. 55-61.

[RAVIN et al.1997] RAVIN, Yael et WACHOLDER, Nina. Extracting names from natural-language text. IBM Thomas J. Watson Research Division, 1997.

[REI et al. 2016] REI, Marek, CRICHTON, Gamal KO, et PYYSALO, Sampo. Attending to characters in neural sequence labeling models. arXiv preprint arXiv:1611.04361, 2016.

[REI 2017] REI, Marek. Semi-supervised multitask learning for sequence labeling. arXiv preprint arXiv:1704.07156, 2017.

[RITTER el al. 2013] RITTER, Ole et ENEVOLD, Flemming. Ritter. Art People, 2013.

[ROCKTÄSCHEL et al. 2012] ROCKTÄSCHEL, Tim, WEIDLICH, Michael, et LESER, Ulf. ChemSpot: a hybrid system for chemical named entity recognition. Bioinformatics, 2012, vol. 28, no 12, p. 1633-1640.

[SETTLES. 2004] Burr. Biomedical named entity recognition using conditional random fields and rich feature sets. In : Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (NLPBA/BioNLP). 2004. p. 107-110.

[SHON et al. 2022] SHON, Suwon, PASAD, Ankita, WU, Felix, et al. Slue: New benchmark tasks for spoken language understanding evaluation on natural speech. In : ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022. p. 7927-7931.

[Schuster et Paliwal 1997] M. Schuster & K. K. Paliwal, 1997. Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on 45(11), 2673–2681.

[SZARVAS et al. 2006] SZARVAS, György, FARKAS, Richárd, et KOCSOR, András. A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms. In : International Conference on Discovery Science. Springer, Berlin, Heidelberg, 2006. p. 267-278.

[TORISAWA et al. 2007] TORISAWA, Kentaro, et al. Exploiting Wikipedia as external knowledge for named entity recognition. In : Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL). 2007. p. 698-707.

[TORAL et al. 2006] TORAL, Antonio et MUNOZ, Rafael. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. In : Proceedings of the Workshop on NEW TEXT Wikis and blogs and other dynamic text sources. 2006.

[Vapnik 1999] V. Vapnik, 1999. The Nature of Statistical Learning Theory. Springer Science & Business Media.

[VASWANI et al. 2017] VASWANI, Ashish, SHAZEER, Noam, PARMAR, Niki, et al. Attention is all you need. Advances in neural information processing systems, 2017, vol. 30.

[WANG et al. 2021] WANG, Changhan, RIVIERE, Morgane, LEE, Ann, et al. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. arXiv preprint arXiv:2101.00390, 2021.

[WU et al. 2020] WU, Felix, KIM, Kwangyoun, WATANABE, Shinji, et al. Wav2Seq: Pre-training Speech-to-Text Encoder-Decoder Models Using Pseudo Languages. arXiv preprint arXiv:2205.01086, 2022.

[YADAV et al. 2020] YADAV, Hemant, GHOSH, Sreyan, YU, Yi, et al. End-to-end named entity recognition from english speech. arXiv preprint arXiv:2005.11184, 2020.

[YANG et al. 2016] YANG, Zhilin, SALAKHUTDINOV, Ruslan, et COHEN, William. Multi-task cross-lingual sequence tagging from scratch. arXiv preprint arXiv:1603.06270, 2016.

[ZHANG et al. 2013] S. Zhang and N. Elhadad, “Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts,” J. Biomed. Inform., vol. 46, no. 6, pp. 1088–1098, 2013.

[ZHENG et al. 2017] ZHENG, Suncong, WANG, Feng, BAO, Hongyun, et al. Joint extraction of entities and relations based on a novel tagging scheme. arXiv preprint arXiv:1706.05075, 2017

[ZHOU et al. 2017] ZHOU, Peng, ZHENG, Suncong, XU, Jiaming, et al. Joint extraction of multiple relations and entities by using a hybrid neural network. In : Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer, Cham, 2017. p. 135-146.

[ZHU et al. 2005] ZHU, Jianhan, UREN, Victoria, et MOTTA, Enrico. ESpotter: Adaptive named entity recognition for web browsing. In : Biennial Conference on Professional Knowledge Management/Wissensmanagement. Springer, Berlin, Heidelberg, 2005. p. 518-529.

[ZHZHOU et al. 2002] ZHZHOU, GuoDong et SU, Jian. Named entity recognition using an HMM-based chunk tagger. In : Proceedings of the 40th annual meeting of the association for computational linguistics. 2002. p. 473-480.