Growing Trees on Sounds: Assessing Strategies for End-to-End Dependency Parsing of Speech

Anonymous ACL submission

Abstract

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 in 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.

1 Introduction

Dependency parsing is a central task in natural language processing (NLP). In the NLP community, it has mostly been addressed on textual data, either natively written texts or sometimes speech transcriptions. Yet, speech is the main form of communication between humans, as well as arguably one of the most realistic types of linguistic data, which motivates the design of NLP systems able to deal directly with speech, both for applicative purposes and to construct corpora annotated with linguistic information. When parsing speech *transcriptions*, most prior work has focused on disfluency detection and removal (Charniak and Johnson, 2001; Johnson and Charniak, 2004; Rasooli and Tetreault, 2013; Honnibal and Johnson, 2014; Jamshid Lou et al., 2019), in an effort to 'normalize' the transcriptions and make them suitable input for NLP systems trained on written language. Using only transcriptions as input is a natural choice from an NLP perspective: it makes it possible to use off-the-shelf NLP parsers 'as is'. However, predicted transcriptions can be very noisy, in particular for speech from spontaneous conversations. Furthermore, transcriptions are abstractions that contain much less information than the speech signal. The prosody, and the pauses in the speech utterances are very important clues for parsing (Price et al., 1991) that are completely absent from transcriptions. Hence, we address speech parsing using only the speech signal as input. With the popularization of self-supervised method and modern neural network architecture (pretrained transformers), both speech and text domains now use similar techniques (Chrupała, 2023). This convergence of methodology has raised interest in other applications of speech models to go beyond 'simple' speech recognition. Thus, addressing classical NLP tasks directly on speech is a natural step and design NLP tools able to deal with spontaneous speech, arguably the most realistic type of linguistic production. In short, Our contributions are the following:

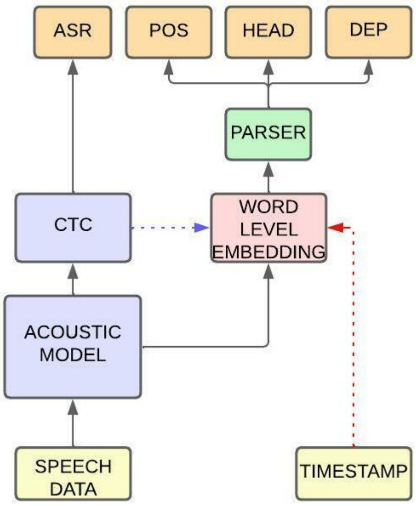
- we introduce a graph-based end-to-end dependency parsing algorithm for speech;
- we evaluate the parser (and compare it to pipelines as well as a parsing-as-tagging parser) on Orféo, a large treebank of spoken French that features spontaneous speech;
- we release our code at `git.anonymized`.

2 Parsers and pre-trained models

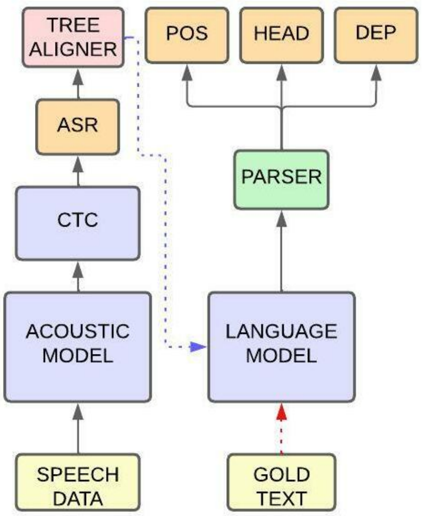
We define speech parsing as the task of predicting a dependency tree from an audio signal corresponding to a spoken utterance.[1]

Our parser is composed of 2 modules (Figure 1a): (i) an acoustic module that is used to predict transcriptions and a segmentation of the signal in words

[1] For the sake of simplicity, we will use the term 'sentence' in the rest of the article, even though the very definition of a sentence is debatable in the spoken domain.



(a) The two models based on audio features, blue arrow is **AUDIO**, red arrow is **ORACLE**.



(b) The two baseline models based on a pretrained language model, blue arrow is **PIPELINE** (predicted transcription), read arrow is **TEXT** (gold transcriptions).

Figure 1: Overview of architectures with the 4 settings described in Section 4.

(ii) a parsing module that uses the segmentation to construct audio word embeddings and predict trees.

Word level representations from speech To extract representations on raw speech, we use a pretrained wav2vec2 model trained on seven thousand hours of French speech: `LeBenchmark7K`[2] (Parcollet et al., 2024). Parsing requires word-level representations. We use the methodology of Pupier et al. (2022) to construct audio word embeddings from the implicit frame level segmentation provided by the CTC speech recognition algorithm (Graves et al., 2006). The method consists in combining the frame vectors corresponding to a single predicted word with an LSTM.

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

Graph based parsing We use the audio word embeddings –whose construction is described above– as input to our implementation of a classical graph-based biaffine parser (Dozat and Manning, 2016): (i) compute a score every possible arc with a biaffine classifier (ii) find the best scoring tree with a maximum spanning tree algorithm.

Sequence labeling The sequence labeling parser follows Pupier et al. (2022) and is based on the *dep2label* approach (Gómez-Rodríguez et al., 2020; Strzyz et al., 2020), specifically the `relative POS-based encoding` (Strzyz et al., 2019). This method reduces the parsing problem to a sequence labelling problem. The head of each token is encoded in a label of the form ±Integer@POS. The integer stands for the relative position of the head considering only words of the POS category. Eg., `-3@NOUN` means that the head of the current word is the third noun before it.

3 Dataset

We use the CEFC-Orféo treebank (Benzitoun et al., 2016), a dependency-annotated French corpus composed of multiple subcorpora (CLESTHIA, 2018; ICAR, 2017; ATILF, 2020; Mathieu et al., (2012-2020; André, 2016; Carruthers, 2013; Cresti et al., 2004; DELIC et al., 2004; Francard et al., 2009; Kawaguchi et al., 2006; Husianycia, 2011), and released with the audio recordings. The treebank consists of various types of interactions, all of which feature spontaneous discussions, except for the French Oral Narrative corpus (audiobooks). Orféo features many types of speech situations (eg. commercial interactions, interviews, informal discussions between friends) and is the largest French spoken corpus annotated in dependency syntax. The annotation scheme has been designed specifically for Orféo (Benzitoun et al., 2016) and differs from the Universal Dependency framework in many regards (in particular: its POS tagset is finer-grained, whereas the syntactic function tagset has only 14 relations). The syntactic annotations of Orféo were done manually for 5% of the corpus and automatically for the rest of the corpus. The train/dev/test split we use makes sure that the test section only contains gold annotations. Nevertheless, the subcorpora with gold syntactic annotations correspond to low-quality recordings, which makes them a very challenging benchmark.

4 Experiments

Experimental settings Our experiments aim at: (i) comparing our graph-based parser to the seq2label model (ii) comparing to pipeline approaches with text-based parsers (iii) assessing the robustness of word representations with control experiments: using word boundaries (provided in the corpus) as input for the audio models and gold transcriptions for the text-based model. We compare the following settings (illustrated in Figure 1):

- AUDIO: Access to **raw audio** only, the model creates word-level representation from the acoustic model as described in Section 2.
- ORACLE: Access to **raw audio** and **silver**[3] **word-level timestamps**, making it easier to create word representations and mitigating the impact of the quality of the speech recognition on parsing.
- PIPELINE: Access to **predicted transcriptions** from the acoustic model only, then a language model uses the transcriptions as input for parsing. The training trees are modified to take into account any deletion and insertion of words. However, as for the speech approach, deletion or insertion penalizes the global score of the model since the model is evaluated against the gold transcriptions and not the modified one. The drawback of this approach is that no information about prosody or pauses is available.
- TEXT: Access to **gold transcriptions**: this unrealistic setting provides an upper bound performance in the ideal case (perfect ASR).

Both PIPELINE and TEXT settings use a French BERT model: `camembert-base`[4] (Martin et al., 2020) to extract contextualized word embeddings. For PIPELINE and TEXT settings, on top of our implementations, we use `hops` (Grobol and Crabbé, 2021), an external state-of-the-art graph-based parser. The `hops` parser uses a character-bi-LSTM in addition to BERT to produce word embeddings, whereas our implementation does not (in an effort to make both versions of our parser, text-based and audio-based, as similar as possible).

[3]The corpus contained word-level timestamps that have been automatically constructed through forced alignment.

[4]https://huggingface.co/almanach/camembert-base

Each parsing method for each modality is trained with the same number of epochs, the same hyper-parameters (see Table 4 and 5 of Appendix A), and approximately the same number of parameters. We select the best checkpoint on the validation set in each setting for the final evaluation. Our implementations use speechbrain (Ravanelli et al., 2021).

Metrics We use classical evaluation measures: *Word Error Rate* (`WER`) and *Character Error Rate* (`CER`) for speech recognition, *POS accuracy* (`POS`), *Unlabeled Attachment Score* (`UAS`), and *Labeled Attachment Score* (`LAS`) for dependency parsing.

We report results in Table 1 for the full corpus, and in Table 2 for a subcorpus of the test set (Valibel) for which speech recognition is easier.

Results: Speech recognition effect on parsing quality In Table 1, we observe that both graph-based and seq2label-based approaches give similar results when using no additional information, which shows that the limiting factor of the model is the speech recognition, rather than the parsing.

It is important to note that due to the nature of the speech corpus (spontaneous discussions), the `WER` is higher than what is typically expected on ASR benchmarks (usually containing 'read' speech). The ASR module used in our model reaches around 8 `WER` when trained and evaluated on CommonVoice5.1 (Ardila et al., 2020).

Further evidence of this is shown in Table 3: changing the number of parameters of the graph-based parser does not significantly alter performance. Additionally, in Table 2 we observe a clear improvement in all the parsing metrics when evaluating on a test corpus with better speech recognition performance. The model's speech recognition ability directly affects the number of predicted tokens (some words may be deleted or added), which in turn impacts parsing.

Results: Difference between sequence labeling approach and graph-based approach It is somewhat surprising that on the text modality (PIPELINE), the sequence labeling parser outperforms the graph-based approach since this is not the case on the other modality (AUDIO). However, it does not outperform a larger graph-based model with an additional character-bi-LSTM such as `hops`. The character bi-LSTM may mitigate the impact of out-of-vocabulary words produced by misspelling errors from the ASR.

On AUDIO and ORACLE settings, the graph-

Model	WER↓	CER↓	POS↑	UAS↑	LAS↑	Parameters	Pre-training
AUDIO SEQ2LABEL	35.9	22.3	73.0	65.7	60.4	315M + 34.9M	Wav2vec2
AUDIO GRAPH	**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 GRAPH	**35.6**	**22.2**	**77.4**	**73.3**	**67.5**	315M + 34.9M	Wav2vec2
PIPELINE SEQ2LABEL	35.6	22.0	70.8	63.8	58.4	314M + 110M + 39.2M	Wav2vec2 + CamemBERT
PIPELINE GRAPH	35.6	22.0	69.3	60.5	53.1	314M + 110M + 41.4M	Wav2vec2 + CamemBERT
PIPELINE HOPS	35.6	22.0	**72.4**	**65.8**	**61.0**	314M + 110M + 100M	Wav2vec2 + CamemBERT
TEXT SEQ2LABEL	0	0	96.9	88.8	85.7	110M + 39.2M	CamemBERT
TEXT GRAPH	0	0	95.1	87.4	84.0	110M + 41.4M	CamemBERT
TEXT HOPS	0	0	**98.2**	**90.3**	**87.7**	110M + 100M	CamemBERT

Table 1: Evaluation on the full Orféo test set with the settings described in Section 4.

Model	WER↓	CER↓	POS↑	UAS↑	LAS↑	Parameters	Pre-training
AUDIO SEQ2LABEL	31.0	18.4	**77.1**	70.2	65.2	315M + 34.9M	Wav2vec2
AUDIO GRAPH	**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 GRAPH	31.4	19.2	**79.8**	**76.0**	**70.4**	315M + 34.9M	Wav2vec2
PIPELINE SEQ2LABEL	30.5	18.2	74.7	67.7	62.4	314M + 110M + 39.2M	Wav2vec2 + CamemBERT
PIPELINE GRAPH	30.5	18.2	73.5	64.2	57.3	314M + 110M + 41.4M	Wav2vec2 + CamemBERT
PIPELINE HOPS	30.5	18.2	**76.3**	**69.4**	**64.6**	314M + 110M + 100M	Wav2vec2 + CamemBERT
TEXT SEQ2LABEL	0	0	94.5	86.7	83.1	110M + 39.2M	CamemBERT
TEXT GRAPH	0	0	96.8	88.3	84.5	110M + 41.4M	CamemBERT
TEXT HOPS	0	0	**98.2**	**90.3**	**87.1**	110M + 100M	CamemBERT

Table 2: Evaluation on the Valibel corpus (a subset of the test set).

	WER↓	CER↓	POS↑	UAS↑	LAS↑	Parameters
Graph-tiny	35.74	22.32	72.97	65.86	60.79	314M + 11.7M
Graph-base	35.63	22.10	73.13	66.05	60.90	314M + 34.9M
Graph-large	35.60	22.02	73.17	65.96	60.67	314M + 67.6M

Table 3: Comparison of parsing metrics with the graph-based architecture and different number of parameters.

based model seems more robust to noise than the simpler sequence-labelling approach.

A hypothesis about the graph-based model performance on **AUDIO** and **ORACLE** settings may be that it is more robust to noise (due to its global decoding) than simpler approaches such as sequence labeling. The largest gap between the two parsing approaches occur when more information about speech segmentation is given to the models (**ORACLE**), reducing the overall influence of the speech recognition task on parsing.

Transcribe then parse or directly parse ? The **PIPELINE** approach with hops does reach a similar performance as the **AUDIO** model with our graph-based parser. However, hops is a more complex model not fully comparable to our graph-based parser. Moreover, it has 50% as many parameters as the model working directly on audio, requires 2 pretrained models, and is thus more expensive to train.

Lastly, Table 2 shows that the **AUDIO** approach outperforms the **PIPELINE** approach when the quality of the speech recognition improves. This result suggests that parsing benefits from **AUDIO** as soon as ASR reaches reasonable quality.

5 Conclusion

We introduced a graph-based speech parser that takes only the raw audio signal as input and assessed its performance in various settings and in several control experiments. We show that a simple graph-based approach with wav2vec2 audio features is on a par with or outmatches a more complex pipeline approach that requires two pretrained models.

From control experiments (**ORACLE**), we show that acquiring good quality word representations directly from speech is the main challenge for speech parsing. We will focus future work on improving the quality of word segmentation on the speech signal.

Limitations

We only evaluate our parsers on French, due to the availability of a large treebank, hence our conclusions should be interpreted with this restricted scope. We plan to extend to other languages and treebanks in future work.

We did not do a full grid search for hyperparameter tuning, due to computational resource limitations, although we dedicated approximately the same computation budget to each model in a dedicated setting. However, we acknowledge that not doing a full hyperparameter search may have affected the final performance of the parsers.

References

Virginie André. 2016. Fleuron: Français langue Étrangère universitaire–ressources et outils numériques.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 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.

Christophe Benzitoun, Jeanne-Marie Debaisieux, and Henri-José Deulofeu. 2016. Le projet orféo: un corpus d'étude pour le français contemporain. *Corpus*, (15).

Janice Carruthers. 2013. French oral narrative corpus. Commissioning Body / Publisher: Oxford Text Archive.

Eugene Charniak and Mark Johnson. 2001. Edit detection and parsing for transcribed speech. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Grzegorz Chrupała. 2023. Putting natural in natural language processing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7820–7827, Toronto, Canada. Association for Computational Linguistics.

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

Emanuela Cresti, Fernanda Bacelar do Nascimento, Antonio Moreno Sandoval, Jean Veronis, Philippe Martin, and Khalid Choukri. 2004. The c-oral-rom corpus. a multilingual resource of spontaneous speech for romance languages. pages 26–28.

Equipe DELIC, Sandra Teston-Bonnard, and Jean Véronis. 2004. Présentation du corpus de référence du français parlé. *Recherches sur le français parlé*, 18:11–42. Equipe DELIC.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Michel Francard, Philippe Hambye, Anne-Catherine Simon, and Anne Dister. 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.

Carlos Gómez-Rodríguez, Michalina Strzyz, and David Vilares. 2020. A unifying theory of transition-based and sequence labeling parsing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3776–3793, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Loïc Grobol and Benoit Crabbé. 2021. Analyse en dépendances du français avec des plongements contextualisés (French dependency parsing with contextualized embeddings). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, pages 106–114, Lille, France. ATALA.

Matthew Honnibal and Mark Johnson. 2014. Joint incremental disfluency detection and dependency parsing. *Transactions of the Association for Computational Linguistics*, 2:131–142.

Magali Husianycia. 2011. *Caractérisation de types de discours dans des situations de travail*. Theses, Université Nancy 2.

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

Paria Jamshid Lou, Yufei Wang, and Mark Johnson. 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)*, pages 2756–2765, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Johnson and Eugene Charniak. 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)*, pages 33–39, Barcelona, Spain.

Yuji Kawaguchi, Susumu Zaima, and Toshihiro Takagaki, editors. 2006. *Spoken Language Corpus and Linguistic Informatics*. John Benjamins.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Avanzi Mathieu, Béguelin Marie-José, Corminboeuf Gilles, Diémoz Federica, and Johnsen Laure Anne. (2012-2020). Corpus ofrom – corpus oral de français de suisse romande. Université de Neuchâtel.

Titouan Parcollet, Ha Nguyen, Solène Evain, Marcely Zanon Boito, Adrien Pupier, Salima Mdhaffar, Hang Le, Sina Alisamir, Natalia Tomashenko, Marco Dinarelli, Shucong Zhang, Alexandre Allauzen, Maximin Coavoux, Yannick Estève, Mickael Rouvier, Jerôme Goulian, Benjamin Lecouteux, François Portet, Solange Rossato, Fabien Ringeval, Didier Schwab, and Laurent Besacier. 2024. Lebenchmark 2.0: A standardized, replicable and enhanced framework for self-supervised representations of french speech. *Computer Speech Language*, 86:101622.

Patti J Price, Mari Ostendorf, Stefanie Shattuck-Hufnagel, and Cynthia Fong. 1991. The use of prosody in syntactic disambiguation. *the Journal of the Acoustical Society of America*, 90(6):2956–2970.

Adrien Pupier, Maximin Coavoux, Benjamin Lecouteux, and Jerome Goulian. 2022. End-to-End Dependency Parsing of Spoken French. In *Proc. Interspeech 2022*, pages 1816–1820.

Mohammad Sadegh Rasooli and Joel Tetreault. 2013. Joint parsing and disfluency detection in linear time. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 124–129, Seattle, Washington, USA. Association for Computational Linguistics.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. 2021. SpeechBrain: A general-purpose speech toolkit. ArXiv:2106.04624.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable dependency parsing as sequence labeling. 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)*, pages 717–723, Minneapolis, Minnesota. Association for Computational Linguistics.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2020. Bracketing encodings for 2-planar dependency parsing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2472–2484, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Training Details

Table 4 and 5 describe in more detail the hyperparameters used for each parser for the different sets of modalities.

Parser	SEQ	GRAPH
Epoch	30	30
Batch size	8	8
Tuning parameters		
Learning rate	0.0001	0.0001
Optimizer	AdaDelta	AdaDelta
Model name	LeBenchmark7K	
Encoder		
Encoder layer	3	3
Dropout	0.15	0.15
Encoder Dim	1024	1024
Activation	LeakyReLU	LeakyRelu
Fusion LSTM		
Layer	2	2
Dim	500	500
Bidirectional	False	False
Bias	TRUE	True
LSTM parser		
Layer	2	3
Dim	800	768
Bidirectional	TRUE	TRUE
Labeler (SEQ2LABEL)		
Dim	1600	
Layer	1	
Linear head dim arc	846	
Linear head dim POS	23	
Linear head dim label	19	
Arc MLP (GRAPH)		
Dim		768
Layer		1
Linear head dim		768
Label MLP (GRAPH)		
Dim		768
Layer		1
Head dim		768
POS MLP (GRAPH)		
Dim		768
Linear head dim		24

Table 4: **AUDIO** and **ORACLE** SEQ2LABEL and GRAPH hyperparameters.

Parser	SEQ2LABEL	GRAPH	HOPS
Epoch	40	40	40
Batch size	32	32	32
Tuning parameters			
Learning rate	0.001	0.001	0.00003
optimizer	Adam	Adam	Adam
Embedding	Last layer	Last layer	Mean First 12 layers
Embedding dim	768	768	768
BERT	camembert_base		
Char Bi-LSTM HOPS			
Embedding dim			128
Word Embedding HOPS			
Embedding dim			256
LSTM parser			
Dim	768	768	512
Layers	3	2	3
Bidirectional	True	True	True
Labeler (SEQ2LABEL)			
Dim	1536		
Layer	1		
Linear head dim arc	846		
Linear head dim POS	23		
Linear head dim label	19		
Arc MLP (GRAPH and HOPS)			
Dim		768	1024
Layer		1	2
Linear head dim		768	768
Label MLP (GRAPH)			
Dim		768	1024
Layer		1	2
Head dim		768	768
POS MLP (GRAPH)			
Dim		768	1024
Linear head dim		24	24

Table 5: **PIPELINE** and **TEXT** SEQ2LABEL, GRAPH and **PIPELINE** hyperparameters.