Automatic Summarization

Matthew Calderwood
University of Washington
calderma@uw.edu

Kirk LaBuda
University of Washington
kwlabuda@uw.edu

Nick Monaco
University of Washington
nickmon@uw.edu

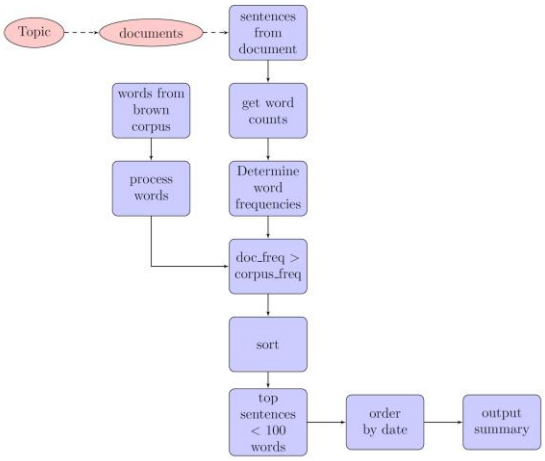
Abstract

The purpose of this project is to participate in a shared task of automatic summarization. Our goal is to break a large problem into manageable subtasks, incrementally improve our system, and evaluate the results at each step. The data comes from the TAC 2010 Guided Summarization shared task, and ROUGE is used for evaluation. Our system has three main components: preprocessing, content selection, and content realization. Our model is extractive, so focus falls primarily on content selection. Preprocessing is used to collect the text and weed out any irrelevant information. Content realization is trivial, since we simply output the extracted sentences in chronological order. However, we may modify this process in order to shorten sentences or improve the readability of our summaries.

1 Introduction

With the vast amount of publically available text information on the internet, one of the most important uses of language processing is to help us query and extract meaning from these large repositories (Jurafsky and Martin, 2008). This project aims to implement a working system for automatic summarization, based on the information presented in class and our knowledge from the program thus far. According to Mani (2001), the goal of automatic summarization is to take an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's needs. While our implementation will not have any practical applications outside of the classroom, we hope to improve our system over time, to the point where it can produce coherent summaries.

2 System Overview



Our system architecture is fairly similar to the diagram in the course notes. First, topics and their relevant document IDs are extracted from the TAC 2010 Guided Summarization task data file. News articles for each topic are located by searching for the document IDs within the aggregated data sets of different publishers. Once found, the article is parsed to extract only the plain text. Paragraph markers are discarded, though this information may be used in the future.

Each article is segmented into sentences and words. The sentences are kept intact since our model is extractive, while the word segmentation is used to store frequency counts for the words within each document.

We decided that it would be convenient to avail ourselves of NLTK's stopword set and the Brown Corpus in Python. We used the WordNet lemmatizer to lemmatize the brown corpus and access the frequency distribution based on these lemmas.

From here, we computed and stored the log probability of every sentence in our topic set from word frequencies within the topic documents. We compared our computed likelihood of every sentence with the probability based on the lemmatized Brown corpus probability using tf*idf. This gave us a relative idea of how important or thematic a given word was to our summary.

Where the difference was greatest, we knew that we had a sentence that was valuable for our summary. We then decreased the importance of these words to try and eliminate repeats of the same information. We stored this computed difference for all sentences. The sentences with the greatest difference were used first in our summaries, and we continued to add sentences while the summary was less than 100 words.

After finishing our summary, we used the heuristic of reordering the sentences by the date of the article they occurred in: earlier publication meant the sentence would occur earlier in our summary. This was just a heuristic, but it seemed intuitively correct to us: logically, earlier occurring articles will be more general - the greater the distance between the occurrence of the event and the publication of an article about it, the more fine-grained and detailed the article is likely to be. This is very likely an area we could tweak or experiment with to improve our results.

After this reordering, we simply output our results and used our rouge script to evaluate the results.

3 Approach

As stated above, we selected content based on our tf*idf computation. Our system computed term frequency in the document set for a given topic, and computed the probability of a given sentence. Separately, we computed the probability of that sentence based on a lemmatized version of the Brown corpus and NLTK's frequency distribution tool. We then chose the sentences that had the greatest difference between these two values as the best representatives for the document set.

As discussed above, after the content selection phase we ordered the sentences within the summary based on the date of the article they occurred in. This assignment didn't have too much required for content realization - we simply made sure that our summaries were less than 100 words as requested.

As a starting point, we're happy with the rough system we have right now. It gives us relatively coherent extractive output, though there is room for improvement. We had minor snags figuring out how to preprocess the documents and consolidate articles together, coordinating our output with the rouge configuration file, and figuring out what the best heuristics for content selection and information ordering were, but we think we found reasonable solutions given the timeframe and the goals of this assignment.

4 Results

Our ROUGE scores dropped consistently from ROUGE1-ROUGE4. ROUGE1 scores were our highest, and our Average_P value was consistently the highest out of F, R and P. Our average R-scores are shown below.

1 ROUGE-1 Average_R: 0.10987 (95%-conf.int. 0.09229 - 0.12813)
1 ROUGE-2 Average_R: 0.01891 (95%-conf.int. 0.01412 - 0.02389)
1 ROUGE-3 Average_R: 0.00502 (95%-conf.int. 0.00317 - 0.00720)
1 ROUGE-4 Average_R: 0.00129 (95%-conf.int. 0.00039 - 0.00242)

5 Discussion

Class 7 has some baseline ROUGE-2 scores for LEAD and MEAD. It seems the average of the LEAD/MEAD scores is ≈.05. Our average ROUGE-2 result was ≈.02 (.01891). For a first pass, we are relatively satisfied with our results and confident that we can improve on these scores. Potential areas for improvement are tweaking our heuristics for content selection and information ordering. For the latter it may make sense to dispense with our date-of-publication heuristic if we come upon other ideas. Additionally, we are planning on implementing a machine learning approach by aggregating features such as those discussed in class (discourse analysis, sentence length, average and sum of probabilities, etc.) and using those as datapoints for each sentence. We then will use combinations of these datapoints (sentences) and run a subset of them through an ML algorithm (to be determined) and use their computed Rouge scores as the target values. We will then use this model to select the best combination of sentences based on our model. This idea is in progress, and the fine details are not hashed out, however this is the idea at a high level.

6 Conclusion

This project is very much a work in progress. At present, the architecture of our system is laid out and we have a baseline to measure future results against. Our goal is to work collaboratively to implement NLP ideas and techniques, improve our automatic summarizer, and learn from our experience in a shared task.

References

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, NJ.

Inderjeet Mani. 2001. *Automatic Summarization*. John Benjamins Publishing, Philadelphia, PA.