ANLP 2020/21; Uni Potsdam; D. Schlangen, B. Aktas

Work Sheet for Week 01

Introduction

As explained in the intro session, there will be a worksheet for you to work through every week. The idea of this is to give you some guidance in doing *active learning*, i.e., in taking the material from the lectures and directly applying it to questions.

The worksheet for the week will be released sometime on the Monday of the week. You can (and should) work through it in parallel to doing the reading for the week / watching the week's lecture. It will also be the basis for the group work during the practical sessions on Thursday. Lastly, you will also be asked to submit your answers by the end of each Friday. Based on your submissions (and the questions on the forum), we may add a short video on Monday with a recap of material from the previous week that raised the most questions.

The exercises / questions really are about your active, collaborative learning. So feel free to discuss them on the forum as well.

For the practical sessions: At the beginning of each practical session, select from this worksheet the exercises that you want to discuss in your group. Work through that list as far as you get. You should have looked at the work sheet before the session, and have attempted some or even all of the exercises. In the group session, you should discuss each other's solutions or ideas.

Technical Preparations

Make sure that you have a working Python environment available. We recommend the Anaconda Python Distribution. You will need Python 3.x, Jupyter, numpy. You should also make sure that you have the `nltk` package installed. (Later, you will also need PyTorch.)

Questions / Tasks for this week

Conceptual questions

[C1] Try to come up with a few potentially useful language tasks. For each task, give an *intensional* description, i.e. a verbal description.) Explain what the input and output would be, and why you would find the task useful. Also think about how you would collect examples for an extensional defintion of the task. Can you identify sub-parts of the task that are in themselves language tasks as well?

[C2] What are the limits of the framing of language use as doing language tasks? Can you think of language activities that aren't that easy (or even are impossible) to frame as a language task?

[C3] In what ways is a natural language different from a programming language?

[C4] What's so funny about the images collected in `language_failures.pdf`? (As we all know, jokes only get better when you explain them…)

Practical exercises

```
>>> from nltk.corpus import brown
>>> brown.categories()
['adventure', 'belles_lettres', 'editorial', 'fiction', 'government', 'hobbies',
'humor', 'learned', 'lore', 'mystery', 'news', 'religion', 'reviews', 'romance',
'science_fiction']
>>> brown.words(categories='news')
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
```

[P1] The code above shows you how you can use nltk to access the contents of the Brown corpus (a collection of English text of various genres). As a warm up exercise (and to test you python / nltk installation), compute the vocabulary size (how many different word types?) of the (news part of the) corpus.

[P2] What is the most frequent word in that corpus? Can you plot the frequency of the words, as a function of their rank? (That is, sort the words by frequency, with the most frequent one ranking highest, then plot the frequencies.) What do you observe?

[P3] How do you compute the dot product of two vectors in numpy? How do you add two vectors, elementwise?

[P4] Let's do some finite-state word morphology. Write a regular expression that accepts all forms of the German word "Mann" (check e.g. here). You can do this in python, or with some online regular expression recogniser (for example https://regex101.com).

[P5] Explore the different *tokenizers* (programms for splitting a text into the word tokens) offered by the NLTK toolkit in the `nltk.tokenize` package: `TweetTokenizer, SpaceTokenizer, ToktokTokenizer, TreebankWordTokenizer`. Create a short text (perhaps allowing yourself to move to genres like social media writing) for which the tokenizers disagree.

[P6] (Bonus Exercise) Use search & replace (with regular expressions) to turn English singular nouns into their plural forms. How well does that work?

[P7] (Super Bonus Exercise) Implement Byte Pair Encoding. (Code on p. 20 in JM3, ch 2.)