LING 575: Summarization Deliverable #3 Report

Ben Cote
University of Washington
bpc23@uw.edu

Mohamed Elkamhawy
University of Washington
mohame@uw.edu

Karl Haraldsson
University of Washington
kharalds@uw.edu

Alyssa Vecht
University of Washington
avecht@uw.edu

Josh Warzecha
University of Washington
jmwar73@uw.edu

Abstract

will be added in a later deliverable

1 Introduction

This paper presents a system for multi-document summarization (the "System"), focusing on the TAC 2010 Guided Summarization shared task. The AQUAINT and AQUAINT-2 corpora serve as the training, development (devest), and evaluation (evaltest) datasets for this task.[1] The specific task is to produce summaries of documents in "Document Set A," which are divided into approximately 44 topics, each of which is classified into one of five categories: accidents and natural disasters, attacks (criminal/terrorist), health and safety, endangered resources, and investigations and trials (criminal/legal/other). The System alternatively selects content using TF-IDF scores or TextRank scores. It relies on an application of the Traveling Salesperson Problem (TSP) to order sentences coherently and a naive mechanism for content realization. Results are evaluated against gold standard summaries using ROUGE-1 and ROUGE-2 metrics.

The rest of this paper is organized as follows. Section 2 presents the overall system architecture. Section 3 details the major subcomponents of the system. Section 4 describes the evaluation results of the system, while section 5 analyzes and interprets those results. Lastly, Section 6 concludes the paper and identifies potential areas of improvement.

2 System Overview

Figure 1 shows the flow of our system, from input configuration through the summarization system, outputting both a collection of summaries and an evaluation of those summaries.

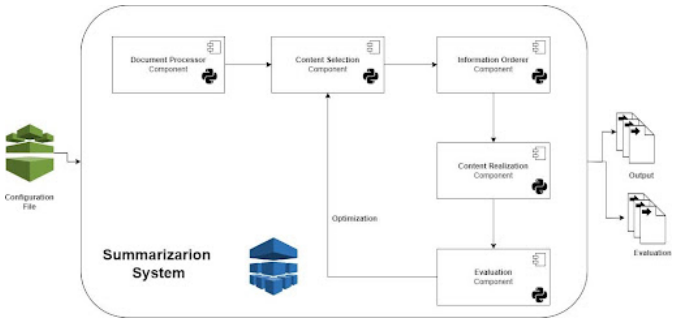

Figure 1: A graphical depiction of our summarization system

3 Approach

Our approach has four major components: (1) Preprocessing; (2) Content Selection; (3) Information Ordering; and (4) Content Realization. This document reflects the state of the system as of its initial implementation (v1.0). Each component is detailed below.

3.1 Preprocessing

The System ingests the raw XML files and processes them for downstream summarization. Ingestion relies on a document index file, which identifies the unique identifier (the document ID) for each relevant document in the AQUAINT, AQUAINT2, and 2009 TAC corpora. Those XML files are thereafter loaded into memory. Once in memory, each document is segmented into sentences using the nltk library. Subsequently, each sentence is tokenized (again using the nltk library. The system then generates processed data files for each document. These files include the document ID along with a handful of metadata. There is a blank line after the metadata. After that blank line, each tokenized sentence is presented on a new line. Para-

[1] The AQUAINT and AQUAINT-2 corpora are collections of English language news articles taken from the New York Times, the Associate Press, and Xinhua News Agency

graphs are separated by a blank line. These files, which are minimally processed, are the basic input for the System's content selection components.

3.2 Content Selection

The system is designed to identify and select the most informative sentences from a collection of documents. It implements two methods for content selection: term frequency-inverse document frequency (TF-IDF) and TextRank (Mihalcea and Tarau, 2004). These methods are implemented in Python, using several well-established libraries to efficiently process and analyze text data.

3.2.1 TF-IDF

TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a corpus (Manning et al., 2008). This relevance is determined by multiplying two metrics: the term frequency (the raw count of a term in a document) and the inverse document frequency (the log inverse of the document frequency of the term across a set of documents).

The system implements TF-IDF for information retreival using the TfidfVectorizer from scikit-learn Python library. The system computes TF-IDF scores for all sentences across the corpus (here, training or devtest). Each sentence is then vectorized, and its TF-IDF score is calculated.It then selects the top n sentences with the highest tf-idf score for each collection of documents. n is set by the user. For the purposes of this article, n=3.

3.2.2 TextRank

TextRank is a graph-based ranking model for text processing inspired by the Lexrank graph-based model for selecting sentences based on relative importance (Erkan and Radev, 2004). It is used for extracting the most important sentences based on the concept of sentence similarity.

The system implements TextRank as a content retriever using the Python libraries scikit-learn and networkx.The CountVectorizer from scikit-learn is used to convert the sentences into a matrix of token counts, which serves as the input for calculating cosine similarity. The system then uses the cosine_similarity library to produce a similarity matrix for the document set. Using the rankings produced by the PageRank algorithm (as implemented in networkx), the system selects the top n sentences from each document as the content for the summary. For the purposes of this article, n=3.

3.3 Information Ordering

The v1.0 system implements an information ordering method inspired by the Traveling Salesperson Problem (TSP) (Conroy et al., 2006) This approach seeks to minimize the "distance" between sentences, where distance is a measure of dissimilarity, thereby ensuring that semantically related sentences are placed closer together in the final summary.

To order sentences effectively, the system first calculates the semantic "distance" between each pair of sentences. This is achieved using the MASI distance metric, which considers the overlap and difference in the semantic content of sentences (Passonneau, 2006). It is calculated using masi_distance from the NLTK Python library.

With the distance matrix established, the system employs the two-opt algorithm, a heuristic for approximating the solution to the TSP (Brodowsky et al., 2023). This algorithm iteratively improves the order of sentences by reversing segments of the route (i.e., the sequence of sentences) if it results in a shorter total distance, which in our context translates to a more coherent summary. The sentences are reordered according to the optimal path identified, resulting in a logically arranged summary where the flow of information mirrors the natural progression of ideas in the original text.

3.4 Content Realization

The v1.0 system relies on a naive method for content realization. There is an object ContentRealizer, which accepts as input the output of the information ordering mechanism. That output is a nested dictionary, which includes for each set of documents a list of words comprising a sentence. For each sentence, ContentRealizer simply concatenates the items into a single string with spaces between words. Each realized sentence is, in turn, concatenated and separated by spaces.

3.5 Evaluation

The system's summaries are evaluated using ROUGE-1 and ROUGE-2 (Lin, 2004). The ACQUAINT corpora includes multiple model outputs for each document set. The system calculates ROUGE-1 and ROUGE-2 for the system's summary vis-a-vis all model summaries for a given document set. Those scores are then averaged for the document set. The system outputs aggregate scores across the development corpus (devtest).

4 Results

Table presents ROUGE-1 and ROUGE-2 recall scores of our base system using our two content selection methods on the devtest set.

Table 1 presents aggregate ROUGE-1 and ROUGE-2 recall scores of the base system using the TF-IDF and TextRank content selection methods on the devtest set. The ROUGE-1 scores for the system using the TF-IDF and TextRank content selection methods are nearly identical, while the ROUGE-2 scores suggest that the system using the TextRank content selection method slightly outperforms the system using TF-IDF.

Table 1: ROUGE-1 and ROUGE-2 recall scores on devtest

Method	ROUGE-1			ROUGE-2		
	min	max	avg	min	max	avg
TF-IDF	0.0880	0.4885	0.2580	0	0.1750	0.0448
TextRank	0.0930	0.3812	0.2578	0	0.1278	0.0485

Table 2 provides ROUGE scores disaggregated by text category[2]. While the differences between content selection methods are less salient, the data show that our system generally summarized texts in categories 2 (attacks) and 5 (investigations and trials) better than those in categories 1 (accidents and natural disasters), 3 (health and safety), and 4 (endangered resources).

Table 2: ROUGE-1 and ROUGE-2 recall scores by Text Category on devtest

Method	Text Category	ROUGE-1			ROUGE-2		
		min	max	avg	min	max	avg
TF-IDF	1	0.0880	0.3139	0.2532	0	0.0876	0.0513
	2	0.2145	0.3835	0.2956	0.0075	0.0992	0.0518
	3	0.1427	0.3274	0.2100	0	0.0842	0.0289
	4	0.1866	0.3052	0.2481	0.0076	0.0746	0.0368
	5	0.2097	0.4885	0.3025	0.0075	0.1750	0.0626
TextRank	1	0.0930	0.2918	0.2377	0.0051	0.0730	0.0479
	2	0.1489	0.3638	0.2778	0	0.0882	0.0567
	3	0.1584	0.2767	0.2301	0.0025	0.0729	0.0327
	4	0.1966	0.3812	0.2534	0.0051	0.1214	0.0473
	5	0.2403	0.3696	0.2954	0.0220	0.1278	0.0637

5 Discussion

Two variants of the system were evaluated–one utilizing TF-IDF for content selection, and the other employing TextRank. In both variants, the information ordering and content realization methods were identical. However, ROUGE scores measure only n-gram overlap with no regard for information ordering and content realization methods, so limiting evaluation of the system to the content selection methods is justified.

The TF-IDF and TextRank methods had very similar ROUGE-1 scores (0.2580 and 0.2578, respectively), suggesting that unigram overlap between the system-generated summaries and the human-generated model summaries was relatively consistent. However, some amount of overlap is expected, as stop words and other common words were not excluded from evaluation. The ROUGE-2 scores are slightly more illustrative, as they rely on bigram overlap. While the difference in ROUGE-2 scores is marginal, TextRank (0.0485) did outperform TF-IDF (0.0448).

Disaggregating the evaluations of system performance on different categories of document sets illuminates some of the strengths and weaknesses of the current system. Summaries generated for document sets whose topic fell into categories 2 (attacks) or 5 (investigations and trials) tended to score better than those generated for document sets with topics falling into categories 1 (accidents and natural disasters) and 4 (endangered resources). Moreover, the system summaries generated for document sets with topics around health and safety (category 3) were consistently worse than those for any other category.

Articles in categories 2 and 5 may be more templatic, as these categories involve concrete events where a line can be drawn neatly from action to result, and similar types of information (e.g., the perpetrator, the action undertaken) are salient in each document set, and perhaps even similarly organized. Though a similar argument could be made for category 1, categories 2 and 5 tend to involve human agents. This distinction suggests that gendered pronouns may play a role in determining content saliency in our system.

Category 3 can be seen as the most general category, where neither the semantic agent nor theme fit any particular template. Thus, content selection becomes more difficult because topics are less saliently marked. Category 4, while also more general than categories 1, 2, and 5, may still be limited enough in scope that relevant information can be selected by our system.

We can postulate a spectrum of category "abstractness" within these document sets, ranging from least abstract (i.e., categories 2 and 5) to somewhat abstract (i.e., categories 1 and 4) to most abstract (i.e., category 3). This aligns with the

[2]Category 1: Accidents and Natural Disasters; Category 2: Attacks (criminal/terrorist); Category 3: Health and Safety; Category 4: Endangered Resources; Category 5: Investigations (criminal/legal/other).

ROUGE scores seen in Table 2. Observing this, it could be useful in refining the content selection methods to first measure the topic variability within a document set and use this information when selecting content to ensure that the most salient sentences are extracted. The greater the topic variability within the document set, the greater the need to select several short sentences in order to cover a wider range of topics, rather than a few long sentences.

Examining the TextRank method more closely, the system generated one summary (D1036G-A) with a ROUGE-2 score of zero. In this case, the system selected sentences that were most similar to the other sentences in the document set, but the selected sentences ended up not carrying much content relevant to the topic. In light of this, the TextRank method could be modified to score sentences not only on their similarity to other sentences, but also by incorporating some other metric and/or feature (e.g., whether or not a sentence contains reported speech, which may be similar to many other sentences but may not provide a coherent summary in isolation).

When using the TF-IDF method, the system generated four summaries with ROUGE-2 scores of zero (D1006A, D1023E, D1026E, and D1030F). It is worth noting that all four of these document sets belong to category 3, the broadest and most abstract category. One flaw of TF-IDF is its inherent bias toward longer strings. The algorithm favors selecting sentences with more words because a longer sentence is more likely to contain a given word than a sentence with fewer words. Even if these words do not function as the semantic topic of the sentence (and are thus unhelpful as a summary), these longer sentences get selected. This can be seen in the summary generated for D1006A, which consists of a single sentence. To remedy this, modifications could be made to the term frequency calculation to normalize for sentence length.

Examination of the other zero-scoring summaries yields additional useful information that can be used to improve the system performance. For example, some document sets include articles that are merely lists of the top headlines of the day or hour (e.g., APW19990224.0002 in D1023E-A) or sets of unrelated questions/answers, rather than articles on a single subject (e.g., NYT19980603.0106 in D1030F-A). In such cases, the system has trouble determining what the topic of the document set is, and is less likely to generate a coherent or relevant summary. In such cases, the ROUGE-1 score is markedly lower than for other summaries, but still non-zero, since common words will likely produce some amount of overlap between the system and model summaries. However, the ROUGE-2 score is often much closer to or equal to zero, since the probability of overlapping bigrams between summaries with differing topics is extremely low.

It may also be worth considering how to account for cases where there is low inter annotator agreement on the model summaries. For example, D1026E-A is broadly about head and brain safety, but articles in the document touch on road safety, helmet laws, concussions, and sports. Each of these is addressed differently in the human-generated model summaries, and no system summary can possibly score well when the model summaries are so diverse.

Lastly, a review of all of the system-generated summaries reveals additional enhancements that can be made. Bylines should be removed from all sentences (though they are nearly always located in the first sentence of the first paragraph), as they have a detrimental effect on ROUGE scores and are unnatural and unnecessary to include in a summary. Additionally, quote handling should be examined further, as direct speech can lead to strange artifacts in the output summaries, such as unmatched quotation marks.

6 Conclusion

The evaluation of our summarization system using ROUGE-1 and ROUGE-2 scores for the TF-IDF and TextRank approaches indicates that both methods are relatively comparable in terms of recall, precision, and F1 scores, with TextRank slightly outperforming TF-IDF, especially in ROUGE-2 scores, suggesting better capture of bi-gram relationships. This points to TextRank's effectiveness in leveraging contextual and relational information between sentences. However, there is considerable room for improvement across both methods. Specifically, content selection could benefit from a more sophisticated analysis that better understands context and thematic significance. Information ordering could be optimized further to enhance logical flow and coherence, addressing the narrative structure more effectively. Most critically, content realization needs significant advancement to improve the synthesis of selected content into more coherent,

fluent summaries that better reflect the complexities and nuances of source documents. These improvements are essential for pushing the boundaries of automated summarization towards producing more informative, readable, and contextually rich summaries.

A Appendix: Workload Distribution

- **Ben Cote**:
  - Prepared/formatted D3-PDF
  - Literature review for D3 content selection
  - Developed TextRank content selection method
- **Mohamed Elkamhawy**:
  - Literature review for D3 content selection
  - Added Document Processor Loading Component
  - Defining data contract for the system
  - Developed tf-idf content selection method
  - Added Evaluation module to the system
  - Reviewed different components code
  - Adding Config file to control different system experiments
  - Adding System Overview to Report/Presentation
- **Karl Haraldsson**:
  - Developed content_realization.py
  - Wrote the Approach section along with preliminary Abstract, Introduction, and Conclusion sections
  - Reviewed and debugged document_processing.py code
  - Reviewed information_ordering.py code
  - Project management
- **Alyssa Vecht**:
  - Further developed the ingestion and preprocessing mechanisms for the system
  - Developed information_ordering.py code
- **Josh Warzecha**:
  - Reviewed content_realization.py code
  - Developed function in main.py to write summaries to output directory
  - Wrote Results and Discussion sections; contributed to Introduction section

B Appendix: Code Repository & Additional Resources

Our team's repository can be found here on GitHub or directly via this URL: `https://github.com/summarization-team/summary`

Additional Resources:

- nltk
- scikit-learn
- networkx
- rouge_scorer

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Ulrich A Brodowsky, Stefan Hougardy, and Xianghui Zhong. 2023. The approximation ratio of the k-opt heuristic for the euclidean traveling salesman problem. *SIAM Journal on Computing*, 52(4):841–864.

John M Conroy, Judith D Schlesinger, Dianne P O'Leary, and Jade Goldstein. 2006. Back to basics: Classy 2006. In *Proceedings of DUC*, volume 6, page 460.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Rebecca Passonneau. 2006. Measuring agreement on set-valued items (masi) for semantic and pragmatic annotation.

Carson Sievert and Kenneth Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA. Association for Computational Linguistics.

Karen Spärck Jones. 2007. Automatic summarising: The state of the art. *Information Processing Management*.

Zhou Tong and Haiyi Zhang. 2016. A text mining research based on lda topic modelling. In *Computer Science Information Technology*, volume 6, pages 201–210.