Marc Fournier
Sabri Amer

Summary Grading

Architecture

Our system has several passes of grading for the summaries. These are described here and shown in the architecture diagram below. The first pass is the most basic, and merely compares the lengths of the documents. After all, the purpose of a summary is to condense information from a long article. If the summary is more than 80% of the length of the original document, then the summary is a failure in that respect.

Next, there is a keyword search to determine the overall relevance of the summary to the original text. We use this as a cursory check to make sure the summary is written on the proper topic. If the summary has too low a score on the keyword search, the system assumes it was not written for the given document and gives it a score of 0. Author and title are also considered in the keyword analysis. Author names and words occurring in the title of the document are given weighting equivalent to appearing as 1.5% of the document text. Words in the title tend to convey the main ideas of the document. Authors' names rarely appear in the text body, but are commonly used in summarization to describe the authors' intent or ideas.

Summaries that pass the keyword search are sent along to semantic triple analysis. The original document and summary are analyzed to find their semantic triples. The triples are then compared across document and summary to give a triples similarity score. This triples score counts for the majority of the points assigned to the summary. After all, a summary should convey as much information as possible from the original, while still being a condensed format. Finally, the document and summary parse trees are analyzed to find similarity. In this portion, similarity counts against score because we do not want to reward plagiarism. If two trees are found to be identical, the system gives a score of 0 for the entire summary. This is an extremely basic form of plagiarism checking (it requires exactly the same sentence in both), but it is unlikely that a sentence will be copied if there is no other plagiarism.

Technical Details

To use the grader, put the document and the summaries to grade in the `res` directory as plain `.txt` files. In the Main class, set `DOCNAME` to the document name and `SUMNAME` to the summary name (e.g., for document `deepblue.txt` and summary `my-summary.txt`, use `deepblue` and `my-summary`). Run the Main class. When it completes, it will print several scores.

As described earlier, our system has several passes of grading for the summaries. We begin with a keyword search. Both the original text and the summary are read in, and their text analyzed to count the number of occurrences of each word in each file. We then trim some common words (such as *the*, *and*, and *is*) and the top 1% of remaining words before running it through our modified Okapi BM25 ranking function. We originally planned to use normal BM25 in our analysis, but decided against it due to preprocessing concerns: each time the program ran, it would need to reprocess every document file, which would take a long time. Instead, we use the BM25 function with a modified IDF function that does not require preprocessing the entire library. The keyword score this generates counts as 30% of the total summary grade. Summaries that score less than 0.05 fail instantly; the rest move on to triples analysis.

Sets of sentences have their triples and parse trees analyzed using the Stanford parser. The Stanford parser does not work well with extremely long strings, so the sets are about 500 words each. Once the triples have been generated, they are compared between the two texts. Triples with simple subjects, such as *this*, *it*, or *she*, are removed from the set. Triples are scored based on how closely summary triples match document triples. A subject-relation-object match scores the most points, followed by subject-object match, subject-relation match, and, least of all, subject match. Total semantic triple score counts for 50% of the total summary grade. The last 20% of the grade comes from the parse tree checks. If parse trees in the summary are too similar to those of the original, the summary is flagged for plagiarism and receives a score of 0.

Analysis and Experimentation

In order to test some of the specific strengths and limitations of our system, we tested a series of experiments which were designed to stress particular features of the system's design. We began by testing the semantic triple comparison and the keyword matching to judge whether or not these components functioned well to grade the quality of the input summary.

The first test we conducted to determine the strengths of these components was to grade the two Deep Blue summaries that were provided to us as examples. This test served to verify that all components of the system were able to function in tandem and to address whether or not the system graded these summaries in a way we agreed with. One of the summaries matched few keywords but still had a mediocre triples score, meaning it was able to keep much of the information of the original without using the same wording. The other summary matched keywords well but matched very few triples, showing that the author did not keep much of the document's information in the summary (at least, not with similar wording). Our system places most of the weight on the semantic triples because we believe that a summary should deliver as much of the same information as possible. However, these two summaries received similar scores due to their opposite natures.

The second experiment that tested the quality determination aspects of the system was to grade our summaries of the Turing and Deep Blue articles and see if the system grades them similarly to the grades we received on the assignments. This test served to help us identify any issues with the grade distribution we placed on the various components of the system. When testing Marc's summaries of the articles, they performed markedly better than the first round of tests on the provided summaries. Both the Deep Blue and Turing summaries achieved high marks on the semantic triples and keyword matching, giving the assignments total grades in the B+-A range, which matches well to the grades received on the assignments.

After stress testing the core grading components of the system we moved on to determining the efficacy of our plagiarism detection software. We tested this component of our system by inputting summaries that we considered plagiarism and observing whether the system's judgement matched. The first example of plagiarism we tested on the system was to give identical documents for the summary and source documents. Initially this simply failed the length cutoff and did not enter the parse tree portion of the code but once we diabled this cutoff for the test the system proved that it was able to determine that this is not a valid summary for the input article and flagged it for plagiarism. We also tested this feature by giving a slightly modified sentence taken from the source document as a summary; the system was also able to successfully mark this example as plagiarism and failed the assignment accordingly.

Evaluation

The keyword score is not always especially accurate. During our testing, we tried using Marc's Turing summary as a summary for the Deep Blue article. It was able to pass frequency analysis by a narrow margin, but received an extremely low score for semantic triples analysis. The low score on triples analysis is expected, as the Turing and Deep Blue articles had wildly different subject material. However, the fact that it was able to pass frequency analysis in the first place was rather surprising. The two articles apparently share similar keywords, if nothing else. The total grade for the Turing summary as a summary of Deep Blue was about 10%, which is higher than we would recommend as an actual grade in that situation.

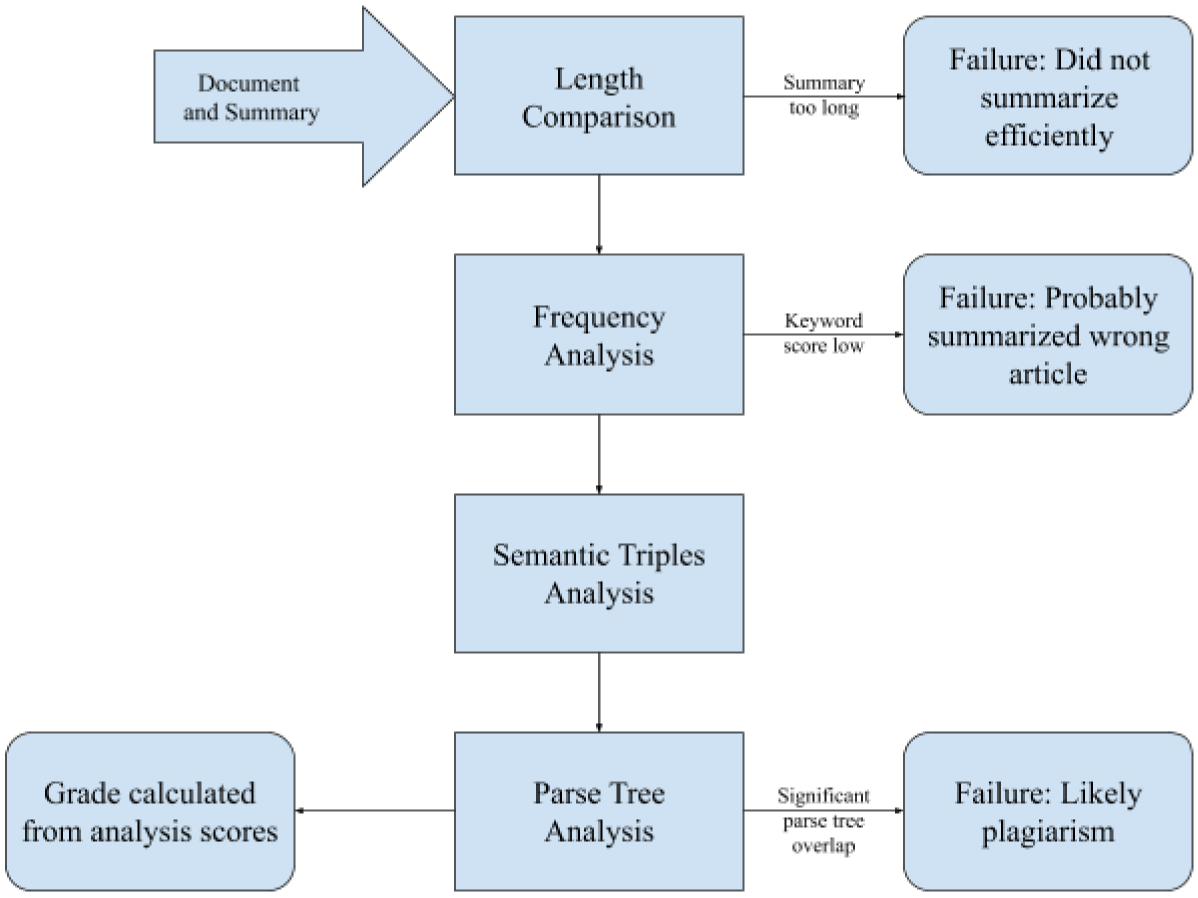
Document and Summary
Length Comparison
Summary too long
Failure: Did not summarize efficiently
Frequency Analysis
Keyword score low
Failure: Probably summarized wrong article
Semantic Triples Analysis
Parse Tree Analysis
Significant parse tree overlap
Failure: Likely plagiarism
Grade calculated from analysis scores

Diagram of the system architecture. Flow follows arrows, branching based on analysis results. Rounded rectangles represent terminal states.