Dual Learning for Query Generation and Query Selection in Query Feeds Recommendation

Kunxun Qi1,2, Ruoxu Wang2, Qikai Lu3, Xuejiao Wang2, Ning Jing2, Di Niu3, Haolan Chen2,5
1School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
2Platform and Content Group, Tencent, Shenzhen, China
3University of Alberta, Edmonton, Canada
kunxunqi@foxmail.com, haolanchen@tencent.com

ABSTRACT
Query feeds recommendation is a new recommended paradigm in mobile search applications, where a stream of queries need to be recommended to improve user engagement. It requires a great quantity of attractive queries for recommendation. A conventional solution is to retrieve queries from a collection of past queries recorded in user search logs. However, these queries usually have poor readability and limited coverage of article content, and are thus not suitable for the query feeds recommendation scenario. Furthermore, to deploy the generated queries for recommendation, human validation, which is costly in practice, is required to filter unsuitable queries. In this paper, we propose TitIE, a query mining system to generate valuable queries using the titles of documents. We employ both an extractive text generator and an abstractive text generator to generate queries from titles. To improve the acceptance rate during human validation, we further propose a model-based scoring strategy to pre-select the queries that are more likely to be accepted during human validation. Finally, we propose a novel dual learning approach to jointly learn the generation model and the selection model by making full use of the unlabeled corpora under a semi-supervised scheme, thereby simultaneously improving the performance of both models. Results from both offline and online evaluations demonstrate the superiority of our approach.

CCS CONCEPTS
• Information systems → Query suggestion, Query intent.

KEYWORDS
Query feeds recommendation; Query generation; Query suggestion

1 INTRODUCTION
Feeds recommendation aims at recommending contents to users by never-ending feeds in mobile applications [35]. In this paper, we study query feeds recommendation (QFR), where a stream of search queries are recommended to users by never-ending feeds as they scroll down in the mobile application under the search box. It has been widely employed in existing search applications to increase user engagement and improve their search experience [7, 17]. Figure 1 shows an example of QFR. Different from related tasks, such as query recommendation [8, 18, 30, 42], QFR recommends queries that appeal to users based on their user profiles without an anchor query, which means that plenty of interesting queries are needed to cover users’ interests.

In general the recommended queries can be easily collected from user search logs [2, 3, 30, 42]. However, these queries are generally unsuited for the QFR scenario. The reason is two-fold. On one hand, the queries generated by users during active search are usually presented as a set of keywords split by space with unlimited length and poor readability. These queries are inconcise and unattractive, and are thus unsuited for the QFR scenario. On the other hand, collecting queries from user search logs is hysteric in time, suffering heavily from the problem of limited coverage and generalizability [8, 36].

In this paper, we study the generation of valuable queries from document titles for QFR in Tencent QQ Browser1, where millions of new content documents are created each day. Query generation

1Tencent QQ Browser has the largest market share in the Chinese mobile browser market with more than 100 million daily active users.
from article titles is able to provide massive amount of interesting and attractive queries for QFR, helping to improve the interest coverage and search experience of users. Nowadays, neural sequence to sequence (seq2seq) models [28] have been dominant for text generation tasks, including text summarization [24] and machine translation [9]. However, common seq2seq models suffer from the problem of the low generalizability in open domain [8, 18], heavily relying on the manually annotated training corpus. In practice, manual annotation of training corpus is laborious and error prone. Thus, an effective strategy for training a query generation model with limited annotated corpus is necessary. Moreover, the generated queries require human validation before incorporation into online QFR. Yet, the crowdsource resources have a daily usage limit. To use the crowdsource resources more efficiently, an automatic strategy for pre-selecting high quality generated queries is necessary.

To tackle the above challenges, we propose TitIE (short for Title Information Extractor), a query mining system for providing valuable queries for QFR at Tencent QQ Browser, as illustrated in Figure 2. For extracting queries from article titles, we propose a query extraction model named QEBERT (short for Query Extraction based on pre-trained language model BERT [5]) to extract valuable queries from article titles. It creates a query by extracting a text span (excerpt) from a given title instead of generating a new query from the title. For some titles, however, there are no suitable spans for which queries can be extracted. To fill this gap, we introduce the pre-trained text generation model mT5 [41] for abstractive query generation, which is complementary to the QEBERT model. In practice, the generated queries require human validation before incorporation into online QFR. Yet, the crowdsource resources have a daily usage limit. To improve the acceptance rate during human validation, as to make full use of the crowdsource resources, we propose a query selection model named QSBERT (short for Query Selection based on pre-trained language model BERT) to pre-select the extracted and generated queries, such that only queries that are predicted as acceptable are presented for human validation. The selected queries are then manually validated by crowdsource to select the acceptable (measured by informativeness, coherence, etc.) queries for recommending to users. To further improve the performance, we carefully design a novel dual learning framework to jointly learn the query generation models and the query selection model, making full use of the unlabeled corpus to mutually improve the performance of both models, as illustrated in Figure 3.

We empirically evaluate our framework through both offline and online evaluations. The offline evaluation is two-fold. First, we create a 50K manually annotated dataset which is collected from document titles in Tencent QQ Browser. It consists of 40K (query, title) tuples for training, 5K for validation and 5K for test. Experimental results on the test set shows that our approach outperforms all baseline models on a number of metrics, including: ROUGE-n [16], BLEU-n [19] and Exact match score. Second, we carry out human evaluation to verify whether our approach can improve the fluency and relevance of generated queries. Results show that our approach outperforms all baseline models. For online evaluation, we verify the feasibility of our system through both A/B Testing and human validation in Tencent QQ Browser. Results from A/B Testing and human validation reveal that the Click-Through Rate (CTR) increases by 26.89% and the acceptance rate increases by 33.0% when TitIE is employed for query generation in our QFR system.

TitIE has been deployed in Tencent QQ browser since January, 2021. Up to today, TitIE has generated more than 5 million high-quality queries from daily article titles, while still growing at a rate of 60K found per day. We make the following contributions in the design of TitIE:

(1) We release a manually annotated dataset, collected from the titles of real-world documents in Tencent QQ Browser, to facilitate the training and evaluation of query generation from document titles for the query feeds recommendation scenario.

(2) We propose a simple but effective query extraction model named QEBERT, based on pre-trained language model BERT, to extract queries from document titles. This overcomes the low generalizability limitation of existing seq2seq models. We also introduce a pre-trained text generation model, mT5, for abstractive query generation, which is complementary with the QEBERT model.

(3) We propose a query selection model named QSBERT for pre-selecting queries that are predicted as acceptable for human validation before incorporation into QFR, thus maximizing the utilization of crowdsource resources.

(4) We propose a novel training framework based on dual learning to jointly learn the query generation model and the query selection model. This allows us to make full use of the unlabeled corpus and mutually improve the performance of both models. To the best of our knowledge, this is the first adaptation of dual training to query generation and selection.

(5) We conduct extensive experiments to verify the effectiveness and the superiority of our approach, including offline evaluations, online evaluations and case study.

2 RELATED WORKS

Our work is mainly related to the research fields of query recommendation, text summarization and dual learning in natural language processing (NLP).

2.1 Query recommendation

Query recommendation (QR), also called query suggestion or query expansion, aims at recommending a set of queries that are more expressive than the anchor query created by users. Early work for QR can be classified into document-based methods [6, 33, 39] and log-based methods [2, 3, 30, 42]. More recently, direct generation of queries from a given query using Seq2Seq [28] models attracts increasing attention. The work [18] proposes to incorporate the pointer network [24] to directly copy words from the input context. The work [15] investigates the problem of feedback-aware query suggestion and proposes a adversarial framework for improving performance. The work [8] proposes a two-stage generation framework by partitioning the QR task into the problems of relevant words discovery and context dependent query generation.

Different from the QR task, the query feeds recommendation scenario recommends a great number of interesting queries to users based on their user profiles without an anchor query, which means that massive attractive queries are needed to cover users’ interests. Thus we study generating queries from document titles in this paper.
2.2 Text Summarization

The approaches for text summarization can generally be categorized to extractive and abstractive.

Extractive Summarization methods usually summarize the given documents by extracting and merging sentences within the documents. Recently, pre-trained language models like BERT [5] are widely used in different fields in NLP, including extractive summarization. The work [40] proposes a discourse-aware neural summarization model based on BERT to reduce summarization redundancy. The work [45] proposes an extractive summarization method based on BERT for text matching. Inspired by the promising result of these works, we propose QEBERT for extracting the text span from a given title as the extracted query via span extraction by answer pointer network [32]. It does not require a large vocabulary for adapting the open domains.

Abstractive Summarization methods usually compose a new summary entirely based on the understanding of the documents. Seq2seq (short for sequence-to-sequence) [28] models have been a performant paradigm in abstractive summarization. The work [24] proposes an attention-based seq2seq model for abstractive summarization which not only generates words from the vocabulary, but also copy words from the source document. The work [31] proposes the Transformer, which relies solely on the Multi-Head Attention [31] mechanism for text understanding and text generation. More recently, pre-trained text generation models like T5 [22] have been dominant for several NLP tasks, including text summarization. Inspired by the promising result of pre-trained text generation models, we also integrate mT5 [41], a multi-lingual version of T5, for Chinese query generation.

2.3 Dual Learning in NLP

Dual learning is a new learning paradigm to leverage the symmetric structure of some learning problems that are emerged in dual forms, such as English->French translation and speech->text transformation [37, 38]. It has been widely employed in many NLP tasks. The work [9] firstly proposes a novel dual learning framework for machine translation. The work [20] proposes a dual learning framework for unsupervised text style transfer. The work [26] extends the emotion-controllable response generation to a dual task to generate emotional responses and emotional queries alternatively by a dual learning framework. Inspired by the promising result of these works, we carefully design a dual learning framework to jointly learn query generation and query selection.

3 THE TITIE SYSTEM

In this section, we describe the details of our proposed TitIE system for query feeds recommendation. The overview of TitIE is illustrated in Figure 2. Given a number of user click logs, TitIE first mines the document titles that are widely clicked by users, namely the high PV titles. These titles are fed into the query generation module that consists of a query extraction model QEBERT, and a query generation model based on the pre-trained mT5, which are described in detail in section 2.1 and 2.2 respectively. The generated queries are selected by the query selection model named QSBERT, which is described in detail in section 2.3. The selected queries are manually validated by crowdsource to filter the unsuitable queries for further recommendation. These include queries that are confusing or inappropriate to user. The validated queries are stored in a large-scale database along with query profile features that are used in the recommendation system. The recommendation system presents attractive and interesting queries to users based on query profiles and user profiles. The search behaviors of the users are recorded through user click logs, thereby forming a closed loop with the recommendation system.

3.1 Formalization of QEBERT

Previous works have shown concerns about the low generalizability of the seq2seq models as the training corpus and the vocabulary size are usually limited in open domain [8, 36]. In practice, the system suffers heavily from cold start problem, wherein the annotation
The architecture of our proposed dual learning framework.

Figure 3: The architecture of our proposed dual learning framework.

of training corpus is laborious and error prone. To tackle these limitations, we propose QEBERT, an efficient and effective neural model for query extraction from document titles based on pre-trained language model BERT [5]. Inspired by the promising result of span extraction [32] for machine reading comprehension [25, 32] and extractive text summarization [40, 45], we also adapt a well-known answer pointer network [32] to extract text spans from titles as extracted queries. Given a document title, QEBERT predicts the start and end position of the potential query within the given title, as formally defined below.

**Lexical Encoder.** Given a document title \( T \), where \( T = (w_1, ..., w_n) \) denotes the token sequence within the title. Following [5], we create a new sequence \( H = (w_1^H, ..., w_{n+2}^H) \) by adding a special token [CLS] in front of the first token \( w_1^H \) and a [SEP] after the last token \( w_n^H \). This new sequence of tokens is fed into a lexical encoder, which is a pre-trained language model such as BERT [5]. By \( X \in \mathbb{R}^{(n+2) \times d} \) we denote the output of the lexical encoder, where \( n + 2 \) is the number of tokens in the new sequence and \( d \) is the dimension of token embedding.

**Answer Pointer Layer.** This layer aims to compute a probability distribution on the start position and a probability distribution on the end position for a query span. Formally, \( p_i^1 \), the probability that the \( i^{th} \) token is the start position, and \( p_i^2 \), the probability that the \( i^{th} \) token is the end position, are defined by

\[
\begin{align*}
p_i^1 &= \frac{\exp(v_i^1)}{\sum_{j=1}^{n+2} \exp(v_j^1)} \\
p_i^2 &= \frac{\exp(v_i^2)}{\sum_{j=1}^{n+2} \exp(v_j^2)}
\end{align*}
\]

where \( x_i \) denotes the \( i^{th} \) vector in \( X \), \( w_1 \in \mathbb{R}^d \) and \( w_2 \in \mathbb{R}^d \) are trainable vectors, \( b_1 \) and \( b_2 \) are trainable biases.

Given a set of training data \( D = \{(H_i, Y_i^1, Y_i^2)\}_{i=1}^N \), where \( H_i \) denotes the input token sequence, \( Y_i^1 \) the labeled start position of the query span within the title, \( Y_i^2 \) the labeled end position of the query span within the title, and \( N \) the total number of training examples, the entire model is fine-tuned by minimizing the cross-entropy loss:

\[
\mathcal{L}_1(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log p_{Y_i^1}^1 + \log p_{Y_i^2}^2
\]

where \( \theta \) denotes all the trainable parameters in the model.

### 3.2 Formalization of mT5 for query generation

For some title, there exists no suitable span for which queries can be extracted. In this case, we need to generate new queries for these titles. As mentioned before, common seq2seq generation models suffer from the problem of the low generalizability in open domain. Thus we propose to use the mT5 model [41] (the multi-lingual version of the T5 [22] model) for query generation to complement the QEBERT model. T5 (short for Text-to-Text Transfer Transformer) is a large-scale Transformer [31] model pre-trained on a large text corpus [22]. The model mT5 is the multi-lingual version of the T5 model. It is pre-trained on a large-scale 750GB multi-lingual corpus [41] covering over 101 languages, including Chinese. Different from the common seq2seq models, mT5 employs the SentencePiece [14] technology to encode text as WordPiece tokens, which means that it does not require a large vocabulary to improve generalizability. Moreover, as mT5 is pre-trained on a large corpus, it can be fine-tuned for the downstream task with fewer training data.

The model details of mT5 are presented in [22, 31, 41]. Given a set of training data \( D = \{(H_i, Q_i)\}_{i=1}^N \), where \( H_i \) denotes the input token sequence, \( Q_i = (w_1^Q, ..., w_m^Q) \) the corresponding query for the title, and \( N \) the total number of training examples, the fine-tuning objective of mT5 can be formulated as:

\[
\theta_{mT5} = \arg \max \sum_{(H, Q) \in D} \sum_{m=1}^{N_m} \log P(w_m^Q | w_{\leq m}^H; \theta_{mT5})
\]

where \( \theta_{mT5} \) denotes the parameters of the mT5 model and \( N_m \) the length of the corresponding query token sequence.
3.3 Formalization of QSBERT

The generated queries (both from QEBERT and mT5) must then undergo human validation by crowdsourcing to filter queries that are confusing or inappropriate to users. However, there is a daily limit to the crowdsourcing usage. To improve the utilization of crowdsourcing resources, a strategy of pre-selecting only good-quality queries for human validation is necessary. To address this challenge, we propose QSBERT, a query selection model based on the pre-trained language model BERT [5] for pre-selecting generated queries that are more likely to be accepted during human validation. It is a 3-way classifier that classifies the input query into one of three categories, including the normal query, the confusing query and the inappropriate query. Note that the context of the title is not utilized in QSBERT with the form like natural language inference (NLI) [1, 34] as the input title is not considered in human validation.

Lexical Encoder. Given a query \( Q \), where \( Q = (w_1^q, ..., w_m^q) \) denotes the token sequence within the query. We create a new sequence \( P = (w_1^p, ..., w_{m+n}^p) \) by adding a special token [CLS] in front of the first token \( w_1^q \) and a [SEP] behind the last token \( w_m^q \). This new sequence of tokens is fed to BERT to calculate its contextual representation. By \( C \in \mathbb{R}^{(m+2) \times d} \) we denote the output of the lexical encoder, where \( m + 2 \) is the number of tokens in the new sequence and \( d \) is the dimension of token embedding. Following [5], we use the contextual embedding of the [CLS] token, namely \( C_1 \), as the contextualized representation for further classification.

Selection Module. This module aims to select the generated queries that are likely to be accepted during human validation. Given a \( d \)-dimensional contextualized representation, a 3-way discriminator is employed to calculate the predicted probabilities of the three categories, formally defined by

\[
y_i = \text{softmax}(Wc_i + b) \tag{5}
\]

where \( c_i \) denotes the contextualized representation \( C_1 \) of the \( i \)th record, \( W \in \mathbb{R}^{L \times d} \) is a trainable matrix, \( b \in \mathbb{R}_L \) is a trainable vector and \( L \) denotes the number of labels. Let \( l \) denotes the label index of the category of the normal query, we use the probability of this category, namely \( y_{i,l} \), as the predicted score for query selection. Given a threshold \( \delta \), the queries that meet the conditions \( y_{i,l} > \delta \) are fed to crowdsourcing for human validation, ordering by \( y_{i,l} \) reversely. If the number of selected queries exceed the daily crowdsourcing capacity, then the top queries with highest scores are selected for human validation.

Given a set of training data \( D = \{P_i, Y_i\}_{i=1}^N \), where \( P_i \) denotes the input token sequence, \( Y_i \) the label of the query and \( N \) the total number of training examples, the entire model is fine-tuned by minimizing the cross-entropy loss:

\[
\mathcal{L}_2(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^L I(j = Y_i) \log y_{i,j} \tag{6}
\]

where \( y_{i,j} \) denotes the \( j \)th element of \( y_i \), \( \theta \) the trainable parameters in the model and \( I(.) \) is an indicator function that returns 1 if the condition is true or 0 otherwise.

Algorithm 1 Dual learning for joint training query generation and query selection.

Require: a set of labeled data \( D_a = \{T_i, Q_i\}_{i=1}^{N_l} \), a set of unlabeled data \( D_b = \{T_i\}_{i=1}^{N_u} \), a set of validation data \( D_v = \{T_i, Q_i\}_{i=1}^{N_v} \) for validation, the number of epochs \( N_T \), the number of episode \( N_E \), a proportion \( \beta \) for sampling subset.

1. Initialize the model parameters \( \theta_{gen} \) and \( \theta_{sel} \) by the pre-trained language models.
2. for epoch from 1 to \( N_T \) do
3. Train the query generation model on \( D_a \) by:
   \( \theta_{gen} \leftarrow \theta_{gen} + \eta \nabla_{\theta_{gen}} \log P(Q|T; \theta_{gen}) \)
5. Predict on \( D_b \) with predicted scores \( z \).
6. Get a set of pseudo examples \( D_b = \{T_i, Q_i\}_{i=1}^{N_l} \).
7. Get \( \beta \% \) subset \( S \) from \( D_b \) by ranking \( z \).
8. for episode from 1 to \( N_E \) do
9. Divide \( S \) into a set of mini-batches \( B = \{B_1, ..., B_{N_b}\} \).
10. Shuffle \( B \).
11. for episode from 1 to \( N_E \) do
12. Select queries from \( B_k \), get selected instances \( B_k^{sel} \).
13. Train the query generation model on \( B_k^{sel} \).
14. Evaluate on validation set \( D_v \), get a sequence of the mean ROUGE scores \( R = \{R_1, ..., R_{N_k}\} \).
15. end for
16. Calculate a sequence of rewards \( \{r_k\}_{k=1}^{N_k} \) by Equation (8).
17. for \( k \) from 1 to \( N_k \) do
18. Train the query selection model by:
   \( \theta_{sel} \leftarrow \theta_{sel} + \eta \nabla_{\theta_{sel}} \log P(y_j|Q; \theta_{sel}) \)
19. end for
20. end for
21. \( D_a \leftarrow \bigcup_{k=1}^{N_k} B_k^{sel} \)
22. \( D_a \leftarrow D_a \cup D_v \)
23. \( D_b \leftarrow D_a \cup D_v \)
24. \( D_b \leftarrow D_v \setminus D_v \)
25. end for

4 THE DUAL LEARNING FRAMEWORK

In this section, we describe the detail of our proposed dual learning framework to jointly learn the query generation model and the query selection model, as illustrated in Figure 3. Given a labeled corpus \( D_a = \{T_i, Q_i\}_{i=1}^{N_l} \), an unlabeled corpus \( D_b = \{T_i\}_{i=1}^{N_u} \) for semi-supervised learning, and a validation corpus \( D_v = \{T_i, Q_i\}_{i=1}^{N_v} \), where \( T = (w_1^t, ..., w_m^t) \) denotes the token sequence within the title, \( Q = (w_1^q, ..., w_m^q) \) the token sequence within the annotated query, and \( N_l, N_u, N_v \) the number of records in the respective corpora, our framework first trains a query generation model using the labeled data \( D_a \). It then generates queries for the unlabeled titles in \( D_b \). The selection model then computes a score for each generated query. A subset of the predicted data with high predicted scores is selected to augment the existing labelled corpus. Thereafter, we concurrently train the generator and selector model in the subsequent training loop. Inspired by promising results of reinforcement learning [9, 43] for sequential decision-making process, we propose a reward based strategy to tune the query generation model and the evaluation scores from the generation model to tune the query
selection model. The entire process of our framework is formalized in Algorithm 1. Different from the work [43], our framework employs two agents (i.e. the generation and selection models) with different rewards, which are formally described below.

### 4.1 Reward for Query Generation Model

In general, the quality of the generated queries varies considerably. This could potentially lead to error accumulation during the training process. To address this limitation, we propose a reward based strategy to guide the learning process of the query generation model, such that examples with higher rewards contribute more during training. We define the rewards of original examples from the labeled corpus as $1.0$. For examples generated from the unlabeled corpus, the rewards are the corresponding scores predicted by the query selection model. Formally, the reward is defined as:

$$ r^\text{gen}_i = \begin{cases} 1.0 & \text{if } y_{i,l} \in A \\ y_{i,l} & \text{otherwise} \end{cases} $$

where $y_{i,l}$ denotes the predicted score of the $i^{th}$ example and $A$ the set of the indices of the original labeled examples in $D_a$.

### 4.2 Reward for Query Selection Model

In our framework, the query selection model aims to augment the training set with generated queries. We observe that the query generation model performs better when high quality queries are selected for training. Thus, we propose a reward strategy for tuning the training process of the query selection model based on the performance of the query generation model on the validation set $D_c$. However, the performance of the query generation model is evaluated after training on the entire dataset, which means that the reward feedback is hysteretic for the selection process. Following the work [43], we address this limitation by regarding the selection process as a sequential decision-making process, wherein the reward score is calculated for each batch of selected queries. For each batch, we train the query generation model using the selected examples and then evaluate on the validation set. We use the mean of ROUGE-n score (ROUGE-1, ROUGE-2 and ROUGE-l) as the evaluation metric. For each episode, we get a sequence of mean ROUGE scores, BLEU-n [19] scores and Exact Match (EM) score as the final score of the example.

The gradients of the expected reward $E[r]$ with respect to parameters $\theta_{\text{gen}}$ and $\theta_{\text{sel}}$ are formally defined as:

$$ \nabla_{\theta_{\text{gen}}} E[r^\text{gen}] = r^\text{gen} \nabla_{\theta_{\text{gen}}} \log P(Q|T; \theta_{\text{gen}}) $$

$$ \nabla_{\theta_{\text{sel}}} E[r^\text{sel}] = r^\text{sel} \nabla_{\theta_{\text{sel}}} \log P(y|Q; \theta_{\text{sel}}) $$

where $r^\text{gen}$ and $r^\text{sel}$ denote the reward of the query generation model and the query selection model, respectively. The parameters of the models are updated by

$$ \theta \leftarrow \theta + \eta \nabla_{\theta} E[r] $$

where $\eta$ denotes the learning rate and $\theta$ the parameters of the model. As the reward score of the query selection model are calculated for each batch of selected queries, following the work [43], the gradient is calculated by

$$ \nabla_{\theta_{\text{sel}}} E[r^\text{sel}] = \frac{r^\text{sel}}{||B_k||} \sum \nabla_{\theta_{\text{sel}}} \log P(y_j|Q; \theta_{\text{sel}}) $$

where $||B_k||$ denotes the number of examples in a batch, $r^\text{sel}_k$ the reward of the batch $B_k$ computed by Equation (8).

### 5 EXPERIMENTS

In this section, we describe the details of our experiments\(^2\). We conducted extensive experiments to verify the effectiveness of our approach, including offline/online evaluations and case study.

#### 5.1 Dataset and Evaluation Metrics

We created a 50K manually annotated dataset from a collection of document titles in Tencent QQ Browser from September to December, 2020. The original dataset is randomly split into a 40K training set, a 5K validation set and a 5K test set. Table 1 reports detailed statistical information on the datasets. For all datasets, a record consists of a title and some associated query, which is either a text span within the title or new query written by a human annotator. Note that for the training set, each title corresponds to only one query, while for the validation and test set, each title may be associated with up to three queries. Note also that the length of each annotated query is limited to 5-13 characters for better performance in the mobile app. We also collected an unlabeled corpus with 500K document titles from Tencent QQ Browser, which is used in our proposed dual learning framework for data augmentation.

For evaluating the performance of query selection, we collected a 10K test set from human evaluation by crowdsourcing.

For evaluation metrics, we used the well-know ROUGE-n [16] scores, BLEU-n [19] scores and Exact Match (EM) score as the performance metrics. If, in the validation or test set, a title has more than one annotated query, we average the scores over the queries as the final score of the example.

\(^2\)The data and the code of our implementations are available at: https://github.com/qikunxun/TitIE

| Table 1: Statistical information on datasets for training the query generation models. |
|---------------------------------|--------|--------|--------|
| | Train | Valid | Test  |
| Size | 40,000 | 5,000  | 5,000  |
| Avg # of tokens in titles | 23.25  | 23.84  | 22.74  |
| Avg # of tokens in queries | 9.56   | 10.01  | 10.03  |
| Avg # of queries for each title | 1.00   | 1.16   | 1.16   |
| # of queries are spans of titles | 38,866 | 3937   | 3944   |
### Table 2: Comparison results for query generation on the test set.

<table>
<thead>
<tr>
<th>Models</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-1</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2seq</td>
<td>55.0</td>
<td>35.2</td>
<td>53.2</td>
<td>49.3</td>
<td>38.7</td>
<td>30.2</td>
<td>24.2</td>
<td>3.9</td>
</tr>
<tr>
<td>Transformer</td>
<td>57.1</td>
<td>47.7</td>
<td>56.8</td>
<td>51.7</td>
<td>46.9</td>
<td>42.3</td>
<td>38.1</td>
<td>4.3</td>
</tr>
<tr>
<td>PGN</td>
<td>65.0</td>
<td>59.2</td>
<td>64.7</td>
<td>59.1</td>
<td>56.0</td>
<td>53.1</td>
<td>50.3</td>
<td>12.1</td>
</tr>
<tr>
<td>QEBERT-1K</td>
<td>72.6</td>
<td>70.7</td>
<td>72.6</td>
<td>69.5</td>
<td>68.6</td>
<td>67.4</td>
<td>66.3</td>
<td>51.0</td>
</tr>
<tr>
<td>QEBERT-10K</td>
<td>80.9</td>
<td>78.8</td>
<td>80.9</td>
<td>77.9</td>
<td>76.9</td>
<td>75.6</td>
<td>74.3</td>
<td>58.5</td>
</tr>
<tr>
<td>QEBERT-40K</td>
<td>83.0</td>
<td>80.8</td>
<td>82.9</td>
<td>79.8</td>
<td>78.7</td>
<td>77.4</td>
<td>76.2</td>
<td>59.9</td>
</tr>
<tr>
<td>QEBERT-40K†</td>
<td>82.8</td>
<td>80.5</td>
<td>82.6</td>
<td>79.6</td>
<td>78.5</td>
<td>77.2</td>
<td>75.9</td>
<td>59.5</td>
</tr>
<tr>
<td>mT5-1K</td>
<td>69.6</td>
<td>63.5</td>
<td>69.6</td>
<td>64.7</td>
<td>61.9</td>
<td>58.9</td>
<td>55.8</td>
<td>28.2</td>
</tr>
<tr>
<td>mT5-10K</td>
<td>80.0</td>
<td>76.8</td>
<td>80.0</td>
<td>76.0</td>
<td>74.6</td>
<td>72.9</td>
<td>71.1</td>
<td>49.8</td>
</tr>
<tr>
<td>mT5-40K</td>
<td>84.2</td>
<td>81.2</td>
<td>84.1</td>
<td>81.0</td>
<td>79.6</td>
<td>77.9</td>
<td>76.2</td>
<td>56.8</td>
</tr>
<tr>
<td>mT5-40K†</td>
<td>84.5</td>
<td>81.4</td>
<td>84.4</td>
<td>81.3</td>
<td>79.7</td>
<td>78.1</td>
<td>76.3</td>
<td>56.5</td>
</tr>
<tr>
<td>DL-QEBERT-1K</td>
<td>75.3</td>
<td>73.1</td>
<td>75.2</td>
<td>72.1</td>
<td>71.0</td>
<td>69.8</td>
<td>68.5</td>
<td>52.2</td>
</tr>
<tr>
<td>DL-QEBERT-10K</td>
<td>82.6</td>
<td>80.3</td>
<td>82.5</td>
<td>79.5</td>
<td>78.4</td>
<td>77.1</td>
<td>75.7</td>
<td>59.3</td>
</tr>
<tr>
<td>DL-QEBERT-40K</td>
<td>84.6</td>
<td>82.3</td>
<td>84.5</td>
<td>81.3</td>
<td>80.2</td>
<td>78.8</td>
<td>77.5</td>
<td>60.5</td>
</tr>
<tr>
<td>DL-mT5-1K</td>
<td>71.8</td>
<td>66.5</td>
<td>71.8</td>
<td>67.1</td>
<td>64.6</td>
<td>61.8</td>
<td>58.9</td>
<td>31.5</td>
</tr>
<tr>
<td>DL-mT5-10K</td>
<td>82.1</td>
<td>78.9</td>
<td>82.0</td>
<td>78.8</td>
<td>77.2</td>
<td>75.4</td>
<td>73.5</td>
<td>53.0</td>
</tr>
<tr>
<td>DL-mT5-40K</td>
<td>85.3</td>
<td>82.5</td>
<td>85.2</td>
<td>82.0</td>
<td>80.7</td>
<td>79.2</td>
<td>77.5</td>
<td>58.6</td>
</tr>
</tbody>
</table>

### 5.2 Experimental Settings

We implemented our models in Tensorflow 1.14.0 and trained all the models with a single Tesla P40 GPU. We applied our proposed dual learning framework on both the extractive generation model QEBERT and the generative generation model mT5. We call the enhanced models DL-QEBERT and DL-mT5, respectively.

For DL-QEBERT, the lexical encoder was initialized by the Chinese pre-trained BERT-base model with 12 transformer layers, which yields output token embeddings with a dimension of 768. The transformer encoder was built with 12 heads. DL-BERT was trained by Adam [13] with 5 training epochs. The initial learning rate was set as 1e-4, the mini-batch size as 128 and the dropout [27] rate as 0.1. Note that only the queries that are text spans of the titles are used for training the QEBERT model.

For DL-mT5, the entire model was initialized by the pre-trained mT5-base model with 12 transformer layers and a hidden dimension of 768. The transformer encoder and the transformer decoder were built with 12 heads. DL-mT5 was also trained by Adam [13] with 8 training epochs. The initial learning rate was set as 2e-4, the mini-batch size as 128 and the dropout [27] rate as 0.1.

For above models, we set the hyper-parameter $N_T$ as 3, $N_E$ as 3, $\beta$ as 10 according to the performance on the development set. Every input title is truncated to 64 tokens and every query is truncated to 13 tokens. Inspired by the well-known policy network pre-training strategy [4, 21, 43] widely used in reinforcement learning, we also pre-trained the QSBERT model on a 100k online data annotated by crowdsourcing.

### 5.3 Baseline Models

We compared our proposed framework against the following baseline models. The hidden dimension was set as 200 and dropout rate was set as 0.1 for all the baseline models. All the baseline models are trained by the entire 40K training set.

### 5.4 Offline Evaluation

Following the previous work [11] for evaluating text summarization, we carried out two forms of offline evaluation, an automatic evaluation and a human evaluation.

**Automatic Evaluation.** Table 2 reports that the comparison results between the baseline models and the variants of our proposed models on the test set. Every value is the test score in percentage. The models QEBERT-40K and mT5-40K are the QEBERT model and mT5 model trained on the entire 40K training set, respectively. QEBERT-10K and mT5-10K are the respective models trained on a 10K subset sampled from the full training set, and QEBERT-1K and mT5-1K.

### Table 3: Results for offline evaluation on query selection.

<table>
<thead>
<tr>
<th>Models</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>QSBERT</td>
<td>75.1</td>
<td>82.0</td>
<td>78.4</td>
</tr>
<tr>
<td>DL-QSBERT</td>
<td>78.3</td>
<td>79.4</td>
<td>78.8</td>
</tr>
</tbody>
</table>

**Seq2seq** [28] is the original sequence-to-sequence generative model. We implement this baseline with a Bi-LSTM [10] encoder and a Uni-LSTM decoder.

**Transformer** [31] is the first sequence transduction model based entirely on attention mechanism. It is widely used for pre-trained language models [5, 22, 41, 44].

**PGN** [24] (short for Pointer Generator Networks) is an attention-based seq2seq model which not only generates text using the target vocabulary but also can copy words from the source text.

**QEBERT-40K†** and **mT5-40K†** are the QEBERT model and the mT5 model that are trained by the well-known self-training [23] framework with the same labeled and unlabeled data used in DL-QEBERT-40K and DL-mT5-40K, respectively. They serve as ablation comparisons to help verify the effectiveness of the proposed dual learning framework.
mT5-1K refer to the models trained on a sampled 1K subset. Results show that our proposed models outperform all the baseline models in all the metrics. The performance difference between DL-mT5-40K and mT5-40K is statistically significant with p-value $2.2e-6 < 0.001$ by a two-tailed t-test. The performance difference between DL-QEBERT-40K and QEBERT-40K is statistically significant with p-value $7.0e-6 < 0.001$ by a two-tailed t-test. These results demonstrate the proposed dual learning framework improves the performances of the original QEBERT and mT5 models by a significant margin. To verify the effectiveness of our proposed dual learning framework, we also conducted ablation comparison by training the QEBERT and mT5 models under the self-training framework [23] with the same unlabeled corpus used in the dual learning framework, as shown by QEBERT-40K$^\ddagger$ and mT5-40K$^\ddagger$. Results show that models trained under our proposed dual learning framework achieves better performance than those trained by the self-training framework. The performance difference between DL-mT5-40K and mT5-40K$^\ddagger$ is statistically significant with p-value $2.0e-4 < 0.001$ by a two-tailed t-test. The performance difference between DL-QEBERT-40K and QEBERT-40K$^\ddagger$ is statistically significant with p-value $6.8e-7 < 0.001$ by a two-tailed t-test. These results indicate that our proposed dual learning framework is more effective for enhancing the performance of query generation. In addition, we also found that the proposed dual learning framework pushes the original models by a higher absolute gain in performance when the labeled corpus is limited, as shown in the comparisons between DL-QEBERT-1K and QEBERT-1K as well as DL-mT5-1K and mT5-1K. It implies that the proposed framework is more suitable for the scenarios with limited labeled data for training.

Table 3 reports that the comparison results between the QSBERT model and the DL-QSBERT on a 10K test set for evaluating query selection. Every value is the test score in percentage. The DL-QSBERT model achieves a precision score of 78.3% on the test set, yielding an absolute gain of 3.2% over QSBERT. As mentioned above, the crowdsourcing resources have a daily usage limit (about 100K queries are evaluated per day). We hope that the selected queries are more likely to be accepted by human evaluation. Thus we focus more on the precision score of the query selection model. The absolute gain in precision score indicates that the DL-QSBERT model is better able to identify potential acceptable queries.

**Human Evaluation.** To evaluate the linguistic quality of the generated queries, we carried out a human evaluation, as shown in Figure 4. For human evaluation, we focused on two main aspects, the fluency and the relevance. The fluency indicator focuses on whether the generated queries are well-formed and grammatically correct while the relevance indicator reflects whether the generated queries cover the salient information within the document titles. Following the previous work [11, 12], we sampled 300 records from the test set and employ two crowdsourcing annotators to inspect the generated query. Each generated query is rated on a scale of 0 (worst) to 2 (best). Each annotator evaluates the generated queries from all of the different models. The final scores are then averaged across the two annotators. Results show that our proposed models outperform all baseline models. The proposed dual learning framework pushes mT5 by relative gains of 6.8% on fluency score and 6.1% on relevance score, pushing QEBERT by relative gains of 7.5% on fluency score and 0.6% on relevance score. These results show that our proposed models are better able to generate valuable queries and the proposed dual learning framework significantly improve the fluency and the relevance for both the mT5 and QEBERT models.

### 5.5 Online Evaluation

We performed large-scale online evaluations to show that the TitIE system helps with maximizing the utilization of crowdsourcing resources and improving the performance of QFR in real-world applications. The online evaluation is two-fold. First, we evaluated the difference in acceptance rate between TitIE and its variants to verify that TitIE significantly improves the online acceptance rate of the generated queries. Second, we evaluated the performance of the recommendation system through large-scale A/B Testing.

**Online Acceptance Rate Evaluation.** Figure 5 reports the average acceptance rate of different systems from online data collected over three days. For each day, 10K generated queries are manually validated by crowdsourcing. Results show that our proposed models DL-QEBERT and DL-mT5 increases the acceptance rate of the baseline models by more than 12.9% (calculated by $(33.7% + 35.6%)/2 - \left(11.1% + 23.0% + 24.3% + 28.7%\right)/4$) on average. Moreover, the acceptance rate increases by 23% when the generated queries were pre-selected by the DL-QSBERT model. Further improvement is attained when the DL-QEBERT model and the DL-mT5 model
are simultaneously deployed in our system, achieving 61.7% average acceptance rate, yielding an absolute gain of 33.0% over the queries mined from user search logs. These results implies that the proposed query generation models are able to generate queries with higher qualities while the proposed query selection strategy significantly improves the acceptance rate in human validation.

Online A/B Testing. We conducted a large-scale online A/B Testing to show that whether the TitIE system helps to improve the performance of QFR. For online A/B Testing, we split users into several buckets, where each bucket contains about 1,200,000 users. Note that both the users and queries are labeled with user profiles and query profiles in our recommender system. We ranked candidate queries by a Click-Through Rate (CTR) prediction model [17], where the candidate queries are retrieved by matching users based on their profile. The ranked queries are recommended to users in the QFR scenario of Tencent QQ Browser. We observed the activities of each bucket for 3 days based on the following metrics:

- **Impression Page View (IPV):** number of times that recommended queries are viewed by users.
- **Impression User View (IUV):** number of users who has viewed the recommended queries.
- **Click Page View (CPV):** number of times that recommended queries are clicked by users.
- **Click User View (CUV):** number of users who has clicked the recommended queries.
- **Click Through Rate on Page View (CTR-PV):** \( \frac{CPV}{IPV} \)
- **Click Through Rate on User View (CTR-UV):** \( \frac{CUV}{IUV} \)

The CTR-UV and CTR-PV metrics are the most important metrics as they show how many queries are attractive to users in the recommender stream. We select two buckets with the most similar activities for A/B testing to ensure that there is no statistical difference between the users in the two buckets. We then ran the A/B testing for 7 days and compared the results by above metrics. For bucket A (i.e., the experiment group), the queries generated from TitIE (about 5 million) are used for recommendation. For bucket B (i.e., the baseline group), the queries mined from user search logs (about 5 million) are used for recommendation.

Table 4 reports the results of the online A/B testing for the above metrics. Results show that the difference between the IPV and IUV metrics reveal no statistical significance (p-value > 0.05) while the CPV, CUV, CTR-PV and CTR-UV metrics increased significantly (p-value < 0.001) when the generated queries from the TitIE system are employed for recommendation, by 27.18%, 23.01%, 26.89% and 22.86%, respectively. These results reveal that the impression difference between bucket A and B is virtually negligible while the queries generated from the TitIE system are more attractive and interesting to users, yielding to higher CTR scores.

### 5.6 Case study

We conducted a simple case study on the generated queries from different sources, as shown in Table 5. For the query mined from user search logs, it is presented as 3 keywords (“英菲尼迪(Infiniti)”, “q50l”, “刹车(brake)”) split by space. We speculate that this form of queries are less attractive to users (supported by the results of A/B Testing). For the input title “NASA照片显示火星异常结构，轮廊清晰 (Photos from NASA show the abnormal structure of Mars with clear outline)”, the generated query from DL-QEBERT is more fluent and informative than that from QEBERT. For the input title “哪個企業和奇瑞相比,誰的技術更好?(Which is better for BYD and Chery, whose technology is better?)”, the generated query from DL-mT5 is more relevant to the title than that from mT5. We believe that these enhancements are attributed to the reward feedback in our proposed dual learning framework.

### 6 CONCLUSIONS

In this paper, we propose TitIE, a query mining system for providing valuable queries for query feeds recommendation at Tencent QQ Browser. A query extraction model based on pre-trained language model BERT and a query generation model based on pre-trained text generation model mT5 are proposed in TitIE for generating valuable queries from article titles. The generated queries are then selected by a proposed query selection model which is able to select the queries that are likely to be accepted during human validation. A novel dual learning framework is proposed to jointly learn the query generation models and the query selection model, thereby mutually improving the performance of both. We conduct both offline and online evaluations to assess the effectiveness of our proposed framework. For offline evaluation, results from the automatic and human evaluations show that the proposed dual learning framework yields significant improvements for both the query extraction and generation models. For online evaluation, results from the acceptance rate evaluation and the online A/B Testing show that the TitIE system helps to improve the acceptance rate of human validation and the performance of query feeds recommendation.

### ACKNOWLEDGMENTS

We thank Hai Wan for helpful discussions on the paper. This work is supported by the National Natural Science Foundation of China (No. 61976232 and 61876204).