María Leva Jiménez and Eloi Singla i Milian
Natural Language Processing

Scottish Gaelic POS tagger

Introduction

For this task, we have trained a part-of-speech (POS) tagger for Scottish Gaelic using a blank multilanguage spaCy model. This decision was motivated by the fact that there are no trained spaCy pipelines for this language and the general lack of resources for its linguistic study. This task required adjusting the tokenizer. Even though the tasks were accomplished with varying degrees of success, the POS tagger's performance is respectable, reaching the expected accuracy baseline levels.

Material and methods

The data used for this task was extracted from the Annotated Reference Corpus of Scottish Gaelic (ARCOSG), compiled by Lamb et al. (2020) and available online at GitHub. ARCOSG is a hand-tagged corpus that uses Brown format tag separators, dividing the elements that comprise each token (the word and its tag) using a slash (eg. 'fhuair/V-s'). The annotation scheme follows the Irish PAROLE tagset, which includes different categories that do not correspond exactly to those included in the spaCy pipelines tagset. The corpus includes samples of text with and without punctuation, which had an impact on our work.

During data preprocessing the most challenging part was dealing with multi-word tokens, since they were separated by spaces from the rest of the "word/tag" compound. In order to extract the relevant data to train the tagger (tokens and their corresponding tag), a regular expression was used to find and group tokens with their tags. The original tagset used by the corpus was adapted into the usual format used in spaCy. Abbreviations and the category of "residuals" were grouped into the spaCy tag "symbols". We also separated the original "conjunction" category into subordinating and coordinating ones.

We turned the corpus into spaCy examples using only the texts that had punctuation, as the tagger processes sentences and we could not find a way to automatically split sentences without punctuation. Approximately 50,000 tokens were used to train the tagger. Data was split into the training-validation-test sets in an 8:1:1 ratio. The training set included 2,019 sentences, the validation set 252, and the test set 253.

As for the SpaCy model, a multilingual blank model was initialized. The tagger was added to it, and the default tokenizer was also adjusted. We did it by removing its apostrophe and dash separation rules, since Scottish Gaelic uses spaces to separate words by orthographical convention, i.e. tokens are already separated properly. We added the multi-word tokens present in the corpus to a list and fed them into the tokenizer so that it would not separate them. Still, it made some mistakes (see Results). The training of the tagger took into account these possible inconsistencies during tokenization.

We tried to make the code deterministic to facilitate reproducibility and testing of hyperparameters, but we could not manage. We set a seed for the *spaCy*, *random*, and *numpy* libraries (the latter of which is not used directly but it could be that some dependencies rely on it), but that would not solve the issue. Our theory is that this is due to the use of the GPU, there are some random elements in assigning different threads, and if one does not control for that, the code cannot be deterministic. We had never run into this issue in other projects that also used the GPU, but it seems based on forum posts that this has happened before.

Results

Regarding the adjustments we made to the tokenizer, it helped to get better results. However, in 12.83% of sentences in the train set there is a mismatch between the number of tokens in the model's output and in the golden labels. Because we measured errors like this, it means that the percentage could be a bit higher, i.e. if the model split one multi-word token but later joined something that was to be kept separate. We believe this could happen for the same reason that the tokenizer is not perfect: the tokenization in the golden labels is not consistent, so that some multi-word tokens are split in some places and joined in others. Therefore, it is impossible to get 0% in errors with this data. Sadly, we have not found what the baseline would be.

The final hyperparameters of the tagger were a 0% dropout rate, because increasing it worsened the results, and a training of 10 epochs (further than this dev accuracy did not increase substantially and we wanted to avoid overfitting, although we do not know in what amount this is possible for this task) and a batch size of 8. After that, the accuracy of the model (taking into account possible mistakes in the tokenization) were as follows: 98.89% train, 93.17% dev and 92.93% test.

Qualitatively, we had the model tokenize a new text and it worked perfectly. However, we did not try with polysemous words due to time constraints. We hypothesize that the model would have struggled, but this could be a further line of research. Another one could be finding a way to use the punctuation-less texts from the corpus, which we only used in our project for the list of multi-word tokens we fed into the model.

To conclude, the final accuracy we got (92.93%) is basically the baseline for tagging if you always guess the most frequent POS of a given word (Jurafsky & Martin, 2025), but it shows that our model is doing at least that. Given our lack of experience with this working pipeline and our lack of data, we are more than satisfied with our results.

References

Jurafsky, D and Martin, J. H. (2025). Sequence Labeling for Parts of Speech and Named Entities. In *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, (3rd ed., pp. 1-27). Online manuscript: https://web.stanford.edu/~jurafsky/slp3/17.pdf

Lamb, W., Arbuthnot, S., Naismith, S., Danso, S. (2020). *Annotated Reference Corpus of Scottish Gaelic* (ARCOSG), 1997-2020 [dataset]. University of Edinburgh. School of Literatures, Languages and Cultures. Celtic and Scottish Studies. Available at https://github.com/Gaelic-Algorithmic-Research-Group/ARCOSG

List of contributions

Coding / checking the writing: Eloi Singla i Milian

Writing / checking the code: Maria Leva Jiménez

Link to code and data repository

https://github.com/EloiSinglaMilian/NLP-Exercise-3/