LING 573 - MultiDocument Summarization System

Audrey Holmes
University of Washington
auholmes@uw.edu

Eslam Elsawy
University of Washington
eslam@uw.edu

Masha Ivenskaya
University of Washington
marliven@uw.edu

Abstract

A short high-level overview of the paper (around 150 words).

1 Introduction

2 System overview

A description of the major design, methodological, and algorithmic decisions in your project. It often includes a schematic of the system architecture.

3 Approach

3.1 Data

We use the TAC-2010 dataset that consists of the following:

- A set of n topics $T = \{T_1, T_2, ..., T_n\}$.
- For each topic T_i, a set of m documents $D_i = \{d_{i,1}, d_{i,2}, ..., d_{i,m}\}$.
- For each topic T_i a vocabulary V_i.

3.2 Content Selection

Sentence Extraction and Preprocessing. For each document, we extract the contents of each document from the SML file using Python's BeautifulSoup module. We then convert the text to lowercase and perform sentence tokenization.

Sentence Similarity. For each document $D_{i,j}$, we convert the sentences into vectors s of length $|V_i|$, where s_v is the number of times word v appears in the sentence. From these vectors, we create an N x $|V_i|$ matrix M_i, where N is the number of sentences within a topic. We then perform a tf-idf transformation on M_i. Finally, we transform M_i into a symmetric N x N matrix of pairwise cosine similarities.

PageRank. We binarize the entries of M_i such that values less than 0.15 are set to 0; all other values are set to 1. We then implement the Google PageRank to estimate sentence importance.

Other Similarity Measures. In addition to tf-idf cosine similarity, we experimented with using Doc2Vec to measure sentence similarity. We trained a doc2vec model on the NLTK Reuters corpus and then used this model to calculate similarity scores between sentences. These scores were then binarized and ran through the PageRank algorithm. We experimented with different binarization thresholds, as well as with removing stopwords from the sentences before measuring similarity. However the Rouge scores obtained through this approach were lower than the results obtained using tf-idf cosine similarity. Our hypothesis is that the contextual similarity captured by doc2vec is not the right match for the document summarization task - i.e. if we are measuring how connected a sentence is to other sentences in the topic, we are interested in actual overlap of entities and concepts, not broad semantic relatedness (i.e. a sentence about dogs being related to a sentence about cats).

3.3 Information Ordering

For our baseline appraoch, we ordered the sentences by descending PageRank score and cut off each summary at the maximum word limit. We also implemented a cosine similarity reranker, skipping sentences that are too similar (threshold of .20) to sentences already in the summary.

3.4 Content Realization

We truncate the summaries so that they do not exceed 100 words.

Table 1: Results

Similarity Measure	Information Ordering	Precision	Recall	F-score
Cosine Similarity	Dummy			.13
TF-IDF Cosine Similarity	Dummy			.26
Doc2Vec	Dummy	.18	.23	.20
TF-IDF Cosine Similarity	Cosine Reranker			.28
Doc2Vec	Cosine Reranker	.20	.22	.21

4 Results

The table below summarizes our results using different similarity measures and reordering algorithms.

5 Discussion

Next steps will be to improve information ordering and content realization. In particular, we will work on removing redundant sentences from the final summaries. Since we got most of our improvements from improving pre-processing and parameter tuning, we will do more of that in the PageRank portion of the code. We noticed that the gold standard summaries seemed to have shorter sentences than our summaries, so we will look deeper into this to see if there is something we can improve.

6 Conclusion

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.