NLP Homework 2: Semantic Role Labeling

Marco Lo Pinto

1 Introduction

In the field of Natural Language Processing, Semantic Role Labeling is the task of automatically extract predicate-argument structure from a given sentence. The selected predicate defines an action/event with its arguments, each of them with a specific role. Recent progress have shown a significant performance gap between different languages, mainly because of the difference in the quantity of resources available (Conia and Navigli, 2020).
The goal of this paper is to automatically recognize and disambiguate the predicates in a sentence, with their respective arguments correctly identified and classified, using the UniteD-SRL dataset (Tripodi et al., 2021). Moreover, cross-lingual performance operations are used, in order to improve results for French and Spanish (which contain a small subset of the English training set). First tests were done on the English training/dev dataset.

2 Argument identification and classification

2.1 Dataset preprocessing

Because there could be more than one predicate in a sentence, multiple approaches can be found to deal with this problem. In order to simplify the model and the training part, the most straightforward solution is to replicate the sentence for each predicate-argument pair. The labels for this project (for both predicates and roles) were automatically extracted from the VerbAtlas (Di Fabio et al., 2019) documentation and then cross-validated using the datasets and the baselines file to check for missing labels. The distribution of roles for each dataset (figure 1) is unbalanced: this is because agent, theme and patient are the most recurrent labels in common phrases (e.g. ”I ate an apple”: ”I” is agent and ”apple” is patient).

2.2 Choice of the Embedding

As already stated in my previous homework (Lo Pinto, 2022), one of the most important parts of a model is the choice of the embedding. For the sake of demonstrability, the first test for a possible implementation in this paper was made by extracting and using embeddings from GloVe (Pennington et al., 2014) with a vocabulary of 400’000 words (excluding <pad> and <unk> special tokens), then a Bi-LSTM composed of 3 layers and hidden size of 256 and finally a fully-connected layer. The Bi-LSTM processed the embedding output concatenated with a one-hot encoding of the predicates labels and a flag denoting which vector is the target predicate (for multiple predicates in the sentence, it needs to be passed multiple times in order to obtain the roles for each of them). As opposite, a pre-trained version of BERT (Devlin et al., 2019) was used in place of GloVe embedding, thanks to the huggingface repository (Wolf et al., 2020). After some training for both models (figure 2), the latter implementation provided better results, as shown in table 2.

2.3 BERT-based approach

Because of the superior results given by the pre-trained version of BERT, different configurations and approaches were tried with the transformer part in order to improve results. Therefore, instead of using it just as a fixed pre-trained embedding, it was fine-tuned along with a fully connected layer. More precisely, the last four layers from BERT were summed and passed to the classification layer: this improved the results. The way the text was encoded and passed onto it was inspired from another paper (Shi and Lin, 2019): the input is represented as [[CLS] sentence [SEP] predicate [SEP]], in order to focus the attention mechanism on the target predicate in the sentence. Because of how huggingface tokenizer for BERT encodes the words (generating subwords), two different approaches were tried for the transformer output: the former was to consider only the first vector representation of a word composed as multiple subwords, while the latter was to use the first vector but computed as the mean of all the subword vectors for that particular word, using peculiar tensor computations. Both solutions generated the same results.

3 Predicate identification and disambiguation

3.1 Dataset preprocessing

In this part, the data was not split in multiple sentences for each predicate because the model needs to classify each word with a respective predicate (if the word is identified as a predicate).

3.2 BERT-based approach

Different tests were done in order to determine which information could be useful for the model. Starting from a similar architecture chosen for the argument identification and classification part, the model was fine-tuned in order to do both parts. The input words are represented as [[CLS] sentence [SEP]]. Then, in order to improve the disambiguation part, the model was modified so to receive as optional input the result from the identification part. This improved the final f1 score for the disambiguation part of around 5%. Other tests were done in order to enrich the model with syntactic information about the input sentence, such as using Part-of-Speech. Unfortunately, the increment was minimal, so the PoS part was removed from the final implementation.

4 Cross-lingual implementation

All tests and results reported so far were done using only the English dataset part. In order to generate good results for the limited training datasets of French and Spanish, clever solutions needs to be found. Because BERT is a transformer model pre-trained on a large corpus of English data, another type of transformer was needed. The final choice is XLM-RoBERTa (Conneau et al., 2019), a model pre-trained on 2.5TB of filtered data containing 100 languages (including French and Spanish). A first fine-tuning done only on English showed an increment on the f1 score w.r.t BERT-based models (but doubling the number of trainable parameters). For the final model implementation, two main approaches have been hypothesized: one in which the weights for the model would work for all languages (language independent) and another in which the model would load the weights fine-tuned for the selected language. The latter strategy was implemented.

4.1 Deep transfer learning

In the first test, the model was trained only with the Spanish dataset, in order to see the performances of the model with little data, generating very poor performances. After that, the training was done with all the datasets of the three different languages unified as one, generating an f1 score of argument identification and classification of 0.8730 and 0.7482 respectively (the score was computed using all three dev datasets). In order to not lose performances in the English part, a different strategy was tried: instead of fine-tuning the model with all the datasets at once, the fine-tuned version made for the English dataset was used and trained for only another language. Different parts of the model were freezed and other left to train. The best performances were obtained by fine-tuning all layers of the model. This generated a resulting f1 score of argument identification and classification of 0.8673 and 0.7138 respectively for the Spanish dataset and of 0.8570 and 0.7006 respectively for the French dataset.

4.2 Silver data creation

Given the fact that the datasets used in this project (Tripodi et al., 2021) have been created by using VerbAtlas (Di Fabio et al., 2019), the latter can be exploited in order to create high-quality automatic annotations on top of an unannotated corpus (for example, using part of Wikipedia). In order to evaluate the newly generated data w.r.t existing one, the sentences from the English training data were used and performances compared (using the original dev dataset and a BERT-based model for argument identification and classification. As we can see from figure 3, there is a significant f1 score difference for the argument identification part (0.8951 in the original versus 0.8544 in the generated) and a much higher difference in the classification part (0.8356 in the original versus 0.6988 in the generated). It can be assumed that the error was not in the generated data per se, but mostly due to the fact that both datasets contains different annotations (and errors) for the same sentence, as is evident from

figure 4. Same tests were done on the predicate identification and disambiguation part and, as we can see from figure 5, there is a great f1 score difference only in the predicate disambiguation part (0.8347 in the original versus 0.6418 in the generated), whereas for the predicate identification part it was relatively minimal (0.9422 in the original versus 0.9147 in the generated).
Nonetheless, an attempt to use big unnanotated corpus was done: starting from the Wikipedia dumps for each language in the dataset (Wikimedia) they were extracted, cleaned up and finally each phrase was preprocessed and saved. Then, by using the same approach described before, samples were generated and passed onto the networks to be trained. As expected from the preliminary results with the original English dataset, deep transfer learning remain the best option. An attempt of combining both techniques degraded the performances of the pretrained model, as we can see from table 3.
A possible observation on using Wikipedia dumps and its scarse performances is that the model could specialize to a specific domain if the sentences are not accurately selected. Even if this was the case, the model still performed worse with the original dataset w.r.t the autogenerated one with the same sentences, so the strategy of silver data creation was abandoned, otherwise a wise choice would be to use the UN Parallel Corpus (Ziemski et al., 2016) (because the UniteD-SRL is based on that).

5 Final Model

The final model was designed with modularity in mind: each part can be used separately or jointly with the others. The flow of the data for each possible input is reported in figure 9. The final scores for each part in the English dev set can be found in table 4, 5, 6, 7, 8, 9, 10, 11 and 12. For the Spanish dev set, they are in table 13, 14, 15, 16, 17, 18, 19, 20 and 21. For the French dev set, they are in table 22, 23, 24, 25, 26, 27, 28, 29 and 30.

6 Results and Conclusions

The confusion matrices for each language (that can be found on figure 6, 7 and 8) shows that the model perform better if evaluated on English sentences rather than Spanish or French. Moreover, some labels that are present in the VerbAtlas documentation are never encountered from train and/or dev datasets. Different approaches for cross-lingual implementation were tried, starting from simple fine-tuning on the existing training data, then by using deep transfer learning and also by generating silver data, highlighting pros and cons for each technique.

References

Simone Conia and Roberto Navigli. 2020. Bridging the gap in multilingual Semantic Role Labeling: A language-agnostic approach. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 627–637, Hong Kong, China. Association for Computational Linguistics.

Marco Lo Pinto. 2022. NLP homework 1: Named Entity Recognition.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Rocco Tripodi, Simone Conia, and Roberto Navigli. 2021. UniteD-SRL: A unified dataset for span- and dependency-based multilingual and cross-lingual Semantic Role Labeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2293–2305, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wikimedia. Wikimedia Downloads. Accessed 2022-07-05.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

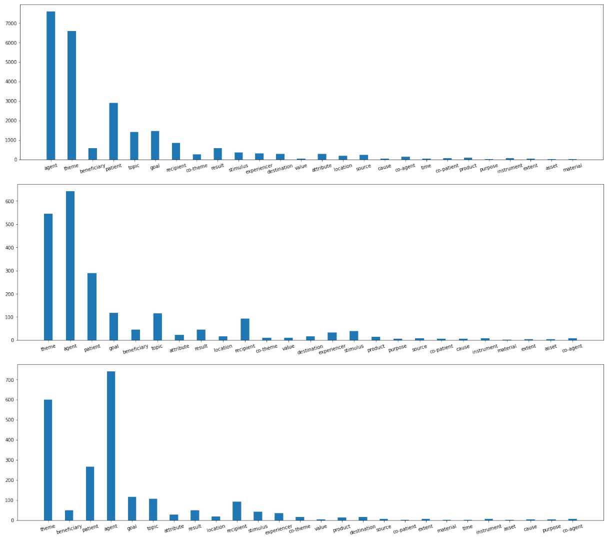


Figure 1: The roles labels distribution for each dataset: English (top), Spanish (middle) and French (bottom). Note: tags that are not present in the dataset are not shown in the distribution.

Model argument part (EN)	**F1-iden**	**F1-class**
GloVe* + Bi-LSTM + 2fc(s)	0.8006	0.6845
BERT* + Bi-LSTM + 2fc(s)	0.8618	0.7229
BERT* + 2fc(s)	0.5890	0.4825
BERT + Bi-LSTM + 2fc(s)	0.8938	0.8294
BERT + fc	0.8951	0.8356
XLM-RoBERTa + fc	0.8983	0.8427

Table 1: Summary of architectures and their f1 scores on the dev set for argument identification and classification in the English dev dataset (predicate identification and disambiguation were skipped). The "*" indicates that the component was freezed during training. The number before the fully-connected part indicates how many layers were used.

Model predicate part (EN)	**F1-iden**	**F1-disam**
BERT + fc	0.9422	0.8347
BERT + fc (no iden)	- (1.0)	0.8817
BERT + fc (+ PoS info)	0.9436	0.8377
XLM-RoBERTa + fc	0.9422	0.8411
XLM-RoBERTa + fc (no iden)	- (1.0)	0.8950

Table 2: Summary of architectures and their f1 scores on the dev set for predicate identification and diambiguation in the English dev dataset. The "-" indicates that the model didn't do the identification part (informations already computed).

ES Model training with	**F1-iden**	**F1-class**
original dataset only	<0.100	<0.100
transfer learning	0.8673	0.7138
wikidump	0.7148	0.4718
transfer learning + wikidump	0.7318	0.5056

Table 3: Summary of different types of trainings for the same network and their f1 scores on the dev set for argument identification and classification in the Spanish dev dataset (predicate identification and disambiguation were skipped).

ARGUMENT IDENTIFICATION (EN, model34)

	Gold Positive	Gold Negative
Pred Positive	4538	519
Pred Negative	475	

Precision	0.8974
Recall	0.9052
F1 score	0.9013

Table 4: Argument identification evaluation for the argument identification + argument classification model.

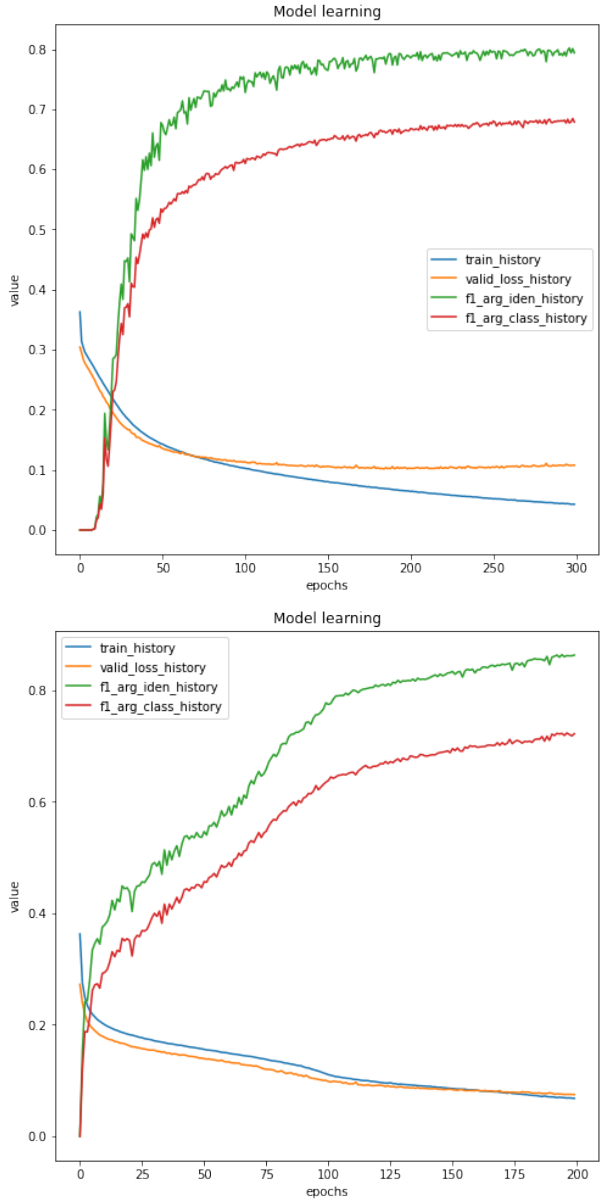


Figure 2: The training history of GloVe* + Bi-LSTM + fc (top) and BERT* + Bi-LSTM + fc (bottom).

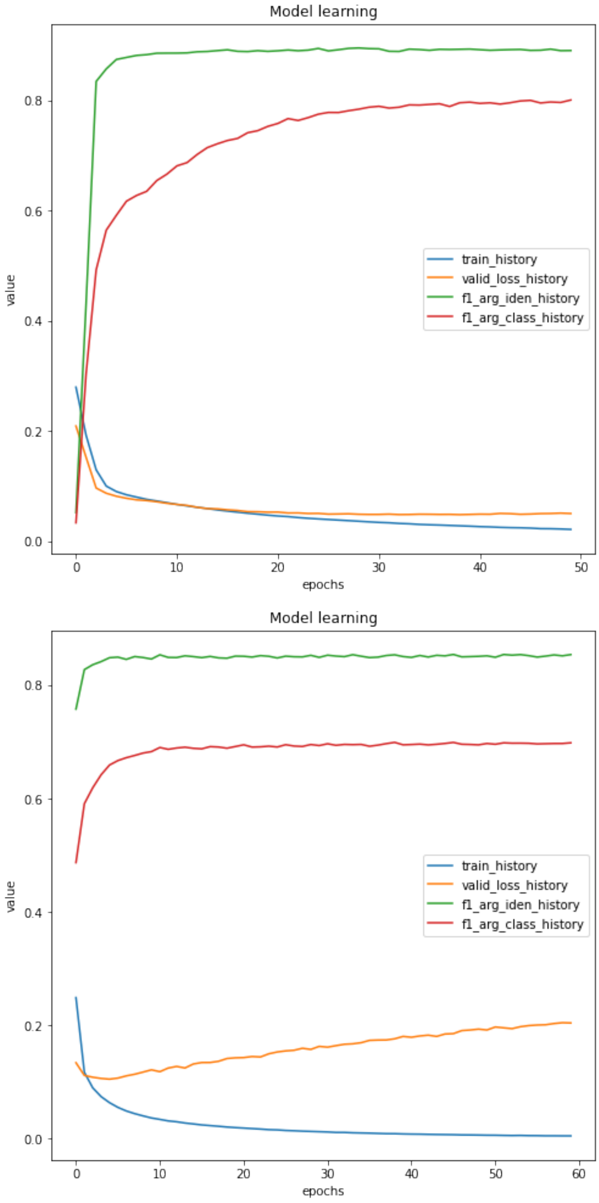


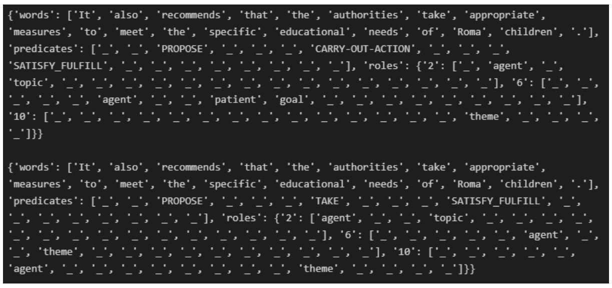
Figure 3: The same BERT-based model but trained with the original data (top) and with an autogenerated one (bottom).

ARGUMENT CLASSIFICATION (EN, model34)

	Gold Positive	Gold Negative
Pred Positive	4236	821
Pred Negative	777	

Precision	0.8377
Recall	0.8450
F1 score	0.8413

Table 5: Argument classification evaluation for the argument identification + argument classification model.



```
{'words': ['It', 'also', 'recommends', 'that', 'the', 'authorities', 'take', 'appropriate',
'measures', 'to', 'meet', 'the', 'specific', 'educational', 'needs', 'of', 'Roma', 'children', '.'],
'predicates': ['_', '_', 'PROPOSE', '_', '_', '_', 'CARRY-OUT-ACTION', '_', '_', '_',
'SATISFY_FULFILL', '_', '_', '_', '_', '_', '_', '_', '_'], 'roles': {'2': ['_', 'agent', '_',
'topic', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_'], '6': ['_', '_',
'_', '_', '_', 'agent', '_', '_', 'patient', 'goal', '_', '_', '_', '_', '_', '_', '_', '_', '_'],
'10': ['_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_', 'theme', '_', '_', '_',
'_']}}

{'words': ['It', 'also', 'recommends', 'that', 'the', 'authorities', 'take', 'appropriate',
'measures', 'to', 'meet', 'the', 'specific', 'educational', 'needs', 'of', 'Roma', 'children', '.'],
'predicates': ['_', '_', 'PROPOSE', '_', '_', '_', 'TAKE', '_', '_', '_', 'SATISFY_FULFILL', '_',
'_', '_', '_', '_', '_', '_', '_'], 'roles': {'2': ['agent', '_', '_', 'topic', '_', '_', '_', '_',
'_', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_'], '6': ['_', '_', '_', '_', '_', 'agent', '_',
'_', 'theme', '_', '_', '_', '_', '_', '_', '_', '_', '_', '_'], '10': ['_', '_', '_', '_', '_',
'agent', '_', '_', '_', '_', '_', '_', '_', '_', 'theme', '_', '_', '_', '_']}}
```

Figure 4: Data sample differences from original (top) and generated (bottom) for argument identification and classification.

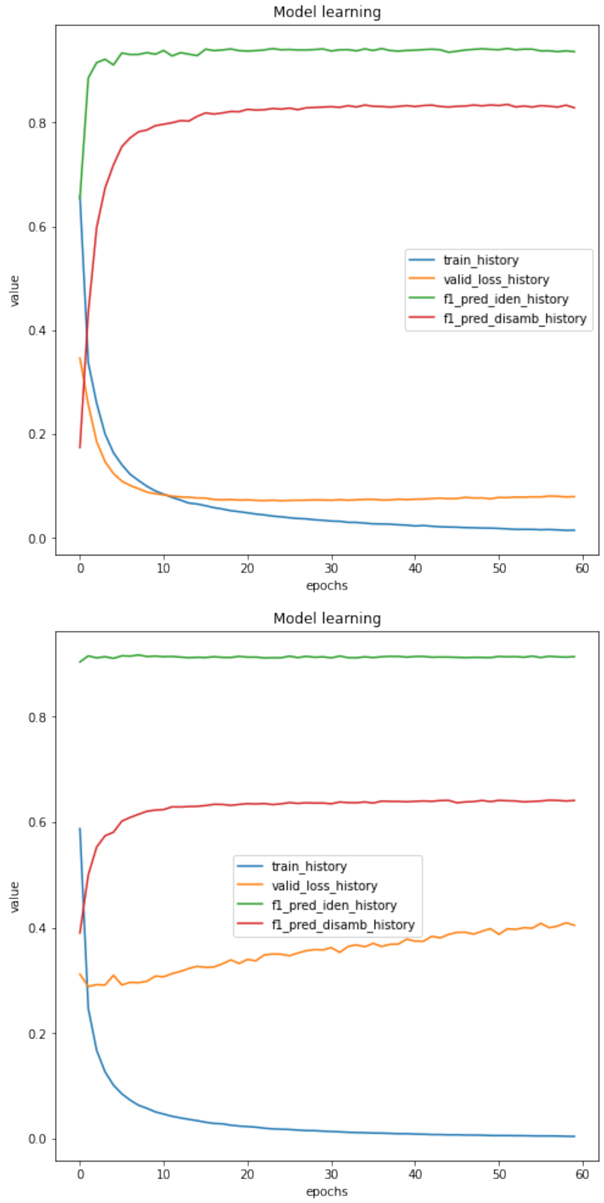


Figure 5: The same BERT-based model but trained with the original data (top) and with an autogenerated one (bottom) for predicate identification and disambiguation.

PREDICATE DISAMBIGUATION (EN, model234)

	Gold Positive	Gold Negative
Pred Positive	2285	268
Pred Negative	268	

Precision	0.8950
Recall	0.8950
F1 score	0.8950

Table 6: Predicate disambiguation evaluation for the predicate disambiguation + argument identification + argument classification model.

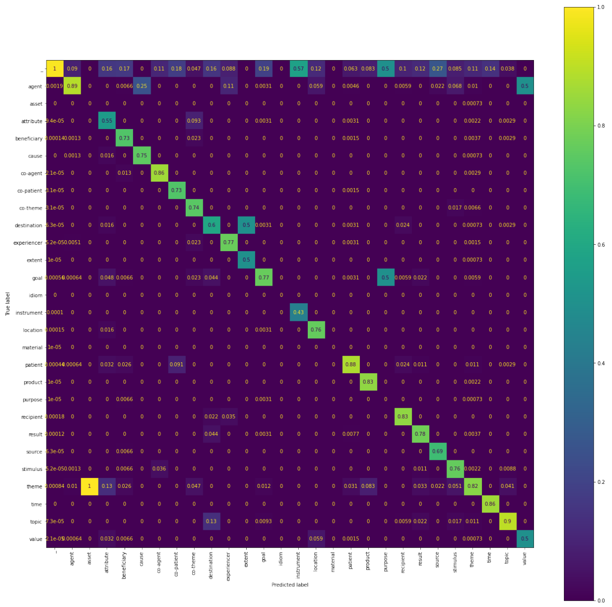


Figure 6: Confusion matrix for argument identification + argument classification evaluated on the English dev dataset (normalized over predictions).

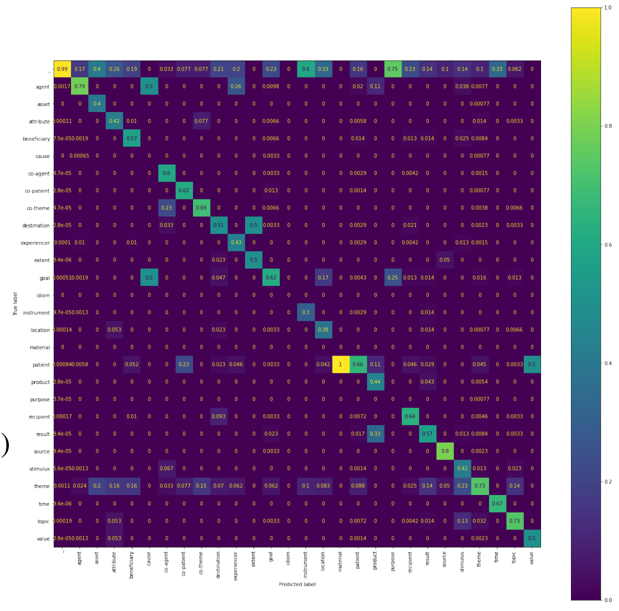


Figure 7: Confusion matrix for argument identification + argument classification evaluated on the Spanish dev dataset (normalized over predictions).

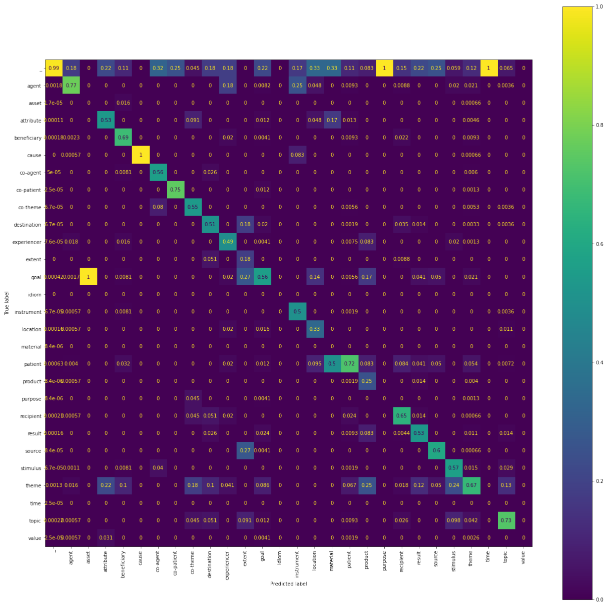


Figure 8: Confusion matrix for argument identification + argument classification evaluated on the French dev dataset (normalized over predictions).

ARGUMENT IDENTIFICATION (EN, model234)

	Gold Positive	Gold Negative
Pred Positive	4538	519
Pred Negative	475	

Precision	0.8974
Recall	0.9052
F1 score	0.9013

Table 7: Argument identification evaluation for the predicate disambiguation + argument identification + argument classification model.

ARGUMENT CLASSIFICATION (EN, model234)

	Gold Positive	Gold Negative
Pred Positive	4236	821
Pred Negative	777	

Precision	0.8377
Recall	0.8450
F1 score	0.8413

Table 8: Argument classification evaluation for the predicate disambiguation + argument identification + argument classification model.

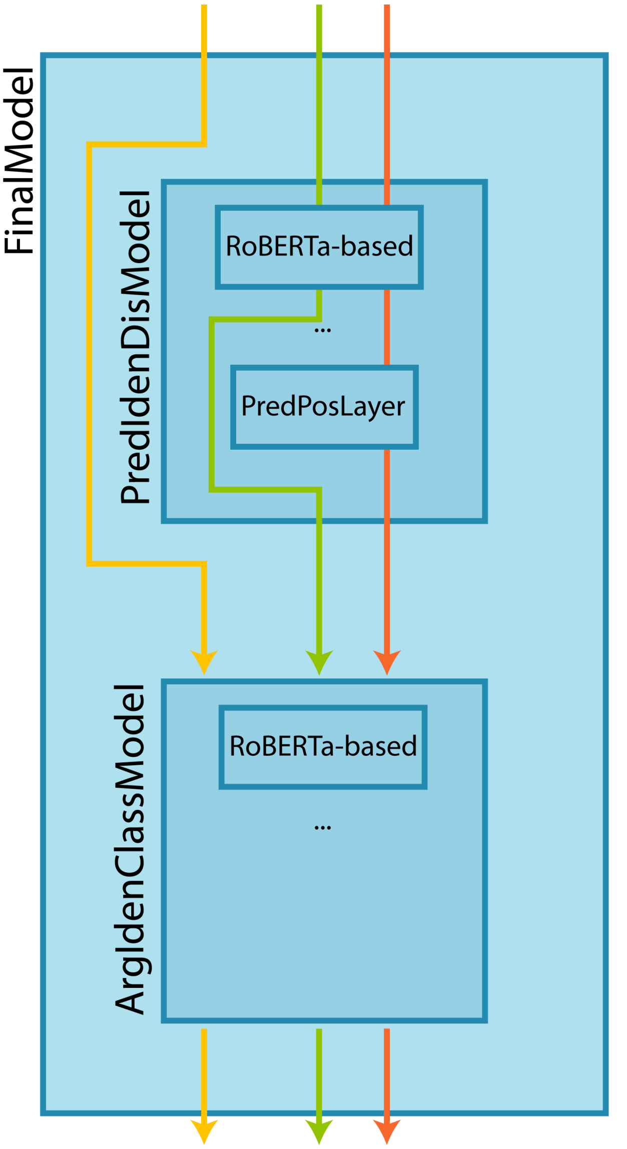


Figure 9: Final model flow. Orange = argument identification + argument classification; red = predicate disambiguation + argument identification + argument classification; green = predicate identification + predicate disambiguation + argument identification + argument classification

PREDICATE IDENTIFICATION (EN, model1234)

	Gold Positive	Gold Negative
Pred Positive	2460	209
Pred Negative	93	

Precision	0.9217
Recall	0.9636
F1 score	0.9422

Table 9: Predicate identification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

PREDICATE DISAMBIGUATION (EN, model1234)

	Gold Positive	Gold Negative
Pred Positive	2196	473
Pred Negative	357	

Precision	0.8228
Recall	0.8602
F1 score	0.8411

Table 10: Predicate disambiguation evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

ARGUMENT IDENTIFICATION (EN, model1234)

	Gold Positive	Gold Negative
Pred Positive	4416	849
Pred Negative	597	

Precision	0.8387
Recall	0.8809
F1 score	0.8593

Table 11: Argument identification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

ARGUMENT CLASSIFICATION (EN, model1234)

	Gold Positive	Gold Negative
Pred Positive	4143	1122
Pred Negative	870	

Precision	0.7869
Recall	0.8265
F1 score	0.8062

Table 12: Argument classification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

ARGUMENT IDENTIFICATION (ES, model34)

	Gold Positive	Gold Negative
Pred Positive	4161	732
Pred Negative	589	

Precision	0.8504
Recall	0.8760
F1 score	0.8630

Table 13: Argument identification evaluation for the argument identification + argument classification model.

ARGUMENT CLASSIFICATION (ES, model34)

	Gold Positive	Gold Negative
Pred Positive	3435	1458
Pred Negative	1315	

Precision	0.7020
Recall	0.7232
F1 score	0.7124

Table 14: Argument classification evaluation for the argument identification + argument classification model.

PREDICATE DISAMBIGUATION (ES, model234)

	Gold Positive	Gold Negative
Pred Positive	1678	817
Pred Negative	817	

Precision	0.6725
Recall	0.6725
F1 score	0.6725

Table 15: Predicate disambiguation evaluation for the predicate disambiguation + argument identification + argument classification model.

ARGUMENT IDENTIFICATION (ES, model234)

	Gold Positive	Gold Negative
Pred Positive	4161	732
Pred Negative	589	

Precision	0.8504
Recall	0.8760
F1 score	0.8630

Table 16: Argument identification evaluation for the predicate disambiguation + argument identification + argument classification model.

ARGUMENT CLASSIFICATION (ES, model234)

	Gold Positive	Gold Negative
Pred Positive	3435	1458
Pred Negative	1315	

Precision	0.7020
Recall	0.7232
F1 score	0.7124

Table 17: Argument classification evaluation for the predicate disambiguation + argument identification + argument classification model.

PREDICATE IDENTIFICATION (ES, model1234)

	Gold Positive	Gold Negative
Pred Positive	2443	42
Pred Negative	52	

Precision	0.9831
Recall	0.9792
F1 score	0.9811

Table 18: Predicate identification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

PREDICATE DISAMBIGUATION (ES, model1234)

	Gold Positive	Gold Negative
Pred Positive	1658	827
Pred Negative	837	

Precision	0.6672
Recall	0.6645
F1 score	0.6659

Table 19: Predicate disambiguation evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

ARGUMENT IDENTIFICATION (ES, model1234)

	Gold Positive	Gold Negative
Pred Positive	4099	786
Pred Negative	651	

Precision	0.8391
Recall	0.8629
F1 score	0.8509

Table 20: Argument identification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

ARGUMENT CLASSIFICATION (ES, model1234)

	Gold Positive	Gold Negative
Pred Positive	3393	1492
Pred Negative	1357	

Precision	0.6946
Recall	0.7143
F1 score	0.7043

Table 21: Argument classification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

ARGUMENT IDENTIFICATION (FR, model34)

	Gold Positive	Gold Negative
Pred Positive	4315	752
Pred Negative	689	

Precision	0.8516
Recall	0.8623
F1 score	0.8569

Table 22: Argument identification evaluation for the argument identification + argument classification model.

ARGUMENT CLASSIFICATION (FR, model34)

	Gold Positive	Gold Negative
Pred Positive	3527	1540
Pred Negative	1477	

Precision	0.6961
Recall	0.7048
F1 score	0.7004

Table 23: Argument classification evaluation for the argument identification + argument classification model.

PREDICATE DISAMBIGUATION (FR, model234)

	Gold Positive	Gold Negative
Pred Positive	1617	938
Pred Negative	938	

Precision	0.6329
Recall	0.6329
F1 score	0.6329

Table 24: Predicate disambiguation evaluation for the predicate disambiguation + argument identification + argument classification model.

ARGUMENT IDENTIFICATION (FR, model234)

	Gold Positive	Gold Negative
Pred Positive	4315	752
Pred Negative	689	

Precision	0.8516
Recall	0.8623
F1 score	0.8569

Table 25: Argument identification evaluation for the predicate disambiguation + argument identification + argument classification model.

ARGUMENT CLASSIFICATION (FR, model234)

	Gold Positive	Gold Negative
Pred Positive	3527	1540
Pred Negative	1477	

Precision	0.6961
Recall	0.7048
F1 score	0.7004

Table 26: Argument classification evaluation for the predicate disambiguation + argument identification + argument classification model.

PREDICATE IDENTIFICATION (FR, model1234)

	Gold Positive	Gold Negative
Pred Positive	2474	103
Pred Negative	81	

Precision	0.9600
Recall	0.9683
F1 score	0.9641

Table 27: Predicate identification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

PREDICATE DISAMBIGUATION (FR, model1234)

	Gold Positive	Gold Negative
Pred Positive	1560	1017
Pred Negative	995	

Precision	0.6054
Recall	0.6106
F1 score	0.6080

Table 28: Predicate disambiguation evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

ARGUMENT IDENTIFICATION (FR, model1234)

	Gold Positive	Gold Negative
Pred Positive	4211	891
Pred Negative	793	

Precision	0.8254
Recall	0.8415
F1 score	0.8334

Table 29: Argument identification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.

ARGUMENT CLASSIFICATION (FR, model1234)

	Gold Positive	Gold Negative
Pred Positive	3451	1651
Pred Negative	1553	

Precision	0.6764
Recall	0.6896
F1 score	0.6830

Table 30: Argument classification evaluation for the predicate identification + predicate disambiguation + argument identification + argument classification model.