Multi-purpose neural network for French categorial grammars

Gaëtan Margueritte[1], **Koji Mineshima**[2], **Daisuke Bekki**[3]
[1]ENSEIRB-Matmeca Engineering School, [2]Keio University, [3]Ochanomizu University
gamargueritte@gmail.com, mineshima@abelard.flet.keio.ac.jp,
bekki@is.ocha.ac.jp

Abstract

Categorial grammar (CG) is a lexicalized grammar formalism that can be used to identify and extract the semantics of natural language sentences. However, despite being used actively to solve natural language understanding tasks such as natural language inference or recognizing textual entailment, most of the tools exploiting the capacities of CG are available in a limited set of languages. This paper proposes a first step toward developing a set of tools enabling the use of CG for the French language by proposing a neural network tailored for part-of-speech and type-logical-grammar supertagging, located at the frontier between computational linguistics and artificial intelligence. Experiments show that our model can compete with state-of-the art models while retaining a simple architecture.

1 Introduction

Categorial grammar (CG) is a formalism whose foundations come from Ajdukiewicz (1935) and Bar-Hillel (1953). From there, we can find two major lines of research that were created, namely, combinatory CG (CCG) (Steedman, 2000) and type-logical grammar (TLG) (Moortgat, 1997; Morrill, 1994) which itself can be divided into two subtheories, namely, displacement calculus (Morrill et al., 2011) and multi-modal CG (Moortgat, 1997). Other theories that build upon those theories also exist, such as hybrid TLCG (Kubota and Levine, 2020) and abstract CG (de Groote, 2001).

Using these syntactic theories offers knowledge about each word passed in an input sentence. Using the appropriate resources, the great amount of information provided by a *supertag* (Bangalore and Joshi, 1999) attributed to a given word in a sentence can be parsed efficiently to solve natural language understanding tasks such as natural language inference or recognizing textual entailment. This syntax-semantic interface can then be used by machines in order to answer various kinds of challenges, such as question answering and text summarization.

The continuous development of CCG and TLG led to the progressive appearance of several annotated corpora in various languages, such as German (Hockenmaier, 2006), Italian (Bos et al., 2009), Japanese (Uematsu et al., 2013), and of course English (Hockenmaier and Steedman, 2007). However, the number of treebanks and tools is very limited for the French language. Because CG has a close affinity to lambda calculus, logic, and natural deduction proofs, we are motivated to develop the current state-of-the-art in this field for the French language.

In this work, we propose a simple supertagger for part of speech (POS) and TLG tagging by exploiting the capacities of deep bidirectional encoder representation from transformers (BERT) (Devlin et al., 2018) for unlabeled input sentences. We demonstrate that integrating into our architecture a small long short-term memory (LSTM)-based variational autoencoder (VAE) while adapting the training pipeline allows us to increase the word-wise supertagging accuracy of our model. We also show experimentally that joining the training of both POS and TLG supertagging offers slightly increased overall accuracy while reducing the accuracy of tags seen rarely during training.

2 Related works

French TLG and POS supertagging The TLGbank (Moot, 2015) is a type-logical treebank for French, developed from the French Treebank, a lexical and syntactic resource by Abeillé et al. (2003). Because both corpora have been manually verified and rectified by their respective authors, they can be considered as the gold standard for French CG. Alongside his TLGbank, Moot presented the supertagger DeepGrail,[1] which is an LSTM layer that uses ELMo (embeddings from language models) vector embeddings of the unlabeled input data. This model successfully assigns 93.2 percent of words their correct TLG formula and presents an accuracy of 99.1 percent of correct POS supertags.

Since then, state-of-the-art TLG supertagging of this treebank has been achieved by Kogkalidis and Moortgat (2022) with an accuracy of 95.92 percent. Their approach revisits traditional models by proposing a framework based on heterogeneous dynamic graph convolutions and by decomposing the structure of the supertags. By doing so, they presented novel accuracy results on supertags that were rarely seen during the training phase. This generalization effort motivated us to explore different ways to regularize our architecture without losing overall model accuracy.

CamemBERT Our approach is built around the use of CamemBERT (Martin et al., 2020), which is a fine-tuned RoBERTa model (Liu et al., 2019) for French, which itself is based on BERT (Devlin et al., 2018). This model is attractive for the French language because it uses a subword tokenization where each word is divided, so it can exploit the numerous inflections that appear in the French language. In the study reported herein, we found only a few differences between the experimental results of CamemBERT$_{BASE}$ and CamemBERT$_{LARGE}$ models. Therefore, for the sake of computing speed and efficiency, we used only CamemBERT$_{BASE}$ in our model because its architecture is three times smaller than its other version.

3 TLG and POS supertagger model

In this section, we describe the training data and procedure and present the different modules of our model.

3.1 Training data

We manually split the TLGbank with a fixed seed into train/dev/test splits at a ratio of 80:10:10 to have comparable results with the network proposed by Kogkalidis and Moortgat (2022). For each word, the corpus presents its TLG and French POS supertags, allowing us to test several versions of our network using solely the 14,521 parsed sentences of the treebank (411,520 words).

Class	Frequency	Number of words
Frequent	$n \geq 100$	43,861
Uncommon	$100 > n \geq 10$	761
Rare	$10 > n \geq 1$	139
Unseen	$n = 0$	21

Table 1: Supertag classes statistics of the TLGbank.

CGs such as TLG often suffer from a large number of possible supertags. To evaluate the regularization power of our architecture, we group the tags into four classes based on their frequency of appearance in our training split. Table 1 shows the supertag class names, frequency of tags in the train split, and number of different words whose supertag is in this class.

Because POS supertags do not share the same sparsity as TLG supertags (<30 different tags for the French MElt POS tagset), we report only the overall accuracy on this task.

3.2 Model architecture

We develop each part of the model presented in Figure 1 before presenting how the different modules were combined and evaluated. For simplicity, we call the model VAEoTL (variational autoencoder over transfer learning).

CamemBERT CamemBERT was trained originally on the masked language modeling task. Thus, we fine-tune CamemBERT$_{BASE}$ during the training phase while only removing its original head in a classical transfer-learning fashion. For each phase of training described in Section 3.3, the learning rate of CamemBERT is 10 times lower than for the rest of the model in order not to waste its pre-training. CememBERT's subword tokenization requires us to adapt the output size. Because we attribute only one supertag per word (and not per subword), we adapt the training data by attributing the supertag to the first subpart of each word and by padding the other subparts. Accuracy is thus evaluated using a simple mask removing this padding.

BiLSTM A single-layered bi-directional LSTM (BiLSTM) is used after the CamemBERT layer. It is a recurrent network that combines two LSTMs: one reading the sentence from left to right, and one reading the sentence from right to left, thus extracting for each input information coming from

[1] https://richardmoot.github.io/DeepGrail/

its neighbors on both sides.

VAE With the goal of regularizing our network in mind, we tried to add a VAE to our architecture. This module allows us to approximate the output distribution of the BiLSTM by encoding it to a latent space, before decoding it to reconstruct the aforementioned outputs. Doing so allows us to regularize the BiLSTM outputs and to increase the supertagging accuracy, specifically over rare tags. Internally, the encoder and decoder of the VAE module are both composed of BiLSTM linked by dense layers to the latent space. In our case, a latent space of size 200 was the best compromise between speed and efficiency in the final model.

However, integrating this module requires adapting the training procedure because it requires the previous layers to be pre-trained. We differentiate our procedure into three distinct phases as described later in Section 3.3.

Dense+CRF heads The final output of our neural network is tagged by a simple dense layer mapping the hidden dimensions to tagset space in order to produce probability emission for each possible supertag. However, applying a simple softmax activation function to such emissions would imply that each tag is conditionally independent of its neighbor, which is in sharp contrast to the nature of CGs.

While the softmax activation allows us to distribute the probability for each supertag to be chosen given an input word, it sometimes fails to modelize the relationship between adjacent supertags. Instead, we use a conditional-random-field layer (Lafferty et al., 2001), a discriminative model that finds the Viterbi path maximizing the probability of a sequence of possible supertags given an input sequence. This effectively considers the context around each supertag while allowing us to use a simple forward-backward algorithm to compute the negative log-likelihood between network emissions and target outputs.

Two different heads are required because we want to evaluate both the TLG and French POS supertagging tasks. We experimented on two possible applications of this model: *single-headed* or *multi-headed*. In the former, we train only a single head at once, thus dedicating the whole architecture to a single task. In the latter, we share the training of the previous layers between each task, on the hypothesis that overall accuracy should improve because only the most relevant features will be learned, thereby effectively preventing overfitting.

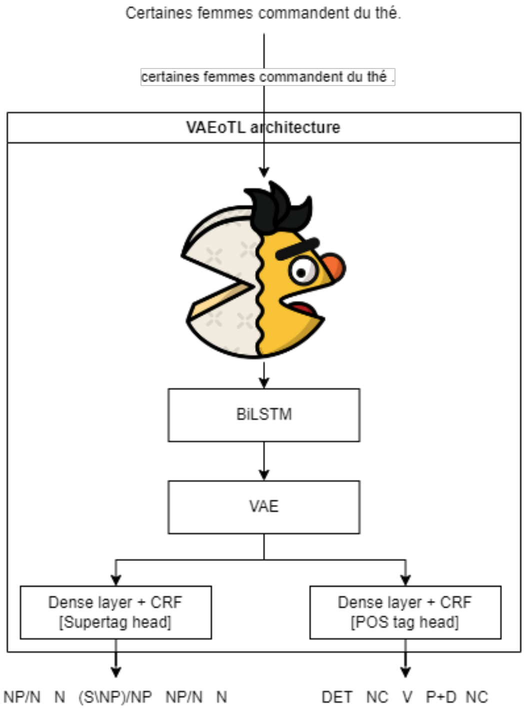

Figure 1: Architecture of the network

3.3 Training procedure

For its training, the VAE module requires an adapted negative log-likelihood with regularizer and to have its previous layers sufficiently trained. Accordingly, we define three distinct phases to our training. The first phase (20 epochs) does not use the VAE module at all, because we do not wish to approximate the outputs of an untrained model. In the second phase, we remove the heads of the model and freeze the training of CamemBERT and the BiLSTM layers in order to train the VAE for 10 epochs, using the mean squared error as a reconstruction criterion added to the Kullback–Leibler divergence in order to compute the loss. In the final phase, we unfreeze all layers and fine-tune the whole model for 10 epochs.

3.4 Implementation

We implement our model using PyTorch,[2] which provides an easy-to-use-and-adapt interface to construct our model, alongside Huggingface,[3] from which we accessed the CamemBERT model.

[2] https://pytorch.org/

[3] https://huggingface.co/

Model	Overall	Frequent	Uncommon	Rare	Unseen
ELMo & LSTM (Moot, 2015)[1]	93.20	95.10	75.19	25.85	0.0
Phase 1 Single-head	95.47	95.90	81.20	41.30	0.0
Phase 1 Multi-head	95.57	96.00	83.57	28.78	0.0
Final Single-head	95.58	96.00	81.20	45.19	0.0
Final Multi-head	95.66	96.13	83.04	28.78	0.0
HDC (Kogkalidis and Moortgat, 2022)[1]	95.92	96.40	81.48	55.37	7.26

Table 2: Model performance in percent for each category of tags (average over five runs). HDC stands for heterogeneous dynamic convolutions. [1]Reported results from the cited paper.

For each phase, we use a different Adam optimizer with $\beta = (0.9, 0.999)$, no weight decay, and a learning rate of 10^{-4} fading to zero with polynomial decay. To regularize the outputs, 40 percent dropout is added during training.

4 Results

In Table 2, we present the wordwise supertagging accuracy compared to the state-of-the-art results published by Kogkalidis and Moortgat (2022) in TLG supertagging. Although our model did not surpass the state of the art, we proved its efficiency despite its simplicity.[4] The first training phase is enough to reach high accuracy, but we observe that adding a VAE module still allows us to improve our accuracy, specifically over rare tags.

We observe that sharing the training between TLG and POS supertagging allows us to improve overall accuracy while sacrificing rare-tags accuracy. This is because the model will learn the underlying correlation between both types of supertags, thus reducing the probability of picking rare TLG supertags knowing the POS supertag of the same word.

Further investigations using this architecture are needed in future work to prove the efficiency of this model. However, its simple nature offers the opportunity to manipulate and adapt it easily, whether by modifying its structure or by simply adding new heads tailored to specific tasks.

Table 3 presents our results on the POS supertagging task compared to MElt tagger results reported by Denis and Sagot (2012). We observe that the model achieves state-of-the-art results, demonstrating that it can learn features relevant for both TLG and POS supertagging.

Model	Accuracy
MElt tagger (Denis and Sagot, 2012)	97.70
Phase 1 Single-head model	99.53
Phase 1 Multi-head model	99.57
Final Single-head VAEoTL	99.55
Final Multi-head VAEoTL	99.56

Table 3: Model performance in percent for French POS tagging on the TLGbank.

5 Contributions and limitations

With the goal in mind to provide a tool allowing to properly represent the syntax of input sentences formulated in natural language, we hope that future works will be able to extend the capacities of this architecture in order to exploit this syntax-semantic interface. While our model has not improved the state of the art of French TLG supertagging, it presents an accessible and simple fine-tuning of existing transformer-based models. Its modular architecture eases the adaptation of other existing techniques such as beam search to obtain more than a single prediction per word.

However, this model fails to modelize the internal structure of the syntactic types in the sense that it does not learn to create new composed types (N/N, S\NP) by assembling atomic types (N, NP, S). The current state of the art presented by Kogkalidis and Moortgat (2022) solves this problem by using a graph-theoretic perspective.

6 Conclusion

In this work, we investigated the different ways to regularize and fine-tune a supertagger for the French language, exploiting pre-trained unlabeled word embedding and a customized procedure utilizing a VAE architecture. We used a gold-standard annotated corpus, TLGbank, to train a simple and adaptable model able to compete with the current state of the art of supertaggers. We have shown experimentally that a VAE can be used

[4] The software used is available at the following github page for reproducibility of results: https://github.com/gaetanmarguerittte/ftlgsupertagger

to improve model regularization and that overall accuracy can be improved by using a multi-headed architecture.

Acknowledgments

We thank both anonymous reviewers for their helpful comments and suggestions. We also want to thank Michael Moortgat for presenting his work and for his insightful comments that allowed us to gain a novel point of view on the supertagging task, and Yuta Takahashi for his continuous support that made easier the researches that led to this paper. This work was partially supported by JST CREST Grant Number JPMJCR20D2, Japan.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. *Building a Treebank for French*, pages 165–187. Springer.

Kasimir Ajdukiewicz. 1935. Die syntaktische konnexität. *Studia Philosophica*, 1:1–27.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Comput. Linguist.*, 25(2):237–265.

Yehoshua Bar-Hillel. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29(1):47–58.

Johan Bos, Cristina Bosco, and Alessandro Mazzei. 2009. Converting a dependency treebank to a categorial grammar treebank for Italian. In *Eighth International Workshop on Treebanks and Linguistic Theories*.

Pascal Denis and Benoît Sagot. 2012. Coupling an annotated corpus and a lexicon for state-of-the-art pos tagging. *Language Resources and Evaluation*, 46(4):721–736.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding.

Philippe de Groote. 2001. Towards abstract categorial grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 252–259, Toulouse, France. Association for Computational Linguistics.

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 505–512, Sydney, Australia. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Konstantinos Kogkalidis and Michael Moortgat. 2022. Geometry-aware supertagging with heterogeneous dynamic convolutions.

Yusuke Kubota and Robert D Levine. 2020. *Type-Logical Syntax*. MIT Press.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

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.

Michael Moortgat. 1997. Categorial type logics. In *Handbook of Logic and Language*.

Richard Moot. 2015. A type-logical treebank for French. *Journal of Language Modelling*, 3(1).

Glyn Morrill. 1994. *Type Logical Grammar: Categorial Logic of Signs*. Springer.

Glyn Morrill, Oriol Valentín, and Mario Fadda. 2011. The displacement calculus. *Journal of Logic, Language and Information*, 20:1–48.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. 2013. Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1042–1051, Sofia, Bulgaria. Association for Computational Linguistics.