

LING 575: Summarization Deliverable #5 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

This paper presents a preliminary exploration of automated multi-document summarization, with a focus on addressing the challenges presented by the Text Analysis Conference (TAC) 2010 Guided Summarization task. It outlines an experimental framework that applies TF-IDF, Enhanced TextRank, and Topic-focused algorithms for content selection. Additionally, the study proposes attempts at information ordering using approaches inspired by the Traveling Salesperson Problem (TSP) and the entity-grid framework of local coherence, both aimed at improving the coherence of the generated summaries. Along with that it introduces Simple (Naive) Content Realization, Advanced (Sentences Compression) and Generative approach which utilizes Generative AI. The evaluation, conducted through ROUGE-1 and ROUGE-2 metrics, indicates a nuanced performance difference between the three content selection methods, with no method demonstrating clear superiority. The findings suggest modest success in bi-gram relationship capture by the Enhanced TextRank and Topic-focused approaches over TF-IDF. Acknowledging the work as ongoing, the paper identifies significant areas requiring further development, including more refined content selection, enhanced information ordering strategies, and improved content realization techniques, along with the need for deeper semantic considerations of the source texts.

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 (devtest), and evaluation (evaltest) datasets for this task.¹ The specific task is

¹The AQUAINT and AQUAINT-2 corpora are collections of English language news articles taken from the New York

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 scores from the TF-IDF, TextRank, or Topic-focused algorithms. Sentences are ordered coherently using either an application of the Traveling Salesperson Problem (TSP), or the entity-grid model of local coherence, and content realization is achieved using a naive mechanism. 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

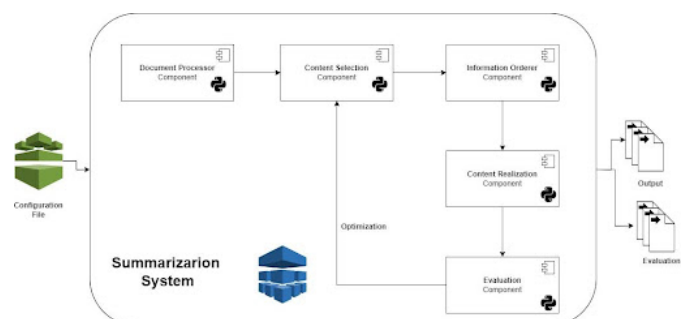


Figure 1: A graphical depiction of our summarization system

Figure 1 shows the flow of our system, from Times, the Associated Press, and Xinhua News Agency

input configuration through the summarization system, outputting both a collection of summaries and an evaluation of those summaries. In this system once the documents have been processed, each set of documents pertaining to an event is treated as the input for the summarization system. Content is selected from these documents, ordered, and then realized according to the methods specified within the configuration file. After evaluation and optimization, the summaries and evaluation results are then returned as the output of the 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 improved implementation (D4). Each component is detailed below.

3.1 Preprocessing

The System ingests the raw XML files and processes them for downstream summarization. The title and category information for each topic in the raw XML files are collected into a text file that is added to the folder for the corresponding docset. Document 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. Paragraphs 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 three methods for content selection: term frequency-inverse document frequency (TF-IDF) (Ramos, 2003), TextRank (Mihalcea and Tarau, 2004), and Topic-focused sum-

marization. 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 retrieval 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.

This project’s implementation is inspired by Enhanced Text Rank (Yulianti et al., 2023), an updated approach that aims to create better sentence representations through word embedding techniques. First, the system selects all sentences in the document above a given word count threshold—in this case, eight words—to keep the top n sentences longer. Next, the system uses the Python libraries torch and Hugging Face transformers to tokenize each sentence and convert each sentence into a vector consisting of the BERT embeddings for each word. The system then implements TextRank as a content retriever using the Python libraries scikit-learn and networkx. The matrix of sentence word embeddings 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, $n=3$.

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, $n=3$.

3.2.3 Topic-focused Summarization

Topic-focused summarization is an approach to refine content selection and ensure summaries closely align with specified topics by integrating semantic embeddings and adjusting LexRank (Erkan and Radev, 2004) for topic relevance. We started by pre-processing documents to include titles in the text, acknowledging their thematic significance, and generated sentence embeddings using Sentence-BERT (Reimers and Gurevych, 2019), capturing the semantic richness of each sentence. A comprehensive topic description was formed by combining the title, narrative in the training data, and category from the description files in the devtest data, which was then encoded into an embedding to serve as a semantic anchor. Semantic similarities between sentences and the topic embedding were calculated to identify topically relevant sentences. We modified LexRank to prioritize sentences based not only on their centrality but also their relevance to the topic, ensuring the selected content was both significant and topically aligned. This methodology allowed us to produce summaries that were not only coherent but also deeply focused on the pre-defined topics, as indicated by improved ROUGE-1/ROUGE-2 scores, demonstrating the effectiveness of integrating semantic understanding into the summarization process.

3.3 Information Ordering

The D4 system implements two major methods for information ordering, in addition to a purely random ordering that can be used as a benchmark.

3.3.1 Random Ordering

This approach randomly shuffles the sentences passed in from the content selection step and returns this randomized ordering without any attempts to maximize discourse cohesion or coher-

ence.

3.3.2 Traveling Salesperson Problem

The D4 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.

3.3.3 Entity Grid

The entity grid is an approach that seeks to maximize the local coherence between sentences by using a model to predict the most likely sequence of entity transitions between sentences (Barzilay and Lapata, 2008). The premise is that coherent discourses display a regular distribution of entity mentions and roles across sentences, and that any system-generated summary should conform to these observed regularities.

The D4 system implements a modified version of the approach laid out by Barzilay and Lapata (Barzilay and Lapata, 2008). Rather than investigating the varying effects of syntax, salience, and coreference of each entity, our approach only examines the presence or absence of entities across sentences.

First, a model is constructed using the training dataset. For each human-generated summary in the training data, the named entities are identified and extracted using NLTK. In cases where no named

entities can be identified, all nouns are extracted instead. Next, an m by n array is constructed, where m is the number of named entities or nouns extracted from the summary and n is the number of sentences in the summary. Cell values are binary, with 1 representing the case where a sentence contains a particular entity, and 0 representing the case where it does not. A vector is constructed from this array by counting the occurrence of the transitions of each entity between sentences (i.e., the counts of each entity’s column-wise transitions from 0 to 0, 0 to 1, 1 to 0, and 1 to 1). Finally, this vector of counts is converted to a probability distribution by dividing each element by the total number of possible transitions $m \times (n - 1)$.

These vectors represent observed transition probabilities in human-generated summaries, which are assumed to be examples of coherent discourse. To generate negative samples, we took each human-generated summary and created multiple copies of it with randomized sentence orderings before repeating the steps above, yielding between six and ten negative samples for each positive sample². From this dataset, we used scikit-learn to build a logistic regression model that classifies each summary as likely being human-generated or system-generated on the basis of the observed transition probability distribution.

To implement this model in our system, we took each set of sentences extracted by the content selection component and generated an array of all possible permutations of these sentences. While this step could become computationally intensive if more sentences were selected, for our purposes, it was possible and feasible to enumerate all possibilities. For each permutation, the entity grid was constructed and the vector of transition probability distributions obtained. The logistic regression model was then used to predict the likelihood of a particular permutation being human-generated or model-generated, with the key assumption that some ordering of sentences would more closely align with a hypothetical human-generated extractive summary. The ordering that yielded the highest probability of being human-generated was then passed to the content realization component.

²Two hyperparameters in the config file are used to specify the minimum and maximum number of permutations to be generated.

3.4 Content Realization

The system implements three methods for content realization: a naive approach, an advanced compression approach, and a generative approach. 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.

3.4.1 Simple (Naive) Content Realization

The naive approach simply removes any sentence whose inclusion in the summary would cause the summary to exceed 100 words. 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.

3.4.2 Advanced (Sentence Compression)

For its “advanced” approach, the system compresses selected and ordered sentences using a sequence-to-sequence transformer inspired by [Nay-eem et al. \(2019\)](#) and [Kamigaito and Okumura \(2020\)](#). This system’s specific implementation utilizes the summarization pipeline object from HuggingFace’s transformers library. Thus, it can accept any sequence-to-sequence model supported by HuggingFace. Here, the recommended model is `t5-base`. Sentences are compressed using the `max_length` argument, which is set to 100. This forces the system to convert the input sequence into a similarly likely sequence with fewer words.

3.4.3 Generative

This is an extractive summarization system. Nevertheless, it also implements a generative model for the content realization step. Specifically, when the generative option is selected, the `ContentRealizer` injects the selected and ordered sentences into a pre-written prompt. The prompt is shown below.

Your task is mutli-document summarization. Specifically, you will be addressing the ”content realization” step of an extractive summarization system. You will be provided a set of sentences that have been extracted from news articles pertaining to a particular topic or event. These sentences have also been ordered. You should return a narrative string. First, edit the sentences to ensure that they flow

coherently. Second, enhance readability and resolve coreferences. Third, compress or fuse the sentences as necessary. Do not add substantive information or context not present in the selected content below. Your output must not exceed 100 words.

The prompt includes language around coherence, readability, and coreference resolution. It also suggests compression or fusion. It specifically indicates that the generated response should not exceed the four corners of the selected and ordered content.

The system utilizes OpenAI’s API, specifically the gpt-3.5-turbo model, with a temperature setting of 0. This temperature parameter controls the model’s creativity or randomness, with a value of 0 leading to deterministic, less varied outputs, ensuring the generated narrative closely adheres to the input content without deviation.

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

This section describes the results of the initial system (D3) and the enhanced system (D4).

4.1 Initial System Results

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: Initial System (D3) ROUGE-1 and ROUGE-2 recall scores on devtest

Method	ROUGE-1			ROUGE-2		
	min	max	avg	min	max	avg
TF-IDF	8.80	48.85	25.80	0	17.50	4.48
TextRank	9.30	38.12	25.78	0	12.78	4.85

Table 2 provides ROUGE scores disaggregated by text category³. 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: Initial System (D3) ROUGE-1 and ROUGE-2 recall scores by Text Category on devtest

Method	Text Cat.	ROUGE-1			ROUGE-2		
		min	max	avg	min	max	avg
TF-IDF	1	8.80	31.39	25.32	0	8.76	5.13
	2	21.45	38.35	29.56	0.75	9.92	5.18
	3	14.27	32.74	21.00	0	8.42	2.89
	4	18.66	30.52	24.81	0.76	7.46	3.68
	5	20.97	48.85	30.25	0.75	17.50	6.26
Text Rank	1	9.30	29.18	23.77	0.51	7.30	4.79
	2	14.89	36.38	27.78	0	8.82	5.67
	3	15.84	27.67	23.01	0.25	7.29	3.27
	4	19.66	38.12	25.34	0.51	12.14	4.73
	5	24.03	36.96	29.54	2.20	12.78	6.37

4.2 Enhanced System Results

The system’s enhanced results (D4), as evaluated on the development dataset, are presented in Table 3. The table compares the ROUGE-1 and ROUGE-2 scores for each content selection method. The D4 system includes two modifications to the D3 content selection approaches and two revised information ordering methods. The latter two have no impact on ROUGE scores, given that ROUGE is largely agnostic to the sequence of sentences.⁴

Table 3: Enhanced System (D4) ROUGE-1 and ROUGE-2 recall scores on devtest

Method	ROUGE-1			ROUGE-2		
	min	max	avg	min	max	avg
TF-IDF	8.80	48.85	25.80	0.00	17.50	4.48
TextRank	16.58	46.09	27.12	0.25	18.57	4.31
	(+7.28)	(+7.97)	(+1.34)	(+0.25)	(+5.79)	(-0.54)
Topic	12.06	43.31	27.15	0.48	14.00	5.30

The D4 system makes no adjustments to the TF-IDF content selector. Thus, its scores are identical to those in D3 above.⁵ D4 did, however, introduce BERT embeddings as the vector representation of sentences for the TextRank content selector. This change resulted in small improvements to ROUGE-1 (+1.34) and a small decrease (-0.54) in average ROUGE-2.

³Category 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).

⁴Though ROUGE-2 may be marginally sensitive to the bigrams formed at the end and beginning of sentences.

⁵See Table 1 for D3 scores.

Table 4: Enhanced System (D4) ROUGE-1 and ROUGE-2 recall scores by Text Category on devtest

Method	Text Cat.	ROUGE-1			ROUGE-2		
		min	max	avg	min	max	avg
TF-IDF	1	8.80	31.39	25.32	0	8.76	5.13
	2	21.45	38.35	29.56	0.75	9.92	5.18
	3	14.27	32.74	21.00	0	8.42	2.89
	4	18.66	30.52	24.81	0.76	7.46	3.68
	5	20.97	48.85	30.25	0.75	17.50	6.26
Text Rank	1	18.10	31.99	25.92	1.27	6.37	3.87
	2	(+8.70)	(+2.81)	(+2.15)	(+1.27)	(-0.93)	(-0.92)
		19.45	38.58	28.48	1.26	8.68	4.71
		(+4.56)	(+2.20)	(+0.70)	(+1.26)	(-0.14)	(-0.96)
	3	16.58	31.69	24.75	0.25	5.65	3.06
		(+0.74)	(+4.02)	(+1.74)	(0.00)	(-1.64)	(-0.21)
	4	20.44	37.28	28.01	0.98	8.66	4.60
		(+0.78)	(-0.84)	(+2.67)	(+0.47)	(-3.48)	(-0.13)
	5	18.65	46.09	28.96	1.43	18.57	5.52
		(-5.38)	(+9.13)	(-0.58)	(-0.77)	(+5.79)	(-0.85)
Topic	1	12.06	33.65	24.05	1.27	7.79	5.18
	2	16.64	40.31	29.35	1.27	10.92	6.07
	3	17.85	31.17	24.38	0.48	7.67	3.75
	4	16.41	32.15	25.93	1.00	12.13	4.47
	5	20.97	43.31	32.32	0.75	14.00	7.52

D4 also introduced a Topic-focused summarization method for evaluation. That method achieves the highest average ROUGE-1 and ROUGE-2 recall scores (27.15 and 5.30, respectively).

Table 3 provides D4 system ROUGE scores disaggregated by text category. The addition of BERT embeddings to the TextRank content selector resulted in increased average ROUGE-1 scores for categories 1 (accidents and natural disasters), 2 (attacks), 3 (health and safety), and 4 (endangered resources). Category 5’s (investigations and trials) average ROUGE-1 score dropped by 0.58. ROUGE-2 scores increased for category 1 (accidents and natural disasters) and category 2 (attacks), but it decreased slightly for the remaining categories.

TF-IDF garnered the highest average ROUGE-1 score for category 2 (attacks). TextRank had the highest average ROUGE-1 for categories 1 (accidents and natural disasters), 3 (health and safety), and 4 (endangered resources). The Topic-focused content selector had the highest average ROUGE-1 score for category 5 (investigations and trials). The Topic-focused content selection method, however, performed better on ROUGE-2 scoring. It achieves the highest ROUGE-2 scores across all categories except for 4 (endangered resources). TextRank achieves the highest ROUGE-2 score for that category.

4.2.1 Final System Results

The system’s final results (D5), as evaluated on the development dataset, are presented in Table 5. The table compares the ROUGE-1 and ROUGE-2 scores for several runs of the system with various methods for content selection, information ordering, and content realization. The table is

sorted by average ROUGE-2 score. Of the fourteen experiments conducted, the five runs employing the generative content realization method consistently outperformed the baseline approach, based on ROUGE-2 scores. Using ROUGE-2 as a basis, all other experiments performed worse than the baseline approach.

Table 6 compares the top-performing combination (tfidf-entity grid-generative), the bottom-performing combination (topic-focused-TSP-advanced), and the baseline performance by text category. We continue to observe the same variance in performance across text categories regardless of the choice of system components, with texts in category 5 (investigations and trials) receiving the highest average ROUGE-2 scores and those in category 3 (health and safety) receiving the lowest average ROUGE-2 scores.

Table 5: Summarization Performance

CS	Method		ROUGE-1			ROUGE-2		
	IO	CR	min	max	avg	min	max	avg
tfidf	eg	gen	13.07	45.81	33.80	0	17.71	7.31
tfidf	TSP	gen	21.37	47.71	34.26	1.53	16.11	7.06
top-foc	TSP	gen	20.63	47.95	33.31	0.78	16.83	7.05
txtrk	eg	gen	17.34	48.94	32.91	0	18.57	6.63
txtrk	TSP	gen	14.83	44.50	33.24	0	16.81	6.52
base	base	base	10.05	48.85	26.04	0	18.50	5.66
top-foc	TSP	simp	10.30	42.03	27.36	0.25	18.51	5.42
tfidf	TSP	adv	8.46	33.93	21.65	0.23	13.43	5.05
tfidf	eg	adv	10.81	30.43	21.26	0.25	10.38	4.96
txtrk	TSP	simp	11.31	47.70	26.91	0.50	19.02	4.73
tfidf	TSP	simp	8.80	48.85	25.11	0	18.50	4.47
tfidf	eg	simp	8.80	48.85	25.60	0	18.24	4.42
txtrk	TSP	adv	8.05	32.17	20.10	0.26	12.17	4.31
top-foc	TSP	adv	9.33	29.41	18.48	0	9.30	3.86

Table 6: ROUGE Scores by Text Category

Method	Cat.	ROUGE-1			ROUGE-2		
		min	max	avg	min	max	avg
TF-IDF, eg, gen	1	13.07	40.39	34.74	0	12.65	7.81
	2	26.25	44.25	34.99	2.90	14.71	7.60
	3	18.83	39.90	29.92	1.24	8.82	4.41
	4	23.94	45.81	34.81	2.32	17.71	7.71
	5	27.13	44.05	35.97	5.48	14.88	9.85
Base	1	10.05	31.12	24.26	0.25	8.76	5.66
	2	18.85	38.82	29.27	1.75	11.09	6.50
	3	14.03	31.81	22.34	0	8.15	3.67
	4	20.45	38.12	26.75	2.06	10.91	5.32
	5	17.06	48.85	28.73	3.61	18.50	7.80
Top-foc, TSP, adv	1	15.60	22.50	18.34	1.78	5.74	3.72
	2	11.83	29.41	19.03	0.52	9.30	3.88
	3	9.33	27.68	15.51	0	6.04	2.50
	4	11.90	25.54	19.95	0.51	8.92	4.71
	5	14.06	29.24	20.29	0.75	8.80	4.73

Based on average ROUGE-2 scores of the system run on the development dataset, we determined our top five systems and evaluated them on the evaluation dataset, along with the baseline system. The performance of these runs is presented in Table 7, along with comparisons against the five highest-performing systems in 2011 TAC Shared Task. The table compares only the ROUGE-2 scores and is sorted by average ROUGE-2 score. None of our systems outperform the five top-performing sys-

tems in the 2011 TAC Shared Task. Of the 50 systems ranked in the shared task, our best-performing system outperforms 10, though we note that the two bottom-ranked shared task systems both received ROUGE-2 scores of 0.

Table 7: Average ROUGE-2 scores for different systems

System	Average ROUGE-2
TAC-2011-43	13.44
TAC-2011-17	12.99
TAC-2011-25	12.82
TAC-2011-24	12.31
TAC-2011-4	12.13
tfidf, entity grid, generative	7.31
tfidf, TSP, generative	7.06
topic-focused, TSP, generative	7.05
textrank, entity grid, generative	6.63
textrank, TSP, generative	6.52
baseline	5.66

4.2.2 Human Evaluation Results

In addition to the system’s evaluation, human evaluations were also performed on a random subset of ten gold standard summaries from the development set of data, and ten system-generated summaries from the development set of data. For this evaluation, two group members agreed upon a series of metrics, ensured cross-evaluator agreement on a set of test documents, then split the summaries evenly between them to get broader coverage. The five metrics used were informativeness (“Does the summary capture the key information of the issue?”), coherence (“Is the summary logically organized, does it flow?”), fluency (“How grammatically correct is the summary, does it feel natural?”), content selection (“How well is information conveyed without unnecessary repetition, redundancy, and verbosity?”), and overall quality (“What would you rate it overall?”). The human evaluator scores were then compared with the ROUGE-1 and ROUGE-2 scores for the same summaries, results shown in ???. In general, there is a slight positive correlation between each of the metrics and the ROUGE scores. However, the p-values associated with these correlations are high, and none of them are close to reaching significance. While our human evaluation was far smaller than our system evaluation through ROUGE, it is still interesting to note that high-scoring summaries don’t necessarily correlate with high ROUGE scores, and the same with low-scoring summaries and low ROUGE scores. This indicates that, while ROUGE does capture important word frequency information, it is not a holistic indicator of a good summary.

Table 8: Correlation between system and human eval.

Metric	ROUGE-1		ROUGE-2	
	corr.	p-value	corr.	p-value
Informativeness	0.3484	0.3238	0.0134	0.9706
Coherence	0.3114	0.3810	0.3241	0.3608
Fluency	0.0721	0.8430	-0.3343	0.3450
Content Selection	0.2288	0.5248	0.0635	0.8615
Overall Quality	0.1236	0.7335	-0.2148	0.5511

5 Discussion

The system can be run with three distinct content selection methods, three distinct information ordering methods, and three distinct content realization methods. However, as there are so many possible permutations and ROUGE scores measure only n-gram overlap with no regard for information ordering, so beginning our discussion with content selection methods is justified. The Topic-focused and TextRank content selection methods had very similar ROUGE-1 scores (27.15 and 27.12, respectively). They were only slightly higher than the ROUGE-1 score that TF-IDF achieved (25.32). The close proximity of all three methods suggests that the 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 the evaluation. The ROUGE-2 scores are slightly more illustrative, as they rely on bigram overlap. While the difference in ROUGE-2 scores is marginal, the Topic-focused content selection method (5.30) outperformed both TextRank (4.31) and TF-IDF (4.48).

The system appears to be stronger when summarizing some topics than others. Summaries generated for document sets whose “category” 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.

We initially speculated that summaries in categories 2 and 5 may be more ‘templatic’, as these categories tend to 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.

Category 3 (health and safety) 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 therefore 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 ROUGE scores seen in Table 2.

The superior performance of the TextRank and Topic-focused content selection methods over TF-IDF lends credence to the postulation regarding abstraction. The incorporation of BERT embeddings in the TextRank approach led to improvements in ROUGE-1 in categories 1, 2, 3, and 4 and ROUGE-2 in categories 1 and 2. The Topic-focused content selector also uses embeddings. It achieved the highest performance in ROUGE-2 across all categories except category 4.

This divergence in performance underscores the limitations of traditional keyword-based methods in handling the nuances of language and meaning. TF-IDF, while powerful for identifying surface-level content, struggles with the depth of semantic representation required for more abstract or nuanced content. In contrast, TextRank’s use of BERT embeddings allows for a richer, context-aware representation of text, enabling it to capture the essence of content with greater precision. Similarly, the Topic-focused approach benefits from embeddings that encapsulate sentence-level semantics, offering a more nuanced representation of text than what individual word tokens can provide. These findings support the abstractness postulation by suggesting that methods that exceed lexical analysis to incorporate context and semantic meaning may be better equipped to handle the variability and complexity of abstract content.

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 three of these four 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.

Examination of the other zero-scoring summaries yields additional useful information that can be used to improve the system’s 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.

Interestingly, the somewhat inferior performance of TF-IDF as a content selection system is what makes it a better selection method for the end-to-end system. When using a naive content realization approach, the more powerful TextRank and Topic-focused selection methods produce better ROUGE scores. However, we base evaluation solely on the ROUGE performance of our various models, then the model combination of tf-idf, entity grid, and generative realization is our highest performing model. This title could arguably go to the model combination of tf-idf, the Traveling Salesperson method, and generative realization because it has a higher ROUGE-1 score (34.26 compared to 33.80), but the tsp-entity_grid-generative method’s higher ROUGE-2 score (7.31 compared to 7.06) can be considered a higher-priority measure since it considers bigram counts instead of unigrams. Our two best systems both use the TF-IDF method. This is likely because it is the best selection system for identifying surface content, which the generative realization method can then work within to create a better summarization.

Considering our content realization methods, the generative approach exceeds the scores of any other method. Looking at , the lowest ROUGE-1 score with a generative realization method (33.24) is higher than the best score with the baseline method (26.04), simple method (27.36), and advanced method (21.65). This pattern is the same for ROUGE-2, with the lowest generative method (6.52) out-performing the baseline (5.66), simple (5.42), and advanced (5.05) methods. OpenAI’s gpt-3.5-turbo model, when prompted to return a coherent, readable summary, is able to handle many different content realization problems simultaneously, including coreference resolution and sentence compression, whereas our advanced realization method solely addressed the issue of sentence compression. Therefore, it is logical that the increased flexibility of using an LLM-based generative approach would produce higher ROUGE scores.

Our Human Evaluation does call the use of ROUGE scores as a proxy for summary quality into question. The lack of correlation between ROUGE scores and human scores across all five evaluation metrics indicates that in our experiment, the evaluators did not find ROUGE-1 or ROUGE-2 to accurately indicate the quality of a summary. Unigram and bigram counts are just one measure of similarity between the gold standard summaries and their system-generated counterparts, and while frequency counts do indicate shared information, they do not reflect attributes like fluency or sentence ordering that do have an effect on the perceived quality of a writing sample.

6 Conclusion

The evaluation of our summarization system using ROUGE-1 and ROUGE-2 scores for the TF-IDF, TextRank (original and embedding-based), and Topic-focused summarization approaches indicates that the three methods are relatively comparable in terms of recall, precision, and F1 scores, with Topic-focused summarization and TextRank slightly outperforming TF-IDF, especially in ROUGE-2 scores, suggesting better capture of bi-gram relationships. Both the Topic-focused and TextRank approaches utilize BERT embeddings, pointing to the effectiveness of such embeddings in leveraging deeper semantic information between sentences. However, when integrated into the full system, TF-IDF provides the greatest amount of

information for the generative content realization system to use when constructing a summary. Still, there is considerable room for improvement in the system. Specifically, content selection could benefit from a more sophisticated analysis that better understands context and thematic significance. Information ordering could be further optimized to enhance logical flow and coherence, addressing the narrative structure more effectively, perhaps through a more nuanced implementation of the entity-grid approach that identifies and models semantic roles rather than mere entity presence. 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. However, we created a system that outperformed ten of the Shared Task systems, even if there still is much that could be improved.

A Appendix: Workload Distribution

- **Ben Cote:**

- Prepared/formatted D5-PDF (in LaTeX)
- Human Evaluation of summaries
- Calculation of correlation for Human Evaluations
- Wrote/updated the system overview, discussion, and conclusion sections of D5
- Project management

- **Mohamed Elkamhawy:**

- Continued development topic-focused content selection method
- Reviewed different components code.
- Updated the abstract and introduction
- Updated Config parameters to allow easier experimentation setup.
- Reviewed pull requests for doc processor and content selection method.

- **Karl Haraldsson:**

- Added two new content realization approaches to content-realizer.py
- Improved various system components, including adding GPU support for transformers

- Wrote content realization section of Approach in D5.pdf
 - Review pull request for information ordering
 - Ran all methods to produce devtest and evaltest results
 - Updated README
- **Alyssa Vecht:**
 - Human Evaluation of summaries
 - Developed information_ordering.py code
 - Developed TSP information ordering method.
 - **Josh Warzecha:**
 - Further developed entity-grid information ordering method
 - Wrote entity-grid information ordering section in D5.pdf
 - Filled out the results table and calculated values for our system performances
 - Drafted initial deck for D5_presentation.pdf

B Appendix: Code Repository & Additional Resources

Our team’s repository can be found [here on GitHub](https://github.com/summarization-team/summary) or directly via this URL: <https://github.com/summarization-team/summary>

Additional Resources:

- nltk
- scikit-learn
- networkx
- rouge_scorer
- Hugging Face transformers
- pytorch

References

- Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.
- 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.
- Hidetaka Kamigaito and Manabu Okumura. 2020. [Syntactically look-ahead attention network for sentence compression](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8050–8057.
- 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.
- Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2019. Neural diverse abstractive sentence compression generation. In *Advances in Information Retrieval*, pages 109–116, Cham. Springer International Publishing.
- Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2021. [Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models](#). *CoRR*, abs/2108.08877.
- Rebecca Passonneau. 2006. Measuring agreement on set-valued items (masi) for semantic and pragmatic annotation.
- Juan Enrique Ramos. 2003. [Using tf-idf to determine word relevance in document queries](#).
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.
- 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.

Evi Yulianti, Nicholas Pangestu, and Meganingrum Arista Jiwanggi. 2023. [Enhanced textrank using weighted word embedding for text summarization](#). *International Journal of Electrical and Computer Engineering*, 13(5):5472–5482.