Natural Language Processing
Dr. Ali Bahrainian
Published: 02.05.2024

University of Tübingen
Summer Term 2025

Assignment 2

Instructions

This assignment is ungraded, so there is no need to share your solution. Please complete this exercise sheet to practice what you learned in the lecture.
The assignment will be discussed in the tutorial on 09.05.2024 starting 15:50.

Exercise 1 - Implementing a tokenizer

Implement a basic whitespace tokenizer in Python from scratch without the use of any NLP libraries. This tokenizer should drop whitespaces and create tokens for the following cases:

(a) End-of-sentence (EOS) symbols, brackets and separators

(b) Abbreviations - Assume those are only one of the following: Ph.D., Dr., M.Sc.

(c) Special characters as in prices separated (i.e. $45.55)

(d) Dates - Assume that they follow the format dd/mm/yy (i.e. 01/02/06)

(e) URLs - Assume that they follow the format:
`http[s]://[...]`, (i.e. https://www.stanford.edu)

(f) Hashtags separated (i.e. #nlproc)

(g) Email addresses - Assume that they follow the format:
name@domain.xyz (i.e. someOne@brown.edu)

Apply your code on the test example below, which should yield the specified tokens:

```
test_text = "He has a M.Sc. in Math and she has a Ph.D. in NLP. A session
   costs 45.55$ or $50.00. As of 01/02/06, please email X/Y at someone@brown.
   edu or visit http://www.stanford.edu and if link does not work try https://
   www.stanford.edu instead. #test#test2#nlproc"
tokenized_text = ['He', 'has', 'a', 'M.Sc.', 'in', 'Math', 'and', 'she', 'has'
   , 'a', 'Ph.D.', 'in', 'NLP', '.', 'A', 'session', 'costs', '45.55$', 'or',
   '$50.00', '.', 'As', 'of', '01/02/06', ',', 'please', 'email', 'X', '/', 'Y
   ', 'at', 'someone@brown.edu', 'or', 'visit', 'http://www.stanford.edu', '
   and', 'if', 'link', 'does', 'not', 'work', 'try', 'https://www.stanford.edu
   ', 'instead', '.', '#test', '#test2', '#nlproc']
```

Exercise 2 - Implementing a BPE tokenizer

Implement a Byte Pair Encoder (BPE) tokenizer as shown in the lecture and apply it to a sample text. You are free with your choice of libraries. You can assume that the corpus only consists of a list of words.

Exercise 3 - Using pre-implemented tokenizers

Use an existing tokenizer from the T5 Transformer or any other tokenizer of choice from the HuggingFace library. Apply the tokenizer to a text sample of choice. Compare the output of this tokenizer with the two tokenizers you implemented in the previous questions and explain the similarities and differences.

Exercise 4 - RegEx

Assume we have a lookup table named `lookup` storing abbreviation definitions. Using regular expressions in Python, write code that uses `lookup` to replace abbreviations in any given text with their full-text counterparts. Apply your code to the following snippet so that `example` is transformed to match `target_output`:

```
lookup = {
    'doa': 'dead on arrival',
    'gf': 'girlfriend',
    'bf': 'boyfriend',
    'btw': 'by the way',
    'lol': 'laughing out loud',
}

example = "I was lol when my bf's phone was doa."
target_output = "I was laughing out loud when my boyfriend's phone was dead on
    arrival."
```