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

1 Introduction

2 System overview

Figure 1 shows the architecture of the baseline summarization system. First, sentences from documents belonging to the same topic are extracted. Then, adjacency matrix is built with pairwise tf-idf cosine similarities. After that, PageRank algorithm utilizes the similarity matrix to sort the sentences based on their importance. Finally, sentences and importance scores are fed into the reranker module which picks the sentences to consider in the final summary.

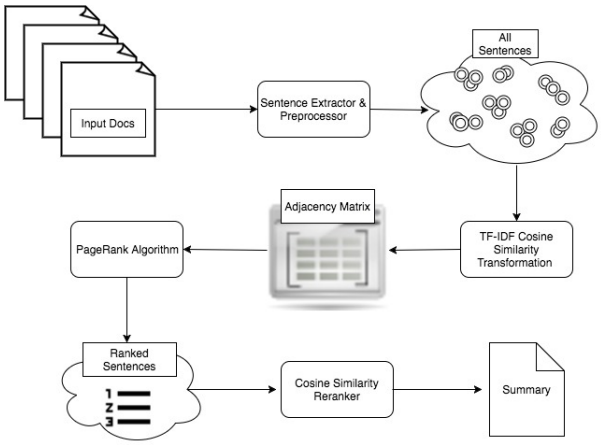

Figure 1: System architecture

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 XML 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 (Page et al., 1999) (Erkan and Radev, 2004) 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 full 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.

Table 1: ROUGE-1 evaluation results for different baseline system configurations

Similarity Measure	Information Ordering	Precision	Recall	F-score
TF-IDF Cosine Similarity	Dummy	.29	.24	.26
Doc2Vec	Dummy	.18	.23	.20
TF-IDF Cosine Similarity	Cosine Reranker Thr=0.2	**.3**	**.26**	**.28**
Doc2Vec	Cosine Reranker Thr=0.2	.20	.22	.21

Table 2: ROUGE-L evaluation scores for baseline

ROUGE-L	Precision	Recall	F-Score
ROUGE-1	0.3	0.26	0.28
ROUGE-2	0.08	0.07	0.08
ROUGE-3	0.03	0.02	0.03
ROUGE-4	0.01	0.01	0.01

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 Improving Content Selection

To improve upon our baseline metrics, we explored several different pre-processing options. First, we removed sentences with with fewer than 35 characters, as these seemed to be sentence fragments or short quotations. We also removed certain types of metadata within the sentences and tried lemmatization and stemming. Finally, we improved upon the *tf-idf* approach.

Metadata Removal. We used regular expressions to remove the sentences that contained phone numbers or websites. We also removed extra location phrases such as **CHICAGO, Illinois (AP) –**.

We also tried removing ages and acronyms in parentheses such as Anna**, 36,** and Papua New Guinea **(PNG)**. However, these changes did not improve performance.

Lemmatization and Stemming. Python's NTLK toolkit has several options for lemmatization and stemming. First we tried the WordNet lemmatizer. However, this option did not seem to have very good coverage, as there were many words in our corpus that did not get lemmatized.

The NTLK package also implements a Porter stemmer and a Snowball stemmer. The Porter stemmer lowered ROUGE scores. We think this is because the stemmer was too aggressive, often removing too much of a word and thus losing important information. The Snowball stemmer, on the other hand, was less aggressive, and provided a slight improvement in ROUGE scores.

Vectorization. We improved upon our baseline *tf-idf* vectorizer by using a binary term frequency for the numerator (keeping the inverse document frequency denominator the same).

We also experimented with using unigrams, bigrams, and trigrams. We were somewhat surprised to find the best results with unigrams. Our hypothesis is that bigrams and trigrams might be more helpful with a larger corpus.

4 Information Ordering

For our baseline approach, 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 (Radev et al.,), skipping sentences that are too similar to sentences already in the summary.

Figure 2 shows that the best achieved results happen at cosine similarity threshold = 0.2

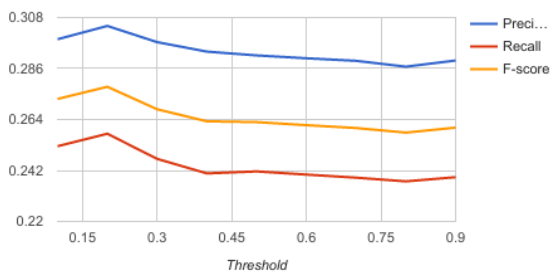

Figure 2: Precision, Recall and F-measure at different reranker cosine similarity thresholds

We also implemented a module that reorders

the sentences that have been selected for the summary. We experimented with several approaches to reordering, summarized below. Reordering of the sentences had little or no effect on the Rouge scores, so we discuss the advantages and disadvantages of each approach based on qualitative analysis of the output summaries.

Chronological Reordering In this approach, we ordered the sentences of the summary based on the publication date and time of the original documents. In those cases when two sentences came from the same document, or when the date/time of publications coincided, sentences were ordered in order of appearance in the documents(s).
This ordering seemed to do well for topics about events that unfold, and get reported on, over time, such as natural disasters, for example:

"the papua new guinea (png) defense force, the police and health services are on standby to help the victims of a tsunami that wiped out several villages, killing scores of people, on png's remote north-west coast friday night. igara said reports so far indicated that a community school, government station, catholic mission station and the nimas village in the sissano area west of aitape had been completely destroyed, where 30 people were dead. the death toll in papua new guinea's (png) tsunami disaster has climbed to 599 and is expected to rise, a png disaster control officer said sunday."

It was problematic for many other topics, as the following summary illustrates:

"for example, new roads will be banned in national forests around the park, servheen said. fish and wildlife service is poised to remove the park's renowned bears from the endangered species list. federal wildlife officials estimate that more than 600 grizzly bears live in the region surrounding yellowstone in idaho, montana and wyoming. grizzly bears in and around yellowstone national park should be removed from the endangered species list after 30 years of federal protection, the u.s. department of interior said tuesday. the only other large population of grizzlies in the united states is in and around glacier national park."

Cohesion Reordering In this approach, we first created all permutations of the sentences within the summary. For each permutation, we calculated a cohesion score as a sum of the cosine similarity scores of adjacent sentences. We then picked the order with the highest cohesion score.
This approach worked well for certain topics, especially those that concern a general issue that doesn't have a strong chronological component:

"in the united states, 21 percent of known species are threatened or extinct. the survey, published online by the journal science, studied the 5,743 known amphibian species and found that at least 1,856 of them face extinction, more than 100 species may already be extinct, and 43 percent are in a population decline many for unknown reasons. the researchers called for efforts to protect the habitat of amphibians and to reproduce the threatened species in captivity. habitat decline, from deforestation to water pollution and wetlands destruction, threatens them because the animals live both on land and in water."

However, it was often problematic for topics where new information was reported over time:

"burke was in the family's boulder home when 6-year-old jonbenet was found beaten and strangled dec. 26, 1996. hunter took the jonbenet case to the grand jury shortly after a former boulder police detective on the case and three former friends of the ramseys publicly demanded that colorado's governor, roy romer, replace hunter on the case with a special prosecutor. although the police chief and district attorney both have said that the ramseys fall under "the umbrella of suspicion," they have not formally named any suspects. police say her parents, john and patsy ramsey, remain under suspicion. "

Entity Grid Reordering

We implemented the apporache described in (Barzilay and Lapata, 2008).

For training we used the training dataset in AQUAINT, For each topic we pick two documents, and for each document we consider the original ordering of sentences and build the feature vector for it as a positive (i.e. good cohe-

Table 3: ROUGE-L evaluation scores for system with improved content selection and information ordering

ROUGE-L	Precision	Recall	F-Score
ROUGE-1	0.31155	0.27056	0.28879
ROUGE-2	0.08850	0.07684	0.08199
ROUGE-3	0.03017	0.02596	0.02780
ROUGE-4	0.00861	0.00739	0.00792

sion) training sample, and then randomly order the sentences and build the feature vector as a negative (i.e. bad cohesion) training sample. Our training dataset has 174 samples. For co-reference we built noun clusters based on lexical match, we used Stanford entity recognizer (Manning et al., 2014) for detecting named entities. For detecting grammatical roles we used Stanford dependency parser (Klein and Manning, 2003), we detect four roles, (S) subject, (O) object, (X) other, (-) missing. We don't distinguish between focused and not focused entity clusters based on frequency.

At run time, we use cosine based similarity approach to pick an initial ordering for the candidate summary, then we consider the original ordering plus 20 different random ordering and pick the summary which achieves the best coherence score. To determine the coherence score we used the model built during the training stage and built a KNN classifier with K=11. We used sklearn KNN classifier which can predict the probability that the test sample belongs to the class of high coherent training samples, then we pick the ordering which achieves the highest probability.

Table 4 shows a sample output of this module, the initial ordering of sentences was 1, 2, 3, 4 which ahcieved coherent score of 0.55 after running this module the best coherent ordering was 2, 1, 4, 3 which achieved coherent score of 0.73

4.1 Content Realization

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

5 Results

Table 1 summarizes our results using different similarity measures and reordering algorithms. The best achieved results occurred when using TF-IDF cosine similarity and cosine sentences reranker with threshold = 0.2.

Table 2 shows best achieved ROUGE-L (Lin,) evaluation scores for our baseline system.

Table 3 shows the best achieved ROUGE-L evaluation scores using improved content selection and information ordering.

Next steps will be to focus on improving content realization. We could improve upon redundant proper nouns and instead opt for appropriate pronouns. In general, we could improve our co-references. We could also make improvements so that more sentences are grammatically correct (fewer fragments, misplaced clauses, etc.).

6 Conclusion

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Dragomir R Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. Mead-a platform for multidocument multilingual text summarization.

Table 4: Sample summary

Id	Sentence
1	the announcement comes just two weeks after merck pulled its painkiller, vioxx, which is in the same class of drugs as bextra, from the market because a study showed that the risk of heart attacks doubled for patients who had taken vioxx for 18 months or longer.
2	vioxx was approved after trials held under the auspices of the food and drug administration showed it to be effective (which it was).
3	gilmartin was clear that the trial should be halted and that the drug might have to be taken off the market.
4	the drug was not pulled at that point.