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 (Kryściński 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 SEGENC method follows an end-to-end framework in which individual 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 models with other baselines and discuss the trade-offs of the end-to-end methods and pipelined approaches. We also perform human evaluation to understand the qualitative differences between the models. Together 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 limited availability of task-specific training data (Dang, 2005). More recent work has taken advantage of the relationship between query-focused summarization and the more data-rich task of question answering for extractive summarization (Egonmwan et al., 2019), reranking documents within a retrieval pipeline (Su et al., 2020), and abstractive summarization (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 summarization 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 further divided into short-document datasets, whose source document length does not exceed the input limits of standard pretrained 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 WikiHowQA (Liu et al., 2018a) proposes the task of answer selection followed by the summarization of individual response articles to queries from the how-to site WikiHow. Within long-document summarization, WikiSum (Liu et al., 2018a) consists 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 summarization (Zhong et al., 2021a), generated final fine-grained summaries based on multiple coarse-grained 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 approaches yet to be applied in query-focused summarization that each achieve state-of-the-art results, including a two-step model and an end-to-end model.

2.2 Long Document Summarization

Long document summarization addresses the setting where source document length exceeds the input limits of standard pre-trained models. Approaches to this task can largely be divided into two categories: two-step extractive-abstractive frameworks, 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, **Ma**sked **R**OU**GE** 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 query and passage and a special token appended to each input that identifies it is 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[1] 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

Model	Lexical Overlap b/w Extractors and References																Span Overlap b/w Extractors and Golden Spans	
	Top-1				Top-5				Top-15				All				All	
	R-1	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$	*Precision*	*Recall*
Gold Spans	15.00	3.80	11.10	60	**20.89**	**6.05**	**15.04**	218	**19.62**	**5.99**	**14.28**	386	**16.09**	**5.60**	**12.47**	660	0.75	1.00
Lead	8.17	0.98	6.30	82	12.84	1.69	9.17	309	13.13	1.81	9.21	463	8.77	1.79	6.77	978	0.09	0.20
DPR	11.31	1.99	8.72	34	17.46	2.86	12.21	156	15.38	2.74	10.64	394	9.75	2.23	7.42	932	0.22	0.27
RelRegTT	23.67	3.34	15.66	82	16.13	3.35	11.18	413	9.65	2.58	7.31	930	9.16	2.52	6.99	994	0.07	0.24
MaRGE	7.13	0.72	5.81	20	13.76	1.39	10.22	92	14.85	1.74	11.09	269	9.21	1.52	7.16	896	0.15	0.21
RelReg	**24.57**	**4.33**	**16.57**	88	17.52	4.11	12.21	418	10.56	3.04	8.06	884	9.62	2.87	7.47	989	0.11	0.28

Table 1: Performance of extractor models on the QMSum validation set. The left section presents the lexical overlap between the utterances retrieved by extractor models and the reference summaries, 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.

based on the `facebook/bart-large` checkpoint, the LED-model was based on the `allenai/led-large-16384` checkpoint, which itself is based on `BART-large`.

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 pre-training 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 abstractor 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 RelReg closely followed by RelRegTT in the *Top-1* evaluation and DPR in the *Top-5*, *Top-15*, and *All* cases. We note that both the RelReg and RelRegTT 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.

Model	R-1	R-2	R-L
DPR	32.79	9.82	28.91
RelRegTT	32.65	9.00	28.57
MaRGE	31.90	9.10	28.17
RelReg	33.43	9.77	29.40
RelReg (256)	**34.67**	**11.53**	**30.66**
RelReg (512)	32.22	10.29	29.49

Table 2: Performance of two-step models on the QMSum 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.

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 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.

Model	Input	Attn	R-1	R-2	R-L
BART	1024	1024	32.42	9.62	28.37
LED	4096	256	31.55	8.89	27.62
		512	32.25	9.27	28.29
		1024	32.16	9.05	28.27
	8192	256	31.79	8.97	27.75
		512	32.76	9.38	28.65
		1024	32.85	9.26	28.73
	16384	256	31.94	9.16	27.73
		512	32.88	9.82	28.90
		1024	32.98	9.60	29.08
SegEnc	4096	256	35.35	10.37	30.91
		512	35.25	10.36	30.85
		1024	34.36	9.85	30.13
	8192	256	36.51	11.36	31.87
		512	36.68	11.71	32.08
		1024	35.48	10.97	31.21
	16384	256	37.21	12.14	32.67
		512	**37.47**	**12.47**	**32.95**
		1024	36.30	11.71	32.01
SegEnc-D	16384	512	36.68	11.97	32.35

Table 3: Performance of end-to-end models on the QMSum 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.

4.2 End-to-End Approaches

We explore hyperparameter choices for two end-to-end architectures described in §3.2: the Longformer Encoder-Decoder (LED) and Segment Encoder (SegEnc). For both models, we consider different choices for input size (4096, 8192, or 16384 tokens) and attention window size[4] (256, 512, or 1024 tokens). For SegEnc, we also consider two different segmentation strategies: overlapping segments (50% overlap) and disjoint segments. Validation set results for both models and a baseline BART model are reported in Table 3.

We notice that both the LED and SegEnc benefit from increasing the input size and perform best with the input limit set to 16,384 tokens. The optimal attention window for LED is 1024, while SegEnc performs best with an attention window of 512 tokens. For SegEnc, using overlapping segments improves performance compared to using disjoint segments, suggesting that the additional context provided by the former approach is helpful for locating relevant content. The SegEnc model achieves the highest performance out of the end-to-end architectures with ROUGE scores of 37.47 *R-1*, 12.47 *R-2*, and 32.95 *R-L* on the validation set.

The results also highlight that while the

[4]For SegEnc, attention window size is equivalent to segment size.

Model	R-1	R-2	R-L
No Transfer	34.42	9.62	28.37
AnswerSumm	34.36	9.64	30.22
AQuaMuse	34.57	9.78	30.42
WikiHowQA	33.08	9.03	28.48
CNNDM	33.87	9.36	28.48
WikiSum	**34.73**	**9.80**	**30.54**

Table 4: QMSum validation-set performance of the end-to-end BART models first 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.

LED model matches or slightly outperforms the BART baseline for higher maximum input and window sizes, it performs substantially worse than SEGENC. This observation is consistent with prior findings on the QMSum dataset (Zhang et al., 2021b). One possible explanation for the lower performance of LED relative to SEGENC is that LED must adapt its parameters for a global attention mechanism that is absent from the backbone BART encoder model, whereas SEGENC relies solely on local self-attention that is aligned with the backbone model. This may be particularly relevant to QMSum given its relatively small size.

Practitioners may wish to consider the computational cost and efficiency of various hyperparameter settings. Computational complexity increases with both input length and attention window size (since attention grows quadratically in attention-window size). Complexity is also greater with the overlapping segment strategy compared to the disjoint segment strategy for the SEGENC model, due to the greater number of resulting segments that are passed through the encoder and decoder modules.

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.

Model	R-1	R-2	R-L
Baselines			
DYLE	*34.42*	*9.71*	*30.10*
SUMM^N	*34.03*	*9.28*	*29.48*
BART	31.87	9.08	27.50
BART-W	32.68	8.97	28.74
BART-W (Gold)	39.54	15.65	35.17
Two-stage			
DPR	32.28	9.73	28.34
RELREGTT	33.02	10.17	28.90
MARGE	31.99	8.97	27.93
RELREG	34.91	11.91	30.73
RELREG-W	**36.45**	**12.81**	**32.28**
End-to-end			
LED	34.18	10.32	29.95
SEGENC	37.05	13.03	32.62
SEGENC-W	**37.80**	**13.43**	**33.38**

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.

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-

Model	Flu.	Rel.	Comp.	Fact.
BART	**4.08**	3.68	3.22	3.31
RELREG-W	3.87	3.81	3.67	**3.70**
SEGENC-W	3.93	**3.87**	**3.81**	3.63

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

Model	R-1	R-2	R-L
Hi-MAP	*30.34*	*14.82*	*26.86*
BART	48.74	33.96	46.02
RELREG-W	54.06	38.51	51.07
SEGENC-W	**63.62**	**51.27**	**61.37**

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).

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: 1) *fluency*, measuring their grammatical quality, 2) *relevance*, assessing their relevance to the input query, 3) *completeness*, evaluating their comprehensiveness considering the input conversation and query, and 4) *factuality*, measuring their factual consistency with respect to the conversation. Scores were assigned on a Likert scale from 1 to 5 (best), where each example was evaluated by 3 judges with the final score averaged. Results are presented in Table 6.

We find that the RELREG-W and SEGENC-W models achieved comparable performance across all of the evaluated dimensions, with summaries generated by SEGENC-W rated as slightly more complete. The BART baseline was rated highest in the fluency dimension, however, it was substantially outperformed by both of the introduced models on completeness and factuality. One possible explanation for the slightly lower fluency scores for the RELREG-W and SEGENC-W models is that they are better able to retrieve content from the source, which itself may have low fluency 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.

6 Conclusion

In this work, we conducted an exploratory study of neural models for query-focused summarization. We studied two categories of models: two-stage and end-to-end, and presented two architectures, RELREG and SEGENC, both of which improve ROUGE performance over prior state of the art by a substantial margin. We also explored task-specific transfer learning, which further improved model performance. Besides model performance, we discussed issues of computational efficiency that practitioners may factor into their modeling choices. Finally, we conducted a human study suggesting that the summaries produced by the best-performing models are more factually correct and complete than a baseline model by a substantial margin. We hope that the analysis and modeling contributions of this paper will be a resource for future research on query-focused summarization.

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

Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, Rheeya Uppaal, Bradford Windsor, Eliot Brenner, Dominic Dotterrer, Rajarshi Das, and Andrew McCallum. 2021. Long document summarization in a low resource setting using pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 71–80, Online. Association for Computational Linguistics.

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.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2021. Summscreen: A dataset for abstractive screenplay summarization. *ArXiv preprint*, abs/2104.07091.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost.

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.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pretraining text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Cui and Le Hu. 2021. Sliding selector network with dynamic memory for extractive summarization of long documents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5881–5891, Online. Association for Computational Linguistics.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Elozino Egonmwan, Vittorio Castelli, and Md Arafat Sultan. 2019. Cross-task knowledge transfer for query-based text summarization. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 72–77, Hong Kong, China. Association for Computational Linguistics.

Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2021a. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 704–717, Online. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019a. 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 Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019b. 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 Kryscinski, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir R. Radev. 2021b. Summeval: Re-evaluating summarization evaluation. *Trans. Assoc. Comput. Linguistics*, 9:391–409.

Alexander R. Fabbri, Xiaojian Wu, Srini Iyer, Haoran Li, and Mona Diab. 2021c. Answersumm: A manually-curated dataset and pipeline for answer summarization.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryściński, Nazneen Fatema Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir R. Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization. *ArXiv preprint*, abs/2105.08209.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. Aquamuse: Automatically generating datasets for query-based multi-document summarization.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. 2020. Pytorch distributed: Experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12):3005–3018.

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.

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.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018a. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018b. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Ye Liu, Jian-Guo Zhang, Yao Wan, Congying Xia, Lifang He, and Philip S. Yu. 2021. Hetformer: Heterogeneous transformer with sparse attention for long-text extractive summarization.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Zhengyuan Liu and Nancy F. Chen. 2021. Dynamic sliding window for meeting summarization.

Potsawee Manakul and Mark J. F. Gales. 2021. Sparsity and sentence structure in encoder-decoder attention of summarization systems.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed H. Awadallah, and Dragomir Radev. 2021. Dyle: Dynamic latent extraction for abstractive long-input summarization.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-sequence rnns for text summarization. *ArXiv preprint*, abs/1602.06023.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116, Florence, Italy. 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. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham Barezi, and Pascale Fung. 2020. CAiRE-COVID: A question answering and query-focused multi-document summarization system for COVID-19 scholarly information management. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

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.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI*, volume 7, pages 2903–2908.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771.

Yujia Xie, Tianyi Zhou, Yi Mao, and Weizhu Chen. 2020. Conditional self-attention for query-based summarization.

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. 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.

Yumo Xu and Mirella Lapata. 2021b. Text summarization with latent queries. *ArXiv preprint*, abs/2106.00104.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed H. Awadallah, Dragomir Radev, and Rui Zhang. 2021a. Summ$^{\mathrm{N}}$: A multi-stage summarization framework for long input dialogues and documents.

Yusen Zhang, Ansong Ni, Tao Yu, Rui Zhang, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz, Ahmed Hassan Awadallah, and Dragomir Radev. 2021b. An exploratory study on long dialogue summarization: What works and what's next.

Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. SEAL: segment-wise extractive-abstractive long-form text summarization. *ArXiv preprint*, abs/2006.10213.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021a. Dialoglm: Pre-trained model for long dialogue understanding and summarization.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021b. 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.

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 RELREG fine-tuned from an Electra-large checkpoint (Clark et al., 2020). For a fair comparison with other metrics, we also fine-tuned RELREG 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 RELREGTT 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 RELREGTT 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 hyperparamters were found to be: `16384` maximum input size, and `1024` attention window size. LED models were trained for `10` epochs, with a batch size `1`, gradient accumulation set to `4` steps, and learning rate set to `0.000005`. The SEGENC 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 hyperparamters were found to be: `16384` maximum input size, and `512` attention window size. The SEGENC 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

Model	Lexical Overlap b/w Extractors and References															
	Top-1				Top-5				Top-15				All			
	R-1	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$
GOLD SPANS	15.00	3.80	11.10	60	**20.89**	**6.05**	**15.04**	218	**19.62**	**5.99**	**14.28**	386	**16.09**	**5.60**	**12.47**	660
LEAD	8.17	0.98	6.30	82	12.84	1.69	9.17	309	13.13	1.81	9.21	463	8.77	1.79	6.77	978
DPR	11.31	1.99	8.72	34	17.46	2.86	12.21	156	15.38	2.74	10.64	394	9.75	2.23	7.42	932
RELREGTT	23.67	3.34	15.66	82	16.13	3.35	11.18	413	9.65	2.58	7.31	930	9.16	2.52	6.99	994
MARGE	7.13	0.72	5.81	20	13.76	1.39	10.22	92	14.85	1.74	11.09	269	9.21	1.52	7.16	896
RELREG	**24.57**	**4.33**	**16.57**	88	17.52	4.11	12.21	418	10.56	3.04	8.06	884	9.62	2.87	7.47	989

Model	Lexical Overlap b/w Extractors and Queries																Span Overlap b/w Extractors and Golden Spans	
	Top-1				Top-5				Top-15				All				All	
	R-1	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$	*R-1*	*R-2*	*R-L*	$\bar{x}$	*Precision*	*Recall*
GOLD SPANS	11.01	2.75	9.90	60	7.30	1.58	6.24	218	4.73	1.10	4.07	386	3.53	0.93	3.05	660	0.75	1.00
LEAD	4.88	0.60	4.49	82	5.51	0.72	4.71	309	3.76	0.64	3.26	463	1.70	0.37	1.55	978	0.09	0.20
DPR	12.41	3.37	11.35	34	8.08	1.74	7.00	156	4.44	0.92	3.90	394	1.97	0.50	1.82	932	0.22	0.27
RELREGTT	9.63	1.58	8.26	82	3.49	0.83	3.09	413	1.81	0.50	1.65	930	1.66	0.46	1.53	994	0.07	0.24
MARGE	7.22	0.81	6.88	20.61	6.86	0.67	6.09	92	4.70	0.61	4.20	269	1.84	0.36	1.70	896	0.15	0.21
RELREG	12.38	3.00	10.61	88	4.32	1.18	3.77	418	2.09	0.61	1.89	884	1.80	0.54	1.65	989	0.11	0.28

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.

an MIT license[5], which aligns with our use for research purposes. Non-identifying names are used in place of real names.

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 use the cleaned, reproduced version of the C4 dataset (Raffel et al., 2020) from the Common Crawl made available by AI2[6]. 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[7], 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 he Conversation.

Figure 1: Instructions presented to annotators for the human studies