PreSumm: Predicting Summarization Performance Without Summarizing

Anonymous ACL submission

Abstract

Although recent advancements in Large Language Models (LLMs) have improved summarization capabilities, not all documents are equally well summarized by these models. Our analysis reveals that documents keep their summarization performance across different summarization systems, suggesting that documents have specific properties that influence summarization performance. Therefore, we ask the question: can we predict a document's summarization performance without actually generating a summary? In this paper, we introduce PreSumm, a system designed to predict how well a general summarization system would perform on summarizing a certain document. By forecasting summary quality, PreSumm may save time and resources while avoiding costly workflows. Additionally, PreSumm reveals what features make a document easy to summarize by current summarization systems. Our baseline model demonstrates a strong correlation to human evaluations, despite not generating summaries. This suggests that certain global document features are consistently relevant across different summarization systems. We further demonstrate the model's utility in other contexts, such as multi-document summarization and limited manual summarization budgets. Overall, PreSumm offers valuable insights into document-level features, addressing limitations of current summarization systems.

1 Introduction

As the volume of textual information grows substantially each year, along with its accessibility, the overwhelming amount of data can make it challenging to extract meaningful insights and information. As a result, summarizing data has become essential for individuals and organizations alike. The recent development of Large Language Models (LLMs) made significant steps in solving this task.

Yet, our preliminary analysis (as presented in 4) reveals that not all documents are equally well summarized by these models. Some documents consistently yield better summaries, while others do not, regardless of the system used. Documents that are difficult for one system to summarize tend to yield similar results across various systems, which may suggest inherent differences in document properties.

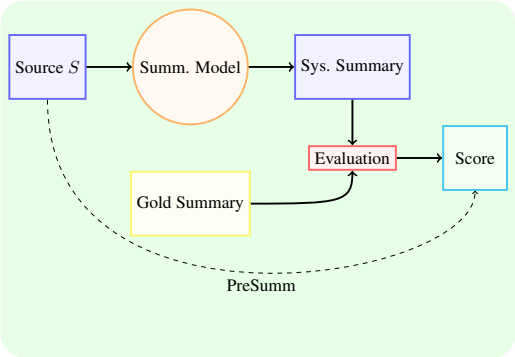

Figure 1: How PreSumm differs from a standard evaluation pipeline.

This analysis raises a compelling research question: Can we predict the performance of summarizing a document without actually generating a summary? In this paper, we aim to answer this question and introduce PreSumm, a system designed to forecast a document's performance in future summarization tasks. We argue that this approach has both practical and theoretical implications.

Practically, PreSumm could save significant time and resources. Summarizing large volumes of documents can be expensive, and as we have shown, the quality of summaries can vary significantly between documents. To evaluate the generated summaries, organizations often rely on automatic metrics or manual human evaluation, which can reveal poor-quality summaries that require further manual summarization—an expensive and time-consuming process. With PreSumm, we could identify low-performing documents in advance, allowing for manual summarization or opting out before en-

gaging in costly summarization workflows. This predictive insight would prevent companies from investing in expensive tools or human resources unnecessarily.

Furthermore, identifying difficult-to-summarize documents could enhance outcomes in various contexts, such as multi-document summarization, where challenging documents might degrade the overall quality. By filtering out these problematic texts, we might get more consistent, high-quality results.

On a theoretical level, PreSumm opens up new avenues for exploring the structure of document space. It helps address an intriguing question: What makes a document difficult to summarize? Understanding these underlying characteristics could deepen our knowledge of current summarization models and provide valuable insights for improving them in the future.

To this end, we developed a baseline model for this task. Surprisingly, the predicted PreSumm scores show a relatively high correlation with human evaluations with respect to automatic metrics, even though no summaries were generated in the process. This finding supports our hypothesis that certain global features of documents are consistently relevant across different systems.

We leveraged this model to demonstrate improvements in a few downstream applications, such as multi-document summarization (Sec. 6.2), and in a case where we have a limited budget for manual summarization (Sec. 6.1). To gain further insight into document-based features, we explore the correlation between various features and our model's scores (Sec. 7.1), and we experiment with different transformations to understand their impact on PreSumm predictions (Sec. 7.2).

Overall, our PreSumm model highlights important document-level features for summarization, providing a clearer understanding of the current limitations in summarization systems and paving the way for future improvements.

2 Related Work

Our work draws significant inspiration from PreQuEL (Don-Yehiya et al., 2022), which introduces a similar approach centered on machine translation. PreQuEL aims to predict translation system performance based solely on the source text. While our motivation and general methodology align with theirs, to the best of our knowledge, we are the first to apply this approach to text summarization. By leveraging PreSumm, we have enhanced summarization-specific applications and explored features more relevant to summarization, making our contributions distinct and novel.

In the past, many summarization systems relied on document-based features, focusing primarily on selecting key source sentences for inclusion in the summary—a task often framed as a classification problem. These features ranged from the presence of cue phrases (Gupta et al., 2011; Kulkarni and Prasad, 2010), the inclusion of numerical data (Prasad et al., 2012; Abuobieda et al., 2012), sentence length (Fattah and Ren, 2009; Abuobieda et al., 2012), and sentence position (Barrera and Verma, 2012; Fattah and Ren, 2009; Abuobieda et al., 2012; Li et al., 2016), to discourse structure (Louis et al., 2010), among others. While these studies employed such features to generate summaries, one may suggest a potential link between these features and the ease of summarization. However, this property was not explicitly examined. In contrast, our work explores *system-independent document-level* features that impact the ease of summarization in both modern *abstractive* and *extractive* systems.

3 Task Definition

Our task is to predict the average performance of summarization systems on a given document, using only the document as input. Averaging performance across multiple systems can reveal key properties of the document itself, while minimizing the influence of system-specific variability or noise. Formally, let $\mathcal{D}$ represent a corpus of N text passages, where each document is denoted as d_i. PreSumm aims to estimate the average quality of summaries generated by multiple systems for each document.

Specifically, consider M systems, denoted as $s_1, \ldots, s_M$, where system s_i produces a summary for document d_j that is scored as $S_{i,j}$. The goal of PreSumm is to first be able to generate a function $f : \mathcal{D} \rightarrow \mathbb{R}$ that predicts the average score assigned to the summaries of document d_j across all systems:

$$d_j^* = \frac{1}{M} \sum_{i}^{M} S_{i,j} \tag{1}$$

This score, d_j^*, reflects the average quality of the summaries generated by different systems for

document d_j. We could leverage this score to rank the documents by their potential performance.

To evaluate the model's performance on ranking documents, we measure the correlation between the predicted scores and the gold-standard scores, following standard practices for assessing summarization metrics (Fabbri et al., 2021).

4 Dataset and Preliminary Analysis

4.1 The RoSE Dataset

The RoSE dataset (Liu et al., 2023) introduces a new method for annotating summarization datasets, which improves annotator agreement via *Atomic Content Units* (ACUs). The protocol tasks an annotator to convert a reference summary into atomic factual statements, then to compare the generated summary to these ACUs. The ACU score is defined as $f(s, A) = \frac{|A_s|}{|A|}$ where $|A|$ is the total number of ACUs for a given reference summary and $|A_s|$ is the number of matched ACUs of the generated summary with respect to the reference.

The authors manually evaluated over 22,000 summary-level annotations across 28 top-performing systems on three datasets. This extensive manual evaluation of diverse systems and datasets aligns well with our task, and we adopt the ACU score as the primary metric to predict. Data statistics can be found in Table 1.

Dataset	Split	#Doc	#Sys.	#ACU
CNNDM	Test	500	12	5.6k
CNNDM	Valid	1,000	8	11.6k
XSum	Test	500	8	2.3k
SamSum	Test	500	8	2.3k

Table 1: Distribution of RoSE Dataset. Taken from Liu et al. (2023).

4.2 Do Systems Fail on The Same Documents?

In this section, we investigate whether different systems tend to consistently fail or succeed across the same documents. Since each system generates summaries for the same set of documents, their performance scores can be used to rank the documents. Our hypothesis is that different systems rank documents similarly, meaning the systems on certain documents consistently perform well or poorly across various systems.

To test this hypothesis, we measure the correlation between the ACU scores of documents on different systems. Specifically, we calculate the correlation between system scores l and k as $corr(S_{1..N,l}, S_{1..N,k})$. We then compute the average correlation across all systems using the formula:

$$\frac{2}{M^2 - M} \sum_{l=1}^{M} \sum_{k=1}^{l-1} corr(S_{1..N,l}, S_{1..N,k}) \quad (2)$$

We obtained a Kendall Tau correlation of 0.446 and a Spearman correlation of 0.565, indicating a moderate agreement in document rankings across systems. This suggests that many documents retain their relative ranking, regardless of the system used for summarization. These findings support our assumption that certain document-specific features significantly influence summarization performance. As a result, it should be possible to predict a document's average summarization performance based solely on its intrinsic characteristics.

5 Experiments

5.1 Models

We trained several models and examined their ability to rank the documents according to the average score across all systems, $d_j^* = \frac{1}{M} \sum_i^M S_{i,j}$. The RoSE dataset was divided into 80% train and 20% test, where we trained all models with a fixed number of 5 epochs.

Regression. In this model, we trained the model to predict the actual d_j^* scores. The input is a document j and the target is d_j^*. We leveraged the Longformer model (Beltagy et al., 2020) to allow a large input length of 4096 tokens required in many documents from our set. We added a regression head on top of the output of the 784 dimensional [CLS] token of the last layer of the Longformer. The regression head contains a feed forward component with an output of one dimension, which should predict the d_j^* score. Additionally, we used a standard MSE loss function for training.

Classification. Since some applications may only care about document rankings, we train an additional model to rank the documents rather than predict their exact d_j^* scores, we focused on ranking them directly based on pairwise comparisons. This approach aligns with our evaluation metric,

which is based on correlation. Instead of predicting individual scores, we aim to determine which document in a pair should be ranked higher, and then aggregate these local decisions to form a global ranking.

In this model, the input consists of a pair of documents, and the task is to classify whether the first document should be ranked higher than the second. We represent each document using embeddings from a pre-trained Longformer model, concatenate the two embeddings, and feed the result into a RankNet model (Burges et al., 2005). The target label for each pair of documents (i, j) is $\delta_{ij} = \mathbb{1}_{d_i^* > d_j^*}$, where $\mathbb{1}$ is an indicator function that equals 1 if document i should be ranked higher than document j. The model is trained using a binary cross-entropy loss function.

We generate all possible n^2 pairs from the training set to fine-tune the model and use all m^2 pairs from the test set during evaluation. To derive the final global ranking, we follow the method outlined by Keswani and Jhamtani (2021), where the final score for document i is defined as $S(i) = \sum_j \hat{\delta} ij$, with $\hat{\delta} ij$ representing the predicted outcome for the document pair (i, j). We then sorted $S(i)$ scores for the final ranking.

Zero-Shot GPT. For zero-shot evaluation, we used GPT-4o to classify a subset of 250 of the Classification model's pairs. We compare the results to the classification model and examine Chat GPT's performance. Additionally, we attempt demonstrate that scores of documents whose true labels are far apart are easier to rank. We do this by giving Chat GPT 800 pairs whose documents ACU score labels are more than 0.9 apart.

Frozen Weights. We examine how a Longformer model (Beltagy et al., 2020) performs with its weights frozen and only training the regression head. The purpose of this model is to see if the embeddings of a chosen Transformer model need significant adjustments before being used for regression or classification and if there are any major differences in between the previously defined models.

5.2 Main Results

All models are evaluated by measuring their correlation with the gold-standard ACU labels, which reflects how well they ranked the documents. The correlation scores are summarized in Table 2. Most models achieve a high correlation, supporting our hypothesis that document ranking can, to some extent, be predicted based solely on the document content.

Model	Kendall Tau	Pearson r	Spearman
Regression	0.321	0.463	0.460
Classification	0.306	0.389	0.389
Frozen Weights	0.279	0.406	0.403
Out of Domain	0.250	0.393	0.358
Zero-Shot GPT (250 samples)	0.0799	0.0799	0.0799
Zero-Shot GPT (0.9 separation)	0.551	0.551	0.551

Table 2: Test set correlations of different PreSumm models

The zero-shot method with GPT over a small subset was not successful, as this is a complicated task to compare between similar documents, especially without fine-tuning. The subset has most document pairs selected be close to each and thus constructing a sound ranking is much harder for a model not trained for this specific task. However, to measure its performance to distinguish between documents with a large performance gap, we sampled pairs with gold scores that are substantially different from each other (more than 0.9) which should in theory be easier to rank. As expected the results are much better, further demonstrating that there are connections between document features and ranking.

The table shows that the Regression PreSumm model outperformed the other baselines in terms of both performance and efficiency. Unlike other models, Regression only requires n samples, whereas the Classification model, for example, uses n^2 samples and requires an additional global aggregation step to consolidate the local pairwise predictions. Given its superior performance and simplicity, we selected the Regression model for the next experiments as our PreSumm method of choice.

5.3 Evaluation on Distantly Ranked Document Pairs

The evaluation presented in Table 2 considers all document pairs in the test set. However, ranking documents that are close in performance may be extremely challenging, or even nearly impossible. Our primary interest lies in correctly distinguishing between far-ranked documents—those with significantly different performance levels. Specifically, we aim to reliably identify high-performing documents from low-performing ones, even if ranking closely-matched documents proves difficult. This

distinction is crucial, as it helps us flag poorly performing documents early on, avoiding a lengthy and time-consuming summarization process for those.

To evaluate the model's ability to differentiate between far-ranked documents, we calculated the correlation for a subset of document pairs where the difference in their d_j^* scores exceeds a certain threshold. Table 3 shows that, as expected, performance improves when focusing on pairs with larger score differences.

Threshold	Percent of Test Set	PreSumm
0	1	0.321
0.3	0.324	0.562
0.5	0.100	0.676
0.6	0.050	0.719
0.9	0.03	0.886

Table 3: Kendall-tau correlation on document pairs with difference $d_i^* - d_j^*$ larger than a certain threshold

5.4 Out-of-Distribution Performance

All PreSumm models (Table 2) were evaluated only on source documents from the same datasets they were trained on. In this section, we examine how well these models perform when applied to documents from a different dataset with distinct properties.

Due to the limited number of ACU annotations and our goal to utilize all available data for training, we did not leave out any dataset in the training of our primary model. However, to estimate how the model would perform on out-of-distribution data, we trained a similar Regression model without including the XSUM documents of RoSE. We first compare this new model, PreSumm$_{\text{OOD}}$, to the original PreSumm model to ensure comparable performance. For this, we evaluate both models on a test set excluding XSUM, which mirrors the training data for PreSumm$_{\text{OOD}}$. As shown in Table 4, the two models perform similarly, confirming their alignment.

Next, we assess PreSumm$_{\text{OOD}}$ on true out-of-distribution data, specifically documents from XSUM. As expected, performance decreases. However, when focusing on far-ranked document pairs (those with substantial differences in performance), the two models' differences are reduced. As shown in Table 5, both PreSumm and PreSumm$_{\text{OOD}}$ improve as the ranking threshold increases. The performance gap between the models also narrows with a higher threshold, suggesting that PreSumm$_{\text{OOD}}$ performs slightly worse than PreSumm, but still remains effective for the critical subset of document pairs we aim to rank.

Model	Test	Kendall Tau	Pearson r	Spearman
PreSumm	w/o XSUM	0.294	0.433	0.423
PreSumm$_{\text{OOD}}$	w/o XSUM	0.296	0.434	0.420
PreSumm$_{\text{OOD}}$	Only XSUM	0.116	0.240	0.177

Table 4: Results of in and out of distribution regression PreSumm models. Where OOD signifies that the PreSumm model was not trained on the XSUM portion of RoSE.

Threshold	Percent of Test Set	PreSumm	PreSumm$_{\text{OOD}}$
0	1	0.246	0.116
0.3	0.453	0.508	0.197
0.5	0.123	0.580	0.386
0.6	0.615	0.713	0.617
0.9	0.005	1	1

Table 5: XSUM only Kendall-tau correlation on document pairs with difference $d_i^* - d_j^*$ larger than a certain threshold

6 Downstream Tasks

In this section, we explore practical applications that can benefit from predicting document performance in summarization. Specifically, we examine two use cases: first, identifying the lowest-performing documents in advance to prioritize for manual summarization, and second, filtering out poorly performing documents in a multi-document summarization setting.

6.1 Filtering Poor Documents

We aimed to assess whether PreSumm can effectively identify low-performing documents in advance, allowing them to be filtered out or prioritized for manual summarization. To this end, we defined a fixed percentage of documents that could be manually summarized within the available budget. Our goal was to select these documents in advance, optimizing the final summarization performance when supplemented with manual summaries.

In addition to PreSumm, we evaluated several baseline methods. In all baselines (except for Random selection), we removed the lowest-ranked documents. Besides Random selection, we tested the reference-free metrics Blanc (Vasilyev et al., 2020)

Model Type	Selection Type	Drop 10%	Drop 20%
Ref. Free	Random	0.428	0.493
	Blanc	0.438	0.512
	SummQa	0.436	0.5
	PreSumm	**0.447**	**0.523**
Ref. Based	Rouge	0.455	0.53
	Bert	0.449	0.52
	Meteor	0.456	0.53

Table 6: New average ACU Scores after dropping the lowest predicted scoring documents.

Model Type	Selection Type	Drop 10%	Drop 20%
Ref. Free	Random	0.342	0.405
	Blanc	0.338	0.409
	SummQa	0.347	0.407
	PreSumm	**0.355**	**0.441**
	$\text{PreSumm}_{\text{OOD}}$	**0.354**	**0.436**
Ref. Based	Rouge	0.359	0.440
	Bert	0.368	0.453
	Meteor	0.363	0.442

Table 7: New average ACU Scores after dropping the lowest predicted scoring documents for only the XSUM portion of the testing dataset.

and SummQA (Scialom et al., 2019). A limitation of these reference-free metrics is that they require system-generated summaries first, whereas PreSumm does not. For comparison, we also included reference-based metrics such as ROUGE-2 F1 (Lin, 2004), BERTScore F1 (Zhang et al., 2019), and METEOR (Banerjee and Lavie, 2005), which represent an upper bound since they rely on both system and reference summaries—resources that are unavailable in real-world scenarios.

We used the previously established test set for evaluation, with the manually annotated ACU score serving as the benchmark. Instead of manually summarizing a selected document, we assigned it an ACU score of 1. In the experiments shown in Table 6, we selected for manual summarization either 10% or 20% of the test set documents.

PreSumm outperformed all other baselines, including the reference-free metrics, and came close to the upper bound set by the reference-based metrics.

To support the conclusion from Section 5.4—that out-of-distribution performance should be similar when focusing on the worst-performing documents—we conducted the same experiment on the XSUM-only test set. As shown in Table 7, both PreSumm and $\text{PreSumm}_{\text{OOD}}$ delivered comparable results, and both continued to outperform other alternatives.

These experiments demonstrate that the PreSumm regression model can effectively identify low-performing documents in advance, thus optimizing the summarization process by saving time and resources.

6.2 Multi-Document Summarization

In a Multi-Document Summarization (MDS) task, a set of documents on the same topic needs to be summarized. However, summarizing a large number of documents presents challenges due to the high input length, which can be costly or limited in certain models. Conventional MDS approaches often concatenate all documents and truncate the input to fit the model's token limit, resulting in the exclusion of some documents. The question arises: can we perform better when using PreSumm to select which documents should be excluded?

To test this, we adopted the MultiNews dataset (Fabbri et al., 2019) as our MDS test set and used the Pegasus summarization model (Zhang et al., 2020), to generate summaries. We conducted experiments with token limits of 256, 512, and 1024. In each experiment, we tested two variations: one where the documents were kept in their original order and truncated once the token limit was exceeded, and another where the documents were reordered based on their PreSumm scores, with the lowest-scored documents placed at the end. It is important to note that reordering the documents has an additional benefit—better document sequencing can enhance summarization quality even without excluding documents, as suggested by Zhao et al. (2022). Thus, we aimed to determine whether combining PreSumm-based document exclusion with PreSumm-based reordering would lead to improved summarization outcomes.

As shown in Table 8, the PreSumm-ordered documents consistently achieved higher ROUGE scores compared to the original document order across all input length limits. For the 1024-token limit, PreSumm showed significant improvement in ROUGE-2 scores only. However, for the 512- and 256-token limits, PreSumm was significantly better across all metrics according to a two-sided Wilcoxon Rank Test (Wilcoxon, 1945). This demonstrates that the PreSumm model is effective not only in identifying high-performing documents individually but also in enhancing multi-document

Token Lim.	Order	R-1	R-2	R-L
1024	original	0.458	0.185	0.243
	PreSumm	**0.461**	**0.189**	**0.246**
512	original	0.439	0.172	0.235
	PreSumm	**0.444**	**0.179**	**0.240**
256	original	0.398	0.147	0.215
	PreSumm	**0.402**	**0.153**	**0.219**

Table 8: Multi-Document Summarization Table. Comparing rouge scores of summarizations generated from the original order of the dataset, to PreSumms order.

Feature	Predictions	Gold Labels
Flesch Reading Ease	0.089	-0.021
Flesch Kincaid Grade	0.0797	-0.0104
Document length	-0.104	-0.010
Count of Numbers	-0.0598	-0.023
# of Unique Named Entities	-0.121	-0.071
Avg Loc of Salient Sent (top 10)	-0.265	-0.121
Avg Loc of Salient Sent (top 5)	-0.305	-0.195

Table 9: Spearman Correlations of document features to regression PreSumm predicted scores and gold label scores.

summarization performance.

7 Analysis

In this section we would like to utilize PreSumm to get some sense of the properties that make a document low-performing for many summarization systems. To that end, we examine the influence of document-based features over PreSumm (Section 7.1). We also examine how PreSumm deals with different corruptions (Section 7.2)

7.1 Document Feature Correlations

To gain deeper insights into which document features most strongly influence the document performance in summarization systems, we analyzed the correlations between various document characteristics and the PreSumm score at the document level. Additionally, we examined the correlation between these features and the gold labels, d_j^*. The features we evaluated are detailed below.

Document Statistics. We explored several basic statistics about the documents, including document length by word, the number of numerical values, and the number of unique named entities identified using the NER module from the NLTK package.

Salient Sentence Location. To determine the location of key sentences within a document, we adopted a method similar to Nallapati et al. (2017); Chen and Bansal (2018), where each sentence in the document is ranked by its similarity to the reference summary, using ROUGE scores as the metric. We selected the top-5 or top-10 most salient sentences, and their positions were normalized by the total number of sentences in the document. The average of these normalized indices represents the typical location of the most important sentences.

Flesch–Kincaid Readability Tests. We applied the Flesch Reading Ease test (Flesch, 1948) to measure how easy the document is to read, with scores ranging from 1 to 100 where higher scores indicate easier readability. Similarly, we used the Flesch-Kincaid Grade Level to estimate the U.S. education grade level required to understand the text, with higher scores corresponding to a more advanced reading level

The correlation results are shown in Table 9. The most strongly correlated feature is the location of salient information. The negative correlation indicates that the earlier the key information appears in the document, the easier it is for the model to summarize. This observation aligns with expectations, as the model does not need to search through the document to locate important content. This is also a well-known characteristic of the news domain, where essential information is typically presented at the beginning.

All basic document statistics show a negative correlation, suggesting that greater document complexity—whether in length, the number of numerical values, or the number of named entities—negatively impacts summarization performance.

The Flesch Reading Ease and Flesch-Kincaid Grade Level scores show only weak correlations with both the gold labels and the predicted scores. This suggests that summarization performance is not strongly tied to the document's readability for humans. One possible explanation is that modern models are capable of processing a wide range of document levels, from elementary to advanced levels, without significant difficulty.

Transformation	Src.	Trans.	Delta
Remove first sentence	0.528	0.506	-0.0225
Replace names w/ spacy name	0.528	0.518	-0.0102
Replace names w/ from bank	0.528	0.512	-0.0163
Remove 5 salient sentences	0.528	0.476	0.0523
Remove 10 salient sentences	0.528	0.428	0.100
Move 5 salient sentences to end	0.528	0.509	0.0199
Move 10 salient sentences to end	0.528	0.502	0.0269
Corrupt Grammar	0.528	0.520	-0.008
Append contradictions	0.528	0.533	0.005
Delete 30% of words	0.528	0.470	-0.0584
Delete 30% of sentences	0.528	0.530	0.00150
Keep first 3 sentences	0.528	0.553	0.0246
Keep last 3 sentences	0.528	0.441	-0.0877
Randomly Shuffle of Sentences	0.528	0.486	-0.0427

Table 10: Predicted scores of documents before a transformation (Src.) and after (Trans.)

7.2 Document-Based Transformations

To further investigate the influence of document-based features on model performance, we explored how the predicted score of a document changes when specific features are perturbed. By comparing the predicted scores before and after these transformations, we can gauge the importance PreSumm places on each feature. Intuitively, the transformations causing the largest change in predicted scores should indicate which features have the greatest impact on document performance in summarization. We applied these transformations to our test set. Below, we detail the transformations applied.

Removing content. We tested several removal strategies: removing the first sentence (often critical for summarization), removing 5 or 10 salient sentences (as defined in Section 7.1), and randomly removing 30% of the words or sentences to disrupt fluency and coherence. Additionally, we removed all sentences except the first three or last three to assess the importance of content location.

Moving content. Given that salient information location was identified as a key feature in Section 7.1, we moved the salient sentences to the end of the document. We also randomly shuffled all sentences to disrupt coherence.

Replacing content. We replaced named entities to test the impact on consistency. We also introduced contradictions by adding negation sentences and corrupted the grammar by converting all verbs to their lemma forms.

As expected, the most impactful corruptions, with the highest changes in predicted scores, were removing the 10 most salient sentences and removing all content except for the last three sentences. Other significant transformations included deleting 30% of the words, removing 5 salient sentences, and shuffling sentences randomly.

Surprisingly, moving the salient sentences to the end of the document had little effect, despite the location of salient information being one of the most influential features in Section 7.1.

Interestingly, content replacement, such as grammar corruption or adding contradictions, did not significantly affect PreSumm's performance. It also appears that strong perturbations, such as deleting 30% of sentences, did not lead to large differences in scores. This might be because these artificial corruptions are not natural and therefore deviate too much from the patterns seen during model training, causing the model to mispredict their impact on summarization performance.

8 Conclusion

In this work, we introduced PreSumm, a novel approach to predicting the performance of summarization systems on specific documents without generating summaries. Our findings demonstrate that certain documents consistently yield better or worse summaries across different systems, suggesting inherent properties that influence summarization systems. By identifying these properties in advance, PreSumm offers practical value by allowing organizations to optimize resources, avoiding costly manual interventions for low-performing documents, and improving the quality of multi-document summarization tasks.

Moreover, this approach opens up new research avenues in understanding the structural features that make a document difficult to summarize, which could lead to more advanced, adaptive summarization models. The strong correlation between PreSumm predictions and human evaluations further validates its potential.

In conclusion, PreSumm not only sheds light on the document-level features that impact summarization quality but also provides a valuable tool for enhancing future summarization models and workflows.

9 Limitations

Although our study covers the RoSE dataset extensively we cannot explore the complete space of summarization systems and are limited to both the datasets and summarization systems that RoSE provides. Because of this, some of the results could be due to other factors that relate to the dataset and would not generalize strongly outside of this study.

In future works, we would seek to train a similar model on a larger dataset. However, using ACU scores would be difficult because of the human labour involved, which could be avoided by using an annotated metric to train on. However, will make us biased towards the chosen metric, which is another limitation.

Out of distribution data is a major factor when it comes to model performance. To mitigate this would require greatly increasing the scope of the experiment and to train on a broader dataset for more accurate predictions.

Overall, overcoming these limitations would necessitate a much larger corpus with either a large set of automated or human annotated metrics to perform a similar study on a much larger set of the space of documents and summarization system.

References

Albaraa Abuobieda, Naomie Salim, Ameer Tawfik Albaham, Ahmed Hamza Osman, and Yogan Jaya Kumar. 2012. Text summarization features selection method using pseudo genetic-based model. *2012 International Conference on Information Retrieval & Knowledge Management*, pages 193–197.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Araly Barrera and Rakesh Verma. 2012. Combining syntax and semantics for automatic extractive single-document summarization. In *Computational Linguistics and Intelligent Text Processing*, pages 366–377, Berlin, Heidelberg. Springer Berlin Heidelberg.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Shachar Don-Yehiya, Leshem Choshen, and Omri Abend. 2022. PreQuEL: Quality estimation of machine translation outputs in advance. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11183, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Mohamed Abdel Fattah and Fuji Ren. 2009. Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. *Computer Speech & Language*, 23(1):126–144.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Pankaj Gupta, Vijay Shankar Pendluri, and Ishant Vats. 2011. Summarizing text by ranking text units according to shallow linguistic features. *13th International Conference on Advanced Communication Technology (ICACT2011)*, pages 1620–1625.

Vishal Keswani and Harsh Jhamtani. 2021. Formulating neural sentence ordering as the asymmetric traveling salesman problem. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 128–139, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Uday Kulkarni and Rajesh Shardanand Prasad. 2010. Implementation and evaluation of evolutionary connectionist approaches to automated text summarization. *Journal of Computer Science*, 6(11):1366–1376.

Junyi Jessy Li, Kapil Thadani, and Amanda Stent. 2016. The role of discourse units in near-extractive summarization. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 137–147, Los Angeles. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference*, pages 147–156, Tokyo, Japan. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Rajesh Shardanand Prasad, Nitish Milind Uplavikar, Sanket Shantilals Wakhare, VY Jain, Tejas Avinash, et al. 2012. Feature based text summarization. *International journal of advances in computing and information researches*, 1(2):15–18.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. *arXiv preprint arXiv:1909.01610*.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the blanc: Human-free quality estimation of document summaries. *arXiv preprint arXiv:2002.09836*.

Frank. Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics*, 1:196–202.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Chao Zhao, Tenghao Huang, Somnath Basu Roy Chowdhury, Muthu Kumar Chandrasekaran, Kathleen McKeown, and Snigdha Chaturvedi. 2022. Read top news first: A document reordering approach for multi-document news summarization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 613–621, Dublin, Ireland. Association for Computational Linguistics.