**Feedback for WS1**

We received 20 submissions for the first worksheet. The responses seem to be logical and mostly interesting. If you are unsure about your response or curious about a question, please try to discuss them with your classmates on Piazza. Here is some short feedback on the first submissions:

* C1: If you are unsure about the difference between intensional vs extensional task description, please refer to the first lecture's slides in course material.
* C4: Error analysis (i.e., detecting the errors and understanding the reasons of the errors) is crucial for improving the performance of implemented systems. There are syntactic, semantic, phonological, etc. ambiguities in the given images/texts which can result in unintended interpretations of the sentences. This exercise can be helpful for defining the (ambiguity) categories from instances.
* P1: In the question, we ask vocabulary size but many of the responses were finding the corpus size instead. Please be sure that what your code gives as an outcome is what the specifications describe. For the vocabulary size, you need to compute the number of unique words.
* P5: There are number of tokenizes implemented in nltk or in other nlp libraries. They serve for similar purposes but the outcome can be different depending on the text. For instance TweetTokenizer can recognize the smiley ";)" as 1 token whereas the same string is recognized by TreebankWordTokenizer as a combination of two tokens: ";" and ")". So, be careful in using the libraries and be sure that you use function/class the most suitable to the domain you are working on.
* P6: Computing the plural forms is known to be hard. The point of this exercise was more to make you see that even simple seeming tasks (with a very simple intensional description) can become quite hard to implement in the form of rules. Language is notorious for looking like it's rule-governed, but when you actually have to write down the rules, you discover that there are many many exceptions. Regular expressions are useful for capturing the patterns in strings. This is a good exercise to capture both regular patterns (boat --> boats, book--> books) and irregular exceptions (child-->children, sheep-->sheep) in the solution (as much as possible).
* P7: It seems that the question was a bit ambiguous. We simply intended to asked you to run the code for Byte Pair Encoding (from the text book) on your environment and look what it does.