Learning to Rank Salient Content for Query-focused Summarization

Anonymous ACL submission

Abstract

This study examines the potential of integrating Learning-to-Rank (LTR) with Query-focused Summarization (QFS) to enhance the summary relevance via content prioritization. Using a shared secondary decoder with the summarization decoder, we carry out the LTR task at the segment level. Compared to the state-of-the-art, our model outperforms on QMSum benchmark (all metrics) and matches on SQuALITY benchmark (2 metrics) as measured by Rouge and BertScore while offering a lower training overhead. Specifically, on the QMSum benchmark, our proposed system achieves improvements, particularly in Rouge-L (+0.42) and BertScore (+0.34), indicating enhanced understanding and relevance. While facing minor challenges in Rouge-1 and Rouge-2 scores on the SQuALITY benchmark, the model significantly excels in Rouge-L (+1.47), underscoring its capability to generate coherent summaries. Human evaluations emphasize the efficacy of our method in terms of relevance and faithfulness of the generated summaries, without sacrificing fluency. A deeper analysis reveals our model's superiority over the state-of-the-art for broad queries, as opposed to specific ones, from a qualitative standpoint. We further present an error analysis of our model, pinpointing challenges faced and suggesting potential directions for future research in this field.

1 Introduction

Query-focused summarization (QFS) is gaining prominence in research community. Unlike conventional summarization tasks that aim to capture the overall essence of a document or a set of documents, QFS focuses on generating concise summaries in response to posed queries. This specialization enables a more targeted information retrieval process, offering summaries that directly address the informational needs rather than providing a broad overview of the source material.

The advancements in QFS have been notably driven by the introduction of invaluable datasets such as QMSum (Zhong et al., 2022) and SQuALITY (Wang et al., 2022a), which have facilitated deeper exploration and innovation in this field. These datasets have laid the groundwork for the development of Transformers-based models which have shown strong potential in generating summaries that respond accurately to queries (Su et al., 2021; Laskar et al., 2022; Vig et al., 2022; Pagnoni et al., 2022; Sotudeh and Goharian, 2023; Yu et al., 2023). However, despite this proficiency, their ability to effectively *prioritize information*—assessing its importance relative to a query to enhance summary relevance—remains an area for improvement. This study seeks to address this limitation, aiming to improve the capability of QFS models to deliver summaries with more effective information ranking.

Particularly, in this study, we present a novel enhancement to QFS through the incorporation of learning-to-rank (LTR), a technique with established efficacy in Information Retrieval. Our approach aims to refine the system's capability to discern and prioritize content segments not only by their relevance but also by their relative importance. This methodological advancement ensures that the produced summaries more accurately reflect the query's intent and hierarchically organize information by its significance.

Central to our approach is the augmentation of use of the decoder that *shares parameters* with the summarization decoder [1], specifically designed for executing the LTR task at the segment level. This strategy, inspired by the work of (Zhuang et al., 2022) in adapting the T5 (Raffel et al., 2020) encoder-decoder framework for text ranking in query-document scenarios, is tailored to

[1] Particularly, we use the single decoder for two tasks: summarization and learning-to-rank.

address the nuances of segment ranking within the QFS context. Through the joint fine-tuning of summarization with cross-entropy loss, and LTR task—utilizing listwise cross-entropy softmax loss, our system not only aims to elevate the relevance of generated summaries but also to introduce a nuanced understanding and representation of information importance, which can aid the summarization system at attending to the source content given their relative importance. In short, our contributions are threefold:

- We propose an LTR-assisted system for QFS that integrates the intuition of ranking and relative importance of segments during the summary generation process;
- Our proposed system outperforms across all automatic metrics (QMSum) and attains comparable performance in two metrics (SQuALITY) with lower training overhead compared to the SOTA. Additionally, our system enhances the relevance and faithfulness of generated summaries without sacrificing fluency;
- We undertake an error analysis to discern the challenges faced by our model including label imbalance, and segment summarizer's hurdles, providing insights into potential avenues for further research.

2 Related work

The field of Query-focused Summarization (QFS) (Dang, 2005) has evolved significantly over the years, moving from early unsupervised extractive models (Mohamed and Rajasekaran, 2006; Wan et al., 2007; Zhao and Tang, 2010; Badrinath et al., 2011; Litvak and Vanetik, 2017) to recent approaches leveraging Transformer-based models (Vaswani et al., 2017; Lewis et al., 2020; Zhang et al., 2020). This evolution has been marked by the introduction of various techniques aimed at improving the relevance of summaries. Passage retrieval techniques (Baumel et al., 2018; Laskar et al., 2022; Su et al., 2021; Zhong et al., 2022; Wang et al., 2022a), transfer learning from the QA task (Xu and Lapata, 2020; Zhang et al., 2021; Yuan et al., 2022), query modeling (Xu and Lapata, 2021, 2022; Yu et al., 2023), segment encoding (Vig et al., 2022), and attention mechanisms to capture query-utterance relations (Liu et al., 2023) have all played a pivotal role in this advancement. Furthermore, the adoption of question-driven pretraining (Pagnoni et al., 2022) and contrastive learning (Sotudeh and Goharian, 2023) has introduced new dimensions to the task, simplifying the identification and summarization of salient content. However, the comprehensive modeling of segment importance within the long QFS task remains a less explored area. Our work builds upon these foundational studies and introduces a learning-to-rank (LTR) (Burges et al., 2005; Cao et al., 2007) mechanism to address this challenge, drawing inspiration from the successful application of LTR in broader Information Retrieval contexts (Wang et al., 2022b; Li et al., 2023).

3 Background: Segment Summarizer (SegEnc)

The current state-of-the-art systems for query-focused long summarization are built upon the Segment Encoding (SEGENC) approach (Vig et al., 2022). SEGENC operates by encoding fixed-length, overlapping segments of the source text, which are then integrated into a cohesive summary in an end-to-end manner, leveraging the decoder's ability to simultaneously attend to all encoded segments. To specifically adapt to query-focused summarization framework, SEGENC embeds the query within each segment of the source text. This is achieved through a particular input framing technique, where the query is encapsulated by special markers and placed adjacent to each segment, adhering to the format: `<s>query</s>Segment`. This incorporation of the query into the summarization process is designed to enhance the focus on the query, offering a tailored approach to generating query-focused summaries.

4 Model: LTR-assisted Summarization

This study introduces an extension to the SEGENC summarizer by integrating the Learning-to-Rank (LTR) principles, a notable ranking technique from the realm of information retrieval. This integration enables the summarizer to effectively learn the ranking of the gold segments. The segments' relevance labels are determined using a span probability-based heuristic (details in Section 5.1) during the preprocessing step. An auxiliary LTR task is then formulated to instruct the summarizer in ranking source segments while performing the summarization task.

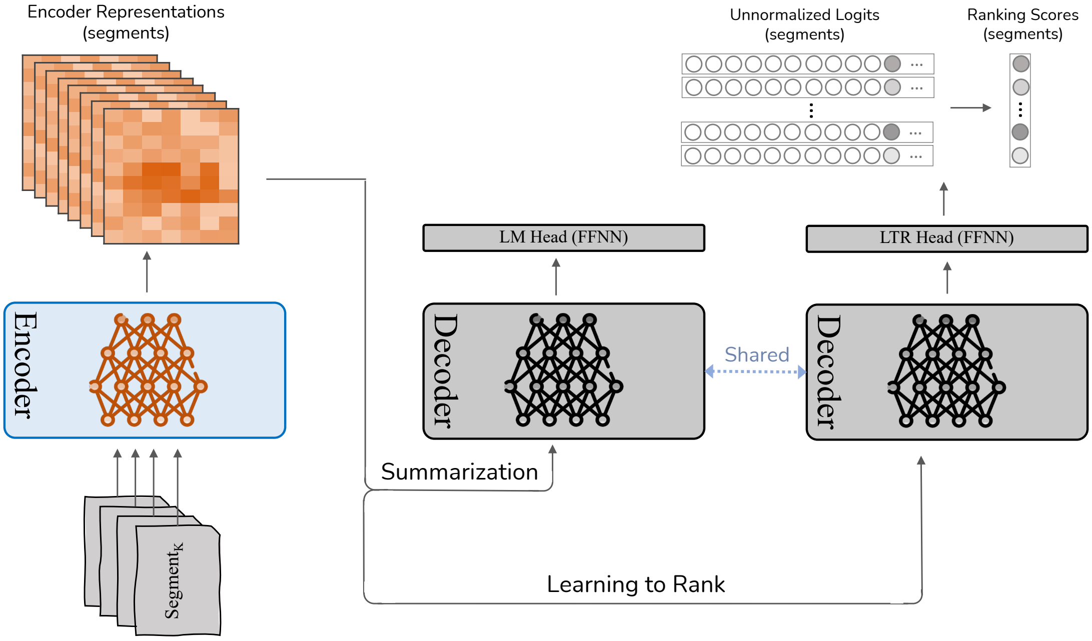

Figure 1: Overview of our proposed system (i.e., LTRSUM). Our system utilizes a *shared-parameter* decoder across two tasks, for the sake of learning to rank source segments (right-side decoder) alongside summarization (left-side decoder). It is important to note that our system uses a single decoder that shares parameters across both tasks, but for visual clarity, two decoders are depicted here.

Figure 1 shows the overview of our proposed system. In particular, we exploit a *shared* decoder to perform two tasks including summarization and learning-to-rank. This shared decoder operates by executing two forward passes, one for each task. For the LTR task, following encoding of each segment (denoted as $\mathtt{Enc(S_i)}$), $\mathtt{Dec_{LTR}}$ takes in the segment encoder representations (i.e., the encoder representations associated with <s> token) and processes them through the LTR-dedicated decoder, followed by an LTR head (i.e., a feed-forward neural network (FFNN)) that is applied to the decoder outputs:

$$\hat{y}_{\mathtt{i}} = \mathtt{FFNN(Dec_{LTR}(Enc(S_i)))} \quad 190$$

wherein $\mathtt{S_i}$ represents the i-th segment, and $\hat{y}_{\mathtt{i}}$ corresponds to the decoder output for the same segment. Furthermore, an additional unused token is defined, analogous to the method described in (Zhuang et al., 2022), and its unnormalized logits are extracted from the decoder output $\hat{y}_{\mathtt{i}}$ to serve as the segment ranking score: $\mathtt{rank_i} = \hat{y}_{\mathtt{i}}$ <extra_token_id>.

Having obtained the ranking outputs for all segments with the above procedure, a listwise softmax cross-entropy function is used to compute the Softmax loss as follows:

$$\ell_{\mathtt{Softmax}}(y_i, \hat{y}_i) = -\sum_{j=1}^{m} y_{ij} \log \left(\frac{e^{\hat{y}_{ij}}}{\sum_{j'=1}^{m} e^{\hat{y}_{ij'}}} \right) \quad 203$$

where y_i and $\hat{y}_i$ are the gold, and predicted relevance, respectively, and m denotes the number of segments. After computing the Softmax loss, we combine it with the generation loss for joint training:

$$\ell_{\mathtt{total}} = \ell_{\mathtt{generation}} + \lambda \ell_{\mathtt{Softmax}}(y_i, \hat{y}_i) \quad 209$$

in which $\ell_{\mathtt{generation}}$ is a cross-entropy loss computed for generation task, and λ is a balancing parameter that should be tuned.

5 Experimental Setup

5.1 Research questions

We seek to address the following research questions:

- **RQ1:** How does integrating the relative importance of segments influence the automatic and qualitative metrics of summaries?
- **RQ2:** How does the specific type of query affect our system's performance compared to the SOTA?

- **RQ3:** What effect does the integration of LTR offer for segment retrieval?
- **RQ4:** What challenges does the model encounter in underperformed cases?

5.2 Datasets

We used two query-focused datasets during our study: (1) The QMSum dataset (Zhong et al., 2021) consists of 1,808 query-focused summaries extracted from 232 multi-turn meetings across different domains. The dataset is split into training, validation, and testing sets with 1,257, 272, and 279 instances, respectively. The average source length is 9K tokens, and the summary length is 70 tokens. (2) SQuALITY (Wang et al., 2022a) is a collection of question-focused abstractive summarization data with 100 stories, 500 questions, and 2,000 summaries. Each question is accompanied by four reference summaries written by trained writers. The dataset provides train/validation/test splits of 39/25/36, equivalent to 195/125/180 document-question pairs with average document and summary lengths of 5.2K and 237 tokens, respectively.

5.3 Relevance labeling

Given the absence of relevance labels within the instances of datasets employed for experiments, we develop a probability-based heuristic to create such pseudo labels, which signifies the extent to which a segment aligns with the gold summary. Initially, the SUPERPAL approach, as mentioned in (**?**), was employed as an external pseudo-labeling heuristic to match summary spans and their originating source spans, represented by a probability value, p. A specified threshold was then empirically determined for p, allowing only spans exceeding this threshold to be considered as gold during the labeling process. Subsequently, the source spans were mapped to their respective segments, and a scoring function was employed to determine the segment score as follows:

$$\mathtt{Score}(S_i) = \sum_j p_j \log(|\mathtt{span}_j|)$$

where S_i denotes the i-th segment, span_j represents the j-th span within the segment S_i, and p_j shows the probability of $span_j$ being aligned to the gold summary. Intuitively, segments that have more common tokens with the gold summary (i.e., $|\mathtt{span}_j|$) and assigned a higher probability by SUPERPAL approach (i.e., p), are more likely to be ranked higher. Following the calculation of segment scores, they were organized in a sequence, and relevance labels were assigned according to the sorted scores.

5.4 Implementation details

We built upon the code base provided by Vig et al. (2022), adhering to the default hyperparameters. The λ hyperparameter was explored within the set $\{0.5, 1, 1.5\}$, and finally tuned to 1. Furthermore, a probability threshold (p) of 40% was employed to filter gold segments. It has to be mentioned that all parameters, including λ and p, were empirically determined and fixed. Our model comprises 406 million parameters. We employed a single NVIDIA A6000 GPU for both training and evaluation. Each experimental training session spanned a duration of two days.

5.5 Comparison

We compare our model to the well-established SOTA baselines on QFS:

- **SEGENC** (Vig et al., 2022): An abstractive summarizer that segments input, encodes and then decodes with joint attention. Versions include: (1) Finetuned on BART large (SEGENC); (2) pre-finetuned on Wikisum (SEGENC-W);
- **SOCRATIC** (Pagnoni et al., 2022): A question-driven pre-training framework for controllable summarization, fine-tuned on SEGENC. Also, a PEGASUS variant pre-trained on *Book3* is presented.
- **QONTSUM** (Sotudeh and Goharian, 2023): A contrastive learning-based summarizer that distinguishes salient content from top-scored non-salient content.

6 Experimental Results

In this section, we present the automatic and human study results, followed by relevant analyses over query type impact, and segment retrieval.

6.1 Automatic evaluation

As shown in Table 1, we compare the performance of our proposed system with existing state-of-the-art summarization techniques on the QMSum and SQuALITY benchmarks, employing ROUGE and BERTSCORE evaluation metrics to address RQ1 on automatic performance. For the QMSum

	RG-1	RG-2	RG-L	BS
SegEnc (Vig et al., 2022)	37.05	13.03	32.62	87.44
+ *Wikisum Pre-Finetuned* (Vig et al., 2022)	37.80	13.43	33.38	-
Socratic Pret. 1M (Pagnoni et al., 2022)	37.46	13.32	32.79	87.54
Socratic Pret. 30M (Pagnoni et al., 2022)	38.06	13.74	33.51	87.63
QontSum (Sotudeh and Goharian, 2023)	38.42	13.50	34.03	87.72
LTRSum (this work)	**38.82**	**14.11**	**34.45**	**88.07**

(a)

	RG-1	RG-2	RG-L	BS
SegEnc (Vig et al., 2022)	45.68	14.51	22.47	85.86
+ *Wikisum Pre-Finetuned* (Vig et al., 2022)	45.79	14.53	22.68	85.96
Pegasus Pret. (Pagnoni et al., 2022)	45.78	14.43	22.90	85.94
Socratic Pret. 30M (Pagnoni et al., 2022)	**46.31**	**14.80**	22.76	86.04
QontSum (Sotudeh and Goharian, 2023)	45.76	14.27	24.14	**86.07**
LTRSum (this work)	46.11	14.68	**24.23**	86.04

(b)

Table 1: Average of Rouge and BertScore (BS) performance of summarization baselines over (a) QMSum and (b) SQuALITY benchmarks. The baseline performances are reported from previous works.

benchmark, LTRSum surpasses state-of-the-art approaches. In particular, when compared with the QontSum, our method achieves relative improvements of approximately 1.0%, 4.5%, 1.2%, on the Rouge-1, Rouge-2, Rouge-L metrics, respectively. Likewise, LTRSum surpasses Socratic Pret. by relative improvements of 2.0% (Rouge-1), 2.7% (Rouge-2), 2.8% (Rouge-L). Additionally, the BertScore for LTRSum slightly edges out both QontSum and Socratic Pret. On the SQuALITY dataset, LTRSum's performance reveals mixed results; over the QontSum model, it slightly improves Rouge-1 and Rouge-2 metrics. However, when compared to Socratic Pret., LTRSum matches on Rouge-1 and Rouge-2 (with relative deficits under 0.01%), demonstrates a remarkable 5.4% improvement in Rouge-L and aligns closely with the BertScore metrics, on SQuALITY benchmark. This is likely due to the challenges in automatically identifying high-quality ground-truth labels in SQuALITY, unlike QMSum, where our system benefits from human-annotated span labels, while the SQuALITY span labels were determined via a heuristic approach. Furthermore, another likely explanation for Socratic's performance boost may be attributed to its pretraining on the Book3 dataset, which likely shares closer linguistic characteristics with the SQuALITY dataset.

It is essential to note that Socratic undergoes a large-scale pre-training process, driven by questions, which encompasses a vast number of examples drawn from the Book3 corpus, amounting uo to 30M pre-training instances. This approach, while effective, is likely resource-intensive. Conversely, our model, LTRSum, bypasses the extensive pre-training stage and centers on learning an auxiliary task during the fine-tuning phase, making it a more resource-efficient alternative.

6.2 Human evaluation

We conducted human evaluations to assess the quality of the summaries generated by LTRSum, in comparison with QontSumand Socraticbaseline systems. The evaluations were performed on the QMSum and SQuALITY benchmarks. Specifically, we randomly selected 64 test cases (QMSum) and 36 cases (entire test set of SQuALITY), resulting in a total of 100 cases. For each case, we provided two annotators [2] with shuffled summaries, including the gold-spans from the source. To prevent bias, we shuffled summaries

[2] Annotators were PhD students in Science and Engineering.

such that the correspondence could not be guessed. We then ask the annotators to score each case on a scale of 1 to 5 (worst to best) in terms of three qualitative metrics listed below, consistent with the ones employed by Sotudeh and Goharian (2023):

- **Fluency:** To gauge the understandability of a summary, focusing on grammaticality, non-redundancy, and coherence aspects;
- **Relevance:** To assess the extent to which a summary is pertinent as an answer to the given query;
- **Faithfulness:** To measure the degree to which the content covered in the source is faithfully reflected in the generated summary.

Table 2 reports the human evaluation scores over QMSum and SQuALITY datasets. As observed, the LTRSum model shows superior qualitative performance as compared to the QontSum and Socratic baselines on both datasets. The improvements are yet more tangible in the relevance and faithfulness metrics, possibly due to the LTRSum's model objective of finding segments that are more relevant to the query with respect to their relative importance. The close performance of the experimented systems over fluency is expected, given the extensive data the language model has encountered during pre-training to learn to generate coherent text.

The inter-rater agreement scores are as follows: for QMSum, 51%, 52%, and 55% and for SQuALITY, 51%, 57%, and 54% across fluency, relevance, and faithfulness metrics, respectively, indicating a moderate level of consensus among evaluators. While automatic improvements are numerically improved, our system still offers benefits in terms of qualitative (over QontSum and Socratic) and training overhead (over Socratic) baselines, as mentioned earlier. This assessment addresses our RQ1 on qualitative performance.

6.3 Query type impact

We observed a potential relation between the system's qualitative performance and the nature of the query (i.e., query type). Specifically, we noticed that **broad queries** like *"Summarize the whole meeting"* tend to have more gold labels as opposed to **specific queries** like *"Why did the Marketing disagree with the Industrial Design when discussing the possible advanced techniques on the remote control?"*, targeting particular details within the source. To explore this, we categorized the evaluation cases from each dataset based on their query type and compared the human-assigned scores to explore any potential links between the query type and the quality of the generated summaries.

	Fluency	**Relevance**	**Faithfulness**
QMSum			
QontSum	4.09	4.03	3.60
Socratic	4.10	4.15	3.72
LTRSum	**4.14**	**4.36**	**3.88**
SQuALITY			
QontSum	4.01	3.58	3.62
Socratic	**4.02**	3.70	3.69
LTRSum	**4.02**	**3.81**	**3.78**

Table 2: Results of the human study on evaluation samples from the QMSum and SQuALITY datasets (64 cases from QMSum and 36 cases from SQuALTIY)

Table 3 presents a comparison of the LTRSum system against QontSum and Socratic systems, categorized by query types across two datasets. For broad queries, LTRSum outperforms QontSum and Socratic, with notable win rates highlighted in bold; e.g., win rates of 37% (QMSum), and 33% (SQuALITY) in terms of relevance against QontSum. However, with specific queries, our system's performance drops, often trailing the QontSum and Socratic baselines, as evidenced by the high lose rates in bold; e.g., 32% (QMSum) and 34% (SQuALITY) lose rates in relevance compared to QontSum. This trend, both highs and lows, is consistent across all qualitative metrics for both datasets. The differential performance of LTRSum vs. QontSum and Socratic across query types can be attributed to the inherent granularity. In other words, broad queries give LTRSummore room to maneuver since they cover a wide range of gold segments, available for ranking by the LTR component of our model. However, specific queries are trickier; they focus on narrow details within narrow segments, where any slight oversight by the model in identifying salient segments leads to a less relevant summary. In the case of Socratic, the outperformance on specific queries can be attributed to its particular pretraining objective, where narrowed questions are generated for document's single sentences, and the language model is forced to learn to ask & answer the generated questions. Likewise, QontSum excels in handling specific queries compared to broad

Dataset	Query type (%)	Flu.	Rel.	Faith.
QMSum	Broad (53%)	**29**/45/26	**28**/55/17	**25**/60/15
	Specific (47%)	21/54/**25**	15/57/**28**	19/53/**28**
SQuALITY	Broad (46%)	**24**/55/21	**28**/56/16	**26**/58/16
	Specific (54%)	19/58/**23**	14/57/**29**	16/55/**29**

(a) LTRSUM vs. QONTSUM

Dataset	Query type (%)	Flu.	Rel.	Faith.
QMSum	Broad (53%)	16/66/**18**	**41**/29/29	**35**/32/32
	Specific (47%)	17/65/**18**	20/38/**42**	27/33/**40**
SQuALITY	Broad (46%)	**21**/58/18	**35**/41/24	**31**/41/**28**
	Specific (54%)	18/60/**22**	26/32/**42**	21/47/**32**

(b) LTRSUM vs. SOCRATIC

Table 3: Query type impact per model and model comparison with respect to query type. The reported numbers show the win/tie/lose % of LTRSUM against the baselines (i.e., QONTSUM and SOCRATIC), respectively.

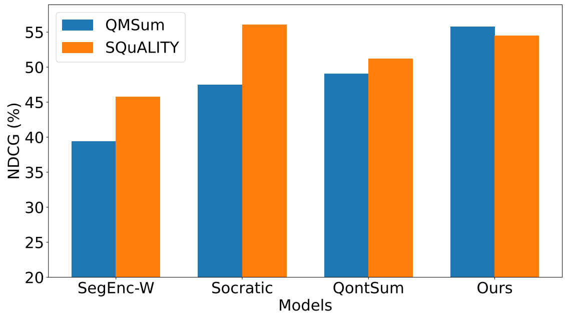

Figure 2: Segment retrieval performance of the models in terms of nDCG score.

queries, suggesting that its contrastive objective is more effective when there are fewer gold segments associated with the query, thereby enhancing the robustness of the objective. This analysis addresses our RQ2.

6.4 Segment retrieval

In order to assess the effectiveness of the summarization system in terms of lifting salient segments w.r.t their relative importance (i.e., ranking), we present a comparative analysis in Figure. 2. To perform this analysis, we first rank the segments per summarization model, given their relative contribution (computed from decoder's attention over the segment tokens) at generating the summary. Subsequently, with the predicted ranked list of segments in hand, we calculate the Normalized Discounted Cumulative Gain (NDCG) score (Wang et al., 2013) as follows:

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2 (i+1)}$$

$$\mathrm{nDCG_p} = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p}$$

where p is a particular ranking position, $rel(i)$ is the relevance score (ranking label) of the segment at position i, and IDCG_p is the ideal cumulative gain (i.e., when the segments are ranked given their gold importance). The relevance scores are obtained by greedily matching the system's ranked segments against the human-annotated important segments. As observed, our system consistently improves the ranking scores on QMSum and is comparable with the best-performing baseline (SOCRATIC) on SQuALITY dataset. This analysis provides support for RQ3.

7 Error Analysis

Two sources of underperformance were identified in response to our RQ4:

Imbalanced Labels. We discovered that in approximately 48% of the underperformed cases, the model exhibited a tendency to misidentify gold segments when generating summaries. Upon further investigation, we observed that these cases were commonly characterized by a label imbalance issue, wherein the model selected segments that contained partially relevant information but were not the actual gold segments. As shown in the example

Query: Why did the Marketing disagree with the Industrial Design when discussing the possible advanced techniques on the remote control?

Human: When discussing adding several advanced techniques onto the remote control to make it more unique, the Industrial Design put forward to using the touchscreen. Notwithstanding the merits of the touchscreen, for instance, a touch screen would make the remote control easier and smaller,the Marketing did not agree to do so just because in that case they would be beyond the budget. *[Written from the 14th segment]*

LTRSum generated: The Marketing believed that it would be too expensive to make a touch screen on the remote control. The Industrial Design believed that the strength of remote controls were most likely to fall down on the ground and get broken. Therefore, it would not be easy to make it fold open and look like a remote control with a touchscreen. *[Written from the 15th segment]*

Query: What did User Interface think about user interface design of remote control?

Human: User Interface found two kinds of remote controls: the multi-functional one and the one easy to use. He emphasized on user-friendliness, but considering that the target people were less than forty years old, multi-function should also be taken into account. *[Written from the 9th segment]*

LTRSum generated: User Interface thought that the remote control should be easy to use with not so many buttons, just a round button which can be pushed in four directions instead of a stick. It should be the same as in the cell phone, just light in the device that shines on all the buttons. *[Written from the 9th segment]*

Table 4: Comparison between human and LTRSum generated summaries for given queries. Left: The model identifies relevant content (highlighted in yellow) from the 15th segment, which is marked in gold due to its 50% overlap with the 14th segment, but also generates irrelevant information from the same 15th segment. Right: The model finds the gold segment (segment 9) but picks up on less relevant parts of the segment.

within Table 4 (left), while both human and LTRSum-generated summaries capture the *budgetary concerns*, LTRSum adds unrelated information about *remote control durability*. This finding sheds light on the challenge of identifying and ranking the gold segments within an imbalanced regime, which may be mitigated in future work through Transfer Learning from a larger dataset (Ruder et al., 2019; Cao et al., 2019).

Segment Summarizer Deficiency. In approximately 39% of the underperformed cases, LTRSum faced challenges in extracting the most pertinent details from the identified gold segments. For instance, as illustrated in Table 4 (right), both the human-written summary and the summary generated by LTRSum drew from the 9th segment (gold). The human summary provided a nuanced understanding of the topic, emphasizing both *user-friendliness* and *multi-functionality* for a *specific age group*. Conversely, the LTRSum summary focused more on the *physical attributes* of the *remote control*, missing out on the *multi-functionality aspect* and the *target demographic*. This observed suboptimality could be attributed to the model's challenges in discerning sentential saliency within the segment which affects the relevancy of the summary. To address this, future work might consider hybrid approaches that combine methods for identifying salient sentences within the identified segments (Pilault et al., 2020).

8 Conclusion

Our method combines Learning-to-Rank with QFS, ensuring content relevance via prioritization. It matches or exceeds SOTA at reduced training costs. Human evaluations highlight improved relevance and faithfulness without compromising fluency. Further analysis suggests that the system outperforms on broad queries while lagging on specific ones, with errors linked to imbalanced labels and segment summarizer challenges.

9 Ethical Considerations

While the proposed summarization system in our paper offers time-saving benefits, it still may produce outputs factually inconsistent with input documents. Such discrepancies risk promoting online misinformation, especially when it is being used on the production scale. This challenge is common in abstractive summarization, necessitating rigorous research and cautious use to prevent false information spread.

References

Rama Badrinath, Suresh Venkatasubramaniyan, and C. E. Veni Madhavan. 2011. Improving query focused summarization using look-ahead strategy. In *European Conference on Information Retrieval*.

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *ArXiv preprint*, abs/1801.07704.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 89–96. ACM.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1565–1576.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 129–136. ACM.

Hoa Trang Dang. 2005. Overview of duc 2005.

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2022. Domain adaptation with pre-trained transformers for query-focused abstractive text summarization. *Computational Linguistics*, 48(2):279–320.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yongqing Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Learning to rank in generative retrieval. In *AAAI Conference on Artificial Intelligence*.

Marina Litvak and Natalia Vanetik. 2017. Query-based summarization using MDL principle. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 22–31, Valencia, Spain. Association for Computational Linguistics.

Xingxian Liu, Bin Duan, Bo Xiao, and Yajing Xu. 2023. Query-utterance attention with joint modeling for query-focused meeting summarization. In *Proc. ICASSP*.

Ahmed Atwan Mohamed and Sanguthevar Rajasekaran. 2006. Improving query-based summarization using document graphs. *2006 IEEE International Symposium on Signal Processing and Information Technology*, pages 408–410.

Artidoro Pagnoni, Alexander R. Fabbri, Wojciech Kryscinski, and Chien-Sheng Wu. 2022. Socratic pretraining: Question-driven pretraining for controllable summarization. *ArXiv preprint*, abs/2212.10449.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Sajad Sotudeh and Nazli Goharian. 2023. Qontsum: On contrasting salient content for query-focused summarization. *ArXiv preprint*, abs/2307.07586.

Dan Su, Tiezheng Yu, and Pascale Fung. 2021. Improve query focused abstractive summarization by incorporating answer relevance. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3124–3131, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Jesse Vig, Alexander Fabbri, Wojciech Kryscinski, Chien-Sheng Wu, and Wenhao Liu. 2022. Exploring neural models for query-focused summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1455–1468, Seattle, United States. Association for Computational Linguistics.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *International Joint Conference on Artificial Intelligence*.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022a. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022b. Simlm: Pre-training with representation bottleneck for dense passage retrieval. In *Annual Meeting of the Association for Computational Linguistics*.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of NDCG type ranking measures. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 25–54. JMLR.org.

Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3632–3645, Online. Association for Computational Linguistics.

Yumo Xu and Mirella Lapata. 2021. Generating query focused summaries from query-free resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6096–6109, Online. Association for Computational Linguistics.

Yumo Xu and Mirella Lapata. 2022. Document summarization with latent queries. *Transactions of the Association for Computational Linguistics*, 10:623–638.

Tiezheng Yu, Ziwei Ji, and Pascale Fung. 2023. Improving query-focused meeting summarization with query-relevant knowledge. *ArXiv preprint*, abs/2309.02105.

Ruifeng Yuan, Zili Wang, Ziqiang Cao, and Wenjie Li. 2022. Few-shot query-focused summarization with prefix-merging. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3704–3714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Weijia Zhang, Svitlana Vakulenko, Thilina C. Rajapakse, and E. Kanoulas. 2021. Scaling up query-focused summarization to meet open-domain question answering. *ArXiv*, abs/2112.07536.

Xiaojuan Zhao and Jun Tang. 2010. Query-focused summarization based on genetic algorithm. *2010 International Conference on Measuring Technology and Mechatronics Automation*, 2:968–971.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11765–11773. AAAI Press.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2022. Rankt5: Fine-tuning t5 for text ranking with ranking losses. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*.