CIS 430/530 Final Project: Summarization

Jason S. Mow
jmow@seas.upenn.edu

Nate Close
closen@seas.upenn.edu

1 Basic Systems

Below we illustrate our approaches and choice of parameters for the three basic summarization systems.

1.1 Centroid-Based System

The Centrality Summarizer has the following parameters to configure it's functionality:

- Vector Feature Weight Representation
- Similarity Comparison Approach
- Sentence Length Limits (short and long)
- Redundancy Removal Approach

We chose a binary representation for our sentence vector feature weight. We did this because it was the simplest to compute and yielded strong results in our preliminary testing. Our similarity approach was to use cosine similarity on the sentence vectors.

Our sentence length limit was between 15 and 50 words, tokenized by NLTK. We mitigated redundancy by rejecting any sentences with a cosine similarity greater than 0.75 with any sentence already in the summary. Again, these thresholds were chosen as they yielded the most sensible and well-scoring results from our trials.

1.2 Topic-word Based System

The Topic-Word Summarizer has the following parameters to configure it's functionality:

- Sentence Score Normalization
- Topic Word Cutoff
- Sentence Length Limits (short and long)
- Redundancy Removal Approach

For sentence vector feature weight, we chose the third representation which calculates weight as (# of topic words / # of nonstopwords). This choice seemed most logical to us as it doesn't dilute the score with stopwords, and also normalizes for sentence length.

Topic Word Cutoff was set to 0.1. This was the default setting and was not adjusted / tested extensively due to time constraints with regenerating topic word files. We deemed this to be an optimal setting after testing out the results on several different cutoff thresholds.

Our sentence length limit was between 15 and 50 words, tokenized by NLTK. We mitigated redundancy by rejecting any sentences with a cosine similarity greater than 0.75 with any sentence already in the summary. Again, these thresholds were chosen as they yielded the most sensible and well-scoring results from our trials.

1.3 LexPageRank System

The LexPageRank Summarizer has the following parameters to configure it's functionality:

- Edge Similarity Threshold
- LexRank End Criteria - ¡ 0.001 change
- Sentence Length Limits (short and long)
- Redundancy Removal Approach

For the LexRank summarizer, we chose to use TF-IDF representation over binary representation. This produced more accurate vectors and better results from ROUGE in the summarization.

For edge similarity threshold, we chose the value of 0.2. This was suggested in the reference text discussing Lex Page Rank, and we found it to be fairly successful. For this value, too, we had limited ability to vary and continue to experiment as the process of generating summaries was extremely time-consuming.

Our LexRank End Criteria was set such that the iteration would end if all values changed less than 0.001 between iterations. This was a good medium between performance and getting reasonable results. Also, the results did not change much as the threshold was decreased further.

Our sentence length limit was between 15 and 50 words, tokenized by NLTK. We mitigated redundancy by rejecting any sentences with a cosine similarity greater than 0.75 with any sentence already in the summary. Again, these thresholds were chosen as they yielded the most sensible and well-scoring results from our trials.

1.4 Performance

Initial testing was done using only the files in the "input/d30001t_raw" directory. This was done primarily for speed and ease, though we found that our results in these tests did not correlate particularly well with our scores using the whole 50-directory corpus.

System	Centroid	Topic-word	LexPageRank
Rouge-2 Recall	0.11414	0.10670	0.11911

The above recall score for Topic-word was obtained using a topicWordCutoff of 12.5. We intended to use this score for the full scale trials, however the time required to regenerate these files for the full corpus was too much.

As a comparison, running the baseline summaries of the files in the same corpus yielded a Rouge-2 Recall score of 0.09926. As you can see, all three of our summarization implementations out performed the baseline for files in this directory.

2 Custom Summarization System

2.1 System Design

Our system was designed to take advantage of tools that we have learned about, implemented, and used this semester to generate a custom summarizer based on our own heuristics. In designing our system, we decided to emulate and extend the functionality of a summarizer we are aware works quite well - the first sentence of each input document.

The summarization system extracts the first and last sentences of each input document. This expands the corpus from the aforementioned summarizer to be considered for our final summary.

Once the candidate sentences have been collected, they are assigned scores based on their similarity (much like in the Centroid Summarizer) and their use of topic words (much like the Topic Word Summarizer). It is important to note that the centroid and topic words referenced are with regards to all sentences in the corpus, not just our selected subset. This ensures that the selected sentences will be scored based on their reprsentation of the entire text, and not just the limited selection we have started with.

The scored sentences are then considered for validity and processed for word replacement. The final portion of our custom summarizer replaces the least frequent nouns and verbs with replacement synonyms. This was an attempt to delve in the abstractive summarization realm whereas our previously implemented summarizers had all been extractive. We believe that the quality and readability of the summary can be improved by replacing some of these words. At the same time, we opted to replace infrequent words rather than very frequent words as it would have put the clarity of key ideas and concepts in our summary at risk.

2.2 Resources & Tools Used

The tools that we utilized for our custom parser included various NLTK modules for basic parsing and word manipulation, the NLTK WordNet module, the NLTK part-of-speech tagging module.

The WordNet module was used to find SynSets and synonyms that were viable for replacement in our summaries. Likewise the POS tagging module was also used in order to find strong matches amongst synonyms being considered to replace words in the summary. These tools were very valuable and allowed us to explore the possibility of incorporating abstractive concepts and strategies in summarization.

2.3 Performance

3 Discussion and Analysis