Query-aware Tip Generation for Vertical Search

Yang Yang[1] , Junmei Hao[2], Canjia Li[3], Zili Wang[4], Jingang Wang[1], Fuzheng Zhang[1], Rao Fu[1],
Peixu Hou[1], Gong Zhang[1], and Zhongyuan Wang[1]*
[1]Meituan [2]Xi'an Jiaotong University [3]University of Chinese Academy of Sciences [4]Xidian University
{yangyang113, wangjingang02, zhangfuzheng, furao02,wangzhongyuan02}@meituan.com,
{peixu.hou, gong.zhang}@dianping.com,
haojunmei1996@stu.xjtu.edu.cn, licanjia17@mails.ucas.ac.cn, ziliwang.do@gmail.com

ABSTRACT

As a concise form of user reviews, tips have unique advantages to explain the search results, assist users' decision making, and further improve user experience in vertical search scenarios. Existing work on tip generation does not take query into consideration, which limits the impact of tips in search scenarios. To address this issue, this paper proposes a query-aware tip generation framework, integrating query information into encoding and subsequent decoding processes. Two specific adaptations of Transformer and Recurrent Neural Network (RNN) are proposed. For Transformer, the query impact is incorporated into the self-attention computation of both the encoder and the decoder. As for RNN, the query-aware encoder adopts a selective network to distill query-relevant information from the review, while the query-aware decoder integrates the query information into the attention computation during decoding. The framework consistently outperforms the competing methods on both public and real-world industrial datasets. Last but not least, online deployment experiments on Dianping demonstrate the advantage of the proposed framework for tip generation as well as its online business values.

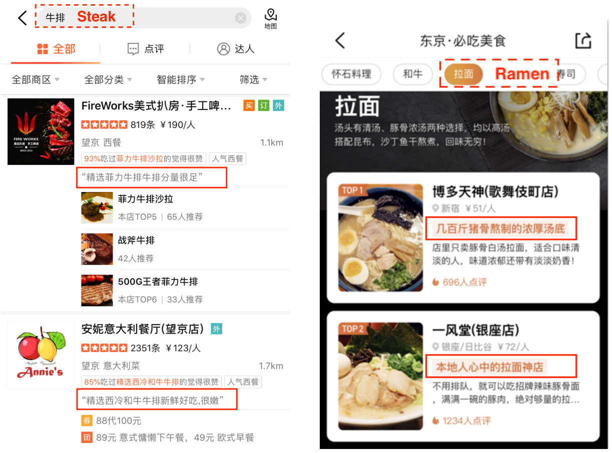
KEYWORDS

Abstractive Tip Generation; Query-aware Generation; Vertical E-commerce Search

ACM Reference Format:
Yang Yang[1] , Junmei Hao[2], Canjia Li[3], Zili Wang[4], Jingang Wang[1], Fuzheng Zhang[1], Rao Fu[1], Peixu Hou[1], Gong Zhang[1], and Zhongyuan Wang[1]. 2020. Query-aware Tip Generation for Vertical Search. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3340531.3412740

1 INTRODUCTION

In generic web search, given a user query, the search engines return a list of relevant documents and usually present title-snippet pairs to users. While in vertical searches, such as Yelp[1] and Dianping[2], to better assist users' decision making, the vertical search engine usually presents some information in addition to search results.



FireWorks美式扒房·手工啤...
819条 ￥190/人
望京 西餐 1.1km
93%吃过菲力牛排沙拉的觉得很赞 人气西餐
"精选菲力牛排牛排分量很足"
菲力牛排沙拉
本店TOP5 | 65人推荐
战斧牛排
42人推荐
500G王者菲力牛排
本店TOP6 | 33人推荐
安妮意大利餐厅(望京店) 外
2351条 ￥123/人
望京 意大利菜 1.7km
85%吃过精选西冷和牛牛排的觉得很赞 人气西餐
"精选西冷和牛牛排新鲜好吃,很嫩"
券 88代100元
团 89元 意式慵懒下午餐，49元 欧式早餐

拉面
汤头有清汤、豚骨浓汤两种选择，均以高汤搭配昆布，沙丁鱼干熬煮，回味无穷！
博多天神(歌舞伎町店)
新宿 ￥51/人
几百斤猪骨熬制的浓厚汤底
店里只卖豚骨白汤拉面，适合口味清淡的人，味道浓郁还带有淡淡奶香！
696人点评
一风堂(银座店)
银座/日比谷 ￥72/人
本地人心中的拉面神店
不用排队，就可以吃招牌辣味豚骨面，满满一碗的豚肉，绝对够量的拉...
1234人点评

(a) Search result page of "Steak". The upper tip is "*The selected filet steak is enough served.*", and the bottom one is "*The selected Japanese sirloin steak is fresh, good taste and tender.*".

(b) An selected channel about "Ramen". The upper tip is "*Pleasant smell and tasty soup made from pork bones*", and the bottom one is "*The best Ramen rated by natives.*".

Figure 1: Two vertical search scenarios on Dianping App. The queries are boxed with dash lines and the tips are boxed with solid lines.

Figure 1 demonstrates two typical vertical search scenarios on Dianping, a popular E-commerce application in China. Figure 1a presents the top 2 restaurants (also known as place of interests (POI)) in the search result page (SRP) of the query "Steak". The SRP not only presents essential information such as price and user ratings of returned restaurants but also provides some one-sentence tips (boxed with red solid lines). Figure 1b shows a selected channel with tips after users clicking the topic "Ramen". The displaying

*Yang Yang, Junmei Hao, Canjia Li and Zili Wang are equal contributions, and Jingang Wang is the corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10...$15.00
https://doi.org/10.1145/3340531.3412740

[1] https://www.yelp.com/
[2] https://www.dianping.com/, a leading vertical service platform in China, whose content including food, entertainment, and travel services, etc.

topics can be treated as potential user queries. These tips, usually as compact and concise feature highlights of the listed POIs, are especially valuable for users to get a quick insight over the search results. Moreover, tips can be utilized to provide fine-grained and more reliable search explanations to help consumers making more informed decisions.

Obviously, it is impractical to manually write tips for millions of POIs indexed by vertical search platforms. Fortunately, large amounts of user-generated reviews have been accumulated for these POIs. Hence it is intuitive to distill relevant information from reviews as tips. Based on user reviews, earlier work generate tips for POIs with natural language generation techniques, such as unsupervised extractive [32, 37] and abstractive methods [11, 12]. As effective as they are, these tips are to some extent sub-optimal as they are generated without taking the user queries into consideration.

This motivates us to focus on producing tips by harnessing both the user query and the POI's reviews. Such query-aware tips can potentially answer user's intent and attract users' attention better than a query-agnostic alternative. For example, given a query "Coffee Latte", a tip "The vanilla latte tastes great!" is more informative than a tip "I love the bubble tea." from the view of user experience.

To this end, we propose query-aware tip generation for vertical search. There are two popular architectures for encoder-decoder framework, *i.e.*, Transformer [28] and RNN [6, 25]. Accordingly, we develop query-aware tip generation encoders and decoders based on them, respectively.

Query-aware Encoder (Qa_Enc). For Transformer, we incorporate the query representation into the self-attention computation. For RNN, we introduce a selective gate network in the encoder to distill query-relevant information from the input sequence.

Query-aware Decoder (Qa_Dec). For Transformer, we similarly incorporate the query representation into the self-attention computation as the encoder. For RNN, we improve the attention mechanism by integrating query representation into the context vector to better direct the decoder.

To the best of our knowledge, this is the first work focusing on query-aware tip generation for vertical e-commerce search. The main contributions can be summarized as follows:

- We propose a query-aware tip generation framework, which is intuitive but effective in vertical search scenarios.
- We introduce query-aware encoders and decoders to enhance the encoder-decoder framework to produce query-aware tips from user reviews, based on Transformer and RNN.
- We evaluate our framework on both public and real-world industrial datasets. Extensive experimental results indicate the effectiveness of our framework. We have also deployed our method in a real-world e-commerce platform and observed better performance than the competing baseline models.

2 QUERY-AWARE TIP GENERATION FRAMEWORK

This section introduces the proposed query-aware encoder-decoder framework in detail. Similar to seq2seq text generation, the objective of query-aware tip generation is to generate a concise tip given a piece of review, except that there exists auxiliary information, i.e., a user query. There are two popular neural network architectures for encoders and decoders, i.e., the Transformer and the RNN. Both of the two architectures are adapted to involve query information. Specifically, the query can be utilized in the encoder and the decoder separately or jointly.

2.1 Problem Formulation

Given a user review $\mathcal{R} = (r_1, r_2, \cdots, r_N)$ of N words, a tip generation system aims to generate a compact tip of length M, namely $\mathcal{T} = (t_1, t_2, \cdots, t_M)$, which is also relevant to user query $\mathbf{Q} = (q_1, q_2, \cdots, q_K)$ of K query words.

2.2 Transformer-based Adaptation

Here we give a brief description of the more recent and arguably more superior Transformer text generation framework. Basically, the Transformer model first projects the tokens in a sequence of length n into the d-dimension embedding space, where these token embeddings are again projected into three different spaces, namely $Q, K, V \in \mathbb{R}^{n \times d}$, via three different projection matrices. Afterwards, the contextualized representations of the entire sequence are computed by multi-head scaled dot-product attention layer:

$$\begin{aligned} \text{Attention}(Q, K, V) &= \text{softmax}(\frac{QK^T}{\sqrt{d}})V \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\ \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \end{aligned} \tag{1}$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d \times n}$, $W_i^K \in \mathbb{R}^{d \times n}$, $W_i^V \in \mathbb{R}^{d \times n}$ and $W^O \in \mathbb{R}^{hd \times d}$, and h denotes the number of the attention heads. In the Transformer, the Q, K, V are from the embeddings of the same input sequence, hence the attention layer is revised to self-attention layer, which helps the Transformer to summarize other words of the input sentence at the current position. The outputs of the self-attention layer are fed to the layer-normalization and position-wise feed-forward neural network. The Transformer encoder block can be stacked one by one to obtain the abstract representation of each token. When the Transformer is applied to the text generation task, the outputs of the last encoder block are taken as key and value weights for decoding. Besides the self-attention and feed-forward layers, the decoder block is also equipped with an encoder-decoder attention layer in between to focus on the relevant parts of the input sequence.

The above mechanism is modified in our proposed Transformer-adapted framework, which consists of three components: (1) a review-aware query encoder, (2) a query-aware review encoder, and (3) a query-aware tip decoder.

Review-aware Query Encoder. The user query represents a condensed information need, while the review contains more detailed information for the POI. The heterogeneity makes information sharing and matching difficult. To bridge this semantic gap, we first introduce a review-aware query encoder to represent the query. More concretely, the dot product attention layer (i.e., self-attention layer in Transformer) is adjusted to allow the interaction between the user query and the review.

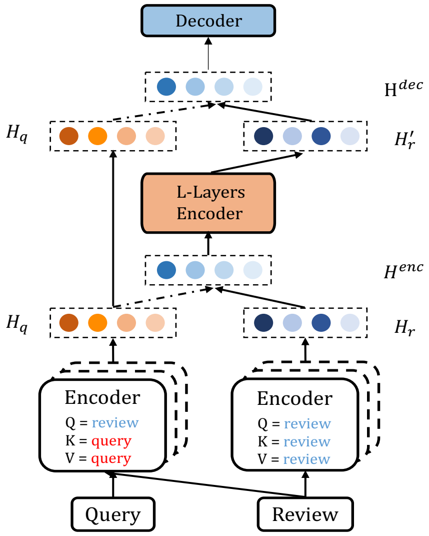


Figure 2: The Transformer-based query-aware tip generation framework. The left encoder encodes review-aware query while the right encoder maintains a self-attention manner to encode review. The two dotted arrows indicate the query information flows to the encoder and the decoder respectively.

Given a query Q and a review $\mathcal{R}$, we first use $E_q = \text{Emb}(Q)$ and $E_r = \text{Emb}(\mathcal{R})$ to represent their word embeddings. Then the review contextualized information was incorporated into query representation. Formally, the dot product attention layer in Equation 1 can be modified as: $H = \text{MultiHead}(E_r, E_q, E_q)$ $H \in \mathbb{R}^{N \times d}$ is then fed into the feed-forward and layer normalization layers as in the Transformer. The output of the query encoder is represented as $H_q \in \mathbb{R}^{N \times d}$. The output of the query encoder can be adapted into either review encoding or tip decoding in the tailored Transformer-based framework.

Query-aware Review Encoder. Similarly, the contextualized representations of the review $\mathcal{R}$ is obtained by setting $Q, K, V = E_r$. The output is denoted as $H_r \in \mathbb{R}^{N \times d}$.

Intuitively, H_r encodes the user review's sequential representations. H_q digests the query information under the context of the review. To combine these two complementary representations, a feed-forward network layer is adopted, namely, $H^{\text{enc}} = [H_q; H_r]W$, where $W \in \mathbb{R}^{2d \times d}$, $H^{\text{enc}} \in \mathbb{R}^{N \times d}$. Subsequently, H^{enc} is fed into several Transformer encoder layers to further extract the query-aware representation for each token.

Query-aware Tip Decoder. When adapting query into tip decoding, the review and query are met at the early stage and composed by stacked Transformer encoder layers. To distinguish these two information flows and encourage the decoder to obtain the query message directly during decoding, stacked Transformer encoder layers are only applied to review hidden representation H_r to obtain a deeper contextualized representation H'_r. During decoding, H'_r and H_q are combined to generate the key and value matrices, namely, $H^{dec} = [H_q; H'_r]W$, where $H^{dec} \in \mathbb{R}^{N \times d}$.

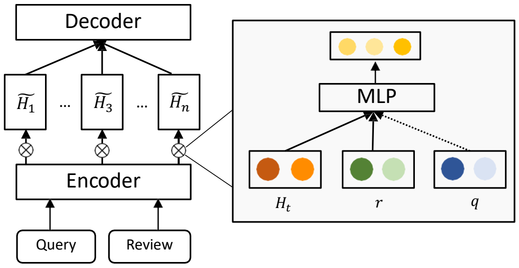


Figure 3: The RNN-based query-aware tip generation framework.

The above encoding and decoding mechanisms are illustrated in Figure 2. Please note that H^{dec} is used as the key and value matrix for decoder layers during decoding.

2.3 RNN-based Adaptation

Besides the Transformer, another family of text generation model is the RNN-based encoder-decoder network. Such models can be integrated into our tip generation framework conveniently. The RNN-adapted framework also consists of three components, i.e., (1)a query encoder, (2)a query-aware review encoder, and (3)a query-aware tip decoder.

Query Encoder. A Bidirectional Long Short-Term Memory (Bi-LSTM) [35] is adopted to generate a hidden representation $h_q \in \mathbb{R}^{2d}$ of the user query, where h_q is the concatenation of the last hidden state of the forward pass and the first hidden state of the backward pass.

Query-aware Review Encoder. Similarly, another Bi-LSTM generates the hidden states $H \in \mathbb{R}^{N \times 2d}$ of all input review tokens. Afterwards, the review hidden representation $h_r \in \mathbb{R}^{2d}$ is obtained in the same way as h_q.

At each time step t, to distill the query-relevant information from the review sequence, a selective gate [36], denoted as g_t, is calculated as follows.

$$g_t = \sigma\left(W_r\left[H_t; h_r\right] + W_q h_q + b_g\right) \tag{2}$$

where $W_r \in \mathbb{R}^{2d \times 4d}$, $W_q \in \mathbb{R}^{2d \times 2d}$ and $b_g \in \mathbb{R}^{2d}$ are learnable parameters, and σ is sigmoid activation function. Thereby, the hidden representation of each time step t is updated as

$$\tilde{H}_t = g_t \circ H_t \tag{3}$$

where $\circ$ is an element-wise multiplication. Such a gate acts like a "soft" selector selecting those tokens in the review that are relevant to the query.

Query-aware Tip Decoder. During decoding, the unidirectional LSTM decoder receives the output token from last step and has a hidden state s_t. To encourage the decoder to generate a word that is relevant to the query, an intuitive but effective approach is to introduce the query representation to guide the decoding process, i.e., the attention a_t over different input tokens. To implement it, at each decoding step t, we have:

$$\begin{aligned} c_i &= W_c[\tilde{H}_i; s_t] + W_q h_q + b_c && (4) \\ a_i^t &= \text{softmax}\left(v^T \tanh(c_i)\right) && (5) \end{aligned}$$

where $W_q \in \mathbb{R}^{d \times 2d}$, $W_c \in \mathbb{R}^{d \times 3d}$, $\mathrm{b}_c \in \mathbb{R}^d$, $\mathrm{v} \in \mathbb{R}^d$. The attention distribution a^t varies at each time step, assuring the model to leverage the query and decoder hidden state into the contextual vector computation. The resulting time-dependent encoder hidden state is calculated as the following:

$$h_t^{\text{dec}} = \sum_i a_i^t \tilde{H}_i \qquad (6)$$

which is used for subsequent decoding. The above encoding and decoding mechanisms are illustrated in Figure 3.

2.4 Model Training

At each time step t, the output of the decoder is denoted as $h_t^d \in \mathbb{R}^d$. The soft-max function is adopted to normalize the distribution.

$$P_t = \text{softmax}(W_v h_t^d) \qquad (7)$$

where $W_v \in \mathbb{R}^{V \times d}$ maps the hidden state into a V-dimension vocabulary space. P_t provides a final normalized distribution for token prediction. During training, the negative log likelihood loss for each training sample is defined as follows:

$$\mathcal{L} = -\frac{1}{M} \sum_{t=1}^{M} P_t(y_t) \qquad (8)$$

where y_t is the index of ground-truth.

3 DATASETS

To evaluate the effectiveness of the proposed query-aware encoders and decoders thoroughly, we conduct an extensive set of experiments on two datasets, Debate and Dianping.

3.1 Debate

The first dataset used is an open-source query-based English summarization dataset [17], denoted as **Debate** for simplicity. The dataset is created from Debatepedia[3], an encyclopedia of pro and con arguments and quotes on critical debate topics. The queries associated with the topic, the set of documents and an abstractive summary associated with each query which is not extracted directly from the document are crawled from Debatepedia. The dataset[4] includes 13, 719 (*query, document, summary*) triplets in total[5]. Table 1 presents an example.

Table 1: A (document, query, summary) example in Debate.

Document	The "natural death" alternative to euthanasia is not keeping someone alive via life support until they die on life support. That would, indeed, be unnatural. The natural alternative is, instead, to allow them to die off of life support.
Query	"non-treatment" : Is euthanasia better than withdrawing life support?
Summary	The alternative to euthanasia is a natural death without life support.

3.2 Dianping

Due to the lack of public tip generation datasets for e-commerce search, we create an in-house dataset by crawling the search log of the Dianping App. Dianping is a leading Chinese vertical E-commerce platform where customers can write reviews for POIs such as restaurants, hotels, etc. The dataset is denoted as **Dianping** for simplicity. Currently, existing tips can be categorized as manually-extractive, manually-abstractive, and template-based (e.g., *The best [placeholder] in town.*). As their names suggest, these tips are created by human experts or manual rules, normally based on concrete reviews, and organized in a one (POI) to many (tips) manner, respectively. These tips can be presented in various scenarios. For instance, for query-agnostic recommendation, when a POI is recommended to the user, one of its tips is randomly selected. In scenarios involving queries, to enhance the user experience, Dianping selects a tip that is the most similar to the query, as shown in Figure 1. We collected query logs with such query-aware tips for our experiments.

For a query-aware tip, we associate it with its review as follows. If a tip is created from a specific review, the original review is simply retrieved. For tips such as template-based ones, or those whose original review is unidentifiable, we would use the tip itself to retrieve the most relevant review from the reviews related to the POI. The relevance score is calculated with the BM25 weighting method [19]. If the relevance score is lower than a pre-defined threshold, which means there does not exist a potential informative review to produce the tip, the tip is removed from our dataset. To facilitate the training process, some additional pre-processing is performed. Only reviews containing 100 ∼ 150 Chinese characters are kept. The tips containing more than 15 characters are also filtered out, as they cannot be displayed properly on a mobile phone screen.

We processed the search log spanning from July 1st to August 1st, 2019. Finally, the corpus is composed of 224, 730 (*POI, Review, Query, Tip*) tuples and the last three fields are used in our experiments. Figure 4 presents an example. Note that the words "*candlelight dinner*" and "*stars*" in the tip are all relevant to the query word "*romance*" in Chinese culture.

The statistics of datasets are listed in table 2. These tuples of both datasets are further randomly split into approximately 80% for training, 10% for validation and 10% for testing. Note that the queries (questions) in Debate and Dianping vary a lot. Moreover, to demonstrate the necessity of queries, two baselines without regard to queries are established in experiments. Hence, for these baselines, we remove queries from both datasets and then remove duplicates, resulting in their query-agnostic counterparts.

4 EXPERIMENTS

4.1 Implementation Details

In Transformer-based framework, all the Transformer variants are implemented as 6 identical layers with a hidden size of 512. In

[3] http://www.debatepedia.org/en/index.php/Welcome_to_Debatepedia%21

[4] https://github.com/PrekshaNema25/DiverstiyBasedAttentionMechanism

[5] Note that the quantities of triplets released on Github is not consistent as depicted in [17]

POI	J Prime牛排海鲜餐厅	J Prime Steak & Seafood Restaurant
Review	果然是顶级牛排，口味超赞！前餐例汤也很有特色，无酒精鸡尾酒很解腻服务非常周到，每道菜细心讲解，吃的放心，环境也特别美，星空下烛光晚餐感觉棒极了朋友很满意！	It was indeed a top-grade steak, and its taste was superb! The soup before the meal was very special. The non-alcoholic cocktail really rid myself of the greasy feeling. The service was very satisfactory, and each dish was explained carefully. The environment is also very beautiful, the candlelight dinner under the stars is fantastic, my friends are very satisfied!
Query	浪漫 晚餐	Romance Dinner
Tip	星空下烛光晚餐感觉棒极了	The candlelight dinner under the stars is fantastic

Figure 4: A (POI, review, query, tip) tuple example in our corpus. The column 3 is the translation of the column 2.

Table 2: Statistics of both datasets. The average length are calculated by words in English and characters in Chinese respectively. 'w/o' represents without query while 'w' means with query.

Dataset	Avg_Len			Query	Train	Valid	Test
	Review (Doc)	Query	Tip (Summary)				
Debate	72.61	11.54	9.93	w/o	10,846	1,356	1,356
				w/	10,975	1,372	1,372
Dianping	101.50	3.39	12.20	w/o	137,208	17,151	17,151
				w/	179,784	22,473	22,473

RNN-based framework, both the encoder and decoder are implemented as 1 layer of (Bi-)LSTM with a hidden size of 256, and the word embedding size is set as 128. The word embedding is randomly initialized and learned during training. For optimization, we use Adam [10] with initial learning rate 0.001 and the batch size is empirically set as 128. For Dianping dataset, the maximum encoding lengths for a review and a query are 150 and 5, respectively. Due to the limited screen size of mobile-devices, the maximum length of decoded tip is set as 15. For the Debate dataset, the maximum encoding lengths for a review and a query are 160 and 30, and the maximum length of decoded tip is set as 30.

4.2 Comparison Models

Several competitive models are implemented to evaluate the performance of our query-aware tip generation framework. Please note that both query and review are taken as model inputs.

Query_LEAD. Taking the leading sentence(s) of a document is reported to be a strong baseline in summarization [21]. Here, the first sentence that contains the query in a review is extracted as the tip. If such a sentence can not be found, the leading sentence of the review is selected instead.

Extract_BM25. This is an unsupervised extractive baseline. Given the query, the sentences in the review are ranked by their BM25 scores and the top one is favored.

Extract_Embed. This is another unsupervised extractive baseline. The sentences in the review are ranked by their embedding-based cosine similarities with the query. We use the publicly largest pre-trained Chinese word embedddings [23] for **Ours** and Glove[6] for **Debate**.

RNN. An abstractive baseline utilizing the pointer generator implementation, regardless of the query.

Trans. An abstractive baseline utilizing the Transformer-based encoder-decoder implementation, regardless of the query.

RNN(Trans) + Qa_Enc/Qa_Dec/Both. The RNN(Trans) with the proposed Qa_Enc and Qa_Dec separately or jointly.

4.3 Automatic Evaluation

In automatic evaluation, the generated tips are assessed in terms of query-relevance and coherency.

Metrics. For query-relevance, two metrics are used. First, the cosine similarities between the embeddings of the generated tips and the corresponding queries are calculated, denoted as **Semantic** [22]. The embedding of tip or query is obtained by maximizing over the embeddings of the tokens in the sequence. Second, the number of co-occurring tokens in the generated tip and query divided by the query length is used as a lexical proxy of the relevance. This metric is denoted as **Lexicon**. The coherency is measured by **BLEU** [18].

Results. The results are reported in Table 3. For the ease of presentation, all the metrics are multiplied by 100. In general, the proposed method Trans + Both outperforms all the comparing algorithms in terms of almost all the metrics across both datasets. In comparison to the 3 query-aware retrieval-based baselines, the query-aware abstractive models perform almost better on both query-relevance and coherency criteria due to the flexibility of abstractive models. Even the query is not explicitly mentioned in the review, the query-aware abstractive models can generate fluent tip relevant to the query. In terms of **Lexicon** metric, the semantic-based retrieval method outperforms the other abstractive methods on the Debate. We speculate it is caused by the long-query characteristic of the Debate dataset. Recall that the queries in Debate are long questions in essence and **Lexicon** measures lexical similarity between the query and the final tip. Retrieval-based methods focus on literal matching between the query and the extractive tip, while abstractive methods dedicate to generating fluent tips by attending to the key information in the given query. In

[6] http://nlp.stanford.edu/projects/glove/

Table 3: Automatic Evaluation Results on Dianping and Debate datasets.

Group	Methods	Debate			Dianping		
		Semantic	Lexicon	BLEU	Semantic	Lexicon	BLEU
Retrieval	Query_LEAD	-	10.23	2.23	-	40.70	23.20
	Extract_BM25	-	14.39	1.12	-	47.18	27.59
	Extract_Embed	-	**14.43**	1.13	-	37.04	28.29
RNN	RNN	83.87	8.91	11.02	60.08	40.94	40.74
	RNN + Qa_Enc	84.37	9.23	15.72	62.65	41.37	48.29
	RNN + Qa_Dec	84.17	9.07	15.37	62.77	43.92	46.88
	RNN + Both	84.43	9.34	16.58	64.86	44.11	48.38
Transformer	Trans	87.17	10.52	30.41	65.64	47.49	48.71
	Trans + Qa_Enc	86.07	13.17	32.03	67.00	49.79	50.39
	Trans + Qa_Dec	84.70	13.46	32.52	62.70	42.61	52.66
	Trans + Both	**88.06**	13.43	**32.93**	**69.79**	**53.75**	**54.20**

addition, Transformer-based models outperform RNN-based models in terms of all the metrics across both datasets. This is reasonable considering Transformer is better at handling the long-range dependencies in user reviews. For RNN-based and Transformer-based models, incorporating query-aware information including Qa_Enc and Qa_Dec improve the overall performance compared with the vanilla RNN and Transformer, which further verifies the importance of query-aware information.

4.4 Manual Evaluation

Due to the high cost of manual assessments, we only conduct manual evaluation of the proposed framework on Dianping dataset. In manual evaluation, the generated tips of different models are assigned to 5 annotators with a related background. They are instructed to score each generated tips with respect to 3 perspectives: Readability, Relevance and Usefulness. For Usefulness and Relevance, the majority annotating result is adopted as the final assessment, while for Readability the average annotating result is adopted.

Metrics. Among the 3 metrics, **Readability** measures whether a generated tip is fluent and grammatical, **Relevance** indicates whether a generated tip is relevant to the query, and **Usefulness** demonstrates whether the generate tip is helpful for the user to make a decision. In particular, Relevance and Usefulness are assessed by a binary score (i.e, 1 for true and 0 for false), and Readability is assessed by a 3-point scale score from 1 (worst) to 3 (best).

Results. The results are reported in Table 4. Overall, the tips generated by Transformer-based models achieve better readability and query-relevance than RNN-based models. The proposed method Trans + Both performs best on all the metrics.

The introduction of query information into RNN and Transformer improves the relevance performance in both cases. In terms of Usefulness, all the query-aware variants generate tips that are more informative for the users.

Table 4: Manual Evaluation Results on Dianping dataset.

Methods		Read.	Rel.	Useful.
RNN	RNN	2.12	32.00%	43.56%
	RNN + Qa_Enc	2.67	40.33%	43.13%
	RNN + Qa_Dec	2.71	52.31%	41.49%
	RNN + Both	2.63	53.50%	44.50%
Transformer	Trans	2.55	39.25%	44.38%
	Trans + Qa_Enc	**2.88**	60.52%	47.36%
	Trans + Qa_Dec	2.80	60.53%	47.37%
	Trans + Both	**2.88**	**63.72%**	**54.35%**

4.5 Case Studies

An illustrative case from the test set in Dianping dataset is presented in Figure 5. Due to the limited space, among query-aware models, only the results of RNN + Both and Trans + Both are presented. It is obvious that only the two query-aware models generate tips related to the user query **cake**. What's more, the tip of Trans + Both mentions the shop owner's service attitude, which may be more informative to users.

5 ONLINE DEPLOYMENT

We deploy the query-aware tips in a production environment to test its online performance. It is an A/B test in the SRP scenario (which is initiated by a user query) of the aforementioned App (with ~10 million daily queries). It is noteworthy that the reviews belong to a POI are ranked by their number of "likes" by users. Given a query, we take the top-ranked review of each returned POI to generate tips. The A/B testing system diverts 10% total query traffic and splits it equally into 4 separate buckets. All the other settings of these buckets are identical. The tips displayed in the 4 buckets are generated with the following strategies: (1) No tip is displayed with POIs, (2) The tips are generated by Trans, (3) The tips are generated by Extract_BM25, and (4) The tips are generated by Trans + Both.

The online test lasted for one week. CTR is adopted to test the performance, which is calculated as $CTR = \frac{\#Clicks\_in\_SRP}{\#Query}$, where $\#Query$ is the count of the user queries, and $\#Clicks\_in\_SRP$ is the total clicks in the SRP triggered by the queries. Higher CTR therefore implies that users are more likely to browse and click the POI. Given a query, several clicks may occur in the same triggered SRP, $\#Clicks\_in\_SRP$ is counted as one in this case. The averaged CTR in the 4 buckets are 65.72%, 65.74%, 65.77% and 65.80%, respectively. In comparison to the No-tip baseline, even the query-agnostic tips improve the CTR. Both extractive and abstractive query-aware models (i.e., Extract_BM25 and Trans +

POI	Nana蛋糕手工烘焙	Nana’s cake handmade baking
Review	公司周年庆吃到的。从外观来说，不错，看得出是裱花的技术还是很好的。因为早上是想9点半拿到，但是因为塞车的原因，晚了一点，这里老板很用心，会提前跟我说，让我们不要着急。另外在味道很不错，我比较喜欢吃草莓那边的，巧克力好吃，哈哈	Eaten at the company‘s anniversary celebration. From the appearance, it is good, it can be seen that the technology of decorating is very good. I wanted to get it at 9:30 in the morning, but cause of the traffic jam, I was a bit late. The boss is very attentive and will tell me in advance do not worry about it. In addition, it tastes very good. I prefer to eat strawberry flavor. The chocolate flavor is delicious, haha.
Query	蛋糕	cake
Tip (Query_LEAD)	公司周年庆吃到的。	Eaten at the company's anniversary celebration.
Tip (RNN)	裱花的技术还是很好的	The technology of decorating is very good.
Tip (Transformer)	这里老板很不错	The boss is very nice.
Tip (RNN-Both)	蛋糕很好吃	The cake tastes very good.
Tip (Transformer-Both)	这里老板很用心蛋糕很松软	The boss is very nice and the cake tastes very fluffy.

Figure 5: Examples of tip generation from Dianping dataset. The column 3 is the translation of column 2.

Both) achieve higher CTR than query-agnostic Trans. We conduct a two-tailed paired t-test, and the improvements are significant with $p < 0.05$. The result is quite impressive, if we consider the fact that tips on a search result page occupy a relatively small space and thus only partially affect the users’ decision.

6 RELATED WORK

Our work touches on two strands of research within Query-focused text summarization (QFS) and constrained sentence generation.

Query-focused Summarization. QFS aims to summarize a document cluster in response to a specific user query or topic. It was first introduced in the Document Understanding Conference (DUC) shared tasks [8, 13]. Successful performance on the task benefits from a combination of IR and NLP capabilities, including passage retrieval and ranking, sentence compression [4, 31], and generation of fluent text. Existing QFS work can be categorized into extractive and abstractive methods. Extractive methods, where systems usually take as input a set of documents and select the top relevant sentences as the final summary. Cao et al. [3] propose AttSum to tackle extractive QFS, which learns query relevance and sentence saliency ranking jointly. Abstractive methods attract more attention due to their flexibility in text summarization. Rush et al. [20] first employ sequence-to-sequence (seq2seq) model [26] with attention mechanism [1] in abstractive summarization and achieve promising results. Further improvements are brought with recurrent decoders [7], selective gate network [36], abstract meaning representation [27], hierarchical networks [16] and variational auto-encoders [14]. In terms of QFS, Nema et al. [17] introduce a query attention model in the encoder-decoder framework, and a diversity attention model to alleviate the problem of repeating phrases in summary. Query relevance, multi-document coverage, and summary length constraints are incorporated into seq2seq models to improve QFS performance [2]. Most QFS work involves long natural language questions as the queries, while we focus on short search queries in this paper.

Constrained Sentence Generation. Constrained seq2seq sentence generation, considering external information during generation, are widely studied in human-computer conversation systems and e-commerce applications. Mou et al. [15] propose a content-introducing approach to dialogue systems, which can generate a reply containing the given keyword. Yao et al. [34] propose an implicit content-introducing method that incorporates additional information into the seq2seq model via a hierarchical gated fusion unit. Xing et al. [33] consider incorporating topic information into a seq2seq framework to generate informative responses for chatbots. Sun et al. [24] propose a multi-source pointer network [29] by adding a new knowledge encoder to retain the key information during product title generation. In e-commerce search scenarios, a query generation task is proposed to improve long product title compression performance in a multi-task learning framework [30]. Chen et al. [5] propose a knowledge-based personalized (KOBE) product description generation model in the context of e-commerce which considers product aspects and user categories during text generation. Duan et al. [9] propose a query-variant advertisement generation model that takes keywords and associated external knowledge as input during training and adds different queries during inference. Abstractive tip generation is first studied and deployed in recommendation systems [12], where tip generation is jointly optimized with rating prediction using a multi-task learning manner. Some researchers also capture the intrinsic language styles of users via variational auto-encoders to generate personalized tips [11]. To take the query impact into account, this paper proposes query-aware tip generation for vertical search. We consider query information in both encoder and decoder sides to generate query-aware tips, that are intuitive but effective and of great business values in vertical search scenarios.

7 CONCLUSION

Vertical search results are devoted to a certain media type or genre of content. Taking Dianping as an example, given a query, the vertical search engine usually returns a list of relevant POIs (i.e., restaurants) to users. To improve the user experience, some extra information need to be presented together with the search results. Tip, a concise summary of genuine user reviews, is an intuitive and complementary form to help users get a quick insight into the search results. This paper studies the task of query-aware tip generation for vertical search. We propose an intuitive and effective query-aware tip generation framework. Two specific adaptations for the Transformer and the RNN architectures are developed. Extensive experiments on both public and realistic datasets reveal the effectiveness of our proposed approach. The online deployment experiments on Dianping demonstrate the promising business value of the query-aware tip generation framework.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints* abs/1409.0473 (Sept. 2014). https://arxiv.org/abs/1409.0473

[2] Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704* (2018).

[3] Ziqiang Cao, Wenjie Li, Sujian Li, Furu Wei, and Yanran Li. 2016. AttSum: Joint Learning of Focusing and Summarization with Neural Attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 547–556.

[4] Yllias Chali and Sadid A Hasan. 2012. On the effectiveness of using sentence compression models for query-focused multi-document summarization. *Proceedings of COLING 2012* (2012), 457–474.

[5] Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Towards Knowledge-Based Personalized Product Description Generation in E-commerce. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 3040–3050.

[6] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[7] Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 93–98. https://doi.org/10.18653/v1/N16-1012

[8] Hoa Trang Dang. 2005. Overview of DUC 2005. In *Proceedings of the document understanding conference*, Vol. 2005. Citeseer, 1–12.

[9] Siyu Duan, Wei Li, Cai Jing, Yancheng He, Yunfang Wu, and Xu Sun. 2020. Query-Variant Advertisement Text Generation with Association Knowledge. *arXiv preprint arXiv:2004.06438* (2020).

[10] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[11] Piji Li, Zihao Wang, Lidong Bing, and Wai Lam. 2019. Persona-Aware Tips Generation? *The World Wide Web Conference on - WWW ' 19* (2019). https://doi.org/10.1145/3308558.3313496

[12] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 345–354.

[13] Sujian Li, You Ouyang, Wei Wang, and Bin Sun. 2007. Multi-document Summarization Using Support Vector Regression.

[14] Yishu Miao and Phil Blunsom. 2016. Language as a Latent Variable: Discrete Generative Models for Sentence Compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 319–328. https://doi.org/10.18653/v1/D16-1031

[15] Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to Backward and Forward Sequences: A Content-Introducing Approach to Generative Short-Text Conversation. arXiv:cs.CL/1607.00970

[16] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 280–290. https://doi.org/10.18653/v1/K16-1028

[17] Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *ACL*.

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 311–318. https://doi.org/10.3115/1073083.1073135

[19] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[20] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 379–389. https://doi.org/10.18653/v1/D15-1044

[21] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1073–1083. https://doi.org/10.18653/v1/P17-1099

[22] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, Satinder P. Singh and Shaul Markovitch (Eds.). AAAI Press, 3295–3301. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14567

[23] Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 175–180.

[24] Fei Sun, Peng Jiang, Hanxiao Sun, Changhua Pei, Wenwu Ou, and Xiaobo Wang. 2018. Multi-Source Pointer Network for Product Title Summarization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. ACM, New York, NY, USA, 7–16. https://doi.org/10.1145/3269206.3271722

[25] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 3104–3112.

[27] Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 1054–1059.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*. 6000–6010. http://papers.nips.cc/paper/7181-attention-is-all-you-need

[29] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 2692–2700. http://papers.nips.cc/paper/5866-pointer-networks.pdf

[30] Jingang Wang, Junfeng Tian, Long Xin Qiu, Sheng Li, Jun Lang, Luo Si, and Man Lan. 2018. A Multi-Task Learning Approach for Improving Product Title Compression with User Search Log Data. In *AAAI*.

[31] Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2016. A sentence compression based framework to query-focused multi-document summarization. *arXiv preprint arXiv:1606.07548* (2016).

[32] Ingmar Weber, Antti Ukkonen, and Aris Gionis. 2012. Answers, not links: extracting tips from yahoo! answers to address how-to web queries. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 613–622.

[33] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[34] Lili Yao, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan. 2017. Towards Implicit Content-Introducing for Generative Short-Text Conversation Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2190–2199. https://doi.org/10.18653/v1/D17-1233

[35] Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*. Shanghai, China, 73–78. https://www.aclweb.org/anthology/Y15-1009

[36] Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective Encoding for Abstractive Sentence Summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1095–1104. http://aclweb.org/anthology/P17-1101

[37] Di Zhu, Theodoros Lappas, and Juheng Zhang. 2018. Unsupervised tip-mining from customer reviews. *Decision Support Systems* 107 (2018), 116–124.