INFORMATION RETRIEVAL & SEARCH ENGINES

TERM PROJECT

AUTOMATIC TEXT SUMMARIZATION

ERHAN TEZCAN
14011062

INSTRUCTOR: Mehmet S. Aktaş

About the Project

This project aims to implement a text summarization technique, and use them on texts that were taken in a Q&A format, where there is a question and an answer, and we try to summarize the answer. Text Summarization is "Extraction" based, as "Abstraction" based summarization is harder and probably out of reach of this class.

Project Timeline as Stated in Project Proposal

TASK	FROM	TO
Extracting dataset	10 April	21 April
Implementing algorithms	24 April	12 May
Designing a GUI	15 May	26 May
Presentation Week	29 May	2 June

About the Dataset

The texts are in the form of Q&A (Question & Answer). They are taken from https://www.reddit.com/r/askscience/top/?sort=top&t=all&count=200. They were taken by hand, and are stored in a PostgreSQL DB. I chose to store this way because I thought it would be easier, and since the texts are not enormous it wouldn't be a problem. A data consists of {QuestionId, Question, Answer, Author, Topic}. Author is the user who asks the question, and answer is selected by me by reading the answers given there. Its not always the top answer which I chose, because sometimes they were too short or too technical (consist of numbers and symbols instead of words)

Algorithms

2 algorithms were used in this project.

1- Taken from the internet which can be found at this link: https://github.com/mark-watson/java_practical_semantic_web/blob/master/src/com/knowledgebooks/nlp/KeyPhraseExtractionAndSummary.java
2- An algorithm that is written by me, and using Porter Stemmer as the only code that isn't mine. When writing this algorithm, I mainly related to [1] and [2]. Other papers were researched too (see [3], [4])

About My Algorithm

My algorithm works very similar to the one written in [1]. We can write the steps like this:

1) Preprocessing
    1.1) Lower casing
    1.2) Sentence Segmentation
    1.3) Word Tokenization
    1.4) Stop Word Removal
    1.5) Stemming (Using Porter Stemmer)
2) Sentence Scoring
    2.1) Word Frequency
    2.2) Title Similarity (Question – Answer Similarity)
    2.3) Sentence Position
    2.4) Question & Author Related Words (Optional)
3) Sentence Ranking

As we can see, first we do preprocessing to lower the amount of words we are going to work with, and make things easier. Sentence Segmentation returns a list of sentences, and Word Tokenization returns the list of words for each sentence. While doing Word Tokenization if a word is Stop Word it is removed, and if not it is stemmed.

Then, to find out which sentence has more meaning we have few scores to look at. Word Frequency basically means: if a word has higher frequency it has higher meaning. If a word in answer is also in question it might also carry more meaning. Sentence Position gives a score such that the first sentence has higher meaning than the ones that come after it. Question and Author related words point to words that might be words that we use when answering a question, or referencing the one who asked it.

Finally we rank these sentences according to the score. After the ranking, we construct the summary according to few rules, that can be selected by the user from GUI.

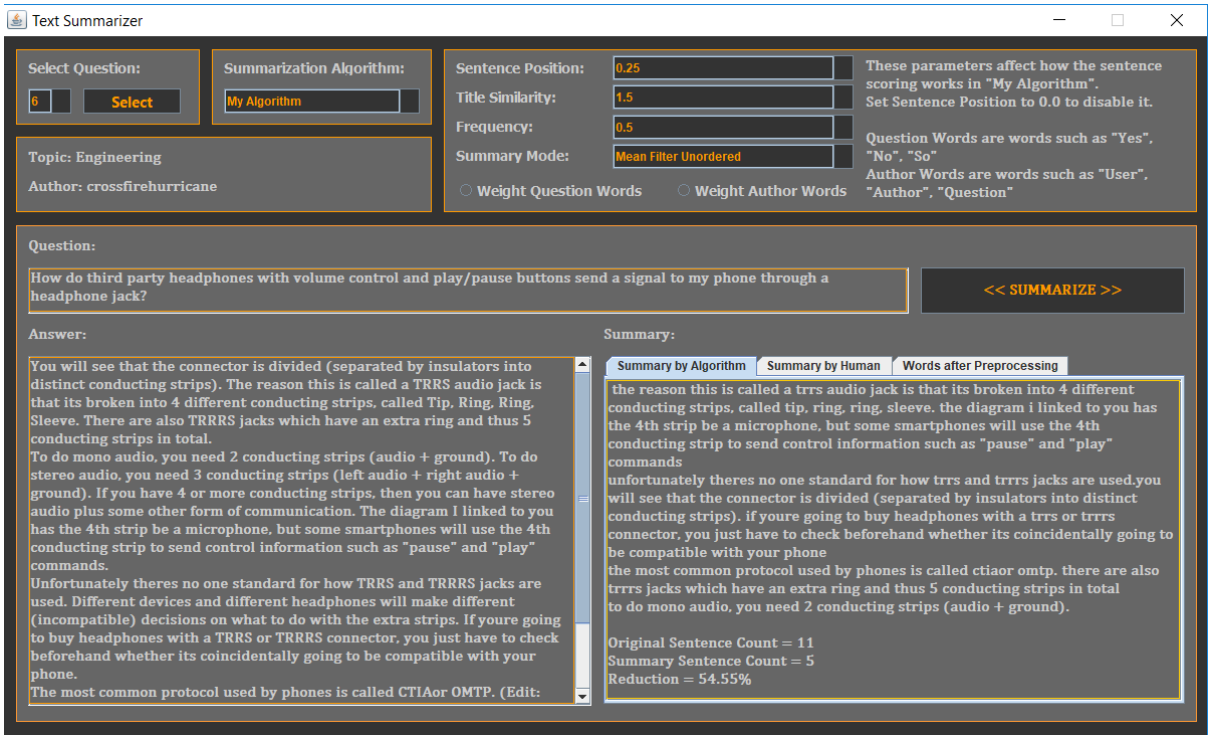
The rules are:

1- Mean Filter Unordered
   The first sentence has the highest score, and all the sentences above a mean value of scores are included in the summary. The sentences are in order by their score.
2- Mean Filter Ordered
   Like the first rule, but in this one the sentences are ordered according to the order they appear in the answer.

3- Sentence Ratio
No mean value is considered, instead it tries to make the summary have half of the sentence count of original answer. The sentences are ordered according to their score.

About the GUI

GUI is easy to use, here is a screenshot:



References

[1] : Priyanka Sarraf, Yogesh Kumar Meena, Summarization of Documents using JAVA, International Journal of Research & Engineering
[2] : Reginald Long, Michael Xie, Helen Jiang, Extraction Based Text Summarization
[3] : Rafeeq Al-Hasemi, Text Summarization Extraction System (TSES) Using Extracted Keywords
[4] : Tomek Strzalkowski, Jin Wang, Bowden Wise, A Robust Practical Text Summarization