QuerySum: A Multi-Document Query-Focused Summarization Dataset Augmented with Similar Query Clusters

Yushan Liu[1], **Zili Wang**[2*], **Ruifeng Yuan**[3]
[1]Fudan University
[2]INF Technology (Shanghai) Co., Ltd.
[3]The Hong Kong Polytechnic University
yushanliu21@m.fudan.edu.cn, ziliwang.do@gmail.com, ruifeng.yuan@connect.polyu.hk

Abstract

Query-focused summarization (QFS) aims to summarize the source document(s) with regard to a specific aspect of information given in a query. It plays an important role in presenting users with a concise answer summary from a set of query-relevant documents retrieved by the information retrieval system. Nonetheless, the QFS research has long been hampered by the lack of adequate datasets in terms of both quality and quantity. In this paper, we introduce a large-scale multi-document query-focused summarization dataset, called **QuerySum**, which contains 27,041 data samples covering diverse topics and its quality is guaranteed through human verification. Unlike some previous QFS datasets constructed directly from the question answering datasets, 74% queries in our dataset are the challenging non-factoid What-, Why-, and How- questions. More importantly, we also provide a set of similar queries together with the corresponding summaries pairs for each query as the retrieved context, presenting a new feature of QuerySum. We aim to encourage research efforts in query intention understanding in the context of QFS. Leveraging QuerySum's depth, we propose a model for query-aware multi-document summarization and set a new QFS benchmark.

Introduction

Text summarization as the process of generating a shorter version of the source document(s) while preserving important context information is one of the most challenging NLP tasks. Sequence-to-sequence neural models have recently obtained significant performance improvements on text summarization for news articles. The existence of large-scale datasets is the key to the success of these models. In this work, we focus on an important variant of text summarization called query-aware summarization. Text documents are multi-faceted and users may only prefer a certain aspect of information. In this case, query-aware summarization requires the model to generate a summary focusing on part of source documents based on a given query (usually a question). This makes it impossible to get a natural reference summary directly from the title or headlines like the news summarization. Therefore, obtaining a high-quality larger-scale dataset is always a crucial problem for query-aware summarization.

Query-aware summarization has crucial application in augmenting information retrieval (IR) system. For example, suppose a user has a question, "why X is so famous", then an effective IR system can find a set of documents related to A across the web and then a query-aware summarization model can generate a concise summary from these documents as the respond to the question. Hence, under the setting for general web IR applications, an ideal query-aware summarization dataset would have the following two features. (1) It is a multi-document summarization dataset since the query-related information typically spreads across multiple documents. (2) The query refers to a certain aspect of information rather than a specific entity. Otherwise, it becomes simple question-answering. However, most existing QFS datasets do not comply with the above two requirements or are inadequate in scale. For example, Debatepedia (Nema et al. 2017a) and PubMedQA (Jin et al. 2019) are single-document summarization datasets in specific domains and Wikihow (Koupaee and Wang 2018) contains only how questions. AQUAMUSE is proposed in (Kulkarni et al. 2020) to extract QA pairs with long answers from a question-answering dataset for multi-document QFS, but most of its queries are factoid-like questions. Although DUC 2005-2007 (Dang 2006) dataset meets the two requirements, it contains only hundreds of samples and is not large enough to train a sequence-to-sequence neural model.

To alleviate the QFS dataset shortage problem, and push forward the research development in this field, we build a new large-scale query-focused multi-document summarization dataset called QuerySum. The dataset contains in total 27,041 data samples covering diverse topics. Each data sample includes a human-written query collected from Answers.com[1] and Google, up to 10 relevant Wikipedia pages as the source documents and a summary retrieved by Google. The data samples are manually checked to guarantee their quality. A large proportion of the queries in our dataset are non-factual questions such as What-, How- and Why- questions that require summarization. Unlike a factual question which targets at a specific entity answer, non-factual questions bring challenges in understanding the in-

*Corresponding Author
Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]https://www.answers.com/

tention of the query. To encourage the research efforts to explore query intention understanding for QFS, we provide an additional annotation called query cluster. In the QuerySum dataset, queries have two sources: seed queries from Answers.com and their similar queries extended by Google Search Engine. Hence, each query belongs to a query cluster composed of some similar queries and their corresponding summaries. We annotate the relationship between these queries as synonymous, related, and unrelated. These extended set of queries enable the model to acquire an enriched understanding of a target query and learn the intention behind it.

Finally, we proposed a model for query-aware multi-document summarization and compare it with the existing extractive and abstractive system as benchmarks for further studies. The contribution of this work is three-fold:

- We introduce a new large-scale query-aware multi-document summarization dataset focusing on non-factual queries. The extended query clusters together with the corresponding summaries in our dataset provide a chance to explore the intention behind the queries for QFS.
- We present the new challenges that brought by our dataset and discuss the different features of the dataset.
- We analyze the performance of a set of summarization systems, including a model proposed by ourselves, on the dataset as benchmarks for further studies.

Our dataset is available on github[2].

Existing Datasets

There are several existing datasets, all of which are related to query-aware text summarization. We will briefly describe the composition and characteristics of these datasets.

DUC 2005-2007: The Document Understanding Conference (DUC) dataset is a classic text summarization dataset and is first proposed in 2004. It is a series of traditional text summarization datasets. Duc 2005-2007 contains data designed for multi-document query-aware summarization, which consists of 500 news articles and corresponding summaries. One main advantage of this dataset is that the summaries are written by humans. However, the limited data is not enough to train sequence-to-sequence models with a large number of parameters. This makes the DUC dataset can only be used as a testing dataset in query-aware summarization for neural models.

Debatepedia: This dataset was crawled from Debatepedia, a website containing the records of debate topics and corresponding discussions. It was first used for single-document query-aware summarization by (Nema et al. 2017b). Due to the limitation its debate nature, the average length of the source document, summary, and query is relatively short. Moreover, we find there is a serious data leakage problem in the Debate dataset, which is further explained in the Experiment Section.

PubMedQA: This dataset mainly consist of medical-related question and articles. It is originally proposed as a question-answering dataset, and the answers contain 1 to 3 sentences as explanations for yes/no medical questions. The dataset is adopted by (Deng, Zhang, and Lam 2020) for single-document query-focused summarization. Although the dataset contains more than 160,000 data samples, the medical domain and the yes/no question type limit its wide application in summarization.

[2]https://github.com/613lys/QuerySum

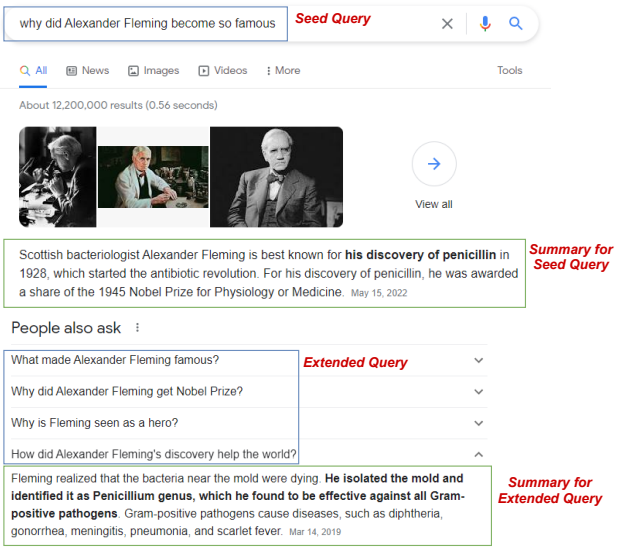

Figure 1: Example of extracting query-summary pairs from Google search engine.

AQUAMUSE: The dataset is automatically constructed for multi-document query-focused summarization from the Google Natural Questions dataset by choosing the questions with long answers. As it is essentially a question-answering dataset, many queries are factoid-like questions that target at specific entries, such as locations. By contrast, we tend to build up the dataset with What-, How- and Why-questions for which abstractive answers are necessary.

WikiHow: This dataset is built up based on Wikihow, a famous how-to instruction website. The queries in the dataset are mainly how questions athe How-questions and the source documents are the multi-step instructions. For each step, there is a concise human-written description, which can be considered as the summaries. Although the dataset provides corresponding queries for the summaries, it is initially designed for generic summarization. Hence, even without queries, a summarization model can still achieve a reasonably good performance. This makes WikiHow not suitable for query-aware summarization. Further details are displayed in the Experiment Section.

QuerySum Dataset

Dataset Construction

The **QuerySum** dataset is constructed by utilizing multiple online resources, including Answers.com, Google Search Engine and Wikipedia. There are three parts of this dataset: queries, summaries and source documents and we introduce the construction of the dataset with the following steps.

Statistic	QuerySum	AQUAMUSE	DUC	Debate	WikiHow	PubMedQA
Number of QA Pairs	27041	5519	150	12695	23843	273500
Avg. doc number	5.5	6.1	23.5	1	1	1
Avg. doc length	423.8	1597	343.8	66.4	579.8	238.9
Avg. summary length	37.2	106	249	11.2	62.1	43.2

Table 1: The comparison of the existing datasets, including the number of question and answer pairs, the average document length, and the average summary length.

Query-Summary Pair Extraction We first extract queries from Answers.com, an Internet-based knowledge exchange website formerly known as WikiAnswers. The website has a large number of user-generated queries covering diverse topics ranging from science to health and law etc. As shown in Figure 1, We take the queries from Answers.com as the seeds and retrieve top-N related queries through Google to extend the number of available queries. The seed query and the extended queries together naturally form a cluster of similar queries. The powerful Google search engine also provides the answers to all these seed and expanded queries. The answers are usually the most important paragraphs extracted from the online documents by Google. We regard these answers as the summaries in response to the queries.

Source Document Extraction Considering Wikipedia is one of the largest knowledge base on the Internet, we believe that the supporting documents for the query-summary pairs can be found from it. We adopt Yake (Campos et al. 2020), a unsupervised key entity extractor, to generate salient entities for each query-summary pair. We conduct a fuzzy search in the Wikipedia knowledge base and find the most relevant Wikipedia page for each salient entity. Here, we select the first paragraph of a Wikipedia page as the source document. For a query-summary pair, the obtained documents about the 10 most important entities are regarded as its information sources. The source document will be removed if the summary appears directly in it. For the documents related to the same query, we calculate the F1 values of ROUGE-1, ROUGE-2, and ROUGE-L for the two most similar source documents. They are 0.348,0.163 and 0.320, which indicates the overlap between the source documents is relatively low.

Data Filtering To ensure the quality of the dataset, we remove a data sample if the recall of Rouge-1 score between its summary and source documents is lower than a threshold (we adopt 0.7 in this work). Meanwhile, the data samples with certain types of questions, such as the factual question asking for time or number, how long or how many that are not suitable for QFS are also removed. In this stage, we reduce 1,490,000 raw data samples to 46,900 potential data samples.

Manual Check and Annotation We employ 7 helpers who are fluent English speakers in this step. They manually check potential data samples by answering four questions. (1) Is the query appropriate for QFS? Inappropriate queries are revised or removed. (2) Whether the source documents are related to the query? Unrelated documents are removed. (3) Whether the summary can answer the query? Bad data samples are removed. (4) Whether the summary can be summarized from the source documents? Bad data samples are removed. Finally, 52% of potential data samples are removed through the manual check and only high-quality data samples are preserved. For the data samples preserved, only 965/27041 queries share the same set of source documents with other queries. For the queries related to totally same source documents, the maximum F1 scores of ROUGE-1 between their answers are 0.976. This shows a small proportion of queries are synonyms and share the same answer.

Meanwhile, we also ask helpers to manually inspect the similar query clusters. It is noted that not all of the queries in the same cluster are equally relevant. Distinguishing the relationship between the queries are crucial for the further research. We ask the helpers to label the relationships between the seed query and the related queries in a query cluster for three cases, (1) synonyms where the two queries have almost the same meaning; (2) related where the information from one query is helpful for understanding the other one; and (3) unrelated where the information from one query contributes nothing to the understanding of the other one. In total, 10,700 query clusters are annotated. The proportion between synonyms, related and unrelated is 29:45:26.

Characteristics of QuerySum

The statistics of **QuerySum** is shown in Table 1. In order to better understand the advantages of our proposed dataset, we analyze it from the following several aspects.

The Scale of the Dataset QuerySum contains 27,041 query-summary pairs. To the best of our knowledge, it is the largest multi-document QFS dataset. Its scale is large enough to train sequence-to-sequence neural models. Following the previous work, We randomly extract 15% of the data samples as the validation set and another 15% as the test set.

The length of the source documents and summaries QuerySum is a query-oriented multi-document summarization dataset containing 8 to 10 documents for each data point. Its documents and summaries have a proper length, which is similar to the classic multi-document query-aware summarization dataset DUC. The proportion between the length of summaries and source documents is much smaller than that of previous Debate and PubMedQA datasets. This indicates that only a small part of contents from the source documents are effective for the summary generation. It leaves a challenge for how to accurately capture the query-relevant information and compress it into the summary.

Types of Queries We present the distribution of the 6 query types covered in QuerySum in Figure 2. It is not a

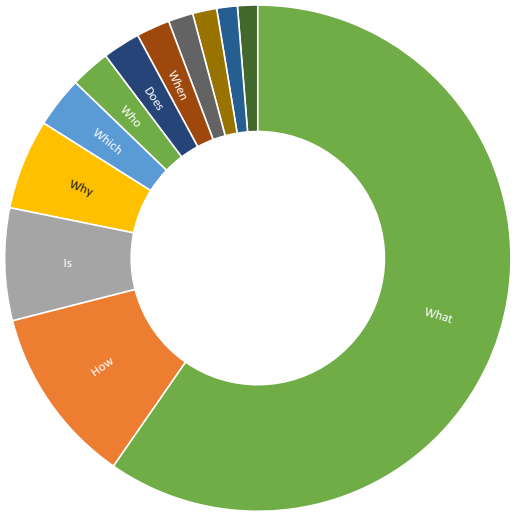

Figure 2: Distribution of question types for the QuerySum dataset

surprise that What-questions are the majority. It is also the most common non-factoid questions on the Web. The How-questions and Why-questions are considered the most appropriate queries for QFS, taking the second and fourth places. For the Yes/No questions starting with "is", the answers can be regarded as the summaries of the evidences supporting the judgments. These non-factoid questions usually target a wide range of information, which bring in high demand for summarizing important information.

Potential Applications of Query Clusters The way to utilize the query clusters to assist QFS remains unexplored. On the one hand, the extended query may help the summarization model to learn the hidden topics behind a query and capture more effective information in summarization during training. On the other hand, the researchers can use the similar query clusters as retrieved context obtained from the Web and explore more in-depth query understanding mechanisms to enhance QFS.

Method

Problem Formulation

Given a query $Q = (w_1^q, w_2^q, \cdots, w_o^q)$ of o query words and a set of documents $R = \{R_1, R_2, \cdots, R_m\}$, in which each document consists of a sequence of words $R_i = (w_1^{r_i}, w_2^{r_i}, \cdots, w_n^{r_i})$, the query-aware generation system aims to generate a compact query-aware summarization $T = (w_1^t, w_2^t, \cdots, w_p^t)$. The multiple documents R, leveraged by the query Q, are consumed as source information to produce a summary that is not only helpful but also relevant to a query.

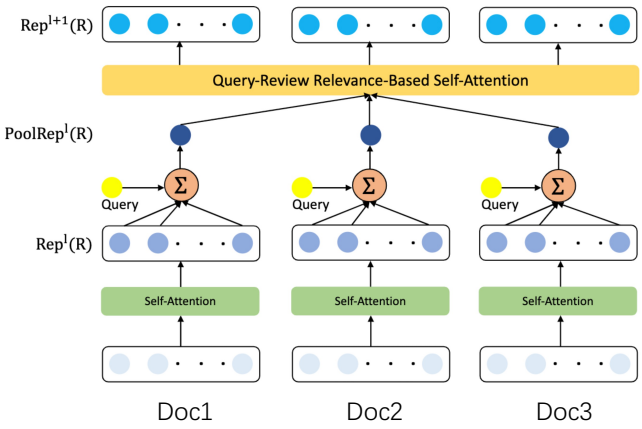

Figure 3: The architecture of our model. The contextualized representations $Rep^l(R)$ of each document are obtained by the self-attention mechanism. The query semantic-based pooled representations $PoolRep^l(R)$ are then computed, followed by a query-document relevance-based self-attention mechanism among multiple documents to refine the document term representations $Rep^{l+1}(R)$.

Model Architecture

We propose a qu**E**ry-**A**ware document **G**en**ER**ation model that distill relevant information from multiple user documents, noted as our model for short. Generally, our model consists of two basic components, i.e., encoder and decoder, as in many seq2seq language generation models. The framework of our model is demonstrated in Figure 3. In terms of the encoder, the first 6 layers are borrowed directly from the vanilla transformer, but we augment 2 more self-attention layers to take query-awareness into account. In terms of the decoder, we adopt the identical implementation of the vanilla transformer.

Document Representation

In this section, we first introduce the multi-document interaction framework. For each document R_i, several transformer encoders are stacked to compose the sequence internally, results in document term contextualized representation $\mathbf{Rep}(R_i) = (\mathbf{c}_1^{r_i}, \mathbf{c}_2^{r_i}, \cdots, \mathbf{c}_n^{r_i})$, which is further utilized to produce query-aware document representation in the next section. A natural way to obtain the global representation of multiple documents is to concatenate different document contextualized representations, namely $concat[\mathbf{Rep}(R_1), \mathbf{Rep}(R_2), \cdots, \mathbf{Rep}(R_m)]$[3], which is also studied as a baseline in the experiments.

Compared with single document models, one advantage of using the multiple documents is that it encourages the model to fully utilize the information in other parallel documents, making document information share possible. To im-

[3]We have also tried to add position embedding at document level but found degraded performance, since there is not relative position information among documents compared with word positions within a document.

plement that, the document pooled representations are computed firstly and then shared across documents. To obtain a document representation $\mathbf{PoolRep}^l(R_i)$, one may average the term representations $\mathbf{Rep}(R_i)$ within the document or add a linear layer to learn the weight of each term then weight average on all terms, as in (Liu and Lapata 2019).

Therefore, the m independent document representations $\mathbf{PoolRep}^l(R) = [\mathbf{PoolRep}^l(R_1),$ $\mathbf{PoolRep}^l(R_2), \cdots, \mathbf{PoolRep}^l(R_m)] \in \mathcal{R}^{m \times d}$ are fed to the inter-document attention layer, as in, to obtain the inter-document representations. where $\mathbf{W_Q^l}, \mathbf{W_K^l}, \mathbf{W_V^l} \in \mathcal{R}^{d \times d}$ are linear transformation weights. These representations are further utilized by the two following sub-layers,

$$\text{HerRep}^l(R_i) = FFN(\mathbf{Rep}^l(R_i) + \mathbf{InterRep}^l(R_i))$$
$$\mathbf{Rep}^{l+1}(R_i) = LayerNorm(\mathbf{Rep}^l(R_i) + \mathbf{HerRep}^l(R_i))$$

where FFN is a two-layer linear transformation with Relu activation in between. $\mathbf{HerRep}^l(R_i)$ allows cross-document information to broadcast to each document term. The above layers can also be stacked as in the transformer encoder. The concatenated representation of the final encoder output is served as key and value in encoder-decoder attention during decoding. We tailor our query-aware summary generation model with two mechanisms: semantic-based document pooling and relevance-based inter-document attention.

Semantic-based Pooling

When summing up the document term representations, there is no explicit mechanism to discriminate term importance. Though one may learn the term weights from term embedding, as in (Liu and Lapata 2019), there could also be rare supervise signal on term weights in the embedding space. To obtain the query-aware document representation, the similarity of document terms $sim(R_i, Q) \in \mathcal{R}^m$ is obtained, where the k-th element is computed as follows:

$$sim(w_k^{r_i}, Q) = \sum_j^o cosine(\mathbf{E}(w_j^q), \mathbf{E}(w_k^{r_i})) \quad (1)$$

The similarity scores are then normalized at document level and served as weights when summing up the term representations, as shown in Equation 2.

$$\mathbf{PoolRep}(R_i) = softmax(sim(R_i, Q))\mathbf{Rep}(R_i) \quad (2)$$

After pooled by the user query information, $\mathbf{PoolRep}(R_i)$ is fed into the inter-document attention layer as in Section Document Representation.

Relevance-based Attention

As the inter-document attention layer enables each document to attend to other documents, the attention score is obtained in the semantic space without consideration of query-document relevance. To mitigate the gap of query-agnostic attention among the documents, we introduce the query-document relevance to inform the model of how relevant is a document to the user query. The kernel-based relevance model, as introduced in previous section, is utilized to produce the relevance score of the documents $\mathbf{F}(R, Q) = [f(R_1, Q), f(R_2, Q), \cdots, f(R_m, Q)] \in \mathcal{R}^m$, which modifies the dot product attention layer into where $\oplus$ denotes matrix concatenate, $\mathbf{W} \in \mathcal{R}^{2 \times 1}$ is a learnable weight that learns to leverage the information between the semantic relationship among documents and relevance degree to the user query.

In summary, the semantic-based pooling method produces condensed document representation from term level, as a complementary, the relevance-based attention further informs the model how relevant a document is to the user query and how much attention to pay for the document from document level.

Recently, generative pre-trained models have shown excellent performance in natural language generation (NLG) including text summarization. Therefore, we apply the pre-trained model on the query-aware summarization task. PEGASUS(Zhang et al. 2020) has achieved the SOTA performance on text summarization.

Experiment

We evaluate the performance of QuerySum dataset using the existing summarization models for a better understanding of the dataset. For all experiment, we concatenate the query with source documents and truncate the input to 1024 tokens.

Implementation Details

We adopt PEGASUS-LARGE or PEGASUS-BASE as the base model for the proposed model. ALL the hyper-parameters are adjusted on the development set. For optimization, the batch size is set to 16. We use dropout with the probability of 0.1 and label smoothing (Szegedy et al. 2015) with smoothing factor 0.1. The optimizer is Adam (Kingma and Ba 2014) with a learning rate of 0.001. In addition, we apply warm-up with the first 10% steps, and learning rate decay of 0.95.

Evaluated Systems

Extractive Baselines: We display two basic extractive baselines: Random and Ext-oracle. As for Random, we randomly select 4 sentences from the source documents for each query as a summary. Besides, we implement the extractive oracle summaries as Ext-oracle, where a greedy algorithm is used to select a group of sentences with the highest Rouge score. This can be regarded as the upper bound of an extractive summarization system on the dataset.

Transformer: A baseline model for self-attention based seq2seq generation (Vaswani et al. 2017), including 6 encoder blocks and 6 decoder blocks. To take the query into account, we select two common ways to combine the query and document, including **CONCAT** and **ADD**. **CONCAT** means that the query embeddings are concatenated with the document embedding, and **ADD** means the query embedding adds the document embedding. Then, the processed query and document are fed to the model to generate query-aware summarization.

Model	Rouge-1	Rouge-2	Rouge-L
Random	12.26	5.67	11.34
Ext-oracle	25.20	7.67	24.12
Transformer(CONCAT)	26.18	8.15	25.07
Transformer(ADD)	26.89	8.53	25.68
BERTSUM	28.14	8.93	26.19
CTRLSUM	28.37	8.86	26.32
QS-BART	32.08	10.43	27.19
QS-PEGASUS	32.11	10.43	27.18
Our model	36.73	12.94	31.12

Table 2: The result of comparison systems and our model in terms of Rouge-F1(Lin 2004) score in QuerySum dataset. There are two categories of previous work: (1) Adopting sequence-to-sequence architecture and training from scratch. (2) Fine-tuning the pre-trained model in query-aware summarization task.

Model	Rouge-1	Rouge-2	Rouge-L
Full-model	36.73	12.94	31.12
-sentence level	35.44	12.46	30.13
-word level	36.25	12.65	30.74

Table 3: The results of different parts of our model in terms of Rouge-F1(Lin 2004) score in QuerySum dataset. The "-sentence-level" means the model removes part of the sentence level, and the "-word level" means our model removes part of the word level.

BERTSUM: It is the first model to integrate a pre-trained model(model) into text summarization and can be used for both extractive and abstractive summarization(Liu 2019). BERTSUM focuses on sentence-level information by adding [CLS] tokens before every sentence as the sentence representation. In the experiment, we adopt the abstractive version of BERTSUM.

CTRLSum: It is a novel framework for controllable summarization(He et al. 2020). The approach enables users to control multiple aspects of generated summaries by interacting with the summarization system through textual input in the form of a set of keywords or descriptive prompts. Using a single unified model, CTRLsum is able to achieve a broad scope of summary manipulation at inference time without requiring additional human annotations or pre-defining a set of control aspects during training.

Query-aware pretrained model: We import two state-of-the-art transformer-based pre-trained models, BART(Lewis et al. 2019) and PEGASUS(Zhang et al. 2020) , for comparison. BART has been wildly used for many text generation tasks and PEGASUS is specifically designed for abstractive summarization. In order to capture the relevance between the query and the source, we concatenate the query and the source documents as the input of the model and split them with the [CLS] token.

Results

Automatic Results We adopt ROUGE score as the evaluation metric. As shown in Table 2, Our model significantly

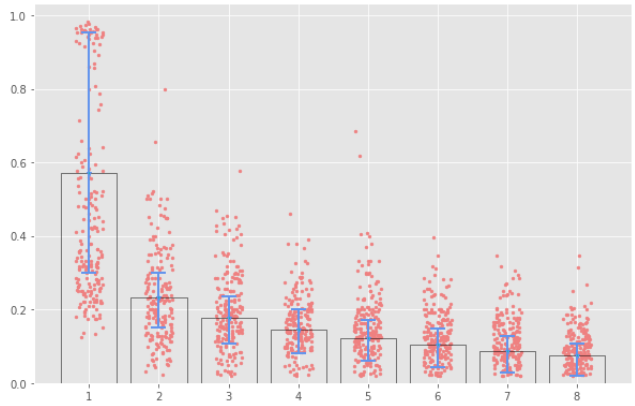

Figure 4: Distribution of effective information across multiple source documents

outperforms the models. Compared with the models utilizing pre-training, ours improves the ROUGE-1 and ROUGE-L scores by a large margin. We use the transformer as the basic model, and we can find that the combination of Add's query and document is more effective than the combination of concat. In addition, just using a document as input, the rouge-2 score of this model is higher than that of the transformer model with query and document input, which reflects the effectiveness of pre-training. The results of PEGASUS and QS-PEGASUS indicate that query and document need to be combined, which reflects the necessity of query. Finally, our model far outperforms the QS-PEGASUS model, which demonstrates the effectiveness of our model. Next, we will study the various parts of the model, which we analyze in the way of an ablation study.

Ablation Studies

Table 3 shows the results of the existing comparison system. To further analyze our results, we conducted further analysis on multiple parts of the model. From the results, we can find that after removing the sentence-level model, the effect of the word-level model only has some improvement compared with the "QS-PEGASUS" model, indicating that the word-level supervision information can obtain the text information related to the query. Besides, we find that the model with the

	QuerySum	Debate_standard	Debate_reform	WikiHow	AQUAMUSE
w/o query	30.50 / 8.87	57.98 / 43.62	53.09 / 16.10	43.06 / 19.71	28.34 / 13.12
w query	32.11 / 10.43	59.02 / 44.59	53.12 / 16.09	43.21 / 19.75	30.34 / 14.82
improvement	1.61 / 1.56	1.04 / 0.97	0.03 / -0.01	0.15 / 0.04	2.00 / 1.70

Table 4: The result of different dataset in terms of ROUGE-1 and ROUGE-2 score with and without query. The model we used for QuerySum and Debatepedia is BART, and we use the result of Hi-MAP (Fabbri et al. 2019) on AQUAMUSE from its own paper.

sentence level has a large improvement, which shows the effectiveness of our query-aware attention, which can effectively improve the performance and generate query-related summaries.

Analysis

Whether the query is correlated with the summary? For a query-aware summarization dataset, the summaries are required to be closely tied to the given query. In Table 4, taking Bart as the summarization model, we display the result of taking queries as the input (concatenate query and source document) and the result without queries in different datasets. In this case, a greater improvement shows the closer connection between the queries and summaries in one dataset. As stated previous, there is a data leakage problem in the debate dataset and more than 70% percent of summaries in testing data appear or have similar ones (the difference is less than 2 tokens) in the training set. We use both the standard division of the dataset (debate_standard) and the reformed version with no data leakage problem (debate_reform).

Whether the summary utilizes effective information from all source document? In multi-document summarization, the necessary information that forms the summary is expected to be scattered into multiple source documents. In order to analyze the distribution of crucial information across the documents, we calculate the recall between the summary and each source document in a data point and rank them from high to low. As shown in Figure 4, we show the average recall of top-8 source documents.

Conclusion

We present QuerySum, a new large-scale summarization dataset consisting of diverse articles. The large scale, proper length of source documents and summaries, abundance of non-factoid queries of the dataset and potential applications of query clusters discussed in the paper can create new challenges to the summarization systems. We evaluate the efficacy of several summarization systems, incorporating our own proposed model, using the dataset as a reference point for future research. We hope that the new dataset can attract researchers' attention as a choice to evaluate their systems.

Limitations

Although we have adopted human annotation to ensure the quality of the dataset, most annotation tend to remove bad data point rather than generate good ones. In this case, the quality of dataset is not comparable with the dataset containing human-written summaries such as CNN/DailyMail (Nallapati et al. 2016).

References

Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; and Jatowt, A. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509: 257–289.

Dang, H. T. 2006. DUC 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, 48–55.

Deng, Y.; Zhang, W.; and Lam, W. 2020. Multi-hop inference for question-driven summarization. *arXiv preprint arXiv:2010.03738*.

Fabbri, A. R.; Li, I.; She, T.; Li, S.; and Radev, D. R. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*.

He, J.; Kryściński, W.; McCann, B.; Rajani, N.; and Xiong, C. 2020. Ctrlsum: Towards generic controllable text summarization. *arXiv preprint arXiv:2012.04281*.

Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W. W.; and Lu, X. 2019. PubMedQA: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *Computer Science*.

Koupaee, M.; and Wang, W. Y. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Kulkarni, S.; Chammas, S.; Zhu, W.; Sha, F.; and Ie, E. 2020. Aquamuse: Automatically generating datasets for query-based multi-document summarization. *arXiv preprint arXiv:2010.12694*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Liu, Y. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Liu, Y.; and Lapata, M. 2019. Hierarchical Transformers for Multi-Document Summarization. In *Proceedings of the*

57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, 5070–5081.

Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B.; et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Nema, P.; Khapra, M.; Laha, A.; and Ravindran, B. 2017a. Diversity driven attention model for query-based abstractive summarization. *arXiv preprint arXiv:1704.08300*.

Nema, P.; Khapra, M. M.; Laha, A.; and Ravindran, B. 2017b. Diversity driven attention model for query-based abstractive summarization. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1063–1072. Association for Computational Linguistics.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2015. Rethinking the Inception Architecture for Computer Vision. *CoRR*, abs/1512.00567.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, 11328–11339. PMLR.