Actes de JEP-TALN-RECITAL 2024
Toulouse, France, 8 au 12 juillet 2024

Une approche par graphe pour l'analyse syntaxique en dépendances de bout en bout de la parole

Adrien Pupier, Maximin Coavoux, Benjamin Lecouteux, Jérôme Goulian
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
first.last@univ-grenoble-alpes.fr

RÉSUMÉ

Effectuer l'analyse syntaxique du signal audio –plutôt que de passer par des transcriptions de l'audio– est une tache récemment proposée par Pupier *et al.* (2022), dans le but d'incorporer de l'information prosodique dans le modèle d'analyse syntaxique et de passer outre les limitations d'une approche cascade qui consisterait à utiliser un système de reconnaissance de la parole (RAP) puis un analyseur syntaxique. Dans cet article, nous effectuons un ensemble d'expériences visant à comparer les performances de deux familles d'analyseurs syntaxiques : (i) l'approche par graphe et (ii) la réduction à une tâche d'étiquetage de séquence ; directement sur la parole. Nous évaluons notre approche sur un corpus arboré du français parlé. Nous montrons que (i) l'approche par graphe obtient de meilleurs résultats globalement et (ii) qu'effectuer l'analyse syntaxique directement depuis la parole obtient de meilleurs résultats qu'une approche par cascade de systèmes, malgré 30% de paramètres en moins.

ABSTRACT

A graph-based parser for end-to-end dependency parsing of speech

Direct dependency parsing of the speech signal –as opposed to parsing speech transcriptions– has recently been proposed as a task (Pupier *et al.*, 2022), as a way of incorporating prosodic information into the parsing system and bypassing the limitations of a pipeline approach that would consist of using first an Automatic Speech Recognition (ASR) system and then a syntactic parser. In this article, we report on a set of experiments aiming at assessing the performance of two parsing paradigms (graph-based parsing and sequence labeling based parsing) on speech parsing. We perform this evaluation on a large treebank of spoken French, featuring realistic spontaneous conversations. Our findings show that (i) the graph based approach obtain better results across the board, (ii) parsing directly from speech outperforms a pipeline approach, despite having 30% fewer parameters.

MOTS-CLÉS : Analyse syntaxique, parole, modèle pré-entraîné, bout-en-bout.

KEYWORDS: Syntactic parsing, Speech, Pre-trained model, end-to-end.

1 Introduction

L'analyse syntaxique est une tâche centrale en traitement automatique des langues (TAL). Au sein de la communauté du TAL, celle-ci a été plutôt effectuée sur des données textuelles, ou plus rarement sur des transcriptions de la parole. Pourtant, la parole est la principale forme de communication entre humain·es, et constitue le type de données linguistiques le plus naturel, ce qui motive le développement de systèmes de TAL pour la parole, à la fois pour des objectifs applicatifs (par exemple : le sous-titrage

31ème Conférence sur le Traitement Automatique des Langues Naturelles,
volume 1 : articles longs et prises de position, pages 234–244.

automatique) que pour des objectifs de recherche (par exemple : construire des corpus de grande taille annotés en syntaxe). La majorité des travaux existant sur l'analyse syntaxique de transcriptions de la parole se concentrent sur l'anglais et sur la détection et la suppression de disfluences (Charniak & Johnson, 2001; Johnson & Charniak, 2004; Rasooli & Tetreault, 2013; Honnibal & Johnson, 2014; Jamshid Lou *et al.*, 2019). Autrement dit, il s'agit de 'normaliser' les transcriptions de manière à pouvoir les utiliser en aval avec des systèmes de TAL entraînés sur du texte 'standard'. Utiliser uniquement des transcriptions en entrée est un choix naturel d'un point de vue du TAL : il est possible d'utiliser des analyseurs syntaxiques état de l'art sans modifications. Cependant, les transcriptions prédites peuvent être très bruitées, particulièrement dans le cas des conversations spontanées. De plus, les transcriptions sont des abstractions contenant beaucoup moins d'informations que le signal audio. La prosodie, ainsi que les pauses dans un énoncé, sont des indices importants pour l'analyse syntaxique (Price *et al.*, 1991) qui sont complètement absentes des transcriptions. Ainsi, nous effectuons cette tâche en utilisant seulement le signal audio en entrée. En particulier nous proposons un nouvel algorithme d'analyse syntaxique en dépendances pour la parole utilisant l'approche par graphe (*graph-based*).

Avec la popularisation des méthodes auto-supervisées et les réseaux neuronaux profonds, les domaines de la parole et du texte utilisent désormais une méthodologie similaire (Chrupała, 2023) : affiner les paramètres d'un modèle auto-supervisé générique sur une tâche applicative particulière. Cette convergence méthodologique a suscité l'intérêt d'autres applications des modèles de l'audio pour aller au-delà de la 'simple' reconnaissance de la parole. Ainsi, aborder les tâches classiques du TAL directement depuis l'audio est un enjeu important pour concevoir des outils de TAL robustes au bruit présent dans la parole. De plus, de nombreuses langues n'ont pas de forme écrite, ce qui proscrit d'utiliser des outils classiques du TAL et motive le développement d'outils de documentation travaillant directement sur le signal audio.

En résumé, nos contributions sont les suivantes :

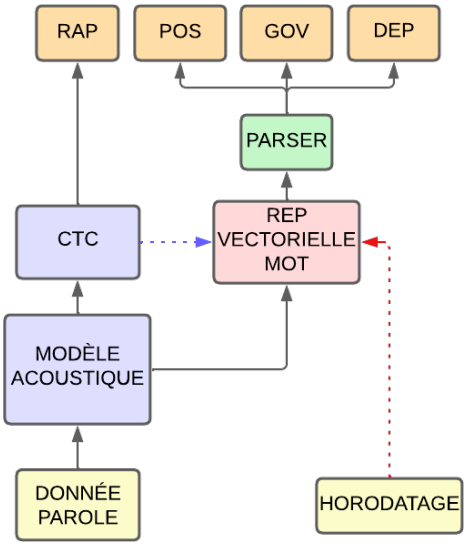
— Nous présentons un algorithme bout-en-bout d'analyse syntaxique basé sur les graphes ;
— Nous évaluons notre analyseur sur Orféo (Benzitoun *et al.*, 2016), un corpus arboré du français parlé contenant essentiellement de la parole spontanée ; nous comparons le parser à une approche cascade ainsi qu'à un analyseur basé sur une réduction du problème à l'étiquetage de séquence (Strzyz *et al.*, 2019) ;
— Nous publions le code à l'adresse suivante : `https://github.com/Pupiera/Growing_tree_on_sound`.

2 Analyseur syntaxique et modèle pré-entraîné

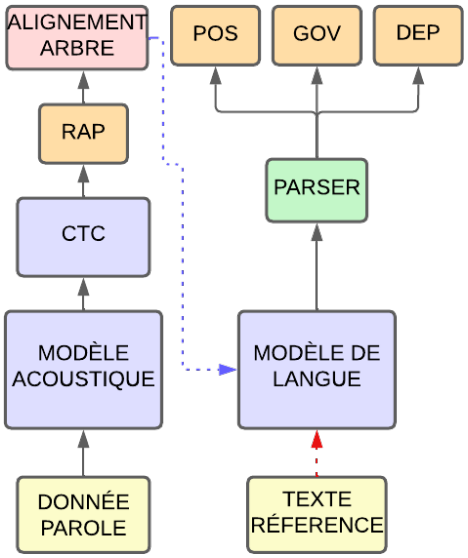
Nous définissons l'analyse syntaxique de la parole comme la tâche consistant à prédire un arbre de dépendance depuis un signal audio correspondant à un énoncé [1]. Les nœuds de l'arbre syntaxique étant constitués des tokens, nous réalisons la reconnaissance automatique de la parole de manière conjointe à l'analyse syntaxique.

Notre analyseur est composé de deux modules (figure 1a) : (i) un modèle acoustique qui prédit les transcriptions et la segmentation du signal en mots et (ii) un module d'analyse syntaxique qui utilise cette segmentation pour construire des représentations vectorielles de mots et prédit les arbres. Ces

1. Dans le reste de l'article, nous utiliserons le terme *phrase* pour garder la terminologie issue de la littérature en analyse syntaxique automatique, même si ce terme est sujet à débat lorsqu'il s'agit de langue parlée.



(a) Les deux modèles basés sur des représentations de l'audio. La flèche bleue désigne la configuration **AUDIO**, et la flèche rouge **ORACLE** (cf section 4).



(b) Les deux modèles de référence basés sur un modèle de langue pré-entraîné. La flèche bleue désigne la configuration **CASCADE**, et la flèche rouge **TEXTE**, (transcriptions manuelles de référence).

FIGURE 1 – Vue générale de l'architecture avec les 4 configurations décrites dans la section 4.

deux modules sont entraînés de manière simultanée.

Construire des vecteurs de mots directement à partir du signal de parole Pour extraire des représentations depuis le signal de parole, nous utilisons un modèle wav2vec2 (Baevski *et al.*, 2020) pré-entraîné sur 7000 heures de parole en français : `LeBenchmark7K`[2] (Parcollet *et al.*, 2024). Ce modèle wav2vec2 construit une suite de représentations vectorielles, chaque vecteur correspondant à un intervalle de 25ms dans le signal de parole. On appelle cette suite de vecteurs la trame du signal.

L'analyse syntaxique nécessitant des représentations vectorielles mot. Nous utilisons la méthodologie de Pupier *et al.* (2022) pour construire des représentations de mots à partir de ces représentations du signal. Nous utilisons l'algorithme de reconnaissance de la parole CTC (*connectionist temporal classification,* Graves *et al.*, 2006), qui a la particularité d'étiqueter chaque vecteur de la trame avec soit une lettre soit un caractère de séparation ou un caractère spécial 'vide' qui permet d'obtenir plusieurs fois la même lettre à la suite. Il est ainsi possible, en identifiant le caractère spécial de séparation (espace) de segmenter la trame du signal audio en mots. Ensuite, la méthode consiste à combiner les vecteurs de trames correspondant à un seul mot à l'aide d'un LSTM, afin d'obtenir une représentation vectorielle de taille fixe pour chaque mot. Intuitivement, cette procédure peut être considérée comme l'analogue des bi-LSTM de caractères classiquement utilisés pour le texte, mais où l'on utiliserait la forme acoustique d'un mot au lieu de sa forme orthographique.

Analyse syntaxique par graphe Nous utilisons les représentations de mot audio –dont la construction est décrite ci-dessus– comme entrée de notre implémentation d'un analyseur syntaxique par

2. https://huggingface.co/LeBenchmark/wav2vec2-FR-7K-large

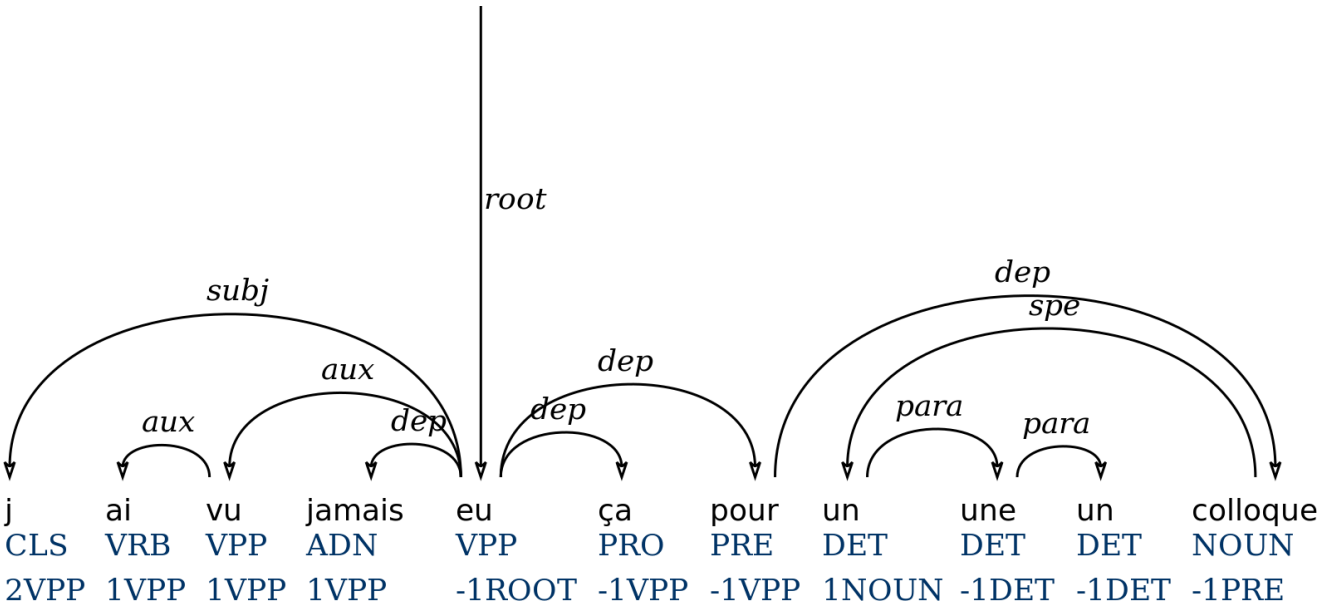


FIGURE 2 – Illustration d'une phrase extraite d'Orféo étiquetée en dep2label ainsi que l'arbre syntaxique en dépendance correspondant.

graphes classique (Dozat & Manning, 2016). Cet analyseur consiste à (i) calculer un score pour chaque arc possible à l'aide d'un classifieur biaffine puis (ii) trouver l'arbre de meilleur score (c'est-à-dire l'arbre maximisant les probabilités globales) en utilisant un algorithme d'arbre couvrant maximal.

Analyse syntaxique par étiquetage de séquence Nous utilisons le même analyseur syntaxique par étiquetage de séquence que Pupier *et al.* (2022). Cet analyseur est basé sur l'approche *dep2label* (Gómez-Rodríguez *et al.*, 2020; Strzyz *et al.*, 2020). Cette méthode réduit le problème de l'analyse syntaxique à un simple problème d'étiquetage de séquence. Elle nécessite un algorithme réversible pour coder un arbre en dépendances dans une séquence d'étiquettes. Nous utilisons l'algorithme d'encodage par position relative en prenant en compte les parties du discours (appelé *Relative POS-based* par Strzyz *et al.*, 2019). Le gouverneur de chaque mot est codé dans une étiquette de la forme ±Entier@POS. L'entier correspond à la distance relative du gouverneur en prenant uniquement compte les mots ayant la même partie de discours. Par exemple, `-3@NOM` signifie que le gouverneur du mot courant est le troisième nom avant le mot courant. Un exemple de cet étiquetage est présenté dans la figure 2. On note que cette méthode ne garantit pas que l'arbre prédit est un arbre en dépendances bien formé.

3 Jeu de données

Nous utilisons le corpus arboré CEFC-Orféo, un corpus du français annoté en dépendances et composé de plusieurs sous-corpus : CFPP (CLESTHIA, 2018), Clapi (ICAR, 2017) TCOF (ATILF, 2020), OFROM (Mathieu *et al.*, 2012 2020), Fleuron (André, 2016), French Oral Narrative (FON Carruthers, 2013), C-ORAL-ROM (Cresti *et al.*, 2004), Corpus de référence du français parlé (DELIC *et al.*, 2004), Valibel (Francard *et al.*, 2009), TUFS (Kawaguchi *et al.*, 2006), Corpus de réunions de travail (Husianycia, 2011), et mis à disposition avec les enregistrements audios. Ce corpus contient des types

Modèle	WER↓	CER↓	POS↑	UAS↑	LAS↑	Nombre de paramètres	Modèles préentraînés
AUDIO SEQ2LABEL	35.9	22.3	73.0	65.7	60.4	315M + 34.9M	Wav2vec2
AUDIO GRAPHE	**35.6**	**22.1**	**73.1**	**66.0**	**60.9**	315M + 34.9M	Wav2vec2
ORACLE SEQ2LABEL	36.3	22.2	75.6	68.7	62.7	315M + 34.9M	Wav2vec2
ORACLE GRAPHE	**35.6**	**22.2**	**77.4**	**73.3**	**67.5**	315M + 34.9M	Wav2vec2
CASCADE SEQ2LABEL	35.6	22.0	70.8	63.8	58.4	314M + 110M + 39.2M	Wav2vec2 + CamemBERT
CASCADE GRAPHE	35.6	22.0	69.3	60.5	53.1	314M + 110M + 41.4M	Wav2vec2 + CamemBERT
CASCADE HOPS	35.6	22.0	**72.4**	**65.8**	**61.0**	314M + 110M + 100M	Wav2vec2 + CamemBERT
TEXTE SEQ2LABEL	–	–	96.9	88.8	85.7	110M + 39.2M	CamemBERT
TEXTE GRAPHE	–	–	95.1	87.4	84.0	110M + 41.4M	CamemBERT
TEXTE HOPS	–	–	**98.2**	**90.3**	**87.7**	110M + 100M	CamemBERT

TABLE 1 – Évaluation sur le corpus de test d'Orféo avec les configurations décrites dans la section 4.

Modèle	WER↓	CER↓	POS↑	UAS↑	LAS↑	Nombre de paramètres	Modèles préentraînés
AUDIO SEQ2LABEL	31.0	18.4	**77.1**	70.2	65.2	315M + 34.9M	Wav2vec2
AUDIO GRAPHE	**30.6**	**18.2**	77.0	**70.9**	**66.2**	315M + 34.9M	Wav2vec2
ORACLE SEQ2LABEL	**30.9**	**18.6**	78.3	71.9	66.2	315M + 34.9M	Wav2vec2
ORACLE GRAPHE	31.4	19.2	**79.8**	**76.0**	**70.4**	315M + 34.9M	Wav2vec2
CASCADE SEQ2LABEL	30.5	18.2	74.7	67.7	62.4	314M + 110M + 39.2M	Wav2vec2 + CamemBERT
CASCADE GRAPHE	30.5	18.2	73.5	64.2	57.3	314M + 110M + 41.4M	Wav2vec2 + CamemBERT
CASCADE HOPS	30.5	18.2	**76.3**	**69.4**	**64.6**	314M + 110M + 100M	Wav2vec2 + CamemBERT
TEXTE SEQ2LABEL	–	–	94.5	86.7	83.1	110M + 39.2M	CamemBERT
TEXTE GRAPHE	–	–	96.8	88.3	84.5	110M + 41.4M	CamemBERT
TEXTE HOPS	–	–	**98.2**	**90.3**	**87.1**	110M + 100M	CamemBERT

TABLE 2 – Évaluation sur le corpus Valibel (un sous-corpus du jeu de test).

d'interactions variés, ayant toutes des spécificités de la parole spontanée, à l'exception de French Oral Narrative qui contient de la parole lue. Les situations de paroles sont également variées (ex : interactions commerciales, discussions informelles entre amis). Il s'agit du corpus de parole français le plus volumineux annoté en dépendances (196h, 2.5M tokens). Le schéma d'annotation (Benzitoun *et al.*, 2016) contient 20 parties du discours et 14 étiquettes de relations syntaxiques.

Les annotations syntaxiques d'Orféo ont été effectuées manuellement (données *gold* ou de référence) pour environ 5% du corpus, et automatiquement pour le reste (données *silver*). Les jeux d'entraînement/validation/test que nous utilisons dans cet article font en sorte que le jeu de validation ainsi que celui de test ne contiennent que des données annotées manuellement. Cependant, les sous-corpus avec les annotations de référence correspondent aussi à des enregistrements audios de qualité faible, ce qui rend la tâche difficile.

4 Expérimentations

Configurations expérimentales Nos expériences ont pour but de répondre aux questions suivantes :

— Est-ce qu'une approche bout-en-bout est préférable à une approche par cascade de systèmes ?
— Quel algorithme d'analyse (par graphe ou par étiquetage de séquence) est le plus approprié pour l'analyse syntaxique de la parole ?

	WER↓	CER↓	POS↑	UAS↑	LAS↑	Nombre de paramètres
Graph-tiny	35.7	22.3	73.0	65.9	60.8	314M + 11.7M
Graph-base	35.6	22.1	73.1	66.1	60.9	314M + 34.9M
Graph-large	35.6	22.0	73.2	66.0	60.7	314M + 67.6M

TABLE 3 – Comparaison des métriques d'analyse syntaxique avec l'architecture basée sur les graphes et différents nombres de paramètres.

— Est-ce qu'une approche est plus ou moins robuste au bruit inhérent du signal de parole ? Est-ce que les frontières de mot prédites sont un facteur important pour les différents systèmes ? Est-ce que les deux approches sont aussi robustes aux erreurs du module de RAP ?

Pour répondre à ces questions, nous comparons les configurations expérimentales suivantes, également illustrées par la figure 1 :

— **AUDIO** : Utilise uniquement **l'audio** comme entrée, le modèle crée des représentations vectorielles de mot depuis les représentations acoustiques comme décrit dans la section 2.
— **ORACLE** : Utilise **l'audio** ainsi que les **horodatages des mots** disponibles dans le corpus [3], ce qui rend la construction de vecteurs de mot plus facile et réduit l'impact de la qualité de la reconnaissance de la parole sur l'analyse syntaxique.
— **CASCADE** : Utilise seulement les **transcriptions prédites** par le modèle acoustique. Ensuite, un modèle de langue pré-entraîné calcule des vecteurs de mots contextualisés qui sont ensuite utilisés par le module d'analyse syntaxique. Dans cette configuration, nous modifions les arbres d'entraînement pour prendre en compte les suppressions ou insertions de mot. Cependant, comme pour l'approche basée sur l'audio, les insertions ou suppressions impactent le score global, car le score final est évalué sur les transcriptions manuelles et non sur une version modifiée. L'inconvénient de cette approche est que les informations prosodiques ne sont pas accessibles.
— **TEXTE** : Utilise les **transcriptions manuelles de référence** : cette configuration artificielle permet d'obtenir une borne supérieure pour les performances du modèle dans un cas idéal (RAP parfaite).

Les configurations **AUDIO** et **ORACLE** sont des modèles conjoints multitâches pour la RAP et l'analyse syntaxique. Comme les deux modules (RAP et analyse syntaxique) sont entraînés simultanément dans ces configurations, les résultats en RAP ne sont pas identiques d'une expérience à l'autre.

Les configurations **CASCADE** et **TEXTE** utilisent un modèle BERT du français : `camembert-base` [4] (Martin *et al.*, 2020) pour extraire des représentations vectorielles contextualisées. En plus de nos implémentations, nous utilisons également `hops` (Grobol & Crabbé, 2021), un analyseur syntaxique état-de-l'art. L'analyseur `hops` utilise également un bi-LSTM de caractères en plus de BERT pour produire des représentations vectorielles des mots, contrairement à nos implémentations qui sont conçues pour varier minimalement d'une configuration à l'autre.

Chaque méthode d'analyse pour chaque modalité est entraînée avec le même nombre d'époques, les même hyperparamètres (voir tables 4 et 5) et approximativement le même nombre de paramètres. Nos implémentations utilisent la bibliothèque `speechbrain` (Ravanelli *et al.*, 2021). La liste complète

3. Le corpus contient des horodatages construits automatiquement via un alignement forcé, c'est-à-dire que l'on dispose pour chaque mot des instants correspondant à son début et à sa fin.
4. https://huggingface.co/almanach/camembert-base

Analyseur	SEQ2LABEL	GRAPHE
Époques	30	30
Taille de *batch*	8	8
Paramètres d'optimisation		
Pas d'apprentissage	0.0001	0.0001
Algorithme d'optimisation	AdaDelta	AdaDelta
Model préentraîné	`LeBenchmark7K`	
Encodeur		
Nombre de couches	3	3
Dropout	0.15	0.15
Dimension de l'encodeur	1024	1024
Fonction d'activation	LeakyReLU	LeakyRelu
Fusion LSTM		
Nombre de couches	2	2
Dimension	500	500
Bidirectionnel	non	non
Biais	oui	oui
LSTM parser		
Nombre de couches	2	3
Dimensions	800	768
Bidirectionnel	oui	oui
Étiqueteur (SEQ2LABEL)		
Dimension	1600	
Nombre de couches	1	
Nombre d'étiquettes dep2label	846	
Nombre d'étiquettes POS	23	
Nombre d'étiquettes syntaxiques	19	
MLP pour les arcs (GRAPHE)		
Dimension		768
Nombre de couches		1
Taille de la couche de sortie		768
MLP pour les étiquettes syntaxiques (GRAPHE)		
Dimension		768
Nombre de couches		1
Taille de la couche de sortie		768
MLP pour les POS (GRAPHE)		
Dimension		768
Taille de la couche de sortie		24

TABLE 4 – Hyperparamètres pour les configurations **AUDIO** et **ORACLE** SEQ2LABEL et GRAPHE.

Analyseur	SEQ2LABEL	GRAPHE	HOPS
Époques	40	40	40
Taille de *batch*	32	32	32
Paramètres d'optimisation			
Pas d'apprentissage	0.001	0.001	0.00003
Algorithme d'optimisation	Adam	Adam	Adam
Modèle préentraîné		`camembert_base`	
Plongements lexicaux	dernière couche	dernière couche	Moy. de 12 couches
Taille des plongements	768	768	768
Bi-LSTM de caractères HOPS			
Taille des plongements			128
Plongements lexicaux HOPS			
Taille des plongements			256
LSTM parser			
Dimension	768	768	512
Nombre de couches	3	2	3
Bidirectionnel	oui	oui	oui
Étiqueteur (SEQ2LABEL)			
Dimension	1536		
Nombre de couches	1		
Nombre d'étiquettes dep2label	846		
Nombre d'étiquettes POS	23		
Nombre d'étiquettes syntaxiques	19		
MLP pour les arcs (GRAPHE and HOPS)			
Dimension		768	1024
Nombre de couches		1	2
Taille de la couche de sortie		768	768
MLP pour les étiquettes syntaxiques (GRAPHE)			
Dimension		768	1024
Nombre de couches		1	2
Taille de la couche de sortie		768	768
POS MLP (GRAPHE)			
Dimension		768	1024
Taille de la couche de sortie		24	24

TABLE 5 – Hyperparamètres pour les configurations **CASCADE** et **TEXTE** SEQ2LABEL, GRAPHE et **CASCADE**.

des hyperparamètres que nous utilisons est présentée dans les tableaux 4 et 5.

Métriques d'évaluation Nous utilisons les métriques d'évaluation classiques : *taux d'erreur mot* (`WER`) et *taux d'erreur caractère* (`CER`) pour la RAP, *exactitude des parties de discours* (`POS`), *Score d'attachement non étiqueté* (`UAS`), et *Score d'attachement étiqueté* (`LAS`) pour l'analyse en dépendances.

Les résultats sont présentés dans le tableau 1 pour tout le corpus, et dans le tableau 2 pour un sous-corpus du jeu de test (Valibel) pour lequel la RAP est moins difficile.

Résultats : effet de la reconnaissance de la parole sur la qualité de l'analyse syntaxique Dans le tableau 1, nous observons que les approches par graphe ou seq2label donnent des résultats similaires quand aucune information additionnelle n'est fournie, ce qui montre que le facteur limitant du modèle est la reconnaissance de la parole plutôt que l'analyse syntaxique.

Il est important de noter qu'étant donnée la nature du corpus (conversation spontanée), le `WER` est plus haut que ce que l'on pourrait typiquement attendre en reconnaissance de la parole sur des corpus classiques de cette tâche (qui contiennent plutôt de la parole lue). Par exemple, le module RAP de notre modèle atteint environ 8 `WER` quand il est entraîné et évalué sur CommonVoice5.1 (Ardila *et al.*,

2020). D'autres indices de cela sont montrés dans le tableau 3 : changer le nombre de paramètres du modèle basé sur les graphes ne modifie pas significativement les performances. De plus, dans le tableau 2, nous observons une claire amélioration des résultats quand on évalue sur un corpus avec des meilleures performances au niveau de la RAP. La qualité du module de RAP affecte directement le nombre de mots reconnus (certains peuvent être supprimés ou ajoutés), ce qui affecte la qualité de l'analyse.

Résultats : différence entre étiquetage de séquence et approche par graphe Il est relativement surprenant que sur la modalité textuelle (**CASCADE**), l'approche par étiquetage obtient de meilleures performances que l'approche par graphe étant donné que ce n'est pas le cas sur d'autres modalités (**AUDIO**). Ainsi, l'approche GRAPHE semble être plus sensible au bruit provenant de l'approche par **CASCADE** que l'approche SEQ2LABEL. Cependant, elle ne parvient pas à surpasser un modèle graphe plus large et plus complexe utilisant un bi-LSTM de caractères comme `hops`. Ce bi-LSTM de caractères réduit probablement l'impact des mots hors vocabulaire produits à cause des erreurs orthographiques de la RAP.

Sur les configurations **AUDIO** et **ORACLE**, le modèle basé sur les graphes obtient de meilleures performances, mais l'écart entre la configuration **AUDIO** et **ORACLE** est plus important que pour le modèle SEQ2LABEL. En effet, cet écart sur le LAS est de 7 points pour le modèle GRAPHE alors qu'il est de seulement 2 points pour SEQ2LABEL. Cela suggère que le modèle GRAPHE est plus efficace pour extraire des informations syntaxique directement depuis le signal, mais qu'il est plus sensible au bruit présent dans le signal de parole. On peut également faire l'hypothèse que le décodage global du système GRAPHE est plus adapté pour extraire les arbres syntaxiques. Le plus grand écart entre les deux approches a lieu quand plus d'informations à propos de la segmentation du signal audio sont données au modèle (**ORACLE**), réduisant l'influence de la RAP sur la qualité de l'analyse. On peut supposer que l'approche GRAPHE est plus adaptée pour l'analyse syntaxique directement depuis le signal de parole, particulièrement quand le `WER` baisse.

Transcrire puis analyser ou analyser directement ? L'approche **CASCADE** avec `hops` atteint des performances similaires à la version **AUDIO** avec notre analyseur par graphe. Cependant, le modèle cascade utilisant `hops` est un modèle plus complexe et n'est pas complètement comparable à notre analyseur. De plus, il possède environ 50% de paramètres en plus que notre modèle, nécessite 2 modèles pré-entraînés ; il est donc plus coûteux à entraîner.

Finalement, le tableau 2 montre que l'approche **AUDIO** surpasse l'approche **CASCADE** quand la qualité de la RAP augmente. Ce résultat suggère que l'analyse syntaxique profite de l'audio dès lors que la RAP atteint une qualité raisonnable.

Limitations Nous évaluons seulement nos analyseurs sur le français, grâce à la disponibilité d'un grand corpus arboré, ainsi nos conclusions doivent être interprétées dans ce cadre restreint. Nous comptons étendre ce travail à d'autres langues dans le futur.

Nous n'avons pas effectué une recherche poussée d'hyperparamètres, à cause du coût computationnel, mais nous avons dédié approximativement le même budget computationnel pour chaque modèle dans une configuration donnée.

5 Conclusion

Nous avons présenté un analyseur syntaxique par graphe prenant uniquement le signal audio en tant qu'entrée et avons évalué ses performances dans des expériences variées et avec plusieurs expériences contrôles. Nous montrons qu'un simple analyseur syntaxique basé sur les graphes avec wav2vec2 obtient des résultats similaires ou surpasse des modèles plus complexe nécessitant deux modèles pré-entraînés. Avec l'expérience de contrôle (**ORACLE**), nous montrons qu'obtenir des représentations vectorielles de mot de bonne qualité est le défi principal pour l'analyse syntaxique de la parole. Nous supposons également que l'approche par GRAPHE est plus adapté pour l'analyse syntaxique directement depuis le signal de parole qu'une approche par étiquetage de séquence. Nous concentrerons nos efforts futurs sur l'amélioration de la qualité de la segmentation du signal audio pour pallier ces problèmes.

Remerciements

Nous remercions les relecteurices anonymes pour leur remarques et suggestions. Ce travail a bénéficié d'un financement du Fond National Suisse (No. 197864) et de l'Agence Nationale de la Recherche, via les projet PROPICTO (ANR-20-CE93-0005) et SynPaX (ANR-23-CE23-0017). Ce travail a également bénéficié d'un accès aux moyens de calcul de l'IDRIS via l'allocation de ressources AD011013463R1 attribuée par GENCI.

Références

ANDRÉ V. (2016). Fleuron : Français langue Étrangère universitaire–ressources et outils numériques.

ARDILA R., BRANSON M., DAVIS K., KOHLER M., MEYER J., HENRETTY M., MORAIS R., SAUNDERS L., TYERS F. & WEBER G. (2020). Common voice : A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, p. 4218–4222, Marseille, France : European Language Resources Association.

ATILF (2020). Tcof : Traitement de corpus oraux en français. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

BAEVSKI A., ZHOU Y., MOHAMED A. & AULI M. (2020). wav2vec 2.0 : A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, **33**, 12449–12460.

BENZITOUN C., DEBAISIEUX J.-M. & DEULOFEU H.-J. (2016). Le projet orféo : un corpus d'étude pour le français contemporain. *Corpus*, (15).

CARRUTHERS J. (2013). French oral narrative corpus. Commissioning Body / Publisher : Oxford Text Archive.

CHARNIAK E. & JOHNSON M. (2001). Edit detection and parsing for transcribed speech. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

CHRUPAŁA G. (2023). Putting natural in natural language processing. In A. ROGERS, J. BOYD-GRABER & N. OKAZAKI, Éds., *Findings of the Association for Computational Linguistics : ACL 2023*, p. 7820–7827, Toronto, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/2023.findings-acl.495.

CLESTHIA (2018). Cfpp2000. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

CRESTI E., DO NASCIMENTO F. B., SANDOVAL A. M., VERONIS J., MARTIN P. & CHOUKRI K. (2004). The c-oral-rom corpus. a multilingual resource of spontaneous speech for romance languages. p. 26–28.

DELIC E., TESTON-BONNARD S. & VÉRONIS J. (2004). Présentation du corpus de référence du français parlé. *Recherches sur le français parlé*, **18**, 11–42. Equipe DELIC, HAL : halshs-01388193.

DOZAT T. & MANNING C. D. (2016). Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

FRANCARD M., HAMBYE P., SIMON A.-C. & DISTER A. (2009). Du corpus à la banque de données. : Du son, des textes et des métadonnées. l'évolution de banque de données textuelles orales valibel (1989-2009). *Cahiers de l'Institut de linguistique de Louvain-CILL*, **33**(2), 113.

GÓMEZ-RODRÍGUEZ C., STRZYZ M. & VILARES D. (2020). A unifying theory of transition-based and sequence labeling parsing. In D. SCOTT, N. BEL & C. ZONG, Éds., *Proceedings of the 28th International Conference on Computational Linguistics*, p. 3776–3793, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.336.

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 '06, p. 369–376, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/1143844.1143891.

GROBOL L. & CRABBÉ B. (2021). Analyse en dépendances du français avec des plongements contextualisés (French dependency parsing with contextualized embeddings). In P. DENIS, N. GRABAR, A. FRAISSE, R. CARDON, B. JACQUEMIN, E. KERGOSIEN & A. BALVET, Éds., *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, p. 106–114, Lille, France : ATALA.

HONNIBAL M. & JOHNSON M. (2014). Joint incremental disfluency detection and dependency parsing. *Transactions of the Association for Computational Linguistics*, **2**, 131–142. DOI : 10.1162/tacl_a_00171.

HUSIANYCIA M. (2011). *Caractérisation de types de discours dans des situations de travail*. Theses, Université Nancy 2. HAL : tel-01749085.

ICAR (2017). Clapi. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

JAMSHID LOU P., WANG Y. & JOHNSON M. (2019). Neural constituency parsing of speech transcripts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 2756–2765, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1282.

JOHNSON M. & CHARNIAK E. (2004). A TAG-based noisy-channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, p. 33–39, Barcelona, Spain. DOI : 10.3115/1218955.1218960.

KAWAGUCHI Y., ZAIMA S. & TAKAGAKI T., Éds. (2006). *Spoken Language Corpus and Linguistic Informatics*. John Benjamins.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty French language model. In D. JURAFSKY, J. CHAI, N. SCHLUTER & J. TETREAULT, Éds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.645.

MATHIEU A., MARIE-JOSÉ B., GILLES C., FEDERICA D. & ANNE J. L. ((2012-2020)). Corpus ofrom – corpus oral de français de suisse romande. Université de Neuchâtel.

PARCOLLET T., NGUYEN H., EVAIN S., ZANON BOITO M., PUPIER A., MDHAFFAR S., LE H., ALISAMIR S., TOMASHENKO N., DINARELLI M., ZHANG S., ALLAUZEN A., COAVOUX M., ESTÈVE Y., ROUVIER M., GOULIAN J., LECOUTEUX B., PORTET F., ROSSATO S., RINGEVAL F., SCHWAB D. & BESACIER L. (2024). Lebenchmark 2.0 : A standardized, replicable and enhanced framework for self-supervised representations of french speech. *Computer Speech Language*, **86**, 101622. DOI : https ://doi.org/10.1016/j.csl.2024.101622.

PRICE P. J., OSTENDORF M., SHATTUCK-HUFNAGEL S. & FONG C. (1991). The use of prosody in syntactic disambiguation. *the Journal of the Acoustical Society of America*, **90**(6), 2956–2970.

PUPIER A., COAVOUX M., LECOUTEUX B. & GOULIAN J. (2022). End-to-End Dependency Parsing of Spoken French. In *Proc. Interspeech 2022*, p. 1816–1820. DOI : 10.21437/Interspeech.2022-381.

RASOOLI M. S. & TETREAULT J. (2013). Joint parsing and disfluency detection in linear time. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 124–129, Seattle, Washington, USA : Association for Computational Linguistics.

RAVANELLI M., PARCOLLET T., PLANTINGA P., ROUHE A., CORNELL S., LUGOSCH L., SUBAKAN C., DAWALATABAD N., HEBA A., ZHONG J., CHOU J.-C., YEH S.-L., FU S.-W., LIAO C.-F., RASTORGUEVA E., GRONDIN F., ARIS W., NA H., GAO Y., MORI R. D. & BENGIO Y. (2021). SpeechBrain : A general-purpose speech toolkit. arXiv :2106.04624.

STRZYZ M., VILARES D. & GÓMEZ-RODRÍGUEZ C. (2019). Viable dependency parsing as sequence labeling. In J. BURSTEIN, C. DORAN & T. SOLORIO, Éds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 717–723, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1077.

STRZYZ M., VILARES D. & GÓMEZ-RODRÍGUEZ C. (2020). Bracketing encodings for 2-planar dependency parsing. In D. SCOTT, N. BEL & C. ZONG, Éds., *Proceedings of the 28th International Conference on Computational Linguistics*, p. 2472–2484, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : 10.18653/v1/2020.coling-main.223.