Tokenization & spelling variation

HANDS-ON EXERCISE

Dialect NLP seminar, LMU Munich 2025/26
Prof. Dr. Barbara Plank, Verena Blaschke, Ryan Soh-Eun Shim

November 20, 2025

1 Introduction

The first processing step of most language models is subword tokenization: splitting the input text into shorter character sequences that are associated with embeddings, which in turn can be used for the calculations within the model. Today's session is a hands-on exercise (plus discussion) about the (lack of) robustness that different subword tokenizers exhibit towards unseen spelling variation. As a motivating example, consider the subword tokenizations for a Standard German and Alsatian sentence pair produced by a German tokenizer:

a.	Wir	sprechen	alemannische	Mundarten	.
	`Wir`	`sprechen`	`al`, `#emann`, `##ische`	`Mund`, `##arten`	.
b.	M'r	redd	alemànnischi	Mundàrte	.
	`M`, `'`, `r`	`red`, `##d`	`al`, `##em`, `##à`, `##nn`, `##isch`, `##i`	`Mund`, `##à`, `##rte`	.

"We speak Alemannic dialects" in Standard German (*a*) and dialectal Alsatian German (*b*), as tokenized by GBERT (Chan et al., 2020) (example via Blaschke et al. (2023)).

In this session, you will try compare the subword tokenization of multiple tokenizers across datasets in different language varieties. We've prepared some datasets, tokenizer choices, and possible questions to focus on below. You are *not* expected to explore all of them! Instead, simply work on the options you find the most interesting. We will compare our findings in the second half of the session.

You can use this hands-on session as a possible starting point for your final project/paper.

2 Resources

2.1 Data

We have prepared data from two different sources:

- Data from Universal Dependencies (Nivre et al., 2020) in varieties related to Arabic, German, Greek, and Italian. You can find the data in the subfolders of the corresponding names. For some languages/varieties, the Universal Dependencies dataset contains multiple different corpora (different genres, text sources). Check the `README`, which explains which file contains which variety, and provides links to more information. You can ignore the 'train/dev/test' notes in the filenames, these only refer to the datasets' original splits.
- Data from CODET (Alam et al., 2024) in varieties related to Arabic, Basque, Bengali, Italian, Kurdish, and Swiss German. For more details on the language varieties and original data sources, check the publication. Many of the sentences in CODET are parallel across related varieties, allowing for more control over the data when comparing the tokenization.

2.2 Tokenizers

The `README` file contains some mono- and multilingual tokenizer options to get you started. You can also try out other tokenizers – this is easiest for tokenizers belonging to models hosted on `https://huggingface.co`.

Once you have already compared some tokenizers, you can optionally try out other types of input representations, as used, e.g., by byte-level models (Xue et al., 2022) via `https://huggingface.co/google/byt5-small`.

2.3 Code

We provide some demo code to get you started:

- `demo_code.py` loads the Universal Dependencies files and shows how to turn them into subword token sequences
- `demo_code.ipynb` does the same, but as a Jupyter Notebook
- `demo_code_paralleldata.ipynb` loads the CODET data.

You should be able to run this code on your laptop. If this doesn't work, you can use `https://colab.research.google.com`. Note that you only need to load the tokenizers, but not the language models associated with them.

Note: parts of the code assume Unix-style file paths when processing the input files. You might need to update them slightly if you use Windows.

3 Comparing subword tokenizations

There is not just one way to compare subword tokenizations. You are free to try out your own ideas and/or to check the "further reading" section for next week's class for inspiration. Some possible measures are:

- Comparing the *subword fertility* (Rust et al., 2021) of a tokenizer on different texts, i.e. the average number of subwords per word.
- Comparing the subword token overlap between pairs of text tokenized by the same tokenizer.

4 Questions to explore & discuss

Some questions to get you started – we *don't* expect you to focus on *all* of them, and you are free to also come up with additional questions:

- How does subword tokenization differ for related standard and non-standard varieties?
- Are these patterns stable across different tokenizers? Is there a difference between mono- and multilingual tokenizers?
- Effect of corpus choice: Are these patterns stable when you not only use one corpus per variety, but instead compare multiple ones? Are these patterns stable across different corpus sizes?

- Is there a difference when applying a monolingual tokenizer on a related standard language vs. a related non-standard variety? (E.g., comparing a German tokenizer on German text with Dutch text vs. German dialect text)
- Do different ways of comparing the subword tokenization (subword fertility, subword token overlap between corpora, ...) show similar trends, or are there interesting differences?
- Are the trends regarding tokenizer choice / corpus choice / type of tokenization comparison stable across languages (e.g., are they similar for standard/dialectal German, Greek, and Kurdish)?

References

Md Mahfuz Ibn Alam, Sina Ahmadi, and Antonios Anastasopoulos (2024). "CODET: A benchmark for contrastive dialectal evaluation of machine translation." In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 1790–1859. Association for Computational Linguistics.

Verena Blaschke, Hinrich Schütze, and Barbara Plank (2023). "Does manipulating tokenization aid cross-lingual transfer? A study on POS tagging for non-standardized languages." In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pp. 40–54. Association for Computational Linguistics.

Branden Chan, Stefan Schweter, and Timo Möller (2020). "German's next language model." In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6788–6796. International Committee on Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman (2020). "Universal Dependencies v2: An evergrowing multilingual treebank collection." In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4034–4043. European Language Resources Association.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych (2021). "How good is your tokenizer? on the monolingual performance of multilingual language models." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3118–3135. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel (2022). "ByT5: Towards a token-free future with pre-trained byte-to-byte models." *Transactions of the Association for Computational Linguistics*, 10:291–306.