Combining Different Parsers and Datasets for CAPITEL UD Parsing

Fernando Sánchez-León[a]

[a]*Independent Academic*

Abstract

This paper describes our experiments on Universal Dependency parsing of a subset of CAPITEL article news corpus, prepared for a competition within IberLEF 2020 Evaluation Forum. Several data-driven systems, using different technologies, are used for the task. Besides the training dataset provided by the organizers, we augment the training set using the training partition from other widely used UD-parsed Spanish corpus —AnCora. On top of this combination of toolkits and corpora, a voting strategy is proposed, where best scoring system predictions are combined to boost final performance. This combined model ranked first in the above-mentioned competition.

Keywords

Universal Dependencies, parsing, Spanish, News articles corpus, CAPITEL, AnCora, data augmentation, parser output combination

1. Introduction

Since the first attempt to build a Universal Dependency Treebank [1], the interest in refining the original proposal, define a universal part-of-speech tagset, and, most prominently, produce corpora for new languages (and/or improve the annotation of the existing ones) and implement multilingual dependency parsers has grown over time.

Universal Dependencies (UD), as a project, has now[1] 163 treebanks, covering 92 languages. The UD family has grown with a new offspring, since the Spanish Government PlanTL for the Advancement of Language Technology, as part of its open R+D lines, has developed CAPITEL, a linguistically annotated corpus of Spanish news articles, from which a small subset[2], syntactically annotated using UD v2, has been manually revised for the purpose of the CAPITEL-UD competition [2].

This paper describes our participation in such competition, where 12 participants were involved. Unfortunately, only two of them finally submitted their results. One of our prediction combinations ranked first in the official scoring table with a LAS score of 88.660, while the second system obtained 88.600 for the same metrics.

Section 2 describes our experiments with different data-driven UD parsing implementations as well as the voting strategy adopted. Results of our submitted systems or system combinations besides a preliminary discussion of them and other issues regarding UD are presented in section 3.

Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)

EMAIL: f.sanchez.lcmcvp@gmail.com (F. Sánchez-León)
ORCID:

CEUR Workshop Proceedings (CEUR-WS.org)

[1] As of May 15, 2020.

[2] "Small subset" should be interpreted here as compared to the size of the whole CAPITEL corpus —that will be several orders of magnitude larger— and not to the mean size of any other corpus parsed for UD.

2. Approach

Rather than developing another UD parsing system, we have opted for the (re)use of well established UD parsers for the estimation of UD parsing models from the CAPITEL-UD datasets. There were a number of reasons for this decision, ranging from the number of existing systems capable of performing UD analysis to our desire to experiment with the combination of various prediction systems, as it is explained below.

The final decision was to use at least three different toolkits to the UD parsing problem, not necessarily the best scoring ones, from the last 2-3 years shared tasks on this issue. The selection of systems was also performed based on criteria of ease of set up and configuration, as well as maturity (and clarity) of online documentation of each candidate system. Finally, the three solutions that have been tested are `UDPipe`, `NLP-Cube` and Stanza. Our experiments with them are described in the rest of this section.

2.1. UDPipe

`UDPipe` [3] is a trainable pipeline for tokenization, tagging, lemmatization and dependency parsing of CoNLL-U files. Program version used in this experiment, 1.2, ranked 8th in *CoNLL 2017 SharedTask: Multilingual Parsing from Raw Text to Universal Dependencies.*[3]

Only the dependency parsing module has been trained, since CAPITEL-UD datasets provide tokenization, tagging and lemmatization for all data distributed to challenge participants. Datasets are provided in CoNLL-U format.

Although its main author has developed a `UDPipe` 2.0 Prototype the CoNLL 2018 UD Shared Task [4] (known as `UDPipe-Future`),[4] we wanted to give a try to the original `UDPipe` implementation, and consider it as our baseline system for the current challenge. `UDPipe` 1.2 uses word embeddings but its implementation is previous to the (now generalized) contextualized word embeddings available in, for instance, `UDPipe` 2.0.

In our experimentation with `UDPipe`, we have used the train dataset with random search for hyperparameters (using the `run=number` option). Besides, with all three packages used, a ten-fold cross validation using training material has been performed, using this to select the model with best performance.

The only remarkable tuning in our experiments with respect to the original setup has to do with form and lemma embeddings. `UDPipe` authors pre-compute form embeddings with `word2vec` using the training data. All other embeddings used by the system are initialized randomly and updated during training. We rather have also used form and lemma embeddings computed from a 500M word corpus of newspaper text we have compiled and POS tagged. Besides, we have used `fastText` word embeddings trained on *Common Crawl* and *Wikipedia* corpora.[5]

Table 1 shows results on development set of a model built with the training material using different word embeddings.[6]

Increasing the number of iterations on the training set above 20 (the value recommended by `UDPipe` authors) does not improve any of the models. Our results for LAS are on a par with those obtained by `UDPipe` developers on UD 2.3 AnCora corpus [5], whose LAS score is 84.6 for raw text, but using

[3] `http://universaldependencies.org/conll17/results.html`.

[4] Available at `https://github.com/CoNLL-UD-2018/UDPipe-Future`.

[5] Available at `https://fasttext.cc/docs/en/crawl-vectors.html`.

[6] The rest of the parameters are those suggested by `UDPipe` developers. We refer to validation dataset as `dev`.

Table 1
Results on dev dataset using UDPipe with different embeddings and iterations

Iterations	Embeddings	UAS	LAS
10	train	87.03	83.23
10	train+dev	86.96	83.23
10	News Corpus	87.78	84.05
10	CC	87.55	83.86
20	train	87.61	83.77
20	train+dev	87.68	84.00
20	News Corpus	88.04	**84.32**
20	CC	87.57	83.90

Table 2
Results for NLP-Cube with different embeddings

Embeddings	UAS	LAS
CC	90.69	**86.84**
News Corpus	90.28	86.16

form and lemma embeddings derived from AnCora itself. It seems that using word embeddings from big enough volumes of same genre text improves results with this toolkit.

2.2. NLP-Cube

NLP-Cube is an open source framework developed by Adobe that competed in CoNLL 2018 (it ranked 9th for LAS score).[7] This toolkit provides an end-to-end text processing solution using neural networks, and it includes modules for sentence splitting, tokenization, lemmatization, part-of-speech tagging, dependency parsing and named entity recognition for more than 50 languages [6].

For this software package, we have used off-the-shelf hyperparametes with Common Crawl embeddings. Early stopping was set to 40 (default is 20), to allow for maximal optimization of the model. This time, performance is lower using News Corpus embeddings, as it can be seen in table 2.

NLP-Cube uses “multiple stacked bidirectional LSTM for the parser layers and project 4 specialized representations for each word in a sentence, which are later aggregated in a multilayer perceptron in order to produce arc and label probabilities.” [6]

Although not tested for this competition (in which POS tagging is already provided by organizers), NLP-Cube is an interesting model to explore since it allows jointly training to also output morphological features. According to the authors, this combined training increases the absolute UAS and LAS scores by up to 1.5%.

2.3. Stanza

Stanza [7] is a Python natural language analysis toolkit developed by Stanford University. Its modules are built on top of the PyTorch library and it provides pre-trained models for 66 languages. It

[7] http://universaldependencies.org/conll18/results.html.

can interface with the popular Stanford CoreNLP Java package from the same institution. For UD, Stanza implements a Bi-LSTM-based deep biaffine neural dependency parser, augmented with two linguistically motivated features that handle *linearization* order of two words in a given language and prediction of the typical distance in linear order between them.

The results obtained with this parser outperform those of UDPipe by nearly 5 points, as it is shown in table 3. For this reason, we also considered the possibility of building another model with this parser, just in case UDPipe results could degrade the overall performance of the joint system.

For the above-mentioned purpose, we have tried to augment the training data pooling together the training partitions from both CAPITEL-UD and AnCora.[8] We were aware of the different treatment of various Spanish language phenomena in CAPITEL-UD and AnCora. These affect tokenization, morphological analysis and dependency relations. A by no means exhaustive list of differences is presented in the following paragraphs.

Tokenization: Words *al* and *del* (preposition+masculine article) are split in CAPITEL-UD whereas they remain as a single token in AnCora, being the former decision compliant with UD v2 guidelines. Other multiword elements like adverbial, prepositional and conjunctive phrases are represented in one token in CAPITEL-UD but, although recognized as a unit in the MISC column of CoNLL-U format, they are split in AnCora, with AnCora practice as UD v2 compliant. Perfect verbal forms, which are syntagmatic in Spanish, are considered one token in CAPITEL-UD but two in AnCora, again being AnCora decision compliant with UD v2 guidelines.

Morphological analysis: Adjectives not showing morphological features for gender (*leve* —either masculine of feminine— as opposed to *recto* —only masculine—) don't include this feature in the parse tree in AnCora, which is however present in CAPITEL-UD analyses; both annotation practices are allowed by UD v2 guidelines in this case.

Dependency relations: Relative pronouns, like the very frequent word *que*, are systematically labeled with the corresponding syntactic function in AnCora while CAPITEL-UD uses a kind of double labeling where both the function and the label `mark` are used; although described in the accompanying CAPITEL-UD documentation, these features are not prescribed in the UD v2 guidelines. Relative clauses are labeled as `acl` in AnCora while CAPITEL-UD uses the language specific `acl:relcl`; both annotations are UD v2 compliant. In complex predicates like *llevar a cabo*, the noun is labeled as `compound` in AnCora (in accordance with the UD guidelines), being an `obl` in CAPITEL-UD.

From this incomplete comparison of annotation styles, let us note in passing that none of the corpora perfectly follow the UD guidelines, although AnCora is closer to full compliance than CAPITEL-UD is.

In spite of these differences, specially those showing two dependency relations coding styles, data augmentation was also considered in order to train a model for Stanza using training material from both CAPITEL-UD and AnCora corpora, in order to have three times more training data than that provided by CAPITEL-UD organizers. Results for both models are presented in table 3.

As can be observed, the pooled model using data augmentation increases performance by 0.11% with respect to CAPITEL-UD set alone. Data efficiency is low, since for this LAS gain, 14,305 new training

[8] This is actually an area of improvement for Stanza suggested by its authors to provide language models more robust for different genres [7].

Table 3
Results for Stanza with different training sets

Dataset	UAS	LAS
CAPITEL-UD training dataset	91.54	88.19
CAPITEL-UD + AnCora training datasets	91.53	**88.30**

Table 4
Results for system combinations

Systems	UAS	LAS
UDPipe, NLP-Cube, Stanza-C, Stanza-CAn	91.65	88.39
UDPipe, Stanza-C, Stanza-CAn	91.63	88.37
UDPipe, NLP-Cube, Stanza-C	91.38	88.02
UDPipe, NLP-Cube, Stanza-CAn	91.41	88.16
UDPipe, Stanza-C, Stanza-CAn	91.63	88.29
NLP-Cube, Stanza-C, Stanza-CAn	91.72	**88.46**

sentences have been used —the whole AnCora training partition. However, this is probably due to the sharp differences in the annotation styles from both datasets. A post-competition experiment using data augmentation with only 500 new sentences from the same genre (analyzed with the Stanza model and hand corrected by the author) delivers a LAS score of 88.46. Note that this is also the best score obtained with our pooled model (see section 2.4).

With this improvement in mind, had we had more time for the competition, we could have implemented an automatic (partial) solution to bring both corpora closer. Unfortunately, in order to outperform other possible competitors, this should have been done by adopting CAPITEL-UD developers decisions, which, as already stated, deviate from the UD guidelines.

2.4. Voting strategy

Our first underlying plan for this competition was that of combining predictions from various UD parsers, as an attempt to further improve our overall performance. Rather than implementing our own solution, we have used for this purpose `conllu-voting.py`, a Python implementation of Chu-Liu-Edmonds minimum spanning tree over a graph of CoNLL-U files, which is part of a set of "[s]cripts for compatibilitising between VISL-CG3, Apertium, CoNLL-X and Universal Dependencies".[9]

There were four system predictions to combine —those of UDPipe, NLP-Cube and two models produced with Stanza. Table 4 shows results on development dataset using several of the possible combinations available.[10]

As table 4 shows, although the combination of all parsing systems improves our overall performance by 0.09%, it is leaving out our lowest performant system, UDPipe, that we get an LAS increase of 0.16%. This is, then, the combination used in one of our runs.

[9] https://github.com/ftyers/ud-scripts.

[10] We use Stanza-C for the system using CAPITEL-UD as training material and Stanza-CAn for the system with CAPITEL-UD and AnCora training datasets.

Table 5
Results for system runs submitted on test dataset

System	UAS	LAS
C	91.72	88.47
CA	91.77	88.53
CACV	91.94	**88.66**

3. Results and discussion

The organizers limited the number of runs allowed to be submitted by each participating team to three. System submissions were then arranged according to performance obtained with development material. Hence, system runs from both models built with Stanza (renamed in table 5 as C and CA) were submitted. Besides, the best of our combinations (CACV) was also submitted. These scores refer to test dataset, and are, thus, our final results in the competition.

It is important to note that none of the models is optimal, since an extremely low early stopping condition of 5 was applied to every model built. Nonetheless, results seem to support the maxim that more data is better data, even in the case of using very heterogeneous training sources. This fact can be observed in the 0.06% LAS improvement of system using both CAPITEL-UD and AnCora training material, a small improvement due to a low data efficiency that can be attributed to the differing (dependency) annotation styles of the corpora used. If a smaller but tighter dataset is used for data augmentation, the above-mentioned maxim seems to be more neatly supported (0.16% LAS improvement with only 500 new sentences).

Most importantly, however, the use of several parsing models built with different software solutions and adequately combined results in a slight, but promising, performance boost. This is clearly an avenue that needs further exploration.

Acknowledgments

We would like to thank two anonymous reviewers for their careful reading of our contribution and their many insightful comments and suggestions.

This work was first presented under the name Martín Lendínez. This is the pseudonym used by the Spanish writer Mariano Antolín Rato in his translations to Spanish of the Beat Generation North-American writers. The name stemmed as a joke with his friends Juan Cueto and Gonzalo Suárez, who had signed a number of works under the name Claudio Lendínez junior. Martín became a kind of hippie brother for Claudio, a brother with a long and wide professional career. We had used his name in this shared task as a tribute to the wonderful work of Antolín Rato in those times of change in Spain, and more modestly to enlarge Martin's CV. Unfortunately, due to reasons beyond the control of the author, it has been impossible to keep this author name during the whole publishing process.

References

[1] R. McDonald, J. Nivre, Y. Quirmbach-Brundage, Y. Goldberg, D. Das, K. Ganchev, K. Hall, S. Petrov, H. Zhang, O. Täckström, C. Bedini, N. Bertomeu Castelló, J. Lee, Universal Dependency annota-

tion for multilingual parsing, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 92–97. URL: https://www.aclweb.org/anthology/P13-2017.

[2] J. Porta-Zamorano, L. Espinosa-Anke, Overview of CAPITEL Shared Tasks at IberLef 2020, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), 2020.

[3] M. Straka, J. Straková, Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe, in: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 88–99. URL: https://www.aclweb.org/anthology/K17-3009. doi:10.18653/v1/K17-3009.

[4] M. Straka, UDPipe 2.0 prototype at CoNLL 2018 UD shared task, in: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 197–207. URL: https://www.aclweb.org/anthology/K18-2020. doi:10.18653/v1/K18-2020.

[5] M. Taulé, M. A. Martí, M. Recasens, AnCora: Multilevel annotated corpora for Catalan and Spanish, in: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), European Language Resources Association (ELRA), Marrakech, Morocco, 2008. URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/35_paper.pdf.

[6] T. Boros, S. D. Dumitrescu, R. Burtica, NLP-cube: End-to-end raw text processing with neural networks, in: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 171–179. URL: https://www.aclweb.org/anthology/K18-2017. doi:10.18653/v1/K18-2017.

[7] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 101–108. URL: https://www.aclweb.org/anthology/2020.acl-demos.14. doi:10.18653/v1/2020.acl-demos.14.