Assignment 1

CS6370: Natural Language Processing

Submitted by:
R Rishaab Karthik (MM19B046),
B Aditi (MM19B022),

Contents

1 Solutions **2**
1.1 Sentence Segmentation: Top down approach 2
1.2 Is it always correct? 2
1.3 NLTK: Punkt Sentence Tokenizer 2
1.4 Sentence Segmentation methods 2
1.5 Tokenization: Top down approach 3
1.6 Word Tokenization methods 3
1.7 Word Tokenization Method 3
1.8 Stemming and Lemmatization 3
1.9 Which method is better? With Justification 4
1.10 Lemmatization 4
1.11 Stopword removal 4
1.12 Bottom up approach - Stopword Removal 4

2 References **4**

Toy search engine

1 Solutions

1.1 Sentence Segmentation: Top down approach

Finding the boundaries between one or more words that make up longer processing units, usually sentences, is known as sentence segmentation. By identifying the places where one phrase stops and the next begins, it seeks to discern between distinct sentences within a document. We know that sentences end with full stops/question marks/ exclamation marks. This is the top down approach which we can use for sentence segmentation in English texts.1

1.2 Is it always correct?

No sometimes there are acronyms/initials ending with a full stop which might be confused for end of a sentence. Thus this naive algorithm may not be a good algorithm Example: Alphabet is the parent company of Google. The CEO of Alphabet is S.Pichai.

1.3 NLTK: Punkt Sentence Tokenizer

This tokenizer divides a text into a list of sentences by using an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences. rather than just simply using full stops, sentence boundaries. It must be trained on a large collection of plaintext in the target language before it can be used.2 The punkt tokeniser in python NLTK library is a pretrained model.3

1.4 Sentence Segmentation methods

a. The top-down method may perform better than the pre-trained Punkt Tokenizer in cases where the text being segmented is noisy. Example: "hi.. how are you? " is correctly segmented into two sentences by the top down approach as ['hi.','how are you']. Whereas the punkt module doesn't identify the boundary between the two sentences.
b. Non standard use of punctuation and complex sentence structures are handled better by the punkt module. Example: "Despite the challenges, the research team was able to come up with a solution; the paper they published received acclaim from the scientific community." Should be divided as ['Despite the challenges, the research team was able to come up with a solution.','the paper they published received acclaim from

the scientific community.']. But the naive approach wouldn't work and segment it into two sentences at all.

1.5 Tokenization: Top down approach

The simplest top-down approach for tokenization of sentences would be to simply split the words based on the presence of space. One way of making this method advanced is by identifying punctuations and ensuring they are also tokenized separately.

1.6 Word Tokenization methods

In Penn TreeBank tokenization, the algorithm uses a Bottom-up approach, where it follows a 4-step method to tokenize words

- First, it splits contractions such as don't → do-n't
- Then, it uses its knowledge bank to seperate chatacters from punctuations for tokenization
- It seperates commas and quotes when followed by whitespace
- Finally, it segments the full-stops at the end of sentences

The first point in particular will not be executed in the naive top-down approach as it does not have access to the English word bank 4

1.7 Word Tokenization Method

1. Penn Treebank tokenizer does not perform well in presence of non-words or words which are particularly not present in the vocabulary. E.g 'u gotta see d doc b4'
2. The instances where Penn TreeBank package approach works better than the naive method would be, as stated in the previous answer, sentences with apostrophe incorporated in the word morphology. E.g. "I don't think he'll agree to this." 4

1.8 Stemming and Lemmatization

Both stemming and lemmatization are word normalisation techniques which aims to obtain the root word or stem from the given set of time or number dependant words.

Stemming is a relatively simple approach which essentially removes prefixes or suffixes to obtain the root word. For example, the word 'swimming' can be stemmed to obtain 'swim'.

Lemmatization on the other hand is an exhaustive method which refers to word morphology and maps all words to their original root as per the rules of english. For example, the word 'mice' is mapped to mouse or 'was' mapped to 'be'. 5

1.9 Which method is better? With Justification

As evident from the previous answer, we can say that lemmatization is better suited for search engine implementation due to its ability to exhaustively map words to their roots while taking account of context in opposition to stemming which is a surface-level rule based method. In addition, given that cranfield dataset consists of passages in the technical context, the presence of complicated jargon is expected yielding to lemmatization being the better suited method.

1.10 Lemmatization

Refer to the code in inflectionReduciton.py

1.11 Stopword removal

Refer to code in stopwordRemoval.py. We have used NLTK stopwords list.

1.12 Bottom up approach - Stopword Removal

As discussed in class a common bottom up approach to stop word removal would be a frequency analysis. Here we can identify the tokens which has more frequency, and classify them as noise words. We can use the inverse document frequency, which is given by the following equation.

$$log_2(\frac{N}{n})$$

$$N \ - \ no \ of \ documents \ in \ the \ collection$$

$$n \ - \ no \ of \ documents \ where \ the \ word \ appears$$

A higher inverse document frequency would imply the word has more importance.

2 References

1. Palmer, David. "Tokenisation and Sentence Segmentation." Handbook for Natural Language Processing, CRC Press, 2000, pp. 11–33.
2. Dhondge, Tanishka. "What Is NLTK PUNKT? - AskPython." AskPython. 2022 www.askpython.com/python-modules/nltk-punkt.

3. “Natural Language Toolkit — NLTK 3.2.5 Documentation.” Natural Language Toolkit — NLTK 3.2.5 Documentation, www.nltk.readthedocs.io/en/latest.

4. “NLTK:: Nltk.Tokenize.Treebank Module.” NLTK:: Nltk.Tokenize.Treebank Module, www.nltk.org/api/nltk.tokenize.treebank.html.

5. “NLTK:: Nltk.Tokenize.Treebank Module.” NLTK:: Nltk.Tokenize.Treebank Module, www.nltk.org/api/nltk.tokenize.treebank.html.