Interspeech 2022
18-22 September 2022, Incheon, Korea

End-to-End Dependency Parsing of Spoken French

Adrien Pupier, Maximin Coavoux, Benjamin Lecouteux, Jérôme Goulian

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, France

first.last@univ-grenoble-alpes.fr

Abstract

Research efforts in syntactic parsing have focused on written texts. As a result, speech parsing is usually performed on transcriptions, either in unrealistic settings (gold transcriptions) or on predicted transcriptions. Parsing speech from transcriptions, though straightforward to implement using out-of-the-box tools for Automatic Speech Recognition (ASR) and dependency parsing has two important limitations. First, relying on transcriptions will lead to error propagation due to recognition mistakes. Secondly, many acoustic cues that are important for parsing (prosody, pauses, ...) are no longer available in transcriptions.

To address these limitations, we introduce wav2tree, an end-to-end dependency parsing model whose only input is the raw signal. Our model builds on a pretrained wav2vec2 encoder with a CTC loss to perform ASR. We extract token segmentation from the CTC layer to construct vector representations for each predicted token. Then, we use these token representations as input to a generic parsing algorithm. The whole model is trained end-to-end with a multitask objective (ASR, parsing) to reduce error propagation. Our experiments on the Orféo treebank of spoken French show that direct parsing from speech is feasible: wav2tree outperforms a pipeline approach based on wav2vec (for ASR) and FlauBERT (for parsing).
Index Terms: speech recognition, dependency parsing, end-to-end, raw signal, spontaneous speech

1. Introduction

Progress on syntactic parsing of written text has been remarkable in the last decade, thanks to the use of bi-LSTM [1] and pretrained language models such as BERT [2]. Most of the work on syntactic parsing of speech has been done on transcribed speech, either from a human annotator (gold) leading to unrealistic parsing settings or from automatic speech recognition tools [3, 4]. Spoken language, as opposed to written language, exhibits specific phenomena, such as disfluencies, hesitations, false starts, that arguably make syntactic parsing harder.

However, parsing transcriptions is unsatisfactory since listeners use prosodic cues to help solve ambiguity [5] and transcriptions no longer provide access to prosodic information. Indeed, using acoustic cues has proven to be a viable way to improve parsing on transcribed speech [6]. Thus, previous work focused mainly on integrating acoustic cues from the signal and using a multimodal model with both text and features extracted from the audio such as word duration and fundamental frequency [7]. Another line of work has focused on automatically detecting disfluencies and ASR mistakes while parsing transcriptions [8, 9, 10, 11, 12]. The disfluencies detection mitigates the error propagation due to the ASR model. Moreover, it allows transition-based parsers to prune disfluency nodes with specific repair actions [10].

We focus on dependency parsing, where each word in a sentence is attached to its head word with a syntactic function (subject, object, modifier,...). In this paper, we introduce wav2tree, an end-to-end neural model for speech parsing. Wav2tree jointly predicts a transcription and its syntactic analysis (a labeled dependency tree and a sequence of part-of-speech tags), using only the raw signal as input. The motivations for designing an end-to-end model for this task are manifold. End-to-end models are known to reduce error propagation thanks to shared error optimization, unlike pipeline approaches, where an error in a previous component will trigger errors in downstream tasks. We compare wav2tree to several pipeline baselines that use pretrained models for both the speech and text modalities. To the best of our knowledge, this is the first attempt to only use signal representations for syntactic parsing. In summary, our contributions are threefold:

- We introduce a new architecture to parse speech directly from the signal;
- We provide an evaluation on our model on a spoken treebank for French;
- We publicly release the code of this architecture online along with pretrained models.[1]

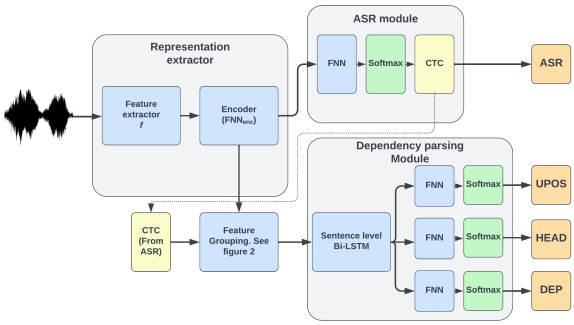


Figure 1: *Wav2tree end-to-end architecture.*

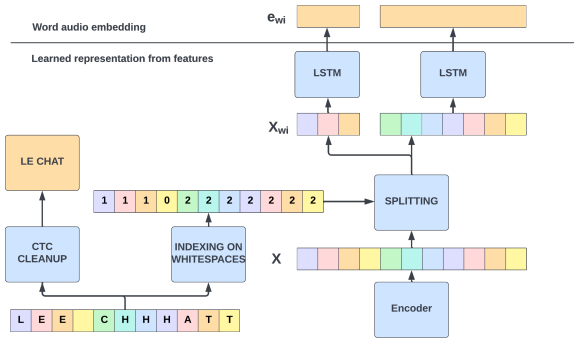
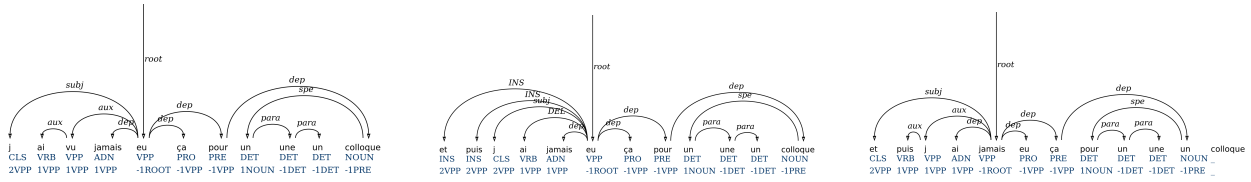


Figure 2: *Feature grouping using CTC segmentation.*

[1] https://gricad-gitlab.univ-grenoble-alpes.fr/pupiera/wav2tree_release

10.21437/Interspeech.2022-381



(a) *Gold sentence dependency tree* (b) *New best tree from ASR output.* (c) *ASR tree without Oracle*

Figure 3: *An example of the tree produced by the oracle with the relative encoding labeling.*

2. Wav2tree: an end-to-end architecture for parsing speech

The objective of our model is to take the raw signal corresponding to a sentence[2] as input and compute the sequence of words and the dependency tree contained within this signal. The architecture of our model is divided into three parts: (i) the representation extractor module, (ii) the ASR module, and (iii) the parsing module, as illustrated in Figures 1 and 2.

The first module of our architecture is the representation extractor. From a signal S, we compute a feature matrix $\mathbf{X}$ as follows:

$$\boldsymbol{X} = \mathrm{FNN}_{\mathrm{enc}}(f(S)), \tag{1}$$

where f is any feature extraction method (MFCC, wav2vec, ...) and $\mathrm{FNN}_{\mathrm{enc}}$ is a feedforward neural network. Another possible instantiation for $\mathrm{FNN}_{\mathrm{enc}}$ is a convolutional recurrent (CRNN) encoder [13], but any neural architecture can be used.

2.1. ASR module

The automatic speech recognition module is composed of a fully connected layer with a softmax activation. The model predicts a character for each audio feature, the sequence of characters is then collapsed through Connectionist Temporal Classification [14, 15, CTC]. We do not use any language model or beam search, despite what is usually done in ASR. Indeed, we are not aware of any pre-trained language model for transcriptions of spontaneous French. Using a language model trained on French books or data crawled from the web would introduce bias, due to linguistic differences between spoken and written French.

2.2. Dependency parsing module

The dependency parsing module is composed of two parts: (i) the audio word embedding extraction and (ii) the parsing model. For the first one, we use the CTC segmentation of the audio features, through the prediction of the whitespace characters, to compute an embedding for each predicted token. Each audio feature, except for the ones corresponding to whitespaces, from the encoder module is part of one word, and every feature between two whitespace characters belongs to the same word. Thanks to this property, we map each feature to a word, and we use this mapping to compute our audio word embedding through an LSTM. This process is described in Figure 2 and is formally described in eqs. (2) and (3):

$$\boldsymbol{X}_{w_i} = \boldsymbol{X}[\boldsymbol{x}_j, ..., \boldsymbol{x}_k], \tag{2}$$

$$\boldsymbol{e}_{w_i} = \mathrm{LSTM}_{\mathrm{hidden}}(\boldsymbol{X}_{w_i}), \tag{3}$$

where $\boldsymbol{X}$ is the representation matrix from the encoder defined in eq. (1), j and k are the boundary of the word i, and $\mathbf{e}_{w_i}$ is the word embedding corresponding to token i.

[2]The term 'sentence' is questionable when referring to spontaneous speech. We keep it here by analogy to written text parsing.

Once we have computed an "audio word embedding" for each token in a sentence, we can use any generic dependency parsing method. For the sake of simplicity and speed, we used the dep2Label method [16], that reduces the parsing problem into a sequence tagging problem. Each token is assigned three labels for (i) its part of speech tag, (ii) the relative position of its head, and (iii) its syntactic function. For example, in Figure 3a, the gold labels for token *j'* (I) are: `CLS`, `2VPP`, `subj`, meaning *j'* is the subject of the second past participle verb (VPP) on its right. Our model computes scores for each token and each subtask as follows:

$$\boldsymbol{Z} = \mathrm{biLSTM}_{\mathrm{output}}(\boldsymbol{E}), \tag{4}$$

$$\boldsymbol{P}_{\mathrm{pos}} = \mathrm{Softmax}(\mathrm{FNN}_{\mathrm{pos}}(\boldsymbol{Z})), \tag{5}$$

$$\boldsymbol{P}_{\mathrm{head}} = \mathrm{Softmax}(\mathrm{FNN}_{\mathrm{head}}(\boldsymbol{Z})), \tag{6}$$

$$\boldsymbol{P}_{\mathrm{dep}} = \mathrm{Softmax}(\mathrm{FNN}_{\mathrm{dep}}(\boldsymbol{Z})), \tag{7}$$

where $\mathbf{E}$ is the matrix containing the sequence of audio word embeddings and FFNs are feedforward networks.

3. Training wav2tree

3.1. Training objective

To train our model, we minimize the sum of the negative log-likelihood on the three classification tasks, and we add the CTC loss for the automatic speech recognition task:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, S, \mathbf{c}, \mathbf{p}, \mathbf{h}, \mathbf{d}) = &-\alpha \cdot \log P(\mathbf{p}|S;\boldsymbol{\theta}) - \beta \cdot \log P(\mathbf{h}|S;\boldsymbol{\theta}) \\ &-\gamma \cdot \log P(\mathbf{d}|S;\boldsymbol{\theta}) - \delta \cdot \log P(\mathbf{c}|S;\boldsymbol{\theta})\end{aligned} \tag{8}$$

where $\boldsymbol{\theta}$ are the model parameters, S is the signal, $\mathbf{p}$ is the target POS sequence, $\mathbf{h}$ is the target sequence of relative positions encoding the dependency tree, $\mathbf{d}$ is the target sequence of syntactic functions, $\mathbf{c}$ is the sequence of gold tokens, and $(\alpha, \beta, \gamma, \delta)$ are hyperparameters.

3.2. Oracle

The token segmentation depends on the ASR frame-level predictions. An error in the segmentation makes the gold tree no longer attainable. As a result, the parsing models cannot always output the correct tree. The parsing model loss should not increase in these cases since the mistakes come from the ASR module. The most common problem is a mismatch in the number of tokens between the supervision and the prediction. In this case, giving the parsing model good supervision is crucial. Using the original supervision to the model would introduce noise when the segmentation is not correct as illustrated in Figure 3c. Thus, we need to compute the best possible dependency tree given (i) the gold tree and (ii) a noisy ASR segmentation. This process is similar to the method used in [11] to transfer the gold annotation to the noisy ASR output. Such a computation requires knowing which tokens are added or missing. To do so, we use an alignment tool: Sclite.[3] Sclite uses the

[3]`https://github.com/usnistgov/SCTK`

transcriptions and the gold text to detect the added or missing tokens. We use the following heuristics to compute the best possible tree:

- Insertions are attached to the root (single root constraint).
- When a token is deleted, all the orphaned tokens are attached to the closest parent of the deleted tokens. Either the parent of the deleted token, or recursively until the root.
- In the case of a missing root, the closest element to the root becomes the new root. If there are several possibilities, the leftmost candidate is chosen.

Thus, when computing the loss in eq. (8), we substitute $\mathbf{p}$, $\mathbf{d}$ and $\mathbf{h}$ by the sequences computed by the oracle. The oracle is illustrated in Figure 3b built from the original tree in Figure 3a.

4. Experiments and result

Questions Our main objective is to assess whether using only the raw signal as input for the syntactic analysis task is feasible in a multitask setting where we simultaneously predict the transcription (ASR), POS tags, and the dependency tree.

Experimental settings We used a pretrained wav2vec2[17] architecture as our feature extractor f for the raw signal, namely `LeBenchmark/wav2vec2-FR-7K-large` [18] available online.[4] Before we train the syntax model, we wait for the ASR module to reach a WER lower than 50 on the dev set (usually, one epoch is enough). Since the syntax module depends on the segmentation of the ASR, training it from the very beginning (with low-quality segmentations) could confuse or slow down learning. The $\mathrm{FNN}_{\mathrm{enc}}$ function has 3 fully connected layers with 1024 units and uses Leaky ReLu activation function. The audio frame $\mathrm{LSTM}_{\mathrm{hidden}}$ that computes audio word embeddings has 500 hidden units and 2 layers. The sentence level bi-$\mathrm{LSTM}_{\mathrm{output}}$ has 800 hidden units and 2 layers. The implementation of our model uses the Speechbrain framework [19]. The loss function weight for each of the tasks is set to $\delta = 0.4$ for ASR (CTC) and $\alpha = \beta = \gamma = 0.2$ for all the sequence labeling tasks (dependency parsing task) based on preliminary experiments. Due to limited resources, we could not tune the value of the hyperparameters, except for the number of epochs (to minimize the validation set WER). Thus, our result might improve with a hyperparameter search. The end-to-end model was trained for approximately 135 hours on a Quadro RTX 8000 GPU. Training the pipeline basline was quicker, with around 100 hours on the same GPU. However, the end-to-end model has several unoptimized computations, such as the call to Sclite during the oracle computation.

Dataset We evaluate our model on the Orféo treebank [20], composed of multiple corpora of spoken French with [5] dependency tree annotations, aligned with audio files at the sentence level. For this paper, we did not consider any sentence containing a multi-word expression, to ensure that the segmentations are comparable for ASR and dependency parsing. The multiple corpora include many domains, most of which feature spontaneous conversations. Domains range from prepared French fairy tales narrated by storytellers (Oral-Narrative) to recordings at checkouts of cheese shops (Clapi) or interviews with Parisian people about their life in the city (Cfpp). The diversity of speaking situations, along with the lower quality of some recordings, makes Orféo harder than standard benchmarks for French ASR. The composition of the train, dev and test set is presented in Table 1.

Table 1: *Subcorpora in the Orféo treebank along with their types of speech and sizes in number of sentences and duration in hours.*

Corpus	Type	Train size	Dev size	Test size
Cfpb	spontaneous	3030 (2.2h)	389 (0.3h)	362 (0.3h)
Cfpp [21]	spontaneous	25500 (19.1h)	3175 (2.3h)	3232 (2.4h)
Clapi [22]	spontaneous	7682 (5.3h)	1001 (0.7h)	967 (0.7h)
Coralrom [23]	spontaneous	10889 (9.6h)	1342 (1.2h)	1376 (1.2h)
Crfp [24]	spontaneous	17357 (15.3h)	2198 (2.0h)	2259 (2.0h)
Fleuron [25]	spontaneous	1779 (1.4h)	207 (0.1h)	217 (0.2h)
Oral-Narrative [26]	prepared/read	8388 (7.3h)	1074(1.0h)	1050 (1h)
Ofrom [27]	spontaneous	11665 (9.3h)	1461 (1.2h)	1476 (1.2h)
Reunions	spontaneous	10067 (8.0h)	1283 (1h)	1245 (1h)
Tcof [28]	spontaneous	16063 (11.6h)	1971 (1.5h)	1997 (1.5h)
Tufs	spontaneous	35990 (24.3h)	4431 (3.0h)	4525 (3.0h)
Valibel	mixed	21095 (17.5h)	2769 (2.3h)	2753 (2.3h)
Total	mixed	169505 (130.9h)	21301 (16.6h)	21459 (16.8h)

Evaluation We compare wav2tree to two pipeline baselines that first perform ASR, and then parse the transcriptions. We first train an ASR model on Orféo (train section) and predict transcriptions for the test section. The pipelines use the same parsing algorithm as wav2tree (see Section 2.2), but the inputs to the parser are embeddings from a pretrained language model representing the predicted tokens. Both pipelines use `flaubert-large` [29] to compute the embedding representation of each word (last embedding of each word in the last layer). We avoid overfitting by saving the model with the best UAS score on the validation set. The first model is trained on gold transcriptions from the training set and tested on the noisy ASR test file. The second baseline, pipeline+oracle, is trained on ASR transcriptions of the train set, whose trees have been processed by the oracle defined in Section 3.2.

We compute all evaluation metrics with a customized version of the 2018 shared task evaluation script[6] based on the token order and not on the form of each word, to account for minor errors (e.g. spelling) in the ASR. The script is also based on Sclite alignment, and allows us to evaluate our results even when the word forms do not match between the gold and predicted trees. We use word error rate (WER) and Character error rate (CER) for ASR, and the traditional metrics for dependency parsing: UPOS (Universal Part of Speech) accuracy, UAS (unlabeled attachment score), and LAS (labeled attachment score).

Results For the first experiment, we train on all the subcorpora in Orféo and report the results in Table 2 broken down by subcorpus. The high WER on Orféo might be due to the type of speech; spontaneous speech is harder to recognize for ASR systems. Moreover, the recording conditions are of low quality for some corpora leading to very high WER in some cases (e.g. clapi). The baseline without the oracle has the worst results, showing that training on ASR output is important for the model and better matches the test settings. Wav2tree outperforms both baselines, showing that using only the signal as input for syntactic parsing is feasible. A notable fact is that this method naturally avoids the out-of-vocabulary problem since our model work on the character level for the ASR module, and our parser module only uses learnt representations extracted from a continuous span of audio representations. The only word-level information used is the boundary of the word. Thus, the only vocabulary needed for our model is the one used by the ASR. In this experiment, it is composed of the Latin alphabet and some diacritics.

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

[5] Around 5% of the Orféo treebank has been manually annotated (gold), the rest of the corpus is automatically annotated (good quality silver data). See the gitlab code release for more information on gold/silver breakdown statistics for each sub-corpus.

[6] `https://universaldependencies.org/conll18/evaluation.html`

Table 2: *Results by corpus with our model and baseline with and without the oracle. The model is trained on all the corpora and then tested on each of them. The model in the pipeline is dep2Label trained for 100 epochs.*

Parser input Pre-trained Parameters	**Wav2tree (end2end)** **Signal** **Wav2vec2** 350M+33M					**Pipeline** **Train: Gold - Test: ASR** **Wav2vec2 + FlauBERT** 350M+373M+32M					**Pipeline + oracle** **Train: ASR - Test: ASR** **Wav2vec2 + FlauBERT** 350M+373M+32M				
Corpus	WER	CER	UPOS	UAS	LAS	WER	CER	UPOS	UAS	LAS	WER	CER	UPOS	UAS	LAS
Cfpb	29.2	18.0	77.6	**73.4**	**68.8**	**28.2**	**17.0**	76.6	71.8	67.5	**28.2**	**17.0**	**78.0**	73.0	68.5
Cfpp	35.7	24.2	**71.8**	**66.6**	**61.8**	**35.5**	**24.2**	69.9	64.8	60.2	**35.5**	**24.2**	70.7	65.0	60.4
Clapi	53.6	35.4	**58.8**	**54.4**	**47.6**	**53.2**	**35.0**	56.4	53.5	46.7	**53.2**	**35.0**	57.4	53.75	47.0
Coralrom	22.5	11.9	**83.9**	**77.8**	**74.5**	**22.0**	**11.8**	82.0	75.3	71.8	**22.0**	**11.8**	83.0	75.9	72.4
Crfp	24.3	14.5	**81.7**	**75.9**	**72.2**	**23.5**	**14.3**	80.3	74.0	70.4	**23.5**	**14.3**	81.3	74.8	71.1
Fleuron	36.1	23.1	71.9	65.3	60.5	**35.5**	**21.6**	71.0	65.1	**61.3**	**35.5**	**21.6**	**72.1**	**66.0**	61.1
Oral-Narrative	11.1	4.7	**93.2**	**87.9**	**85.7**	**10.2**	**4.3**	92.1	86.1	83.4	**10.2**	**4.3**	93.0	86.6	84.2
Ofrom	20.0	11.6	**85.2**	**79.3**	**75.9**	**19.1**	**11.2**	84.2	77.9	74.7	**19.1**	**11.2**	85.1	78.4	75.1
Reunions	**40.9**	26.1	**67.7**	**61.8**	**56.3**	41.3	**26.0**	65.5	60.3	55.1	41.3	**26.0**	66.7	60.8	55.6
Tcof	34.1	20.8	**74.3**	**67.4**	**62.7**	**33.6**	**20.4**	72.3	65.4	60.8	**33.6**	**20.4**	73.2	65.6	61.0
Tufs	33.1	20.8	**75.2**	**69.6**	**65.1**	**32.5**	**20.3**	73.5	67.8	63.5	**32.5**	**20.3**	74.6	68.7	64.1
Valibel	23.0	12.8	**82.8**	**76.9**	**73.2**	**22.3**	**12.5**	81.3	75.4	71.7	**22.3**	**12.5**	82.2	75.7	71.8
Orféo full	31.0	19.4	**77.4**	**71.7**	**67.5**	**29.1**	**17.9**	75.8	70.0	65.8	**29.1**	**17.9**	76.7	70.5	66.2

Table 3: *Results with models trained only on the corpus with the best WER score: Oral-Narrative. The Pipeline model uses dep2Label and FlauBERT language model, and is trained for 100 epochs.*

Models	WER	UPOS	UAS	LAS
Wav2tree (end2end)	**10.9**	**93.18**	**86.60**	**84.31**
Pipeline	**10.9**	91.67	82.94	79.89
Pipeline + Oracle	**10.9**	91.77	83.46	80.66
Pipeline gold test	0	97.21	88.80	86.48

Moreover, the end-to-end model has around 350M parameters, whereas the pipelines use the same model for the ASR task and then use a pre-trained language model with 373M parameters (`flaubert-large`). The end-to-end model outperforms the pipeline while using less than half the parameters. The WER and the parsing results (UAS) are correlated (Pearson coefficient: -0.9648), meaning that noisy or hard-to-understand recordings are harder to parse. Thus, any improvement in the speech recognition process should have a positive effect on dependency parsing.

For the second experiment, we focus on parsing *only* the corpus with the best WER of Orféo, namely French-oral-narrative. We train our model for 50 epochs on this corpus and the pipeline for 100 epochs. This corpus features prepared speech (readings) unlike other Orféo subcorpora; in this case, the corpus is composed of storytellers narrating fairy tales. The objective of this experiment is to see if our model still outperforms the pipeline approach on less noisy corpora. The results, as seen in Table 3, show that wav2tree still outperforms the baseline, even on the corpus favorable to the pipeline approach where the risk of error propagation is lower. Another notable result is that the model only trained on one corpus had worse results than the one trained on all of Orféo for 30 epochs. This is probably due to the sheer size difference between the two train sets. We conclude that the model can leverage information from other corpora.

5. Conclusion

We introduced a new architecture able to parse speech directly from the raw audio signal. We have shown that it outperforms a pipeline approach working on transcribed speech with twice the number of parameters. This model can use any generic parsing algorithm on top of the audio word embedding extractor, allowing for high modularity. We have shown that using only the raw signal for the dependency parsing task is feasible, meaning that the raw signal contains syntactic information. For this purpose, we designed a method to create audio word embedding from the CTC whitespace prediction usable in any task requiring word-level segmentation such as named entity recognition (NER), or word sense disambiguation (WSD). However, further research on the performance of these audio word embeddings is required, for instance comparing them to a baseline using both word embedding and prosody-acoustics features.

Extending this method to use multimodality is a promising idea to improve performance. We could use the predicted text as input for any Bert model and combine the word embeddings from Bert and the audio word embeddings. We also plan to use a language model on top of the CTC loss and experiment with other ASR methods.

6. Acknowledgements

This project was funded by the **A**gence **N**ational de la **R**echerche (ANR) through the project **PROPICTO (ANR-20-CE93-0005)**. We thank Marco Dinarelli for feedback on an earlier version of this article.

7. References

[1] E. Kiperwasser and Y. Goldberg, "Simple and accurate dependency parsing using bidirectional LSTM feature representations," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 313–327, 2016. [Online]. Available: https://aclanthology.org/Q16-1023

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[3] F. Béchet and A. Nasr, "Robust dependency parsing for spoken language understanding of spontaneous speech," in *Proc. Interspeech 2009*, 2009, pp. 1039–1042.

[4] F. Béchet, A. Nasr, and B. Favre, "Adapting dependency parsing to spontaneous speech for open domain spoken language understand-

ing," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[5] P. J. Price, M. Ostendorf, S. Shattuck-Hufnagel, and C. Fong, "The use of prosody in syntactic disambiguation," *the Journal of the Acoustical Society of America*, vol. 90, no. 6, pp. 2956–2970, 1991.

[6] J. K. Pate and S. Goldwater, "Unsupervised dependency parsing with acoustic cues," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 63–74, 2013. [Online]. Available: https://aclanthology.org/Q13-1006

[7] T. Tran, S. Toshniwal, M. Bansal, K. Gimpel, K. Livescu, and M. Ostendorf, "Parsing speech: a neural approach to integrating lexical and acoustic-prosodic information," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 69–81. [Online]. Available: https://aclanthology.org/N18-1007

[8] E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001. [Online]. Available: https://aclanthology.org/N01-1016

[9] F. Jørgensen, "The effects of disfluency detection in parsing spoken language," in *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*. Tartu, Estonia: University of Tartu, Estonia, May 2007, pp. 240–244. [Online]. Available: https://aclanthology.org/W07-2435

[10] M. Honnibal and M. Johnson, "Joint incremental disfluency detection and dependency parsing," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 131–142, 2014. [Online]. Available: https://aclanthology.org/Q14-1011

[11] M. Yoshikawa, H. Shindo, and Y. Matsumoto, "Joint transition-based dependency parsing and disfluency detection for automatic speech recognition texts," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1036–1041.

[12] P. Jamshid Lou, Y. Wang, and M. Johnson, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2756–2765. [Online]. Available: https://aclanthology.org/N19-1282

[13] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2298–2304, 2017.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[15] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1764–1772. [Online]. Available: https://proceedings.mlr.press/v32/graves14.html

[16] M. Strzyz, D. Vilares, and C. Gómez-Rodríguez, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 717–723. [Online]. Available: https://aclanthology.org/N19-1077

[17] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[18] S. Evain, H. Nguyen, H. Le, M. Z. Boito, S. Mdhaffar, S. Alisamir, Z. Tong, N. A. Tomashenko, M. Dinarelli, T. Parcollet, A. Allauzen, Y. Estève, B. Lecouteux, F. Portet, S. Rossato, F. Ringeval, D. Schwab, and L. Besacier, "Lebenchmark: A reproducible framework for assessing self-supervised representation learning from speech," *ArXiv*, vol. abs/2104.11462, 2021.

[19] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[20] C. Benzitoun, J.-M. Debaisieux, and H.-J. Deulofeu, "Le projet orféo: un corpus d'étude pour le français contemporain," *Corpus*, no. 15, 2016.

[21] CLESTHIA, "Cfpp2000," 2018, ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr. [Online]. Available: https://hdl.handle.net/11403/cfpp2000/v1

[22] ICAR, "Clapi," 2017, ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr. [Online]. Available: https://hdl.handle.net/11403/clapi/v1

[23] E. Cresti, F. B. do Nascimento, A. M. Sandoval, J. Veronis, P. Martin, and K. Choukri, "The c-oral-rom corpus. a multilingual resource of spontaneous speech for romance languages," 2004, pp. 26–28.

[24] E. DELIC, S. Teston-Bonnard, and J. Véronis, "Présentation du corpus de référence du français parlé," *Recherches sur le français parlé*, vol. 18, pp. 11–42, 2004, equipe DELIC. [Online]. Available: https://halshs.archives-ouvertes.fr/halshs-01388193

[25] V. André, "Fleuron: Français langue Étrangère universitaire–ressources et outils numériques," 2016. [Online]. Available: https://fleuron.atilf.fr/index.php?lg=fr

[26] J. Carruthers, "French oral narrative corpus," 2013, commissioning Body / Publisher: Oxford Text Archive.

[27] A. Mathieu, B. Marie-José, C. Gilles, D. Federica, and J. L. Anne, "Corpus ofrom – corpus oral de français de suisse romande," (2012-2020), université de Neuchâtel. [Online]. Available: www.unine.ch/ofrom

[28] ATILF, "Tcof : Traitement de corpus oraux en français," 2020, ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr. [Online]. Available: https://hdl.handle.net/11403/tcof/v2.1

[29] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "FlauBERT: Unsupervised language model pre-training for French," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 2479–2490. [Online]. Available: https://aclanthology.org/2020.lrec-1.302