Assignment Part 1

CS6370: Natural Language Processing Release Date: 18/02/2020

Tentative Evaluation Date: 28/02/2020

The goal of the assignment is to build a search engine from scratch, which is an example of an information retrieval system. In the class, we have seen the various modules that serve as the building blocks of a search engine. We will be progressively building the same as the course progresses. The first part of this assignment is to build a basic text processing module that implements sentence segmentation, tokenization, stemming/lemmatization and stopword removal. The Cranfield dataset will be used for this purpose, which has been uploaded separately on Moodle.

1. What is the simplest and obvious top-down approach to sentence segmentation for English texts?
2. Does the top-down approach (your answer to the above question) always do correct sentence segmentation? If Yes, justify. If No, substantiate it with a counter example.
3. Python NLTK is one of the most commonly used packages for Natural Language Processing. What does the Punkt Sentence Tokenizer in NLTK do differently from the simple top-down approach? You can read about the tokenizer here.
4. Perform sentence segmentation on the documents in the Cranfield dataset using:
   (a) The top-down method stated above
   (b) The pre-trained Punkt Tokenizer for English

   State a possible scenario along with an example where:
   (a) the first method performs better than the second one (if any)
   (b) the second method performs better than the first one (if any)
5. What is the simplest top-down approach to word tokenization for English texts?
6. Study about NLTK's Penn Treebank tokenizer here. What type of knowledge does it use - Top-down or Bottom-up?
7. Perform word tokenization of the sentence-segmented documents using
   (a) The simple method stated above
   (b) Penn Treebank Tokenizer

   State a possible scenario along with an example where:
   (a) the first method performs better than the second one (if any)

(b) the second method performs better than the first one (if any)

8. What is the difference between stemming and lemmatization?
9. For the search engine application, which is better? Give a proper justification to your answer. This is a good reference on stemming and lemmatization.
10. Perform stemming/lemmatization (as per your answer to the previous question) on the word-tokenized text.
11. Remove stopwords from the tokenized documents using a curated list of stopwords (for example, the NLTK stopwords list).
12. In the above question, the list of stopwords denotes top-down knowledge. Can you think of a bottom-up approach for stopword removal?

Submission Instructions:

1. **The template for the code (in python) will be provided** in a separate zip file and you are expected to fill in the template wherever instructed to. Note that any python library, such as nltk, stanfordcorenlp, spacy, etc can be used.
2. A folder named 'RollNo1_RollNo2.zip' that contains a zip of the code folder and a PDF of the answers to the above questions **must be uploaded on Moodle by 28/02/2020. Evaluation will be done by the TAs as part of the first progress meeting.**
3. Please include the names and roll numbers of each member of the team in the document.
4. All sources of material must be cited. The institute's academic code of conduct will be strictly enforced.
