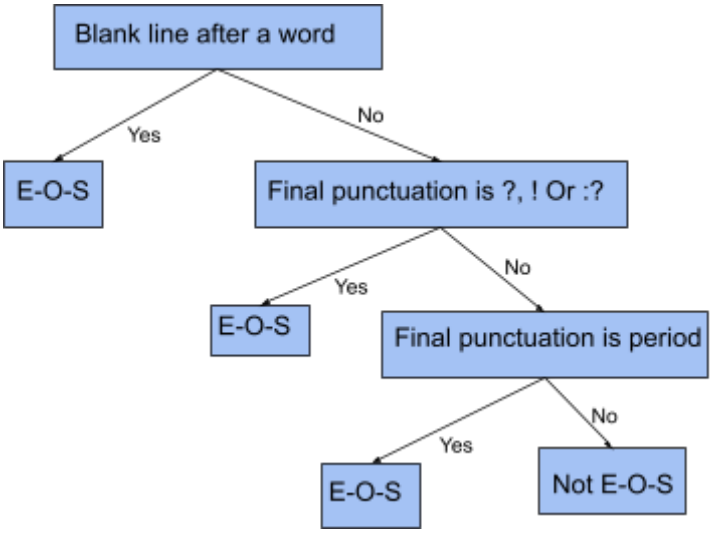
Natural Language Processing

Assignment 1 - Part 1

Team: Abdul Mooizz - cs17b034, Arabhi Subhash - cs17b005

1. A simple top-down approach is to have a decision tree like below



2. Below are a few examples where our top-down approach for sentence segmentation fails
    a. Decimal numbers having a period are treated as EOS (ex: .23, 3.4%)
    b. Abbreviations that end with or contain periods.
    c. Emails/Websites have periods but are not EOS. We need to use regular expressions to handle them
    d. Informal writings without proper punctuation
3. Punkt is designed to learn parameters (a list of abbreviations, etc.) unsupervised from a corpus similar to the target domain. The algorithm of the Punkt system uses an empirical approach in handling abbreviations even though the rules are similar to the top-down approach described above.

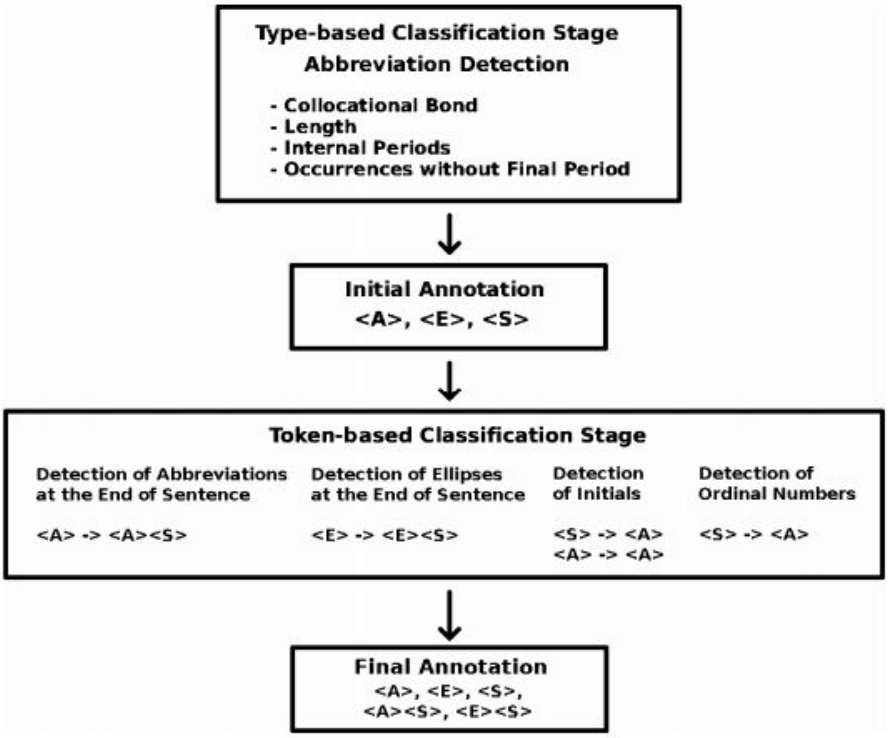


Figure 1
Architecture of the Punkt System.

4.

a) Consider the scenario where the author missed the spaces between the sentences
Example: “he is the one.he is not the one”
″, Top-down approach gives [‘he is the one.’ ,‘ he is not the one’] which is a better segmentation when compared with Punkt which outputs [‘he is the one.he is not the one’].

b) Consider the scenario where Text has abbreviations, Example text = “ Dr. Subbu told that the surgery is at 11 a.m.”
The top-down approach gives [“Dr”,”Subbu told that the surgery is at 11 a”,”m”]. Punkt recognizes the above text as a whole sentence.

5. The simplest top-down approach for tokenizing the words is to break the text at every character which does not belong to [**a-zA-Z0-9_]**.

6. Penn TreeBank tokenizer is a top-down approach that uses regular expressions to tokenize the words using the below rules.

- It splits common English contractions, e.g. don't is tokenized into do n't and they'll is tokenized into ->they 'll,
- It handles punctuation characters as separate tokens,
- It splits commas and single quotes off from words when they are followed by whitespace,
- It splits off periods that occur at the end of the sentence.

7.

a) Consider a simple scenario where the info required from corpus has nothing to do with punctuations. The Top-down approach output doesn’t contain punctuations but Penn Treebank Tokenizer treats them as separate tokens. The only problem is that the list of tokens in Treebank’s output is large which may cause computational problems.

b) Top-down approach fails with word contractions(“don’t”) , decimals(2.3)and hyphenated compound words(“father-in-law”). Treebank performs better in the following example: “Don’t trust the father-in-law”. The top-down approach gives [Don, t , trust , the ,father,in,law] and Treebank gives [Do, n’t, trust , the , father-in-law”]

8. Stemming and Lemmatization are Text-Normalization methods that mainly reduce inflections to base forms with some differences mentioned below.

- Stemming and Lemmatization both generate the root form of the inflected words. The difference is that stem might not be an actual word whereas, the lemma is an actual language word.
- Stemming usually refers to a crude heuristic process and Lemmatization employs a proper morphological analysis to arrive at the lemma.
- A Stemmer is far simpler to build than a lemmatizer. In the latter, deep linguistics knowledge is required to create the dictionaries that allow the algorithm to look for the proper form of the word.

9.

- In applications like search engines, we prefer better recall over precision. Stemming increases recall while harming precision. On the other hand, Lemmatization performs better morphological analysis resulting in precision for Information retrieval requests.
- Stemming methods tend to be simpler and faster than lemmatization methods.

Consider the case where the user searches with the word “celebrities”, as in the plural of celebrity, a stemming engine ends up with a stem of “celebr”. That search could end up with false positives from other words with the same stem as “celebrations”, but with lemmatization utilized by the search engine, the query is correctly interpreted as “celebrity”, not “celebration”, enabling the search engine to deliver precise results. Though this level of precision is not desired in most search engine based applications. One can also try different stemming algorithms which may increase precision.

10. Code

11. Code

12. Stop Words are the words that carry negligible information. This is the case with the most Frequently occurring words(“the”,”a”). One of the bottom-up approaches to remove stop words in a text.

- Calculate frequencies of words.

- Remove the words with a frequency above a certain threshold.
- Removing words that occur once, i.e., singleton words.
- Removing words with low inverse document frequency(IDF). IDF can be calculated inverse matching discussed in the class.

References:

- https://www.aclweb.org/anthology/J06-4003.pdf (Sentence Segmentation)
- https://www.nltk.org/_modules/nltk/tokenize/punkt.html (Tokenizer)
- http://www.ijfrcsce.org/download/browse/Volume_4/April_18_Volume_4_Issue_4/1524218332_20-04-2018.pdf ()