Exploring Neural Models for Query-Focused Summarization

Anonymous ACL submission

Abstract
Query-focused summarization (QFS) aims to produce summaries that answer particular questions of interest, enabling greater user control and personalization. While recently released datasets, such as QMSum or AQuaMuSe, facilitate research efforts in QFS, the field lacks a comprehensive study of the broad space of applicable modeling methods. In this paper we conduct a systematic exploration of neural approaches to QFS, considering two general classes of methods: two-stage extractive-abstractive solutions and end-to-end models. Within those categories, we investigate existing methods and present two model extensions that achieve state-of-the-art performance on the QMSum dataset by a margin of up to 3.38 ROUGE-1, 3.72 ROUGE-2, and 3.28 ROUGE-L. Through quantitative experiments we highlight the trade-offs between different model configurations and explore the transfer abilities between summarization tasks. We also perform human evaluation that suggests the best models produce more comprehensive and factually-consistent summaries compared to a baseline model. Code and checkpoints are made publicly available: https://github.com/anonymized

1 Introduction
Text summarization aims at transforming long documents into short snippets that contain only the most important information from the source document. The field has seen substantial progress driven by the availability of large-scale models pre-trained on vast amounts of data (Devlin et al., 2019; Lewis et al., 2020), the development of summarization-specific pre-training strategies (Zhang et al., 2020; Zhao et al., 2020), and computationally efficient neural architectures (Zaheer et al., 2020).

The majority of recent research efforts in text summarization assume an unconstrained setting in which models are given only a source document as input and are expected to generate a general summary covering the salient aspects from the source. The performance of such models has been evaluated on benchmark datasets spanning various domains: news articles (Nallapati et al., 2016; Narayan et al., 2018; Fabbri et al., 2019a), legal documents (Sharma et al., 2019), scientific writing (Cohan et al., 2018), or creative writing (Kryscinski et al., 2021; Chen et al., 2021). However, it has been shown that summarization in an unconstrained setting is an ill-defined task where multiple generated summaries are equally relevant (Kryscinski et al., 2019). This in turn hinders the ability to evaluate and understand the models’ content selection capacity. In addition, such generic summarization models lack control mechanisms that would allow end users to customize summaries to their particular needs and expectations.

Query-focused summarization (QFS) is a subtask within text summarization that focuses on generating summaries where the summary content is tailored to a user-specified query that is passed alongside the source document as input to the model. Each source document can be associated with multiple unique queries inquiring about different information from that document. In this setting, end users are enabled to explicitly specify their preferences for the summary, and the relevance of the output summary may be evaluated more precisely with respect to the input query. Research on this task has been accelerated by the recently introduced high-quality datasets, such as QMSum (Zhong et al., 2021b) and AQuaMuSe (Kulkarni et al., 2020).

In this work we conduct a systematic, exploratory study of different approaches to query-focused text summarization, considering both two-step and end-to-end neural methods. We present two models, RELREG and SEGENC, which achieve state-of-the-art ROUGE scores on the QMSum dataset by a margin up to 3.38 R-1, 3.72 R-2, and 3.28 R-L. The RELREG model uses a two-step approach to solving the problem, where the first step...
extracts content relevant to the given query and
the next step synthesizes the extracted fragments
into a coherent summary. The SE2ENC method
follows an end-to-end framework in which individ-
ual document segments are separately encoded to
avoid the computational bottleneck of long input
documents, and the decoder jointly attends to all
encoded segments when producing the summary.
Through quantitative studies, we compare our mod-
els with other baselines and discuss the trade-offs of
the end-to-end methods and pipeline approaches.
We also perform human evaluation to understand
the qualitative differences between the models. To-
gether with this manuscript, we share the code base
and model checkpoints to enable future research in
this area.

2 Related Work

2.1 Query-Focused Summarization

Query-focused summarization aims to generate
a summary of a given text conditioned upon a
query. Initial work in this area centered around
unsupervised extractive approaches (Wan et al.,
2007; Litvak and Vanetik, 2017) due to the lim-
lited availability of task-specific training data (Dang,
2005). More recent work has taken advantage of
the relationship between query-focused summa-
rization and the more data-rich task of question an-
swering for extractive summarization (Egonmwan
et al., 2019), reranking documents within a retrieval
pipeline (Su et al., 2020), and abstractive summa-
rization (Su et al., 2021; Baumel et al., 2018; Xie
et al., 2020). Xu and Lapata (2020) introduce a
pipeline consisting of a relevance estimator filter
followed by query-focused evidence and centrality
estimators, while other work converts generic sum-
marization dataset to query-focused training data
(Xu and Lapata, 2021a) or performs latent query
modeling (Xu and Lapata, 2021b).

Recently, several query-focused summarization
datasets have been introduced, which can be fur-
ther divided into short-document datasets, whose
source document length does not exceed the in-
put limits of standard pre-trained models, and long-
document datasets. Within short-document, query-
focused summarization, AnswerSumm (Fabbri
et al., 2021c) is composed of summaries of answers
to queries from StackExchange forums, while Wik-
ishowQA (Liu et al., 2018a) proposes the task
of answer selection followed by the summariza-
tion of individual response articles to queries from
the how-to site WikiHow. Within long-document
summarization, WikiSum (Liu et al., 2018a) cons-
ists of Wikipedia article titles as queries, the first paragraph of the article as the summary, and
documents referenced by the article as the input.
AQuaMuSe (Kulkarni et al., 2020) is a query-
focused multi-document summarization dataset
with user-written queries and human-verified long-
answer summaries from the Natural Questions
dataset (Kwiatkowski et al., 2019), and QMSum
(Zhong et al., 2021b) is a manually-curated dataset
for query-focused dialog summarization. QMSum
and AQuaMuSe are of particular interest to our
study due to the combined challenges of query-
focused and long-document summarization and the
presence of high-quality, curated query-summary
pairs.

Recent work on QMSum has introduced task-
specific denoising objectives for meeting sum-
marization (Zhong et al., 2021a), generated final
fine-grained summaries based on multiple coarse-
grain steps (Zhang et al., 2021a), and treated the
extractive text of an extractive-abstractive model
as a latent variable (Mao et al., 2021). Zhang et al.
(2021b) analyze the challenges of long dialogue
summarization such as the input length, the role of
queries, and domain adaptation. Our work builds
on QA-motivated methods and presents two ap-
proaches yet to be applied in query-focused summa-
rization that each achieve state-of-the-art results, in-
cluding a two-step model and an end-to-end model.

2.2 Long Document Summarization

Long document summarization addresses the set-
ing where source document length exceeds the
input limits of standard pre-trained models. Ap-
proaches to this task can largely be divided into two
categories: two-step extractive-abstractive frame-
works, which first extract a subset of the text
as input to an abstractive model, and end-to-end
models, which process the input within a single
model. The two-step pipeline has been applied to
topic-focused Wikipedia summarization (Liu et al.,
2018b; Liu and Lapata, 2019; Perez-Beltrachini
et al., 2019), low-resource summarization (Bajaj
et al., 2021), and single-document summarization
Chen and Bansal (2018). End-to-end approaches
address the input-length problem using sparse-
attention models. Beltagy et al. (2020) introduce
the Longformer, consisting of local attention as
well as global attention between select input tokens.
Other approaches make use of dynamic attention mechanisms (Zhao et al., 2020; Manakul and Gales, 2021; Cui and Hu, 2021), sliding window strategies (Liu and Chen, 2021), and other mechanisms to introduce sparsity into the model (Huang et al., 2021; Liu et al., 2021). Izacard and Grave (2021) concatenate the outputs of multiple encoders as input to a generator component for the task of open domain question answering. In our work we build on these models for query-focused summarization and perform extensive hyperparameter ablations, achieving state-of-the-art results over other two-step and end-to-end models.

3 Methodology

We present existing methods and propose modeling extensions to address the challenges of query-focused summarization.

3.1 Two-Step Approaches

Two-step approaches consist of an extractor model, which extracts parts of the source document relevant to the input query, and an abstractor model, which synthesizes the extracted segments into a final summary. We consider score-and-rank extractor models, which first score each source passage for relevance to the query and then rank the passages in descending order of relevance, with the concatenated and truncated results passed to the abstractor. In this work we present two types of scoring models: single-encoder models and dual-encoder models, which we describe below. All two-step approaches share the same abstractor, a BART-large model.

3.1.1 Single-Encoder Models

Single encoder models concatenate a query and source passage as input to the scoring function that produces the similarity score. Those models benefit from full cross-attention between query and passage, resulting in richer data representations.

MARGE (Xu and Lapata, 2021a) is a single-encoder, Masked ROUGE extractor that aims to improve upon low-resource query-focused summarization by synthesizing query-focused data from more resource rich, generic summarization datasets. This model is trained to predict the relevance of each passage in the source document with respect to a query, where the proxy for relevance is the ROUGE overlap between the passage and the reference summary. For training on generic summarization datasets, MARGE uses pseudo-queries that are created by masking content words in the reference summaries.

When performing inference using real queries, certain query words (e.g., wh-words) are masked to better align the queries to the pseudo-queries from the training process. Following Xu and Lapata (2021a), we apply MARGE trained for masked relevance prediction on Multi-News (Fabbri et al., 2019b) without training on our target dataset.

RELREG (Motivated by the retrieval component of MARGE, we propose the RELREG (RELevance REGression) model, which trains a relevance prediction model directly on QFS data using the original, non-masked query. Like MARGE, this model is trained to predict the ROUGE overlap between a source passage and the reference summary, using only the passage and query as input. A single-encoder model jointly encodes the delimiter-separated query and passage, and the final layer of the model outputs the predicted relevance value.

3.1.2 Dual-Encoder Models

Dual-encoder models separately encode a query and source passage before calculating the cosine similarity between the embeddings to compute the relevance score. This class of models offers computational benefits, as passage embeddings may be precomputed and stored for a given input, while the single-encoder model must be run over all passages should a new query be introduced.

DPR (Karpukhin et al., 2020) is a dual-encoder model that separately encodes queries and passages into an embedding space optimized for calculating semantic similarity between the two, showing improved results over traditional vector-space models. We fine-tune a DPR extractor model directly on the target dataset. As opposed to other locators that optimize with respect to the continuous ROUGE overlap, DPR uses the ROUGE score between the passage and reference summary to identify binary positive and negative passages and optimizes the negative log likelihood of the positive passages.

RELREGTT (RELevance REGression Two Tower) is a more computationally-efficient version of RELREG that uses a dual-encoder architecture to predict ROUGE-based relevance scores. This model is implemented with a backbone architecture of Sentence-BERT (Reimers and Gurevych, 2019), using a shared-parameter encoder for each of the

3
query and passage and a special token appended to each input that identifies it as either query or passage, following the suggested best practices of Reimers and Gurevych (2019). The final output for the model is based on the inner product of the pooled embeddings for the query and passage.

### 3.2 End-to-End Approaches

Two-step pipelines depend on the strength of the retrieval component, and may still fail to capture all relevant content despite an ideal retriever, due to length limitations of the generation component. This motivates our experiments on end-to-end models that can incorporate longer input texts.

**BART** (Lewis et al., 2020) As a baseline end-to-end model, we consider BART, an encoder-decoder Transformer model pre-trained using a denoising objective. BART is composed of a bidirectional encoder module and an autoregressive decoder model that attends to the encoder’s final layer outputs. Due to the quadratic memory complexity of the encoder’s full self-attention mechanism, the model input size is limited to 1024 tokens. In our experiments, we prepare the input to BART by concatenating the query, a delimiter token, and the source document, and then truncating the combined text to the model’s input size.

**LED** To circumvent the input size limitations of the BART model, we include the Longformer Encoder-Decoder (Beltagy et al., 2020) (LED) in our study LED replaces the quadratic self-attention mechanism of traditional Transformers with a memory-efficient version that combines local attention with sparse global attention. The architecture allowed us to run experiments with input sizes up to 16384 tokens. Based on insights from the original work on tuning the model to the QA task, we configure the global attention mechanism to span the entire query.

**SEGEnc** We also consider a simpler form of sparse attention in the encoder based solely on windowed local attention, combining elements of LED with Fusion-in-Decoder (FiD) (Izacard and Grave, 2021), a model for open-domain question answering. In our Segment Encoder (SEGEnc) model, the source document is split into fixed-length overlapping segments, each of which is separately appended to the query and encoded using a standard Transformer model. Similar to FiD, these encodings are then concatenated into a single embedding sequence and passed to a decoder model that generates the summary. Since there is no cross-attention between the encoded segments, the attention mechanism scales linearly in the number of segments and hence the length of the source document. Nonetheless, the decoder can attend to all encoded segments jointly, enabling the encoder-decoder architecture to operate in an end-to-end fashion. This model is motivated by two hypotheses: 1) query-relevant sections within a source document are often small enough to be processed by standard Transformer models (e.g. 1024 tokens), and 2) each query-relevant section may be understood independently of other sections, removing the need for cross-attention between the segments.

### 3.3 Data

**QMSum** (Zhong et al., 2021b) is a query-focused dialogue summarization dataset consisting of 1,808 query-summary pairs over 232 meetings from product design, academic, and political committee meetings, all conducted in English. QMSum also includes additional annotations such as topic segmentations and highlighted text spans associated with reference summaries. We leverage the provided span annotations to run oracle experiments.

**AQuaMuSe** (Kulkarni et al., 2020) is a query-focused multi-document summarization dataset consisting of 5,519 query-long answer summary pairs from the Natural Questions question-answering dataset (Kwiatkowski et al., 2019) and associated input documents from the Common Crawl\(^2\). Input documents for the original dataset were selected based on embedding similarity with respect to the summary, and hyperparameters can be chosen to control the level of semantic overlap between the input document set and the summary. Data replication details are found in the Appendix.

### 3.4 Experiment Setup

**Implementation** Models were implemented using the PyTorch (Li et al., 2020) and Huggingface (Wolf et al., 2019) libraries. Model weights were initialized from pre-trained checkpoints available through the Huggingface Model Hub\(^3\). All BART models were

---

1 We use segments that are 50% overlapping, though other configurations may be considered.

2 https://commoncrawl.org/

3 https://huggingface.co/models
Training & Inference  Models were trained for 10 epochs with final checkpoints selected based on the average of ROUGE-{1, 2, L} (R-1, R-2, R-L) scores achieved on the validation set. Gradient checkpointing (Chen et al., 2016) was used for the LED and SEGENC models to reduce the memory footprint. Model outputs were decoded using beam search with 4 beams. To ensure high consistency of results, all experiments in §4 were repeated 5 times with results averaged across runs.

Evaluation  Models were automatically evaluated using the ROUGE-{1, 2, L} metrics (Lin, 2004) included in the SummEval toolkit (Fabbri et al., 2021b). Models were also manually evaluated by hired human annotators. Annotators were hired through the Amazon Mechanical Turk platform. Workers were selected from English speaking countries and offered an hourly rate of approximately 12 USD. The study was conducted on 50 model generated examples chosen at random from the test set of QMSum.

4 Results & Analysis  In this section, we first analyze the effects of model-specific architectural and hyperparameter choices on the performance of two-stage (§4.1) and end-to-end models (§4.2). Next, we study the task-specific knowledge transfer capabilities of different pretraining strategies in §4.3. Lastly, we conduct a final evaluation and comparison of all discussed models in §4.4. All experiments and analyses presented in this section were conducted on QMSum.

4.1 Two-Stage Approaches  For two-stage models, we first focus on evaluating the extractor component and comparing performance to baseline heuristics. We quantify extractor performance using two metrics: 1) lexical overlap between the extracted utterances and reference summaries, computed using R-1, R-2, and R-L metrics, 2) span overlap between the extracted and golden spans included with QMSum represented by Precision and Recall scores, with results shown in Table 1. In both cases, we first order utterances of the conversation according to the scores assigned by the extractor models, then concatenate the utterances and finally truncate the result to 1024 tokens (excluding the space reserved for the query) to mimic the input length limits of downstream abstracter models; we present those numbers as the All columns in the table. For the lexical overlap, we also show the scores for the best 1 (Top-1), 5 (Top-5), and 15 (Top-15) utterances.

The results show that the best-performing extractor model is REL. REG closely followed by REL.REGTT in the Top-1 evaluation and DPR in the Top-5, Top-15, and All cases. We note that both the REL.REG and REL.REGTT models tend to select longer utterances than the other extractors; the regression-based training mirrors the ROUGE overlap score which favors longer, more informative utterances. However, despite their strong performance in extracting top-matching utterances, the results also expose a considerable gap between model-based approaches and human annotations when considering the entirety of extracted spans.
This shows a promising topic for future work in this matter. We also notice that despite the simplicity of the LEAD heuristic, which extracts the first $k$ utterances in their original order, it remains competitive with the data-driven extractor models when we consider the All case. An extended version of this study, which includes the lexical overlap between extracted spans and input queries is presented in Table 8 in the Appendix.

Next, we analyze how the performance of the extractor components carries over to the final summarization task. For the best-performing model, we additionally test the effect of varying the input segment size used during training and inference between 256 and 512 tokens. Validation-set results for all models are reported in Table 2.

We find that DPR slightly outperforms RELREGTT for dual-encoder models. Among single-encoder models, RELREG outperforms MARGE by over a full R-1 point, which may be explained by RELREG using more direct supervision based on an in-domain query, rather than creating synthetic queries from an external dataset using masking. We find that the single-encoder RELREG outperforms the best dual-encoder model; the cross-attention term in the single-encoder RELREG model allows it to better attend to the query when determining relevance. Intuitively, the ordering of results corresponds to the span overlap recall with the gold spans; the ability of the extractor to select produce high-recall rankings directly affects abstractor performance. We see that increasing the input segment length used in training and inference for RELREG improves at 256 tokens but decreases at 512 tokens, suggesting that a balance is found between including additional context for ranking versus enabling a greater number of shorter segments that may capture more diverse content from the source.

### Table 2: Performance of two-step models on the QM-Sum validation set, divided into dual-encoder and single-encoder extractors. Input segment lengths are indicated in parentheses, and otherwise the model operates on utterance-level input.

<table>
<thead>
<tr>
<th>Model</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR</td>
<td>32.79</td>
<td>9.82</td>
<td>28.91</td>
</tr>
<tr>
<td>RELREGTT</td>
<td>32.65</td>
<td>9.00</td>
<td>28.57</td>
</tr>
<tr>
<td>MARGE</td>
<td>31.90</td>
<td>9.10</td>
<td>28.17</td>
</tr>
<tr>
<td>RELREG</td>
<td>33.43</td>
<td>9.77</td>
<td>29.40</td>
</tr>
<tr>
<td>RELREG (256)</td>
<td>34.67</td>
<td>11.53</td>
<td>30.66</td>
</tr>
<tr>
<td>RELREG (512)</td>
<td>32.22</td>
<td>10.29</td>
<td>29.49</td>
</tr>
</tbody>
</table>

### Table 3: Performance of end-to-end models on the QM-Sum validation set, across varying input and attention window sizes (in number of tokens). SEGEnc-D is a variant of SEGEnc in which the segments are disjoint rather than overlapping; this ablation was evaluated on the best-performing SEGEnc hyperparameters.

<table>
<thead>
<tr>
<th>Model</th>
<th>Input</th>
<th>Attn</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR</td>
<td>4096</td>
<td>256</td>
<td>31.55</td>
<td>8.89</td>
<td>27.62</td>
</tr>
<tr>
<td>RELREGTT</td>
<td>512</td>
<td>512</td>
<td>32.25</td>
<td>9.27</td>
<td>28.29</td>
</tr>
<tr>
<td>MARGE</td>
<td>1024</td>
<td>512</td>
<td>32.16</td>
<td>9.05</td>
<td>28.27</td>
</tr>
<tr>
<td>RELREG</td>
<td>8192</td>
<td>512</td>
<td>31.79</td>
<td>8.97</td>
<td>27.73</td>
</tr>
<tr>
<td>RELREG (256)</td>
<td>1024</td>
<td>32.65</td>
<td>9.26</td>
<td>28.73</td>
<td></td>
</tr>
<tr>
<td>RELREG (512)</td>
<td>16384</td>
<td>512</td>
<td>31.94</td>
<td>9.16</td>
<td>27.73</td>
</tr>
<tr>
<td>SEGEnc</td>
<td>4096</td>
<td>256</td>
<td>35.35</td>
<td>10.37</td>
<td>30.91</td>
</tr>
<tr>
<td>BART</td>
<td>512</td>
<td>512</td>
<td>35.25</td>
<td>10.36</td>
<td>30.85</td>
</tr>
<tr>
<td>LED</td>
<td>1024</td>
<td>512</td>
<td>34.36</td>
<td>9.85</td>
<td>30.13</td>
</tr>
<tr>
<td>SEGEnc</td>
<td>8192</td>
<td>512</td>
<td>36.51</td>
<td>11.36</td>
<td>31.87</td>
</tr>
<tr>
<td>BART</td>
<td>1024</td>
<td>512</td>
<td>36.68</td>
<td>11.71</td>
<td>32.08</td>
</tr>
<tr>
<td>LED</td>
<td>16384</td>
<td>512</td>
<td>35.48</td>
<td>10.97</td>
<td>31.21</td>
</tr>
<tr>
<td>SEGEnc-D</td>
<td>16384</td>
<td>512</td>
<td>37.21</td>
<td>12.14</td>
<td>32.67</td>
</tr>
<tr>
<td>BART</td>
<td>1024</td>
<td>512</td>
<td>37.47</td>
<td>12.47</td>
<td>32.95</td>
</tr>
<tr>
<td>LED</td>
<td>1024</td>
<td>512</td>
<td>36.30</td>
<td>11.71</td>
<td>32.01</td>
</tr>
</tbody>
</table>

4For SEGEnc, attention window size is equivalent to segment size.
Table 4: QMSum validation-set performance of the end-to-end BART models fine-tuned on related summarization tasks and then further fine-tuned on QMSum data. The model indicates the task first fine-tuned on, and input is truncated to 1024 tokens.

<table>
<thead>
<tr>
<th>Model</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Transfer</td>
<td>34.42</td>
<td>9.62</td>
<td>28.37</td>
</tr>
<tr>
<td>AnswerSumm</td>
<td>34.36</td>
<td>9.64</td>
<td>30.22</td>
</tr>
<tr>
<td>AQuaMuse</td>
<td>34.57</td>
<td>9.78</td>
<td>30.42</td>
</tr>
<tr>
<td>WikiHowQA</td>
<td>33.08</td>
<td>9.03</td>
<td>28.48</td>
</tr>
<tr>
<td>CNNDM</td>
<td>33.87</td>
<td>9.36</td>
<td>28.48</td>
</tr>
<tr>
<td>WikiSum</td>
<td>34.73</td>
<td>9.80</td>
<td>30.54</td>
</tr>
</tbody>
</table>

Table 5: QMSum test-set performance of two-stage and end-to-end models that performed best on the validation set (Tables 2 and 3), including versions fine-tuned from the WikiSum-finetuned checkpoint (denoted by -W). Results reported in prior work are italicized. Also included is an extractive-oracle model that takes the gold spans (§3.3) as input.

<table>
<thead>
<tr>
<th>Model</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baselines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYLE</td>
<td>34.42</td>
<td>9.71</td>
<td>30.10</td>
</tr>
<tr>
<td>SUMM^N</td>
<td>34.03</td>
<td>9.28</td>
<td>29.48</td>
</tr>
<tr>
<td>BART</td>
<td>31.87</td>
<td>9.08</td>
<td>27.50</td>
</tr>
<tr>
<td>BART-W</td>
<td>32.68</td>
<td>8.97</td>
<td>28.74</td>
</tr>
<tr>
<td>BART-W (Gold)</td>
<td>35.54</td>
<td>15.65</td>
<td>35.17</td>
</tr>
</tbody>
</table>

Two-stage
<table>
<thead>
<tr>
<th>Model</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR</td>
<td>32.28</td>
<td>9.73</td>
<td>28.34</td>
</tr>
<tr>
<td>RELREGTT</td>
<td>33.02</td>
<td>10.17</td>
<td>28.90</td>
</tr>
<tr>
<td>MARGE</td>
<td>31.99</td>
<td>8.97</td>
<td>27.93</td>
</tr>
<tr>
<td>RELREG</td>
<td>34.91</td>
<td>11.91</td>
<td>30.73</td>
</tr>
<tr>
<td>RELREG-W</td>
<td><strong>36.45</strong></td>
<td><strong>12.81</strong></td>
<td><strong>32.28</strong></td>
</tr>
</tbody>
</table>

End-to-end
<table>
<thead>
<tr>
<th>Model</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED</td>
<td>34.18</td>
<td>10.32</td>
<td>29.95</td>
</tr>
<tr>
<td>SEGENC</td>
<td>37.05</td>
<td>13.03</td>
<td>32.62</td>
</tr>
<tr>
<td>SEGENC-W</td>
<td><strong>37.80</strong></td>
<td><strong>13.43</strong></td>
<td><strong>33.38</strong></td>
</tr>
</tbody>
</table>

4.3 Task-Specific Transfer

Having determined the best-performing models, we examine whether performance can be further improved by fine-tuning a model that has already been fine-tuned for a different summarization task. We conduct this study using the end-to-end BART on 1024 tokens, as this model is the backbone, albeit in varying ways, of both our two-step and end-to-end models. We test the transferring capabilities of models trained on the news summarization task from CNN/DailyMail (Nallapati et al., 2016) as well as the previously-mentioned query- and topic-focused summarization tasks: AnswerSumm, AQuaMuSe, WikiHowQA, and WikiSum. We compare to fine-tuning from the original BART checkpoint, with results shown in Table 4.

We find that transferring from any of the tasks improves over no transfer in R-1 and R-L. Transferring from any of the constrained, query-focused tasks outperforms transferring from unconstrained news summarization. Furthermore, transferring from WikiSum outperforms transfer from other datasets, which aligns with other work that shows the generalizability of Wikipedia as a source of data for task transfer (Fabbri et al., 2021a).

4.4 Final Results

We now measure the test set performance of the best-performing architectures from §4.1 and §4.2 in combination with the optimal transfer-learning approach from §4.3. Results are presented in Table 5 along with baseline models.

We find that RELREG and SEGENC outperform existing state-of-the-art models by a substantial margin, and that initializing the model from the WikiSum-fine-tuned checkpoint further improves performance, with the best model exceeding current state-of-the-art performance by a difference of 3.38 R-1, 3.72 R-2, and 3.28 R-L. Comparing the best models from each category, we find that the end-to-end approach outperforms the two-stage. Within the two-stage dual-encoder models, RELREGTT outperforms DPR on the test set despite the slightly worse performance on the validation set. We attribute this variation to the small size of the validation set, and our other findings re-
main consistent across validation and test sets. The single-encoder RELREG outperforms the best dual-encoder model, with RELREG-W improving upon the current state-of-the-art performance by a difference of 2.03 R-1, 3.10 R-2, and 2.18 R-L.

5 Further Analysis

In this section we conduct further analysis of the best performing models from Section 4. First, we offer additional insights into the performance of those models on the QMSum dataset through a human-based study. Next, we discuss the generalization abilities of those models by running experiments on the AQuaMuSe dataset.

5.1 Human Evaluation

To gain a better understanding of the performance of the models on the QMSum dataset, human judges were hired and asked to assess the quality of generated summaries. Summaries were evaluated across four dimensions: fluency (Flu.), relevance (Rel.), completeness (Comp.), and factuality (Fact.).

<table>
<thead>
<tr>
<th>Model</th>
<th>Flu.</th>
<th>Rel.</th>
<th>Comp.</th>
<th>Fact.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BART</td>
<td>4.08</td>
<td>3.68</td>
<td>3.22</td>
<td>3.31</td>
</tr>
<tr>
<td>RELREG-W</td>
<td>3.87</td>
<td>3.81</td>
<td>3.67</td>
<td>3.70</td>
</tr>
<tr>
<td>SEGEnc-W</td>
<td>3.93</td>
<td>3.87</td>
<td>3.81</td>
<td>3.63</td>
</tr>
</tbody>
</table>

Table 6: Human evaluation of two best-performing models from Section 4, along with a baseline BART model. Summaries were evaluated across four dimensions: fluency (Flu.), relevance (Rel.), completeness (Comp.), and factuality (Fact.).

due to its conversational nature. The results also highlight a gap between the performance of existing models and perfect scores, which shows that there is potential for improvement in future work.

5.2 Dataset Generalization

To test that the results from the previous section generalize beyond the QMSum dataset, we evaluated the best-performing models on AQuaMuSe, another high-quality dataset for QFS that includes long documents (§2.1, §3.3). Test-set performance for the best-performing two-stage and end-to-end models, along with a baseline BART model, are shown in Table 7. Results are consistent with those for the QMSum dataset (Table 5), with the best end-to-end model (SEGEnc-W) outperforming the best two-stage model (RELREG-W), and both outperforming the baseline (BART) model.

<table>
<thead>
<tr>
<th>Model</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hi-MAP</td>
<td>30.34</td>
<td>14.82</td>
<td>26.86</td>
</tr>
<tr>
<td>BART</td>
<td>48.74</td>
<td>33.96</td>
<td>46.02</td>
</tr>
<tr>
<td>RELREG-W</td>
<td>54.06</td>
<td>38.51</td>
<td>51.07</td>
</tr>
<tr>
<td>SEGEnc-W</td>
<td>63.62</td>
<td>51.27</td>
<td>61.37</td>
</tr>
</tbody>
</table>

Table 7: AQuaMuSe test-set performance of two best-performing models from §4, along with a baseline BART model and previously reported results (in italics) for Hi-MAP (Fabbri et al., 2019a) from Kulkarni et al. (2020). Note that the version of the dataset used for previous results would have been slightly different due to variations in document selection parameters and Common Crawl indices (see Appendix).
7 Ethical Considerations

Dataset Biases  QMSum and AQuaMuSe contain meeting transcripts and documents in English and thus mainly represent the culture of the English-speaking populace. Political or gender biases may also exist in the dataset, and models trained on these datasets may propagate these biases. Additionally, the pretrained BART model carries biases from the data it was pretrained on. We did not stress test these models for biases and request that the users be aware of these potential issues in applying the models presented.

Crowdsourcing Protocols  Workers were compensated $1 per example, calibrated to equal a $12/hour payrate. We use the following qualifications to recruit MTurk workers with good track records: HIT approval rate greater than or equal to 97%, number of HITs approved greater than or equal to 10000, and located in one of the following English native-speaking countries: Australia, Canada, New Zealand, United Kingdom, United States.

Misuse Potential and Failure Mode  When properly used, the summarization models described in this paper can be time-saving. However, the current model outputs may be factually inconsistent with the input documents, and in such a case could contribute to misinformation on the internet. This issue is present among all current abstractive summarization models and is an area of active research.

Environmental Cost  The experiments described in the paper primarily make use of A100 GPUs. We typically used a single GPU per experiment, and the experiments may take up to a day when repeating across random seeds. The largest backbone model used, BART-Large, has 400 million parameters. While our work required extensive experiments, future work and applications can draw upon our insights and need not repeat these comparisons.

References


Yumo Xu and Mirella Lapata. 2021a. 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.


**A Appendix**

**Locator Model Parameters** For MARGE experiments, we apply the original fine-tuned BERT-base checkpoint from Xu and Lapata (2021a), while for DPR, we fine-tune a BERT-base model for both query and passage encoders following Karpukhin et al. (2020).

We report results for REL REG fine-tuned from an Electra-large checkpoint (Clark et al., 2020). For a fair comparison with other metrics, we also fine-tuned REL REG from a BERT-base checkpoint. This version still outperformed DPR by about a point in R-1, R-2, and R-L, demonstrating the advantage of this locator approach beyond the chosen base model.

We apply REL REG TT fine-tuned from a distilled RoBERTa base (Liu et al., 2019) checkpoint initially fine-tuned for the task of entailment. This approach of continuing fine-tuning from an entailment checkpoint is suggested by the sentence transformers library (Reimers and Gurevych, 2019). We also experimented with fine-tuning the REL REG TT model from BERT-base and Electra-large checkpoints, but these locators did not perform better in initial experiments.

**Summarization Model Parameters** In all experiments described in this work, the LED model was initialized from the allenai/led-large-16384 checkpoint. Two model hyperparameters, maximal input size and attention window size, were chosen through a hyperparameter search with candidate models selected based on their performance on the validation set. Best hyperparameters were found to be: 16384 maximum input size, and 1024 attention window size. LED models were trained for 10 epochs, with a batch size 4, gradient accumulation set to 4 steps, and learning rate set to 0.000005. The SegEnt model was initialized from the facebook/bart-large checkpoint. The model hyperparameters, maximal input size and attention window size, were chosen through a hyperparameter search with candidate models selected based on their performance on the validation set, with results reported in the paper. Best hyperparameters were found to be: 16384 maximum input size, and 512 attention window size. The SegEnt models were trained for 10 epochs, with a batch size of 1 and learning rate set to 0.000005.

**QMSum Details** QMSum contains 1,808 query-summary pairs in total, with a train/validation/test split of 1257/272/281. It is made available through
### Table 8: Performance of extractor models on the validation set.

The left and middle sections present the lexical overlap between utterances retrieved by extractor models and the reference summaries and summary queries, accordingly. Lexical overlap is evaluated by means of ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L) metrics. Segments of the section focus on the lexical overlap between the highest ranked 1 (Top-1), 5 (Top-5), 15 (Top-15) utterances, and all utterances truncated to a 1024 token limit (All). The table also includes the average word counts of all extracted utterances, denoted as $\bar{x}$. The right section shows the span overlap between the utterance spans retrieved by the extractor models and those collected from human annotators by the authors of QMSum. The performance is evaluated by means of Precision and Recall scores and uses the highest ranked utterances truncated to the limit of 1024 tokens.

---

**AQuaMuse Details**

We experiment the V3, abstractive version of AQuaMuse, consisting of 7725 query-summary pairs, with a train/validation/test split of 5566/596/734. The original AQuaMuse paper reported results on V2 of the dataset, which contains a slightly different input document set due to variations in the semantic overlap threshold used to retrieve documents. Some input documents could not be retrieved due to differences in the Common Crawl index used; we used the cleaned, reproduced version of the C4 dataset (Raffel et al., 2020) from the Common Crawl made available by AI2. We kept examples for which all input documents were found, which resulted in a dataset of 6896 examples. The natural language questions it contains are made available through an Apache 2.0 license, which aligns with our use for research purposes. This dataset uses publicly available entities from Wikipedia.

**B Human Annotation Interface**

The instructions shown to the annotators during human studies are presented in Figure 1

---

5 https://github.com/Yale-LILY/QMSum/blob/main/LICENSE

6 https://github.com/allenai/allennlp/discussions/5056

7 https://github.com/google-research-datasets/natural-questions/blob/master/LICENSE
Instructions

In this task you will evaluate the quality of summaries of a conversation. The summaries were written to answer a question about the conversation. To correctly solve this task, follow these steps:

1. Carefully read the Question and the Conversation, be aware of the information they contain.
2. Read the proposed summaries A-C (3 in total).
3. Rate each summary on a scale from 1 (worst) to 5 (best) by its fluency, relevance, completeness, factuality.

Definitions

Fluency
This rating measures the grammatical quality of the Summary text, is it well-written and grammatically correct.
Check the quality of individual sentences.
Fluency can be rated without considering the Conversation or the Question.

Relevance:
The rating measures how relevant the Summary is to the Question ignoring the Conversation.
Check whether the content of the Summary is on topic with respect to the Question.
Relevance must be rated considering the Question.

Completeness:
The rating measures how completely/comprehensively the Summary answers the Question.
Check whether all necessary information from the Conversation is present in the Summary.
Completeness must be rated considering the Question and Conversation.

Factuality:
The rating measures whether the Summary is factually correct with respect to the Conversation.
Check whether all the facts listed in the Summary are backed by facts from the Conversation.
Factuality must be rated considering the Conversation.

Figure 1: Instructions presented to annotators for the human studies