Adding Features to a Basque PoS Tagger Model

Xabier de Zuazo Oteiza
University of the Basque Country
(UPV/EHU)
xzuazo002@ikasle.ehu.eus

Carlos Domínguez Becerril
University of the Basque Country
(UPV/EHU)
cdominguez019@ikasle.ehu.eus

Abstract

Part-of-speech (PoS) tagging is a popular Natural Language Processing task (NLP) that refers to categorizing words in a text (corpus) with respect to a specific part of speech, depending on the definition of the word and its context. Part-of-speech tags describe the characteristic structure of lexical terms within a sentence or text. Therefore, we can use them for making assumptions about semantics. In this project, we propose to use the Flair framework to predict the PoS tags for the Basque language in three different ways: predict only the basic tags, predict the basic tags together with another feature, and predict the basic tags along with two additional features.

1 Introduction

In traditional grammar, a part-of-speech (PoS) is a category of words (or, more generally, of lexical items) that have similar grammatical properties. Particularly, words that display similar syntactic behavior are assigned the same part-of-speech tag.

Every language has its own set of part-of-speech categories. The most common parts of speech are nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, interjections, numerals, articles, or determiners. Nevertheless, there are some languages like Slavic languages that do not have articles or the Uralic family languages (Finnish and Hungarian, for example) that completely lack prepositions or have only very few of them. But generally speaking, these categories are mostly consistent across languages, and the scheme is shared between most languages (Nizami et al., 2019).

The language of interest in this paper is Basque, a language spoken by Basques and others of the Basque Country, a region that straddles the westernmost Pyrenees in adjacent parts of northern Spain and south-western France. The case of Basque is particular: linguistically, it is an isolated language, unrelated to any other language. Moreover, Basque is an ergative-absolutive language; that is, the subject of an intransitive verb is in the absolutive case, which is unmarked, and the same case is used for the direct object of a transitive verb. Additionally, the subject of the transitive verb is marked differently, with the ergative case (shown by the suffix -k). This also triggers main and auxiliary verbal agreement, which poses a challenge for machine learning models.

This paper focuses on training and evaluating the prediction of the universal part-of-speech tags for the Basque language using the Flair framework (Akbik et al., 2019). Finally, we will try to combine them with more features, such as the name, the case, the number, and so on, to see if they are helpful or not for predicting the PoS tags.

2 Related Work

To accomplish the requirements of an efficient PoS tagger, researchers have explored the possibility of using Deep Learning (DL) and Machine Learning (ML) techniques. Under the big umbrella of artificial intelligence, both ML and DL aim to learn meaningful information from the given enormous language resources. In order to learn valuable information from the corpus, the techniques differ. On the one hand, the ML-based PoS tagger relies primarily on feature engineering. On the other hand, DL-based PoS taggers are better at learning complex features from raw data without relying on feature engineering because of their deep structure.

There exist also other proposals that put forward different approaches for PoS tagging tasks (Agerri et al., 2020). Their research work includes the use of fast-text and Flair embeddings alongside the Flair framework and the use of a transformer model (Vaswani et al., 2017) obtaining state-of-the-art results with the latter.

3 Dataset Description

The research here uses the Universal Dependency dataset (de Marneffe et al., 2021) for the Basque language. The universal dependency dataset is a project that seeks to develop a cross-linguistically consistent treebank annotation for many languages to facilitate multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective. The annotation scheme is based on universal Stanford dependencies (De Marneffe et al., 2014), Google universal part-of-speech tags (Petrov et al., 2011), and the Interset interlingua for morphosyntactic tagsets (Zeman, 2008). Table 1 shows the contents of each split for the Basque language.

Split	Sentences	Tokens
Training	5,396	72,974
Development	1,798	24,095
Testing	1,799	24,374
Total	8,993	121,443

Table 1: Distribution of data across splits.

Besides, in Figure 1 we can see the distribution of each of the labels in the different splits. Most of the infrequent labels have more or less the same frequencies over the three partitions. However, frequent labels are not so well balanced. For example, the adjectives are pretty common in the train partition and less frequent in the other partitions. Other tags like verbs, particles, adverbs, and numbers also suffer from a certain imbalance between partitions. To remedy this mismatch, the F1 measure will become more relevant when performing an analysis.

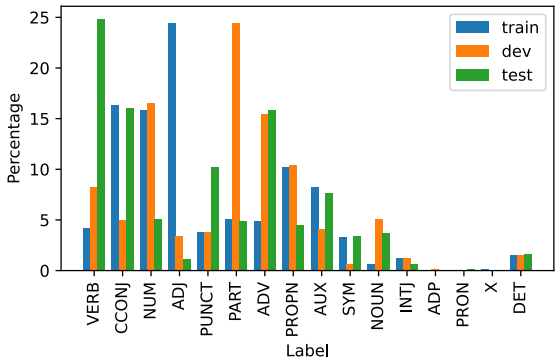

Figure 1: Universal PoS label distribution.

4 Models

This project proposes to train three different types of models using a distinct amount of features. All the models developed here use the same underlying architecture, in particular: contextual word embeddings in a bidirectional LSTM with a hidden size of 256, a word dropout of 0.05, and a locked dropout of 0.5. For reproducibility, all of them were trained for 25 epochs.

4.1 Model Type 1: Basic PoS tagger

The first model uses the basic universal PoS tags. The included tags are the following ones: ADJ (adjective), ADP (adposition), ADV (adverb), AUX (auxiliary), CCONJ (coordinating conjunction), DET (determiner), INTJ (interjection), NOUN (noun), NUM (numeral), PART (particle), PRON (pronoun), PROPN (proper noun), PUNCT (punctuation), SCONJ (subordinating conjunction), SYM (symbol), VERB (verb), and X (other).

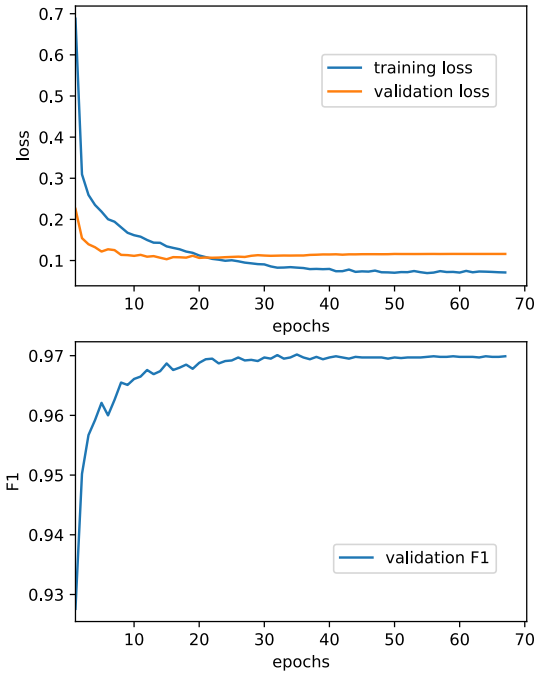

Figure 2: Universal PoS tagger training curve.

The model has been trained from the ground up using the Flair framework (Akbik et al., 2019). Furthermore, three different types of embeddings (Akbik et al., 2018) to represent the text are stacked: Basque word embeddings, forward Basque Flair

embeddings, and backward Basque Flair embeddings. Figure 2 shows the training curve of the model for a maximum of 200 epochs, but the framework stopped at around 70 epochs due to a lack of progress. Checking the curve shows that after 25-30 epochs there is not much improvement in both the loss and the F1 score.

On top of that, analogous models have also been trained on the other label types but limited to 25 epochs. These results will help us in later comparisons along the course of this work. The training F1 scores on these other label types can be seen in Figure 3. This type of bidirectional LSTM model obtains the best performance on the Universal PoS tag classification task, also bringing quite good results for dependency and lemma rules classification tasks. In fact, the latter may have room for further improvement because the curve looks like it can continue to improve after 25 epochs. However, the results on the other type of labels are not so satisfactory. Notably, number, case, and definite tags got remarkably low scores that do not even reach a score of 50 on the F1 value.

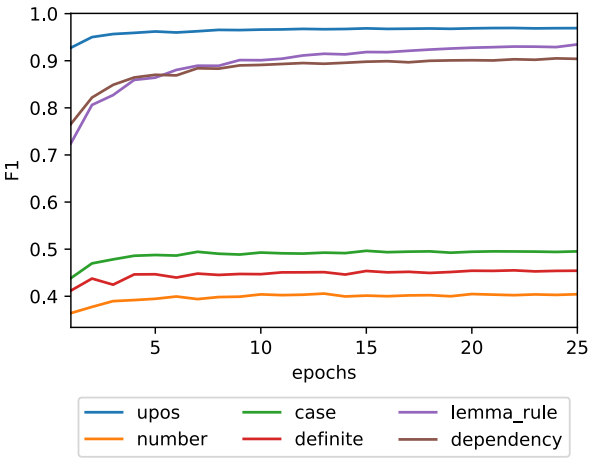

Figure 3: Single-label models' dev training curve.

Special mention requires the lemma label type. The original possible values of this label are big enough to give problems when trained on sequential tagger models due to word variety in the language. To tackle this problem, the lemma values have been transformed to minimum script edit rules, a work based upon UDPipe 2.0 research scripts (Straka, 2018). These rules are used to convert between the original word and the lemma. The change of lemmas to rules reduces the number of possible values of the label, considerably reducing the memory required and consequently making the training of neural models much more affordable.

4.2 Models of Type 2: Pair of Labels

These models have been trained from scratch but having label pairs as the target. To decide which label types to merge, we have selected the most common ones. Specifically, we merged the labels having more than 10,000 annotations in the corpus: lemmas, definite, number, case, and dependency. Then, these label types were merged with the universal PoS tags, picking up instances with both tags present. The final list of targets is the following:

- `upos, lemma_rule`
- `upos, definite`
- `upos, number`
- `upos, case`
- `upos, dependency`

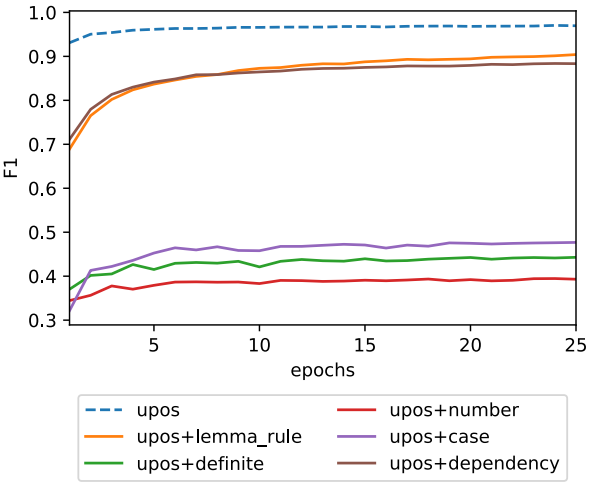

Figure 4: Double-label models' dev training curve, with the single UPoS curve dashed for comparison.

The work conducted here with these double-feature models created a total of five models trained on pairs of labels. In Figure 4, we can see the training curve of the different models. Checking the different results, the tasks that were difficult to predict previously also got low scores when merging with the PoS label. Specifically, number, case, and definite label types scores have substantially decreased, conserving the distance between them, and showing less stability during training. At the same time, dependency and lemma rules also have decreased their scores, but they are still correctly predicted overall and seem to have gained stability. Over and above that, as with the single models, the lemma rules still seem to require more than 25 epochs to complete the training.

4.3 Models of Type 3: Triple Labels

Following the same pattern as in Section 4.2, we merged the most commonly annotated features with the Universal PoS tags, but in groups of three this time. The list of labels types generated is the following:

- `upos`, `definite`, `number`
- `upos`, `definite`, `case`
- `upos`, `definite`, `dependency`
- `upos`, `number`, `case`
- `upos`, `number`, `dependency`
- `upos`, `case`, `dependency`

In Figure 5, we can see the training curves of these models. In this case, all the models' scores dropped considerably, ending with F1 scores between 34 and 43. At the sample time, the learning curves became pretty irregular; in other words, the models are not as stable as before.

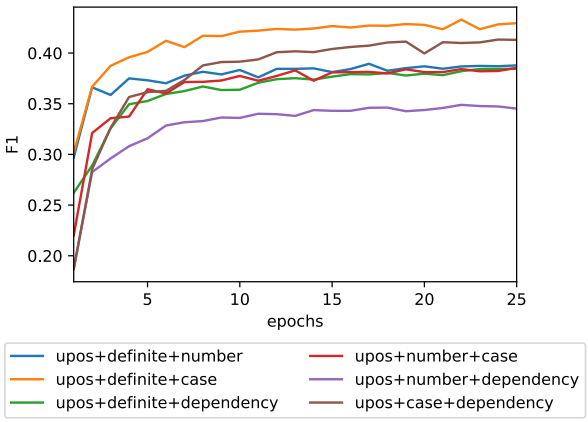

Figure 5: Triple-label models' dev training curve.

In this particular scenario, it should be noted that, due to hardware constraints, the inclusion of lemmas on these latest models has been left for future work. The reason is that the label-value diversity generated by the 3 feature models increases substantially when working with UPoS, lemmas, and a third feature together, demanding more memory.

5 Results

Table 2 shows that the best way for obtaining the universal PoS tags of a text is by predicting them alone, getting a result of 97.07 in the F1 score: that is, without any extra feature. Overall, adding features to the universal PoS tags negatively affects the learning process. In the case of merging with one feature, the model with PoS tags and lemma rules is the least affected by the merge, returning an F1 score of 90.41, followed by the dependency label with a score of 88.34. The rest of the features adversely affected the models. When merging three features, the best model is the one merging the universal PoS tags with de definite and case features, obtaining an F1 score of 42.94; followed by the universal PoS tags merged with definite and dependency, with an F1 score of 41.30.

Inspecting the effect of the addition of features, Table 3 shows the mean and standard deviation in the scores produced by the addition of each of the features to the models. With 2-feature type models, the number is the feature that affects the scores less, but in 3-feature type models, the dependency feature is the one that produces minor damage and instability in the models. Broadly, precision is severely affected as we keep merging features, but not so much the recall. This occurs because the initial label imbalance is accentuated, and the positive class becomes increasingly scarce due to the growth in the variety of label values and annotations reduction.

6 Conclusions and Future Work

In this paper, we have seen three different types of models to predict the PoS tags for the Basque language: predict only the basic tags, predict the basic tags together with another feature, and predict the basic tags along with two additional features.

The results show that the best way to get the correct PoS tag is by predicting it directly without any other features. When merging different features, the scores decrease rapidly, being dependency and number the features that affect the model less.

For future work, we can do more feature combinations to check which ones are more meaningful. Moreover, we want to combine the output of this project with the task of lemmatization and observe if merging features help. Finally, in order to have more robust results, using upsampling methods or creating more extensive and varied corpora for Basque might help balance the classes.

To conclude, all the code to replicate this research, including the training of the models, generating the plots, and calculating the final results, is available online[1].

[1] https://colab.research.google.com/drive/1_Xf4n9psiq1wW-eQL6eHIFjecl3btUgS

Model	Annotations	Loss	Precision	Recall	F1	Accuracy
lemma_rule	72,974	0.3164	93.26	93.26	93.26	93.26
definite	22,822	0.0851	29.92	95.38	45.56	29.92
number	19,419	0.0714	25.52	95.93	40.31	25.52
case	25,450	0.0878	33.31	95.91	49.45	33.31
dependency	72,974	0.3454	90.31	90.31	90.31	90.31
upos	72,974	0.1128	**97.07**	**97.07**	**97.07**	**97.07**
upos+lemma_rule	72,974	0.4670	**90.41**	90.41	**90.41**	**90.41**
upos+definite	22,822	0.1203	29.09	92.72	44.29	29.09
upos+number	19,419	0.0937	24.88	**93.53**	39.31	24.88
upos+case	25,450	0.1412	32.12	92.47	47.68	32.12
upos+dependency	72,974	0.4523	88.34	88.34	88.34	88.34
upos+definite+number	19,150	0.0962	24.48	**93.28**	38.78	24.48
upos+definite+case	22,501	0.1555	**28.11**	90.87	**42.94**	**28.11**
upos+definite+dependency	22,822	0.2849	25.26	80.51	38.45	25.26
upos+number+case	22,501	0.1189	24.32	92.69	38.54	24.32
upos+number+dependency	19,419	0.2226	21.85	82.14	34.52	21.85
upos+case+dependency	25,450	0.3335	27.82	80.11	41.30	27.82

Table 2: The performance results of the final models on development partition.

	Precision		Recall		F1	
Feature	2-label	3-label	2-label	3-label	2-label	3-label
+lemma_rule	-2.8		-2.8		-2.8	
+definite	-0.8	-22.5±28.7	-2.7	-3.2±3.3	-1.3	-18.4±22.3
+number	**-0.6**	-26.3±28.4	-2.4	**-1.8**±3.1	**-1.0**	-22.8±22.0
+case	-1.2	-20.7±28.2	-3.4	-3.64±3.3	-1.8	-16.4±21.7
+dependency	-2.0	**-3.7± 0.5**	**-2.0**	-12.0**±0.4**	-2.0	**-5.7± 0.7**

Table 3: The effects on models when merging each of the features.

References

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for basque. *CoRR*, abs/2004.00033.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Muhammad Nizami, Tafseer Ahmed, and Muhammad Yaseen Khan. 2019. Towards a generic approach for pos tag-wise lexical similarity of languages.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Milan Straka. 2018. 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*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Daniel Zeman. 2008. Reusable tagset conversion using tagset drivers. In *LREC*, volume 2008, pages 28–30.