Computer Speech & Language 66 (2021) 101166

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

QBSUM: A large-scale query-based document summarization dataset from real-world applications

Mingjun Zhao[1,a], Shengli Yan[1,b], Bang Liu[c,*], Xinwang Zhong[b], Qian Hao[b], Haolan Chen[b], Di Niu[a], Bowei Long[b], Weidong Guo[b]

[a] University of Alberta, 116 St. and 85 Ave., Edmonton, AB T6G 2R3, Canada
[b] Platform and Content Group, Tencent, 10000 Shennan Ave, Shenzhen 518057, China
[c] University of Montreal, Apartment 1209, 4998 Boul De Maisonneuve O, Westmount QC, H3Z 1N2, Canada

ARTICLE INFO

Article History:
Received 15 May 2020
Revised 3 September 2020
Accepted 24 October 2020
Available online 28 October 2020

Keywords:
Query-based summarization
Natural language generation
Information retrieval

ABSTRACT

Query-based document summarization aims to extract or generate a summary of a document which directly answers or is relevant to the search query. It is an important technique that can be beneficial to a variety of applications such as search engines, document-level machine reading comprehension, and chatbots. Currently, datasets designed for query-based summarization are short in numbers and existing datasets are also limited in both scale and quality. Moreover, to the best of our knowledge, there is no publicly available dataset for Chinese query-based document summarization. In this paper, we present *QBSUM*, a high-quality large-scale dataset consisting of 49,000+ data samples for the task of Chinese query-based document summarization. We also propose multiple unsupervised and supervised solutions to the task and demonstrate their high-speed inference and superior performance via both offline experiments and online A/B tests. The QBSUM dataset is released in order to facilitate future advancement of this research field.

1. Introduction

Query-based document summarization aims to produce a compact and fluent summary of a given document which answers or is relevant to the search query that leads to the document. Extracting or generating query-based document summarization is a critical task for search engines (Wang et al., 2007; Sun et al., 2005), news systems (Liu et al., 2017), and machine reading comprehension (He et al., 2017; Choi et al., 2017; Wang et al., 2019). Summarizing a web document to answer user queries helps users to quickly grasp the gist of the document and identify whether a retrieved webpage is relevant, thus improving the search efficiency. In practical applications, query-based document summarization may also serve as an important upstream task of machine reading comprehension, which aims to generate an answer to a question based on a passage. The summary can be considered as evidence or supporting text from which the exact answer may further be found.

Existing research on text summarization can be categorized into generic text summarization and query-based text summarization. Generic text summarization produces a concise summary of a document, conveying the general idea of the document (Carbonell and Goldstein, 1998; McDonald, 2007; Gillick and Favre, 2009; Erkan and Radev, 2004). A number of diverse datasets have been developed, including Gigaword (Graff et al., 2003), New York Times Corpus (Sandhaus, 2008), CNN / Daily Mail (Hermann et al., 2015), and NEWSROOM (Grusky et al., 2018), etc. In contrast, query-based document summarization must

*Corresponding author.
E-mail address: bang.liu@umontreal.ca (B. Liu).
[1] These authors contributed equally.

https://doi.org/10.1016/j.csl.2020.101166
0885-2308/

produce a query-biased result answering or explaining the search query while still being relevant to the document content, which is a more challenging task. Datasets created to date for this task are often of a tiny scale such as DUC 2005 (Dang, 2005), or built upon human-crafted rules using web-crawled information (Nema et al., 2017). To the best of our knowledge, there is not yet any publicly available dataset developed for Chinese query-based document summarization.

To address the demand for a large and high-quality query-based document summarization dataset and to facilitate the advancement of related research, in this paper, we present the **Q**uery-**B**ased document **SUM**marization (**QBSUM**) dataset, which consists of ⟨ *query, document, summarization* ⟩ tuples, where the summarization to each query-document pair is a collection of text pieces extracted from the document (with an example shown in Fig. 1) labeled by five professional product managers and one software engineer in Tencent. QBSUM contains more than 49,000 data samples on over 49,000 news articles, where queries and documents are extracted based on queries and search logs of real-world users in QQ browser that serves over 200 million daily active users all around the world.

Beside the large-scale and high-quality data collected from real-world search queries and logs, the QBSUM dataset is also created with considerations to various quality measurements, including relevance, informativeness, richness, and readability. Firstly, the selected summary must be relevant to the query as well as the major focus of the corresponding document. If a query can directly be translated to a question, the summary shall contain the answer to the question if applicable, or provide information that is helpful for answering the question. Furthermore, the summary shall convey rich and non-redundant information related to the query. Lastly, while being concise, the natural language summary must also be fluent for readability purposes—simple concatenation of several text pieces may not always serve the purpose.

To tackle the task of query-based document summarization, we design and implement three solutions: (i) an unsupervised ranking model based on relevance defined in Peyrard (2019); (ii) an unsupervised ranking model based on a range of features; and (iii) a query-based summarization model based on BERT (Devlin et al., 2018). We evaluated the performance and inference efficiency of different models on the QBSUM dataset, and compared with multiple existing query-based summarization baseline methods. Our best model achieves a BLEU-4 score of 57.4% and ROUGE-L score of 73.6%, which significantly outperforms the baseline methods.

Based on the QBSUM dataset, we trained and deployed our query-based document summarization solution into QQ browser and Mobile QQ, two real-world applications involving more than 200 million daily active users all around the globe. Our solution currently serves as the core summarization system in these commercial applications for extracting and presenting concise and informative summaries based on user queries, to improve the search effectiveness and efficiency in these applications. Furthermore, we conducted large-scale online A/B tests on more than 10 million real-world users in QQ browser mobile app. The experimental results suggest that our model is able to improve the search results with web document summaries conforming to user queries and attention. The Click-Through Rate (CTR) increased by 2.25% after our system was incorporated into the search engine. To facilitate advancements in related research and tasks, we open source the QBSUM dataset[2] and will later release our code for experiments as well.

2. Dataset collection and analysis

In this section, we introduce the procedures of constructing the QBSUM dataset, and analyze its specific characteristics. Fig. 1 gives an data example presented in the dataset.

2.1. Data collection

Query and document curation. We collect the queries and documents in our dataset from Tencent QQ Search (http://post.mp.qq.com), one of the most popular Chinese news websites.

We retrieve a large volume of queries and its top clicked articles posted between June 2019 and September 2019. For each article, we perform text segmentation with punctuations [, ? ! ◦], and filter out short articles with less than 15 text segments, as well as long articles with more than 10 paragraphs. The retrieved samples covers a wide range of topics such as recent events, entertainment, economics, etc. After that, we tokenize each query and article, and filter out samples in which the article and the query do not possess token overlaps. Table 1provides the key data statistics and a comparison with two existing query-based summarization dataset. The QUSUM dataset has a vocabulary size of 103,005. The average length (number of words) of the queries, articles, and summaries are 3.82, 378.57 and 53.08, respectively. Comparing to DUC2005 and DUC2006, QBSUM holds a scale two orders magnitude larger and covers a far wider range of subjects, which fills the pressing need of a large-scale dataset in the research field of query-based summarization.

Summary annotation. Next, we annotate the summary of each collected query-document pair. The document is denoted as Dconsisting of $m$text pieces $\mathcal{S}=\{S_1, S_2, \ldots, S_m\}$. And a query $Q$is a sequence of $n$query words $q_1 q_2 \cdots q_n$. The annotators are asked to select at most $k=10$text pieces $S_i \in \mathcal{S}$that are most relevant to $Q$and convey valuable information to users. The summary $Y$is constructed by concatenating the selected text pieces in the order they present in the document. In terms of quality control of the annotated summary, we refer to the following four criteria:

[2] https://www.dropbox.com/sh/t2cp7ml1kb8ako0/AADmS2RMfJvLbukyQbb08CGGa?dl=0

- **Relevance**. The summary must be semantically relevant to the user query. They may contain the keywords of the query.
- **Informativeness**. The summary shall include the answer to the query and provide valuable information or explanations if such information is available in the document. Supposedly, it should reduce user's uncertainty about the query.
- **Richness**. The summary shall contain diverse and non-redundant information that is relevant to the query.
- **Readability**. The summary shall be consistent for a good readability to real-world users.

As an aid to annotators, for each text piece, we estimate its importance by TextRank (Mihalcea and Tarau, 2004) and indicate whether it contains at least one content word in the query. The detailed procedure of annotation is introduced as follows:

1. Check whether the document $D$is relevant to the query Q. Discard this sample if not.
2. Check the text pieces that share the same content words with Q,and read over the context sentences they belong to. Find out the most relevant and important text pieces with respect to Q.
3. If the selected text pieces in the document are not consistent or fluent, decide whether to expand them to include their neighboring text pieces. A neighboring chunk is selected if it helps to improve the readability of summary $Y$and contains relevant and non-redundant information to Q. Each selected text piece $S_i$is expanded with no more than 3 neighboring chunks. The overall number of text pieces in Yshall not exceed k,and the maximum number of words is set to 70.

Annotation Cost. Each worker needs to extract text segments related to the query from documents composed of 49 segments in average, according to their relevance and informativeness. At the same time, the adjacent text pieces need to be carefully selected, such that the requirements of richness and readability can be fulfilled. Due to the high complexity of the annotation process, the time cost of data annotation is relatively expensive with the average cost for annotating a single piece of data being 115 seconds.

Quality control. Our workers comprise 5 professional product managers and 1 software engineer in Tencent. Each worker needs to take an annotation examination to ensure their intact understanding about how to annotate and the ability of extracting qualified summaries according to the multi-aspect criteria. Three rounds of summary annotation are conducted. First, five workers will annotate different groups of the dataset individually. Second, each worker will review 5%of data sampled from the groups annotated by other workers. Third, an expert reviewer will review all the data examples created by workers. This loop is repeated to make sure the accuracy (judged by the expert reviewer) is above 95%.

2.2. Data analysis

Statistical analysis. Table 1 describes the statistical information of QBSUM and compares it with the DUC2005 and DUC2006 datasets. As the DUC datasets are not publicly available, we can only collect the shown information from paper (Dang, 2005). We can see that QBSUM is two orders of magnitude larger than DUC2005 and DUC2006. The average number of characters in a summary in QBSUM dataset is 41 which is notably lower than the average 250 characters in DUC2005 and DUC2006. The reason of this difference is traced back to the characteristics of Chinese language where a Chinese word is usually made up of 1 to 4 characters which is far less than English words.

Query type distribution. The DUC 2005 and 2006 tasks are question-focused summarization tasks which concentrate on summarizing and integrating information extracted from multiple documents to answer a question. In contrast, our QBSUM dataset contains a diversity of queries produced by real-world users, which can be generally categorized into 5 types:

- **Hot topics/events**. This type of queries is focused on recent hot topics or events. The corresponding summaries are mainly descriptions about the event development, location, related entities, time and so forth.
- **Questions**. This category covers user questions such as "what universities are recommended for a GMAT score of 570". The summaries are the answers or evidence sentences to the questions.
- **Exact queries**. Such queries are about some specific concepts or entities, such as "country houses under 100 square meters in Shenzhen". The summaries contain useful information that are relevant to the queries.
- **Fuzzy queries**. This type of queries includes fuzzy or subjective concepts or questions, such as "top ten fuel-efficient cars". The corresponding summaries provide information of relevant entities or the explicit answers to the queries.
- **Unclear/incomplete queries**. For this type, users may be not clear about how to express their intention. Therefore, the summaries cannot give specific answers to the queries.

We sample 500 queries from the QBSUM dataset and manually check their query types where the data distribution is shown in Fig. 2. The largest partition of the query type in QBSUM belong to exact queries and questions while the percentage of topics/events are relatively small, due to the limited occurrences of hot topics/events in our daily lives. However, the Click-Through Rate (CTR) of hot topics/events are actually quite high.

Topic distribution. The QBSUM dataset covers queries and documents in a wide range of topics. Fig. 3 also presents the distribution of document topics in which we can see that a large portion of documents are discussing topics about entertainments, games, fashion or anime.

3. Methods

We developed three models for query-based document summarization including two fast unsupervised methods and a high-performance supervised model based on pre-trained BERT (Devlin et al., 2018).

3.1. Relevance-based summarization

Intuitively, in query-based summarization, a summary is profitable for a user, if it yields effective knowledge of the field which the query is focused on. Formally, *relevance* measures the information loss between the distribution of a summary $Y$and the requested knowledge field (i.e., the query Q) (Peyrard, 2019) which is defined via the cross-entropy $CE(Y, Q)$:

$$Rel(Y, Q) = \sum_{w_i \in Y \cup Q} \mathbb{P}_Y(w_i) \cdot \log\Big(\mathbb{P}_Q(w_i)\Big), \tag{1}$$

where $w_i$is a character appearing in Yor Q. We propose to maximize $Rel(Y, Q)$or $Rel(Q, Y)$under the limitation of summary length. Specifically, we test the following methods based on relevance:

- **Rel-YQ-top6**. Iteratively select the top Ntext pieces of document Daccording to $Rel(Y, Q)$. The iteration stops until $N = 6$or the number of characters in Yreaches or exceeds 70 for the first time.
- **Rel-YQ-top3-expand**. Iteratively select the top 3 text pieces according to $Rel(Y, Q)$. Expand each text piece in order to include its preceding and following text piece until the number of characters in Yreaches or exceeds 70.
- **Rel-QY-top3-expand**. Similar with the above method, except that we replace $Rel(Y, Q)$with $Rel(Q, Y)$.
- **Rel-QY-top2-expand**. Similar with the above method, except that we replace $N = 3$with $N = 2$.

3.2. Ranking with dual attention

Fig. 4 presents the architecture of our online unsupervised summarization model. We first acquire the representations of the query and the document with pre-trained word embeddings. We represent the query by a collection of embedding vectors of the query words it contained. And the document representation is formed by its text pieces, where the embedding vector of each text piece is computed by taking an average of its word vectors. Our model then exploits the derived representations to compute various feature scores with respect to different dimensions of the samples. The main features shown in Fig. 4 include the following:

- **S-D self-attention**. This feature measures the importance of each text piece $S \in \mathcal{S}$to $D$by interacting it with the other remaining text pieces in the document and calculating the self-attention scores of $S \in \mathcal{S}$. We follow Vaswani et al. (2017)for the computation of attention scores. A summary should disclose important information of the document.
- **S-Q co-attention**. This feature measures the relevance of each sentence $S \in \mathcal{S}$to $Q$by calculating the co-attention scores between $Q$and itself. We suppose the summary should be highly relevant to the query.
- **S-Q semantic matching (DSSM)**. This feature measures the semantic relevance between $S$and $Q$by a Deep Structured Semantic Matching (DSSM) model (Huang et al., 2013). The DSSM model is trained in a pair-wise setting with 0.2 billion query-title

查询: 吃什么可以降血糖
Query: What can you eat to lower blood sugar

标题：红薯升血糖还是降血糖？血糖高可以吃哪些食物？
Title: Does sweet potato raise or lower blood sugar? What foods can people eat with hyperglycemia?

正文：高血糖人群的饮食十分严格，只要食物的糖份较高，就不能多吃。大部分人认为红薯、南瓜等食物的含糖量较高，高血糖人群不能食用，那么红薯究竟是降血糖还是升血糖呢？血糖高可以吃哪些食物呢？ 。。。如果混合其他食物一起食用，**红薯能够有效降低食物的消化速度，提高饱腹感，平稳血糖的起伏。所以高血糖人群可以适当吃一些红薯，不但不会升血糖，还有利于血糖控制。**1. 高血糖人群的饮食要以清淡为主，尽量减少高油、高脂、高糖、高盐类的分食物。。。
Content: The diet of hyperglycemic people is very strict, as long as the sugar content of the food is high, they should not eat more. Most people think that foods like sweet potatoes and pumpkins have a high sugar content, and people with high blood sugar can't eat them. So does sweet potatoes lower or raise blood sugar? What foods can you eat with hyperglycemia? ... If mixed with other foods, **sweet potatoes can effectively reduce the speed of food digestion, improve satiety, and smooth blood sugar fluctuations. Therefore, people with hyperglycemia can eat some sweet potatoes, not only will not raise blood sugar, but also conducive to blood sugar control.** First, people with high blood sugar should have a light diet, and try to reduce foods that are high in oil, fat, sugar and salt...

Fig. 1. An example of the query-doc-summary tuple in our QBSUM dataset. The summary is shown in blue text.

pairs from Sougou search engine, where we utilize the queries and the titles of top 1 clicked documents as positive examples, and sample from random combinations of query-title pairs to construct negative samples.
- **S position**. It indicates the sequential order of Sin D. We apply min-max normalization on the sequential number to calculate the position score.
- **S-Q overlap**. It measures the overlap between $S$and Qon word level and is estimated by $\frac{n_o}{n}$,where $n_o$is the number of overlapping query characters in S,and $n$is the total number of query characters.

In our implementation, we use different weights to combine the above feature scores where the weights are tuned as hyperparameters, and derive an unsupervised model that ranks the scores of each Sbelonging to the output summary Y. The feature weights can also be learned by simple logistic regression resulting in a supervised model.

Next, we expand and combine the ranked text pieces to extract a query-based summarization Y. Algorithm 1presents the detailed steps to extract the text pieces belonging to Y. We iterate through each candidate text piece $S_{p_i}$and expand it by its preceding and following text pieces (if available) to get C_{p_i}. We estimate the redundancy of $C_{p_i}$by the ratio of overlapping bi-grams between $C_{p_i}$and S_{p_i}. We discard C_{p_i}if the redundancy reaches or exceeds certain threshold (we use 0.5). Otherwise, C_{p_i}will be appended as a part of Y. We repeat this step until the length (number of words) of $Y$is larger than a threshold (we use 70).

3.3. Query-based summarization based on BERT

Large-scale pre-training models such as BERT (Devlin et al., 2018) have dramatically advanced the performance on a wide range of NLP tasks. We propose a simple BERT-based model with a pre-trained BERT of 110M parameters as encoder, and perform $S-Q$text-pair classification to determine whether each text piece Sbelongs to the summary Y. The architecture of the model is shown in Fig. 5.

In order to distinguish the query and the document in the input of BERT, we use a [*CLS*]token at the start of the input followed by the query tokens, then a [*SEP*]token followed by tokens of document, and another [*SEP*]token is appended at the end of the document. Each token of the input is represented by the summation of its token embedding, segment embedding, and position embedding and is fed to the BERT model to obtain its encoded BERT representation. The self-attention mechanism adopted in BERT allows the learning of correlation between any two tokens in the document or the query. Hereby, the query information is also transmitted to the document representations.

Then, a mean-pooling layer is adopted to collect the sentence vector representation of each text piece in the document by taking the average among all the token vectors within the text piece. Furthermore, on top of the derived BERT sentence representations, a transformer layer is added to model the correlation among text pieces in the document and produce document-aware sentence representations, which are then fed through a linear projection layer that transforms the representation into a scalar score.

Inspired by Rel-QY in Section 3.1 and Rank-DualAttn in Section 3.2, we add several modules in addition to the BERT prediction including: (i) a relevance module which computes the relevance between a given text piece and the query using Eq. (1); (ii) an S-D self-attention module which measures the importance of each text piece regarding to the document, note that we use the derived BERT representation instead of pre-trained word vectors in Rank-DualAttn; (iii) an S-Q co-attention module which calculates the co-attention scores between the query and a text piece with BERT representations. The computed scores are then concatenated together with the BERT prediction score to make the final prediction to decide whether a text piece belongs to the output summary.

4. Experiments

In this section, we evaluate our approaches for query-based document summarization and make comparisons with multiple baselines. We also demonstrate the benefits QBSUM brought to search engines through large-scale online A/B testing.

4.1. Experimental setup

4.1.1. Baseline methods

We compare our model with three baseline methods, Textrank-DNN, tfidf-DNN, and MDL (Litvak and Vanetik, 2017).

In **Textrank-DNN** and **tfidf-DNN**, an undirected graph is created from the document with nodes being the semantic embeddings of text pieces and edges being cosine similarities with the query. The difference is that Textrank-DNN uses Textrank algorithm (Mihalcea and Tarau, 2004) to compute the weights of nodes where tf-idf algorithm is used in tfidf-DNN.

MDL first selects frequent word sets related to the give query, then extracts summaries by selecting sentences that best cover the sets.

For our Bert-QUSUM model, we perform ablation analysis to study the effectiveness of different modules of our proposed model and evaluate the following versions:

- **Bert-QBSUM (no relevance)**. In this variant, we keep the dual attention modules and the transformer sentence encoder, and remove the relevance module.
- **Bert-QBSUM (no self-attention)**. This model variant does not contain the self-attention module.
- **Bert-QBSUM (no co-attention)**. The co-attention module is not contained in this variant.

Table 1
The statistical information of QBSUM and compare it with the DUC datasets.

Datasets	QBSUM	DUC2005	DUC2006
# source sentences	2,175,639	14,410	18,794
# summary sentences	105,751	4,242	5,037
avg #characters/query	7.1	-	-
avg #characters/summary	41	250	250
#unique queries	16,250	-	-
#unique documents	43,762	-	-
#doc-summary-query	49,535	-	-
avg #words/document	378.57	-	-
avg #words/query	3.84	-	-
avg #words/summary	53.08	-	-

- **Bert-QBSUM (no transformer encoder)**. The transformer sentence encoder is removed so that the output of mean-pooling layer is used as representation of text pieces to produce the BERT prediction scores.
- **Bert-QBSUM**. This is the complete version of our BERT-based summarization model.

4.1.2. Metrics

We use ROUGE and BLEU to evaluate model performance.

ROUGE (Lin, 2004) measures the quality of a summary by comparing it to reference summaries by counting the number of overlapping units. In our experiment, we use n-gram recall ROUGE-N with $N = 1, 2$and ROUGE-L based on Longest Common Subsequence (LCS) statistics.

BLEU (Papineni et al., 2002) measures precision by how much an n-gram text in prediction sentences appear in reference sentences at the corpus level. BLEU-1, BLEU-2, BLEU-3, and BLEU-4, use 1-gram to 4-gram for calculation, respectively.

4.1.3. Implementation details

For our experiments, we utilize a part of the QBSUM dataset with around 10,000 data samples as the rest of the data was collected after the experiments were conducted. The dataset is split into a train set, a evaluation set and a test set consisting of 8787, 1099 and 1098 samples respectively. All performances are reported on the test set. In the implementations of our unsupervised models and the baseline models, we use a 200-dimensional Chinese word embedding trained using w2v (Mikolov et al., 2013) on 2 billion queries. The maximum length of generated summaries is set to 70 Chinese words.

4.2. Results and analysis

Table 2 summarizes the performance of all the compared methods on the QBSUM dataset. As the inference speed of models are critical to real-world applications such as search engines or recommender systems, we also report inference time of different methods on a set of 2120 samples in QBSUM. Among all methods, our unsupervised methods achieve both better performance and faster inference when compared with baseline methods. And our supervised BERT-based models produce the best results.

By observing the results, we can see that the Bert-QBSUM model obtains the best performance in terms of all performance metrics as shown in Table 2 and achieves a huge performance gain compared with previous models. The success of our Bert-QBSUM model can be attributed to the powerful encoding ability of BERT as well as the combination of different feature modules in our model. By carefully examining the performance of different variants of QB-SUM, we can see that when the co-attention and relevance are removed, the performance drops notably, from which we can conclude that modeling the relevance between each text piece and the query plays an essential role in query-based summarization. The transformer encoder and the self-attention module are also beneficial to the model performance, as they are capable of modeling the correlation between text pieces in documents on sentence level where BERT only models the information on token level.

However, Bert-QBSUM is quite slow in inference than other methods due to its large number of model parameters. The relevance-based summarization model variants achieve the best inference speed because they are unsupervised methods and require less calculations during inference. As a trade-off between inference speed and performance, our online Rank-DualAttn model achieves better performance within acceptable inference time, making it the most suitable model for real-world applications.

The Rank-DualAttn model is deployed into QQ browser that involves more than 0.2 billion daily active users all around the globe. Fig. 6 demonstrates its effectiveness through an example. When a user inputs a query “Is it reliable to buy things on Pinduoduo?”, our model is able to extract a summary from the top clicked documents which contains the answer to the query and relevant explanations and details. And the performance gain achieved proved the effectiveness of our designed features.

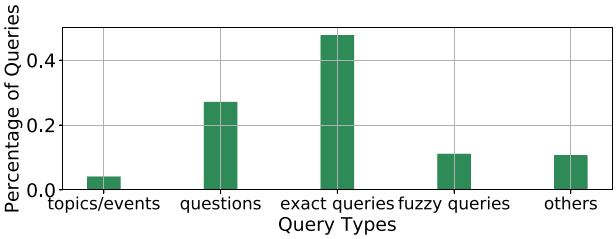

Fig. 2. The distribution of query types in QBSUM.

We performed error analysis on 100 summaries sampled from our applications. Based on our observation, the errors can be mainly divided into two groups: (i) *low relevance*, which means the summary are not quite relevant to the query, and (ii) *incomplete information*, i.e., the summary only contains an incomplete part of the key information relevant to the query.

The main cause of the errors is due to the fact that the semantic relevance estimation between query and document sentences is inaccurate. As supplementary, the model emphasizes more on low-level features such as keyword overlapping or textual similarities. For example, given a query "What is the order of the Chinese zodiac signs", the model is not capable of capturing the hidden information that "Chinese zodiac signs" represent 12 kinds of animals. As a result, the extracted summary usually contains the keyword "Chinese zodiac signs", but not exactly the answer of "the order of the Chinese zodiac signs". Our analysis shows that low relevance errors often happen for the type of exact queries, and incomplete information errors emerge most frequently for question type queries.

4.3. Online A/B testing for query-based document summarization

We perform large-scale online A/B testing to show how query-based document summarization helps with improving the performance of searching in real world applications.

For online A/B testing, we split users into buckets where each bucket contains 5 million users. We first observe and record the activities of each bucket for 3 days based on the following metrics:

- **Click-Through Rate (Global-CTR)**: the ratio of users who clicked on any of the search results to the total users who received the results.
- **Top 1 Click-Through Rate (Top1-CTR)**: the ratio of users who clicked on the top 1 search result to the total users who received the result.
- **Top 2 Click-Through Rate (Top2-CTR)**: similarly, here we use top 2 results.
- **Top 3 Click-Through Rate (Top3-CTR)**: similarly, here we use top 3 results.
- **Selection Rate**: the ratio of user clicked queries to the total queries.

We then select two buckets with highly similar activities where our Rank-DualAttn query-based summarization model is utilized in one of the buckets while in the other buckets the first few sentences of the document are presented to users as the summarization. We run our A/B testing for 3 days and compare the results on the above metrics.

Table 3 shows the results of our online A/B testing. In the online experiment, we observe a statistically significant Global-CTR gain (2.86%) when employing our Rank-DualAttn model. We also detect improvements on other metrics such as selection rate and top-N CTRs. These observations prove that our online model for query-based summarization greatly benefits the search engine and grants users better experience by precisely capture their interested documents. With the help of query-based summarization, we can better capture relevant and helpful information in a document attractive to users and display the summaries before users click into the search results. Such summaries grant users the ability of quickly retrieve the core meaning of a document and locate the document of their true interest, hence improving both the effectiveness and the efficiency of the search engine.

5. Related work

5.1. Existing datasets

In this section, we introduce existing datasets for document summarization and their characteristics.

Generic Document Summarization aims at compressing long documents into human readable short summaries that contain the most important information without specific focuses. In the past years, several large-scale summarization datasets have been introduced to accommodate the advance of this field.

Gigaword (Graff et al., 2003) is a large-scale dataset containing more than 8 million documents from different newswire sources and corresponding headlines which are used as simulated summaries in prior work (Rush et al., 2015; Chopra et al., 2016). This compromise results in shorter summaries than most natural summary text. The New York Times Annotated Corpus

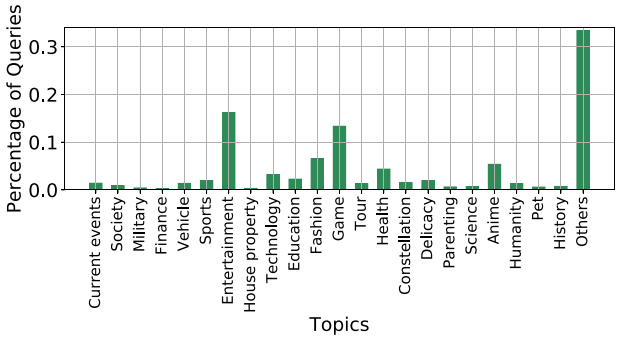

Fig. 3. Topic distribution in QBSUM.

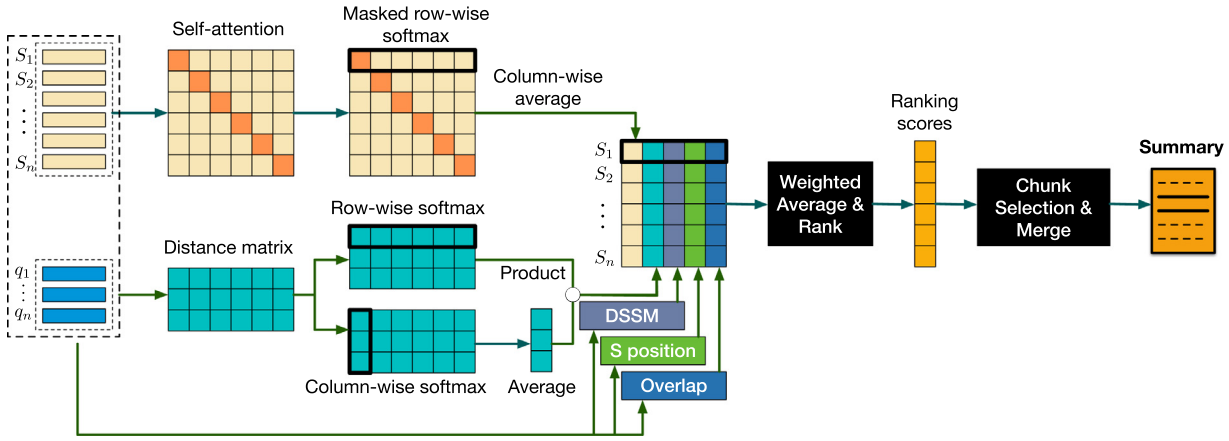

Fig. 4. The architecture of our online query-based summarization model.

(Sandhaus, 2008) is a collection of over 1.8 million articles from the New York Times magazine between 1987 and 2007, with manually written and tagged summaries. It has been used for both extractive document summarization (Li et al., 2016; Xu and Durrett, 2019) and abstractive document summarization (Gehrmann et al., 2018; Celikyilmaz et al., 2018).

CNN / Daily Mail question answering dataset (Hermann et al., 2015) is originally introduced as a Cloze-style QA dataset, but also widely adopted on generic document summarization. It consists of CNN and Daily Mail articles, each associated with several bulletin point descriptions which are concatenated to form a summary. Grusky et al. (2018) presented the NEWSROOM dataset, consisting of 1.3 million articles and summaries written by their original authors and editors. LCSTS (Hu et al., 2015) is a Chinese short text summarization dataset collected from Weibo. It consists of over 2 million short texts with short summaries given by the author of each text.

Query-based Document Summarization, compared with generic document summarization, highlights the points in the document relevant to the context of a query. It is of great value to question answering and search engines. However, due to the lack of datasets, this problem has drawn much less attention.

DUC[3] and CAT[4] have proposed several query-based summarization task in the past years, where each summary is focused on a number of complex questions However, they only provide a small test dataset which is far from satisfactory.

Some researchers studying query-based summarization create datasets themselves by crawling from the web (Nema et al., 2017) or generating queries from hand-crafted rules. However, such datasets often suffer from poor quality control and lack of data diversity.

[3] https://www-nlpir.nist.gov/projects/duc/index.html
[4] https://tac.nist.gov/

ALGORITHM 1: Greedy Sentence Selection

Input: a sequence of sorted text pieces $\mathcal{S} = \{S_{p_1}, S_{p_2}, \cdots, S_{p_m}\}$, where p_i is the position of S_{p_i} in a document D.
Output: Summary $Y = \{S_{y_1}, S_{y_2}, \cdots, S_{y_{|Y|}}\}$.

$Y \leftarrow \varnothing$;
for each $S_{p_i} \in \mathcal{S}$ **do**
  $C_{p_i} = [S_{p_i-1}]S_{p_i}[S_{p_i+1}]$;
  $Redundancy = \frac{Count(Bigram(C_{p_i} \cap S_{p_i}))}{Count(Bigram(C_{p_i} \cup S_{p_i}))}$;
  if $Redundancy \geq 0.5$ **then**
    Continue;
  end if
  $Y = Y \cup C_{p_i}$;
  if The length of Y reaches threshold **then**
    Break;
  end if
end for
Sort the text pieces in Y by their positions.

ALGORITHM 1. Greedy Sentence Selection.

QBSUM contains more than 49,000 ⟨*query, document, summarization*⟩,with considerations of various quality measurements, including relevance, informativeness, richness, and readability. To the best of our knowledge, QBSUM is the first large-scale high-quality Chinese query-based summarization dataset.

5.2. Query-based summarization methods

Document summarization methods can be classified into extractive summarization which summarizes the document by extracting key words and phrases without modification, and abstractive summarization which generates new sentences to form a summary.

Work on generic extractive summarization spans a large range of approaches. Maximum Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) is a widely known greedy approach. McDonald (2007) and Gillick and Favre (2009) formulate this question as an Integer Linear Programming problem. Graph-based models also play a leading role in this field due to its

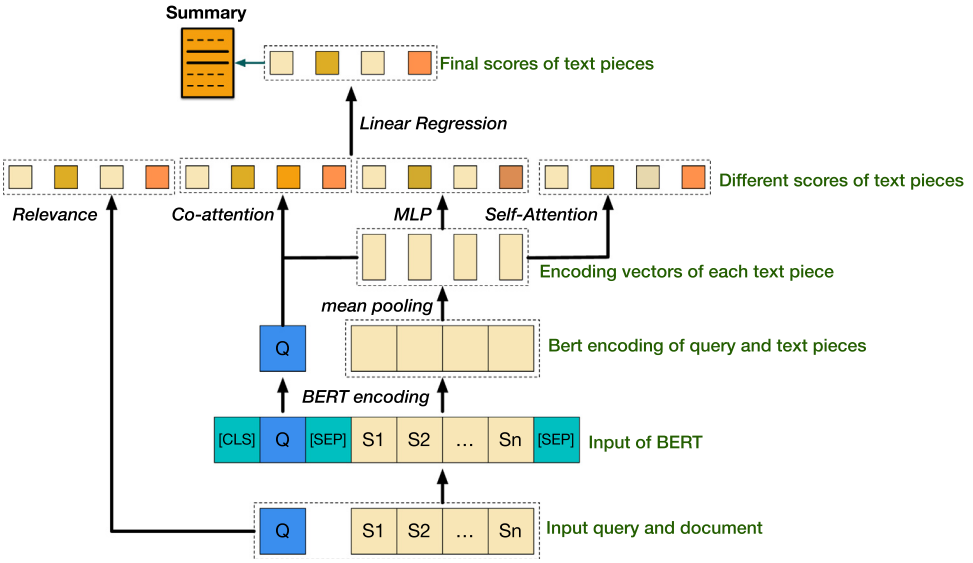

Fig. 5. The architecture of our BERT-based query-based summarization model.

Table 2
Compare the performance of different approaches based on the QBSUM dataset.

Models	BLEU				ROUGE			Inference Time (s)
	1	2	3	4	1-recall	2-recall	L-recall	(2,120 samples)
TFIDF-DNN	41.43	36.38	33.34	31.24	43.92	33.09	37.04	35
MDL	36.48	29.02	24.99	22.41	37.64	23.10	29.87	79
TextRank-DNN	42.89	36.84	33.33	30.96	44.23	32.70	36.67	1,134
Rel-YQ-top6	39.76	34.16	30.77	28.30	41.48	29.41	33.58	30
Rel-YQ-top3-expand	47.15	40.79	36.81	33.85	56.08	41.43	43.67	20
Rel-QY-top3-expand	44.93	39.09	35.49	32.85	61.84	46.01	47.54	20
Rel-QY-top2-expand	47.73	41.69	37.93	35.16	53.64	39.41	42.67	15
Rank-DualAttn	50.78	45.68	42.33	39.77	58.68	45.96	56.17	762
Bert-QBSUM (no relevance)	60.99	58.40	56.88	55.83	72.89	67.03	71.95	12,944
Bert-QBSUM (no co-attention)	61.64	59.03	57.52	56.48	73.17	67.34	72.23	11,391
Bert-QBSUM (no self-attention)	62.20	59.62	58.13	57.12	72.42	66.55	71.46	11,236
Bert-QBSUM (no transformer encoder)	62.03	59.52	58.05	57.03	73.67	67.92	72.62	10,767
Bert-QBSUM	**62.40**	**59.91**	**58.45**	**57.45**	**74.69**	**68.91**	**73.58**	14,000

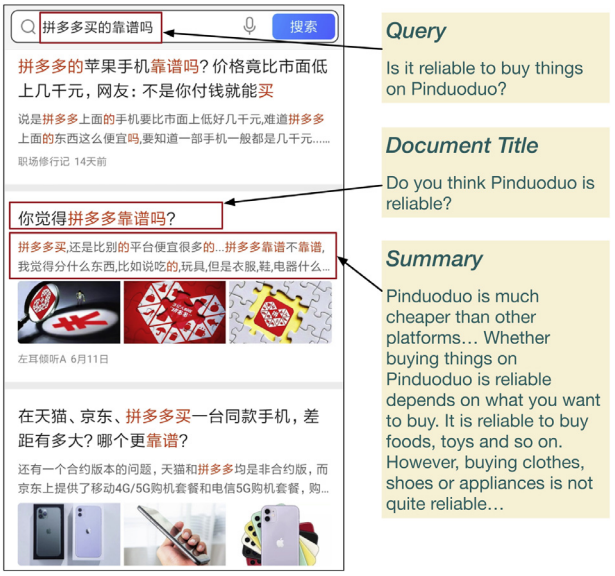

Fig. 6. Example of query-based document summarization in QQ browser.

ability to construct sentence relationships (Erkan and Radev, 2004; Parveen et al., 2015; Parveen and Strube, 2015). Recently, reinforcement learning methods have been applied (Narayan et al., 2018). For example, Narayan et al. (2018) conceptualize extractive summarization as a sentence ranking task and optimize ROUGE through an RL objective.

Many query-based summarizers are heuristic extensions of generic summarization methods by incorporating the information of the given queries. A variety of query dependent features were defined to measure the relevance, including TF-IDF cosine similarity (Wan and Xiao, 2009), WordNet similarity (Ouyang et al., 2011), and word co-occurrence (Prasad Pingali and Varma, 2007), etc. Cao et al. (2016) proposed a joined attention model AttSum to meet the query need and calculate sentence weight.

6. Conclusion

In this work, we introduce a new Chinese query-based summarization dataset called QBSUM, which to the best of our knowledge, is the first large-scale high-quality dataset in query-based summarization. QBSUM contains more than 49,000 data samples collected from real-world applications, and is two magnitudes larger in scale than existing datasets such as DUC2005 and DUC2006. The QBSUM dataset is released and we hope that this dataset will promote future development of this research field.

Table 3
Online A/B testing results.

Metrics	Percentage Lift
Selection Rate	+0.33%
Top1-CTR	+0.58%
Top2-CTR	+1.34%
Top3-CTR	+1.38%
Global-CTR	+2.86%

Additionally, we propose several supervised and unsupervised models which incorporate different properties of queries and documents including relevance, informativeness, and importance. The experiments on our QBSUM dataset demonstrate that our methods surpass other baselines.

In the future, we plan to further study the interactions among the queries, documents and summaries and develop abstractive methods on the QBSUM dataset.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Cao, Z., Li, W., Li, S., Wei, F., Li, Y., 2016. AttSum: joint learning of focusing and summarization with neural attention. arXiv:1604.00125.

Carbonell, J.G., Goldstein, J., 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries.. In: Proceedings of the SIGIR, 98, pp. 335–336.

Celikyilmaz, A., Bosselut, A., He, X., Choi, Y., 2018. Deep communicating agents for abstractive summarization. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1662–1675.

Choi, E., Hewlett, D., Uszkoreit, J., Polosukhin, I., Lacoste, A., Berant, J., 2017. Coarse-to-fine question answering for long documents. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 209–220.

Chopra, S., Auli, M., Rush, A.M., 2016. Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93–98.

Dang, H.T., 2005. Overview of DUC 2005. In: Proceedings of the Document Understanding Conference, 2005, pp. 1–12.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Erkan, G., Radev, D.R., 2004. Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Intel. Res. 22, 457–479.

Gehrmann, S., Deng, Y., Rush, A., 2018. Bottom-up abstractive summarization. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 4098–4109.

Gillick, D., Favre, B., 2009. A scalable global model for summarization. In: Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing. Association for Computational Linguistics, pp. 10–18.

Graff, D., Kong, J., Chen, K., Maeda, K., 2003. English Gigaword. 4. Linguistic Data Consortium, Philadelphia, p. 34.

Grusky, M., Naaman, M., Artzi, Y., 2018. Newsroom: a dataset of 1.3 million summaries with diverse extractive strategies. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 708–719.

He, W., Liu, K., Liu, J., Lyu, Y., Zhao, S., Xiao, X., Liu, Y., Wang, Y., Wu, H., She, Q., et al., 2017. Dureader: a Chinese machine reading comprehension dataset from real-world applications. arXiv:1711.05073.

Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P., 2015. Teaching machines to read and comprehend. Advances in Neural Information Processing Systems, pp. 1693–1701.

Hu, B., Chen, Q., Zhu, F., 2015. Lcsts: a large scale chinese short text summarization dataset. arXiv:1506.05865.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., Heck, L., 2013. Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management. ACM, pp. 2333–2338.

Li, J.J., Thadani, K., Stent, A., 2016. The role of discourse units in near-extractive summarization. In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 137–147.

Lin, C.-Y., 2004. Rouge: a package for automatic evaluation of summaries. Text Summarization Branches Out, pp. 74–81.

Litvak, M., Vanetik, N., 2017. Query-based summarization using MDL principle. In: Proceedings of the Multiling Workshop on Summarization and Summary Evaluation Across Source Types and Genres, pp. 22–31.

Liu, B., Niu, D., Lai, K., Kong, L., Xu, Y., 2017. Growing story forest online from massive breaking news. In: Proceedings of the ACM on Conference on Information and Knowledge Management. ACM, pp. 777–785.

McDonald, R., 2007. A study of global inference algorithms in multi-document summarization. In: Proceedings of the European Conference on Information Retrieval. Springer, pp. 557–564.

Mihalcea, R., Tarau, P., 2004. Textrank: bringing order into text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781.

Narayan, S., Cohen, S. B., Lapata, M., 2018. Ranking sentences for extractive summarization with reinforcement learning. arXiv:1802.08636.

Nema, P., Khapra, M.M., Laha, A., Ravindran, B., 2017. Diversity driven attention model for query-based abstractive summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1063–1072.

Ouyang, Y., Li, W., Li, S., Lu, Q., 2011. Applying regression models to query-focused multi-document summarization. Inf. Process. Manag. 47 (2), 227–237.

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, pp. 311–318.

Parveen, D., Ramsl, H.-M., Strube, M., 2015. Topical coherence for graph-based extractive summarization. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1949–1954.

Parveen, D., Strube, M., 2015. Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence.

Peyrard, M., 2019. A simple theoretical model of importance for summarization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1059–1073.

Prasad Pingali, R.K., Varma, V., 2007. IIIT Hyderabad at DUC 2007. In: Proceedings of the DUC.
Rush, A.M., Chopra, S., Weston, J., 2015. A neural attention model for abstractive sentence summarization. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 379–389.
Sandhaus, E., 2008. The New York Times Annotated Corpus. 6, p. e26752.
Sun, J.-T., Shen, D., Zeng, H.-J., Yang, Q., Lu, Y., Chen, Z., 2005. Web-page summarization using clickthrough data. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 194–201.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in Neural Information Processing Systems, pp. 5998–6008.
Wan, X., Xiao, J., 2009. Graph-based multi-modality learning for topic-focused multi-document summarization. In: Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence.
Wang, C., Jing, F., Zhang, L., Zhang, H.-J., 2007. Learning query-biased web page summarization. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management. ACM, pp. 555–562.
Wang, H., Yu, D., Sun, K., Chen, J., Yu, D., Roth, D., McAllester, D., 2019. Evidence sentence extraction for machine reading comprehension. arXiv:1902.08852.
Xu, J., Durrett, G., 2019. Neural extractive text summarization with syntactic compression. arXiv:1902.00863.