**What does the system do?**

This project tackles text summarization. The system summarizes tech news articles from the BBC Datasets.

**How did I tackle the task?**

The task was tackled in several steps. The first stage aimed to gain an understanding of how summarization is done. For this, I consulted different sources on how to build summaries. I wanted to know what makes a good summary. I found that a good summary conveys the same idea as the original text but using less words. I initially thought to have the summarizer summarize small texts that I would personally annotate. For this I would I read some texts and then manually make a summary of them, to get an idea of the process. After, I wanted to translate this knowledge and process into a computer program. To make the system understand what the text is about and to extract the most important parts of it. This task of making the system understand was more difficult than what I expected so I tried a different approach. I decided to use the BBC Datasets [1] given that they were readily available and had different news texts. I found that news articles contain the most important information at the beginning [2]. The information is in the beginning because readers might not finish reading an article they started. With Zipf’s law I thought it reasonable to produce the summary from a subset of the data of the news article. The subset of the data would be comprised of the headline (title), leading sentence and first 20% of the article (percentage was chosen because it appears in Pareto’s principle without any further justification). Additionally, the chosen subset of the data would hopefully contain the most important information about the article. I also hoped that working with a subset of the data would reduce the complexity of the problem by reducing the amount of text to process.

Having clear that a subset of the data was to be analyzed, the subset of the data would then be passed through a pipeline that would give more information about the text and its structure. The initial pipeline consisted of several steps. First, separating the text into different sentences. Then tokenizing the text. After, perform POS tagging. Then, perform constituency parsing and dependency parsing. After passing the text through the pipeline the summary would be produced. Here, trying to identify the most important parts of a text from the resulting representations in each step of the pipeline proved to be difficult. I opted to devise a heuristic that could produce a summary from a text. For the heuristic, some elements of the pipeline were modified. The heuristic worked upon the result of constituency parsing of individual sentences. The results of the constituency parse were chosen because I thought it gave information about the structure of the text on a sentence level. The system would now not alter the information from the text, but extract parts from the original text to present as a summary. The design of the heuristic was based on manual inspection of the tech news articles of the data set. I noticed that news articles contain additional details that are not necessary for understanding the gist of the article. The devised heuristic aims to reduce these additional details from the text. To understand the parts that were going to be excluded from the summary I used Stanford’s parser [3] with some example sentences. What I was able to gather from the manual inspection is reflected in the heuristic. The heuristic tries to reduce each sentence to its simplest subject and predicate. For each sentence, the heuristic aims to get the first noun phrase and verb phrase. It extracts the first atomic noun phrase found in the sentence (one that does not contain other noun phrases inside of it). For the verb phrases, it includes every part before the first child noun phrase. When it finds a noun phrase it returns the atomic part of the noun phrase and repeats the process. The implementation of the heuristic is included in the code. An example of a sentence of the heuristic in action follows:

Original sentence: Ink helps drive democracy in Asia  
Constituency parse of sentence: [ Root S [ NP [ NNP [‘Ink’]] VP [ VBZ [‘helps’] NP [ NP [ NN [ ‘drive’] NN [‘democracy’] ] PP [ IN [‘in’] NP [ NNP [‘Asia’] ] ] ] ] ] ]   
Processed sentence: Ink helps drive democracy

This was the process of what I was able to do for the text summarizer

**System’s performance**

The system, at the time of writing does not correctly deal with processing of all types of VPs, I believe works in some news texts. It works in news texts that accommodate to the assumptions made. The assumptions that the most important information is at the beginning of the text. Nevertheless, I cannot say the extent of the success of the system since I do not have a metric that could evaluate the output of the system. I think that evaluating the success of the system would require a human annotated metric rather than an automated one like in evaluating a generated utterance. For instance, BLEU does not represent the quality of a generated utterance as a metric. Nevertheless, with human evaluation an idea of the success of the system can be achieved. Humans could evaluate how good the system extracts the most important information of the text. For this system, only manual inspections of the possible output were analyzed. There are texts and sentences where the heuristic returns sensible outputs like the example sentence above. Nevertheless, there are sentences where the result is not adequate. For example, the sentence *“In an effort to live up to its reputation in the 1990s as "an island of democracy", the Kyrgyz President, Askar Akaev, pushed through the law requiring the use of ink during the upcoming Parliamentary and Presidential elections.”* becomes *“the Kyrgyz President pushed IN requiring the use”* which does not make sense by itself.

Looking at some example outputs led to several thoughts regarding the system’s performance. Several issues were identified. First, using a single heuristic to process each sentence is not the best approach. This is because there is not a single way a sentence can be constructed. Sentences that do not follow the structure from which the heuristic was devised are not properly processed. Thus, the heuristic fails to extract the most important parts of the sentence. Additionally, the system might generate summaries where the sentences that compose the summary are not logically connected to each other. This is because the system is oblivious to the meaning of the sentence. Furthermore, the system’s output heavily relies on its input, since it does not transform the input, but extract parts of it. This means that a poorly written input text could lead to a poorly written output.

**Other issues**

While developing the system several challenges emerged. First, I believe my lack of linguistic knowledge made it more challenging to achieve this task. I did not explicitly know the terms for the relations between objects in a sentence. This meant that although I could intuitively summarize the texts, I found it difficult to translate this process to a system. As seen in the class, I realized that as a human the resulting produced summary depends on the target audience. This would mean that the summarizer could have some knowledge about the domain that the text addresses and the target audience to produce a fitting summary. I believe that there is not a universal consensus on what is relevant to include in a summary, it depends on the person.

Another issue with the system was that it relied on constituency parsing. If the constituency parsing was incorrect, then the output might not be appropriate. Moreover, the system cannot handle text that is taken from a conversation. In conversations the produced utterances might not fit a complete sentence or follow a standard structure. Finally, the model assumes that the most important information is present in 20% of the article. There are other texts that are not related to news that do not follow this structure.

**Future work**

I believe there are many ways this simple system could be extended. There are many exciting avenues to explore in the vast field of NLP. Here I include some suggestions that I came up with. As mentioned, the system can only handle certain types of texts with a specific structure and sentence construction. As such an extension could be to have different summarizers that perform well on different domains. Then have a consolidation algorithm that allows the different summarizers to vote on which sentences are the most important. From the result of the vote, the summary could be constructed.

Another extension has to do with tailoring the summary to a target audience. In texts relevant aspects to some people might not be relevant to another group of people. Then, the system could tailor the summaries based on a prototype of the person that is going to read the summary.

An interesting avenue that I would like to explore is trying to incorporate dependency parsing in the system. I believe dependency parsing gives a deeper understanding of the structure of a text or a sentence. Although I do not know exactly how it works, my initial intuition would be to identify the most important parts in each sentence by the number of connections to the other elements of the sentence or text. Then build the summary from the most important nodes.

If I had more time, I would like to invest it in broadening my linguistic knowledge. To be able to understand better how to formalize understanding and meaning in a computer system. I would like to understand the existing approaches for text summarization.

Finally, I believe this short task helped me to get closer to the challenges that are present in the field. To realize the complexity of formalizing language. Even though at first sight the task does not seem complicated. Additionally, it made me realize how great the human brain is, for it to be able to produce utterances almost with no effort. In short, the task gave me more perspective of NLP.

**References**

[1] D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006.

[2] <https://mcccharleston.org/how-to-write-a-newspaper-article-rules-and-examples/>

[3] https://corenlp.run/