Campaign Keyword Augmentation via Generative Methods

Haoran Shi, Zhibiao Rao, Yongning Wu, Zuohua Zhang, Chu Wang
Amazon.com Inc
Seattle, Washington, USA
{haoransh, zhibiar, yongning, zhzhang, chuwang}@amazon.com

Abstract

Keyword augmentation is a fundamental problem for sponsored search modeling and business. Machine generated keywords can be recommended to advertisers for better campaign discoverability as well as used as features for sourcing and ranking models. Generating high-quality keywords is difficult, especially for cold campaigns with limited or even no historical logs; and the industry trend of including multiple products in a single ad campaign is making the problem more challenging. In this paper, we propose a keyword augmentation method based on generative seq2seq model and trie-based search mechanism, which is able to generate high-quality keywords for any products or product lists. We conduct human annotations, offline analysis, and online experiments to evaluate the performance of our method against benchmarks in terms of augmented keyword quality as well as lifted ad exposure. The experiment results demonstrate that our method is able to generate more valid keywords which can serve as an efficient addition to advertiser selected keywords.

1 Introduction

Sponsored search has proved to be an efficient and inspiring way of connecting shoppers with interesting products. Advertisers have the freedom to provide a list of targeting keywords with associated bidding prices to the ad platform, so that their ad campaigns can match to shopper queries either lexically or semantically. The quantity and quality of targeting keywords are fundamental to the performance of the ad campaign: insufficient keywords can hardly get the campaigns with enough exposure; and low-quality ones will match shopper queries with irrelevant ads, leading to low conversion and damages to customer experiences.

Efficient and optimal keyword selection is challenging and time consuming because it requires deep understanding of the ad industry as well as the sponsored search platform. Furthermore, an ad campaign used to be designed for a single product traditionally, but ads with richer information start to appear in the recent years. Nowadays, an ad campaign can contain multiple products, brand stores, or even rich media contents. Consequently, the keyword selection task becomes even more crucial and challenging for advertisers campaign creation and management.

In this paper, we present an end-to-end machine learning solution to generate keywords for ad campaigns. The method applies to single-product campaigns as well as campaigns with any number of products. It only relies on product information like product titles, hence efficient on newly created campaigns without any performance logs in the past. We conduct offline and online experiments on the proposed method and observe significant improvements over traditional statistical methods in terms of keyword quality. Specifically, we highlight our contributions as the following:

- We propose an end-to-end solution for keyword generation. It can be applied to recommendation of high-quality keywords for advertisers as well as semantic augmentation for better ad exposure.
- The keyword generation method relies on product metadata but not historical performance data of ad. Therefore, the method applies to tail or newly-created campaigns.
- Our method is able to handle single-product-campaign as well as multi-product-campaign by leveraging semantic meanings of each product in the latent space.
- The quality and superiority of the generated keywords are validated by human audits, offline analysis as well as online experiments.

Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4), pages 33–37

2 Related Work

Considerable research work has been devoted to keyword augmentation techniques because of its important applications in information retrieval, indexing, and digital library management. The majority of existing work focuses on processing documents with statistical information including term co-occurrence and frequency (Campos et al., 2020). In particular, Rose et al. (2010) proposed RAKE to split the document into candidate phrases by word delimiters and calculate their scores with co-occurrence counts. Ravi et al. (2010) first applied statistical machine translation model for keyword candidate generation and ranking. With rapid development of deep learning models, neural machine translation has surpassed statistical translation in many benchmarks, where recurrent neural networks (RNNs) and gating mechanisms are popular building blocks to model sequence dependencies and alignments (Hochreiter and Schmidhuber, 1997; Cho et al., 2014). However, extracting high-quality and diverse keywords from short document like ad campaigns remains a difficult problem due to the lack of context.

Query expansion for improved product or ad discovery, as an application of keyword augmentation, is crucial to e-commerce search engines and recommender systems. He et al. (2016) applies LSTM architecture to rewriting query into web document index space. However, the long tail distribution of the query space hinders the deployment of complicated generative models. It is well known that infrequent queries account for a large portion of the e-commerce daily queries. In Lian et al. (2019), a lightweight neural network for infrequent queries is trained, incurring even more engineering burdens for deployment. It also proposed the method of using trie-based search to normalize the decoding in the constrained semantic space, which is further investigated in Chen et al. (2020).

Expanding advertiser bidding keywords is another growing research area. Qiao et al. (2017) applies keyword clustering and topic modeling to retrieve similar keywords and Zhou et al. (2019) conducts keywords expansion in the constrained domains through neural generative models. In addition, Zhang et al. (2014) formulates the keyword recommendation problem as a mixed integer optimization problem, where they collect candidate keywords whose relevance score to the ad group exceed a threshold and handle the keyword selection problem by maximizing revenue. Such methods rely on the quality of advertiser bidding keywords. Campaigns with sub-optimal or misused keywords may suffer significantly.

3 Methods

In this section, we present our products-to-keyword framework and algorithm for campaign keyword augmentation. The framework is compatible with any seq2seq components with encoders and decoders. Given an ad campaign C including a set of products $\{p_1, p_2, \ldots, p_n\}$, our goal is to generate a list of relevant keywords $\{k_1, k_2, \ldots, k_m\}$. We will describe how we generate keywords for each product first and later generalize to ad campaigns with multiple products.

3.1 Dataset and Preprocessing

We choose to use organic search click data for model training, which includes the pairs of queries and clicked products in search log. Compared to sponsored search data, it can guide the model to generate more keywords than existing ads system as shown in Lian et al. (2019). We lowercase shopper queries and product titles, and then apply pretrained T5 tokenizer (Raffel et al., 2020) for tokenization. Note that the vocabulary space for shopper queries and product titles are ever-growing, but the subword encoding space is stable. Therefore, subword tokenization is an efficient method to handle the out-of-vocabulary issue which hurts the fluency of generated queries.

3.2 Modeling

In the following, we use $X = [x_1, x_2, ...x_L]$ to denote tokenized product title whose length is L. Let θ be the trainable model parameters, and $Q = [q_0, q_1, q_2, \ldots, q_S]$ as the padded tokenized target query, where q_0 is the special start token and q_S the special end token. For training, we feed the model with the product title X and the first s query token $Q_{<s} = [q_0, q_1, \ldots, q_{s-1}]$, to predict the next query token q_s, where $1 \leq s \leq S$.

We adopt the seq2seq model training with teacher forcing, where multi-layer Gated Recurrent Units (GRU) are used in the encoder and the decoder (Cho et al., 2014). The encoder transforms the tokenized sequence into the latent space with an embedding layer and a GRU encoder. Then the decoder transforms the latent vector back to a predicted distribution over token vocabulary given

all previously decoded tokens as inputs. The token embedding layer for the encoder and the decoder are shared. We use cross entropy loss to maximize the likelihood of the model generating the correct next token for each training data point $(X, Q_{<s}, q_s)$. The objective function is written as

$$L(\theta) = -\sum_{s=1}^{S} \log\ p(q_s|X, Q_{<s}; \theta))). \quad (1)$$

3.3 Keyword Generation

Intuitively, the desired generated keywords should be diverse to accommodate different aspects of the products, and relevant to promote the products to right shoppers. In the model inference phase, the encoding is the same as in training, while in decoding process beam search is usually used for larger search space. However, standard beam search will generate similar sequences with minimal diversity. To resolve this issue, we build the trie T_Q on all tokenized queries in our training dataset to normalize the decoding. Specifically in the i-th decoding step, the decoder outputs the probability of $p(q_s|X, Q_{<s}$ over the vocabulary. Then we extract all children nodes of $Q_{<s}$ in the trie and keep those with highest probability in the candidate beam for future decoding. In this way, it is guaranteed that the generated sequence exists in the canonical query space as a path traversal in the trie ending with the special end token. We define such queries as valid queries since they reflect the word selection of shoppers. The prebuilt Trie and the inference workflow for one product title is illustrated in Figure 1 and 2 respectively.

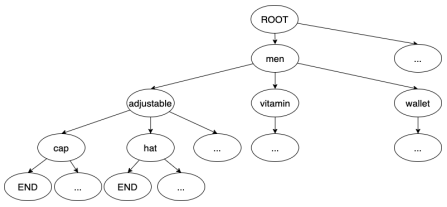

Figure 1: An illustration of the Trie built on queries

Now we discuss the handling of multiple products within one campaign. A naive solution is to generate keywords for each product, and then aggregate all generated keywords. Alternatively, we propose to encode each product title into the latent space, and apply the decoder to the averaged title encodings. These two methods are denoted as Generation by Keyword Aggregation (G-KA) and Generation by Hidden State Mixing (G-HSM).

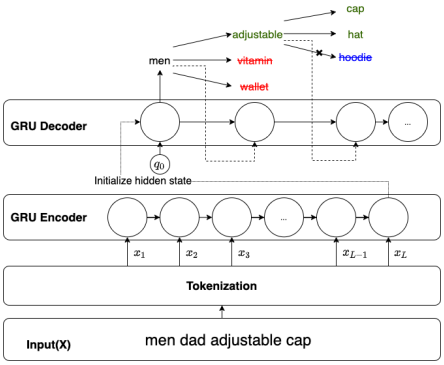

Figure 2: An illustration of the keyword generation process. Tokens in red color with strikethrough line are removed by beam search, and "men adjustable hoodie" is pruned by the query trie. Details of the encoder/decoder are omitted.

4 Experiments

In this section, we compare the performance of the proposed methods with empirical study. In Section 4.1, we explain how we collect our experimental data including training, validation, and testing; then we introduce benchmarking methods and parameter setup in Section 4.2; evaluation metrics are explained in Section 4.3; and eventually in Section 4.4, we illustrate experimental results.

4.1 Data Preparation

We collect query-product pairs in search click logs from September 2020 to March 2021. To reduce the noise, we apply a series of filtering: 1) remove stop-words in queries and product titles; 2) remove tokens with non-alphanumeric characters; 3) remove pairs with empty query or title; 4) remove query-product pairs with less than 1024 clicks.

In total, we collect 6.2M pairs of queries and products, where more than 95% of the queries have less than or equal to 6 tokens. We split them into training set (5.2M) and validation set (1M). To prevent frequent queries dominating the result and encourage diversity, we normalize the weight of all pairs to the same for training stage.

For testing, we use cold campaigns to benchmark the keyword augmentation model performance, which are campaigns with less than 100 impressions from January 2021 to March 2021. Since we use organic search log for training, there is no overlap between training and testing data.

4.2 Benchmarks and Parameters Setup

The benchmark methods include heuristics based on search log as well as trending keyword genera-

tion methods. We use ADV to denote targeting keywords provided by advertisers, and OS to denote keywords generated by organic search logs heuristically. More specifically, we extract those queries which lead to the click of the campaign products in organic search, and collect those distinct queries as keywords for the campaign. We also include RAKE in our comparison which is a popular open-sourced keyword extraction algorithm based on lexical co-occurrence statistics. To achieve better extraction performance, we run RAKE on the concatenation of all product titles in the campaign, and keep the keywords with length between 2 and 6. In addition, we compare the two variants of our proposed solutions, G-KA and G-HSM:

- G-KA: We select top 8 generated queries with lowest perplexity from each product.
- G-HSM: We select top 3 products in terms of sales in each campaign and averaged their latent encodings for decoding. We select top 8 generated queries for each campaign too.

For both variants, the encoder and decoders are 6-layer GRUs with 256 hidden dimensions, and the beam search size is set as 20. We choose the model with the lowest loss on the validation dataset.

4.3 Evaluation Method

We sample 1500 keyword-campaign pairs from each method for human annotations. Each campaign will be associated with a landing page URL including all targeted products. Three different auditors are assigned to label each pair as exactly relevant, partially relevant, and irrelevant. We take the majority decision as the final label of each pair. For simplicity, we merge exactly relevant and partially relevant labels, and report the ratio of relevance for different methods. To evaluate whether the generated keywords are able to effectively promote ad exposures, we calculate the total traffics incurred by generated keywords as a metric, and report the median total traffics as the Exposure column of Table 1. We also report the median value of the number of generated keywords for each method as the Count column, and use Exposure divided by Count to evaluate the traffic incurred by each individual keyword.

In addition, we conduct online A/B testing by enriching the campaign keywords with generated results from G-KA for ad sourcing and comparing with the existing system in terms of total ad impressions. All other components in the system, including relevance and ranking logics, are consistent for control and treatment.

4.4 Results and Analysis

Table 1 illustrates the performance of different methods in terms of the number of generated keywords, relevance ratio and exposure. For the testing campaigns without many impressions, advertisers bid on a few relevant keywords which lead to poor ad exposures. Such impression shortage issue is one of the motivations for our work, and we use this method as the baseline.

RAKE is able to extract relevant keywords from the product titles, but their exposure is quite low. Such results indicate vocabulary gap exists a between product titles and shopper queries. Organic search connects the products to the relevant queries but the amount of queries are much fewer than the baseline. Intuitively, this is because advertisers are aware of historical queries related to their products.

G-KA and G-HSM provide a moderate number of keywords with ads exposure much larger than baseline (+1665% and +2194%), though the relevance rate are lower than standard baseline. The boost of Exposure/Count also demonstrates the effectiveness of the proposed keyword generation methods with seq2seq learning framework and trie-based decoding. In addition, the G-HSM shows superiority over G-KA in terms of keyword relevancy and validity.

In our online experiment, our model increases ad impressions by 5.3%, which demonstrates the contribution from the proposed keyword augmentation methods. Note that relevance and ranking logics are the same for both control and treatment groups. Only augmented keywords not covered by existing advertiser selected keywords with good quality are able to yield additional ad exposures.

5 Conclusion and Future Work

In this paper, we formulate the sponsored search keyword augmentation task as a seq2seq learning problem in the constrained space. We present a general framework which incorporates seq2seq architecture and trie-based pruning for query generation from product titles. We compare the proposed method with baselines and other existing methods, and show that our method is able to generate relevant keywords which bring up the campaign exposure significantly. In the future, we would like to

Method	Count	Relevance	Exposure	Exposure/Count
ADV	12	97.8%	baseline	baseline
RAKE	9	93.1%	-71.66%	-62.22%
OS	2	98.1%	+192.4%	+1654%
G-KA	19	78.1%	+1665%	+1015%
G-HSM	8	88.3%	+2194%	+3341%

Table 1: Performance comparison.

explore more structured decoding strategies combined with trie to improve the generation quality, and take more factors into account when generating keywords including long-tail keywords and keyword competitiveness.

Acknowledgments

We would like to thank to Hongyu Zhu, Weiming Wu, Barry Bai, Hirohisa Fujita for their help to set up the online A/B testing, and all the reviewers for their valuable suggestions.

References

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alipio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Pinzhen Chen, Nikolay Bogoychev, Kenneth Heafield, and Faheem Kirefu. 2020. Parallel sentence mining by constrained decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1672–1678.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to rewrite queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1443–1452.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yijiang Lian, Zhijie Chen, Jinlong Hu, Kefeng Zhang, Chunwei Yan, Muchenxuan Tong, Wenying Han, Hanju Guan, Ying Li, Ying Cao, et al. 2019. An end-to-end generative retrieval method for sponsored search engine–decoding efficiently into a closed target domain. *arXiv preprint arXiv:1902.00592*.

Dandan Qiao, Jin Zhang, Qiang Wei, and Guoqing Chen. 2017. Finding competitive keywords from query logs to enhance search engine advertising. *Information & Management*, 54(4):531–543.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sujith Ravi, Andrei Broder, Evgeniy Gabrilovich, Vanja Josifovski, Sandeep Pandey, and Bo Pang. 2010. Automatic generation of bid phrases for online advertising. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 341–350.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

Ying Zhang, Weinan Zhang, Bin Gao, Xiaojie Yuan, and Tie-Yan Liu. 2014. Bid keyword suggestion in sponsored search based on competitiveness and relevance. *Information processing & management*, 50(4):508–523.

Hao Zhou, Minlie Huang, Yishun Mao, Changlei Zhu, Peng Shu, and Xiaoyan Zhu. 2019. Domain-constrained advertising keyword generation. In *The World Wide Web Conference*, pages 2448–2459.