MASTER OF SCIENCE IN ENGINEERING

AdvNLP - Tokenization

Lab 1 (not graded)

Prof. Dr. Daniel Perruchoud, FHNW, Institut for Data Science

20. February 2025

1 Background

Subword tokenization in NLP is a technique that breaks words into smaller units (subwords), i.e., splitting infrequent words into meaningful subunits while keeping common words intact. This approach ensures that even unseen words can be represented as a combination of known subwords, making it crucial for handling out-of-vocabulary words, improving language modeling, and enabling efficient training in multilingual and low-resource settings.

Methods like **Byte Pair Encoding (BPE)**, **WordPiece**, **Unigram LM** and **SentencePiece** learn subword units based on frequency and probabilistic segmentation.

2 Objective

By coding some of these algorithms yourself and carry out small systematic experiments, you will gain a deeper understanding of how subword tokenization works and why it is essential in modern NLP applications.

3 Tasks

3.1 Byte Pair Encoding

In this lab, you will first implement **Byte Pair Encoding (BPE)** from scratch using basic Python functionalities. You will step through the core concepts of BPE, including vocabulary creation, pair frequency counting, and iterative merging.

Test the correctness of your implementation by reproducing the example in chapter 2.5.2 of “Speech and Language Processing”.

3.2 WordPiece

Second, you will implement **WordPiece** from scratch, again using basic Python functionalities. You may use a similar approach as for BPE, given that both algorithms use an agglomorative merging approach.

Test your implementation by reproducing the same example as in above Task above and compare the results of **BPE** and **WordPiece**. Then, choose additional examples to experiment with and compare the results and describe peculiarities.

3.3 Other implementations

Third, you will compare your implementation with other publicly available implementations of subwords tokenization. Focus again on **BPE** and **WordPiece**… and be open to look at others as well ;o) !

Take a moment and reflect on the different, existing algorithmic options, e.g., tie-breaking strategies for BPE. Choose the same examples to systematically compare and establish sturdy findings and take a a moment to describe them.

3.4 HuggingFace implementations

Fourth, you will utilize existing subword tokenizers from the HuggingFace `transformer` library to extend your comparison with previous results.

Building upon the previous tasks, keep your comparison consistent and take a moment to describe them.

4 Deliverables

Prepare a short documentation of your findings and work as a notebook.

In the first part, you will describe the findings by illustrating them with examples and/or summary statistics. Include references for sources used in the “Other implementations” task. Similarly, if your work can be supported by work of others, please cite the respective articles or blog posts.

In the second part, you will present your code using comments where appropriate.