Analysing PoS Tagging Transfer Scores

Guillaume Wisniewski
guillaume.wisniewski@u-paris.fr

November 2025

1 Submission Guidelines/Requirements

This project consists of 24 questions, which concern either general knowledge in NLP, Machine Learning, Linguistics or Computer Science, or questions requiring code. You must submit, no later than 24th November at 8:00 a.m. (hard deadline), a single PDF file (and only a PDF !) containing your answer to each question. Following a previous painful experience, I must make it clear that : i) the questions must be answered in the given order and ii) your name must appear both in the filename and on the document itself.

For the general knowledge questions, you should provide a written answer of no more than five lines (and not in font size 8). You do not need more than that : as the poete once said, "Ce que l'on conçoit bien s'énonce clairement, et les mots pour le dire arrivent aisément." For the coding questions, you must include the instructions necessary to answer the question, the output of these instructions, and a brief commentary. As talented computer scientists with impeccable taste, you should avoid dreadful screenshots of code and results, and instead learn to present both directly and neatly in your report. This is not to make your life difficult : it is a skill expected from you both in research laboratories and in industry. As emphasised repeatedly during the course, you should normally be able to answer each coding question with a single instruction (two at most).

A crucial point : I assume that you are responsible adults, and therefore I will not be performing any checks regarding the use of AI tools (I am fairly certain ChatGPT can answer all these questions) or regarding plagiarism or cheating. I sincerely hope you will discuss among yourselves if any question lacks clarity, and that you will engage in stimulating and passionate discussions about the various approaches one may take to obtain the best answer. However, you must produce the work individually. You will soon be facing interviews in which your knowledge and skills will most likely be tested (including coding interviews), and it is essential that you are able to answer these types of questions on your own, and efficiently.

The marking scheme is straightforward : a fully correct answer, both in content and presentation, receives 100 % of the marks for the question ; an answer with correct content but poor presentation receives 50 % ; an answer with incorrect content receives 0 %.

2 Introduction

One of the most surprising, and frankly, most astonishing, aspects of modern language models (which, to put it simply, are neural networks capable of building representations of language without any human guidance) is their multilingual ability.

Thanks to these models, it is now possible to train a model (in the statistical-learning sense you encountered in Marie Candito's course) on data from one language A, and then test it on a completely different language B, even though the model has never seen any annotated data in language B. This phenomenon is known as *model transfer*.

This is a highly valuable tool from a practical point of view : it allows us to build systems for languages where annotated data is scarce or unavailable. But it is also fascinating from a theoretical perspective. One key question (at least in my opinion !) is : what linguistic properties enable or enhance transfer between languages ? For example, can we successfully transfer a model from an SOV language to an SVO one ?

3 Getting started with the data

The objective of this project is to use `pandas` to study the performance of model transfer and to investigate whether performance varies according to certain linguistic properties. To do so, we will rely on data from the Universal Dependencies (UD) project, [1] a multilingual initiative providing annotated corpora for a wide range of languages. As indicated on the project website, UD provides separate *train* and *test* corpora.

We focus on the task of Part-of-Speech (PoS) tagging. The idea is straightforward : we train a PoS tagger on each of the n training corpora and evaluate each trained model on each of the m test corpora (even when a test corpus does not correspond to the training corpus).

A key point is that in the UD project, there may be several corpora for the same language. Throughout this practical and in all collected results, we will therefore distinguish :

— the language and UD corpus ID of the model (the corpus on which the model is trained) — columns `model_lang` and `model_ud`

— the language and UD corpus ID of the evaluation data — columns `target_lang` and `target_ud`

1. `https://universaldependencies.org/`

Note that some corpora provide only a test set (in which case no model can be trained, only evaluated), whereas others include both train and test data. There cannot be train-only corpora.

UD also provides different annotation granularities. In this practical, we consider only the scores using the coarsest tagset, corresponding to Universal PoS tags (column `UPos`). Scores are expressed as percentages : a model predicting all reference tags correctly would obtain a score of 100 %.

We therefore obtain $n \times m$ scores, corresponding to three situations :

— **in-domain evaluation** : same corpus used for training and testing
— **out-of-domain evaluation** : same language, but different train and test corpora
— **cross-lingual model transfer** : training and test corpora come from different languages

In the file `pos_tagging_scores.tsv`, you are provided with all PoS tagging scores required for this practical (do not worry : you will later learn how to train and evaluate such models yourself).

1. Briefly describe the task of PoS tagging. Why is the ability to predict PoS tags still useful in the era of ChatGPT-like models ?
2. Why should training and test corpora be kept separate ? Is the use of predefined train–test splits, as in the UD project, a good idea or not ? Motivate your answer.
3. Why are the data stored in a TSV rather than a CSV format ? How can they be loaded into a dataframe ?
4. How many different training corpora are available ?
5. How many different test corpora are available ?
6. How many different languages are represented in the data ?
7. Plot the distribution of the number of corpora per language (x-axis : number of corpora ; y-axis : number of languages with that number of corpora).
8. How many languages have three training corpora ?
9. Which language has the highest number of corpora ?

4 Analysis of Scores

We now focus exclusively on the performance obtained on the French test corpora (purely for scientific reasons, with absolutely no trace of chauvinism).

10. Considering only the in-domain scores (same train and test corpus), compute the mean, median, minimum, and maximum scores.
11. Answer the same questions while considering only the out-of-domain scores (train and test corpora differ, but the language is the same). Comment on your observations.

12. Repeat the same questions for the cross-lingual transfer scores (train and test languages differ).
13. Construct a dataframe that, for each French test corpus, contains the following information :
    — the in-domain performance (if available, otherwise `NaN`),
    — the best out-of-domain performance,
    — the best cross-lingual transfer performance,
    — the language achieving this best transfer score.
    Provide a short commentary.
14. How many different languages achieve the best transfer score (considering all French test corpora) ?

We will now move on the analysis of all transfer scores :

15. Plot the distribution of in-domain, out-of-domain, and cross-lingual transfer scores across all test corpora using a violin plot. Interpret the results.
16. For each language and each test corpus, determine which training language yields the best cross-lingual transfer score.
17. For a given language, plot the distribution of the number of distinct training languages that achieve the best transfer score. Comment on your findings.

5 Impact of Linguistic Typology

The file named `language_family.json` associate each language to its language familly extracted from `Glottolog`. [2]

18. Why is it natural that transfer performance varies with the source language ?
19. Why is it useful to predict which source language will yield the best performance for a given target ?
20. Define the notion *diachronic* and *synchronic* similarities used in typological linguistics. Which one should you use to predict which source language will yield the best performance for a given target ? Is the typological information available on `Glottolog` useful ?
21. How many language families are represented in `language_family.json` ? Which family is most represented ? What is the distribution of languages per family ?
22. Compute, for every test corpus, the family of the best-performing source language. Report the proportion of corpora whose best source comes from the *same* family as the target language. Interpret the result.

2. `https://glottolog.org/`

23. Plot the distributions of transfer scores by grouping pairs into (i) same-family and (ii) different-family. Comment on central tendency and dispersion, and whether within-family transfer confers a systematic advantage.
24. For each corpus where both are available, compute the difference between the in-domain score and the best transfer score obtained from each language family. Analyse the distribution of these deltas (e.g., median, IQR, tails) and discuss which families tend to close the in-domain gap, and in which conditions they fail.