Personalized Query Suggestions

Jianling Zhong
jizhong@linkedin.com
LinkedIn Corporation

Weiwei Guo
wguo@linkedin.com
LinkedIn Corporation

Huiji Gao
hgao@linkedin.com
LinkedIn Corporation

Bo Long
blong@linkedin.com
LinkedIn Corporation

ABSTRACT

With the exponential growth of information on the internet, users have been relying on search engines for finding the precise documents. However, user queries are often short. The inherent ambiguity of short queries imposes great challenges for search engines to understand user intent. Query suggestion is one key technique for search engines to augment user queries so that they can better understand user intent. In the past, query suggestions have been relying on either term-frequency–based methods with little semantic understanding of the query, or word-embedding–based methods with little personalization efforts. Here, we present a sequence-to-sequence-model–based query suggestion framework that is capable of modeling structured, personalized features and unstructured query texts naturally. This capability opens up the opportunity to better understand query semantics and user intent at the same time. As the largest professional network, LinkedIn has the advantage of utilizing a rich amount of accurate member profile information to personalize query suggestions. We applied this framework in the LinkedIn production traffic and showed that personalized query suggestions significantly improved member search experience as measured by key business metrics at LinkedIn.

KEYWORDS

Query suggestion; Sequence-to-sequence model; Personalization; Deep learning model deployment

ACM Reference Format:
Jianling Zhong, Weiwei Guo, Huiji Gao, and Bo Long. 2020. Personalized Query Suggestions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401331

1 INTRODUCTION

Search engines have been the most popular way to retrieve information for users on the world wide web. However, retrieving the right content has been increasingly difficult because of the exponential growth of data volume. The inherent ambiguity of search queries also contributes to this difficulty. Azad and Deepak [1] reported that the majority of search queries are less than five words. This kind of short text lacks context and promotes ambiguity. For example, when a user searches for "engineer jobs", it is not immediately clear whether the user wants to find "software engineer jobs", "mechanical engineer jobs", or jobs in some other engineering domains.

On top of that, well-known issues, such as synonymy (different words sharing the same meaning) and polysemy (one word having multiple distinct meanings), could further confuse search engines. To combat those challenges, practitioners have developed different strategies in information retrieval and ranking. One key strategy that directly addresses the ambiguity of search queries is query suggestion (Q.S.).

Q.S. often works by presenting a few related query suggestions to a user after a search query was issued. Users have the freedom to choose which one they want to explore. Note that this is different from automatic query expansion [13], in which search engines implicitly expand the queries without user feedback. We are primarily interested in the Q.S. problem in this work.

LinkedIn is the largest professional social network. Through LinkedIn's search capability, members interact with many different entities, including (but not limited to) job postings, member profiles, and content postings. The Q.S. functionality (*"Try searching for"*, Figure 1) is a key component of LinkedIn search. In the past, we experimented with sequence-to-sequence (Seq2Seq) model [15] for this functionality. Seq2Seq model overcomes challenges that traditional co-occurrence methods face, such as little understanding of the query semantics, and only known queries get suggestions or will be recommended as suggestions. Seq2Seq model summarizes word semantics through an encoder. Its decoder is a generative model. Therefore the model could understand word meaning and can effectively deal with rare or unseen queries. The decoder could also generate meaningful suggestions that never appeared in the past search history. Those properties are important for improving LinkedIn search experience and helping users discover new opportunities.

However, both traditional methods and vanilla Seq2Seq model have another drawback: Suggestions are not personalized. Personalization is a critical need, especially for users with diverse backgrounds. For example, two LinkedIn users can both search for "microsoft" (Figure 1). However, a software engineer is more likely to be looking for "microsoft azure" while a salesperson would be more interested in "microsoft sales". Without knowing any personal information, a model would recommend suggestions irrelevant to a user's intent, leading to a poor search experience.

In this work, we further improved our Q.S. system by modeling both structured, personalized user features and unstructured text data simultaneously in a Seq2Seq model. Our Q.S. system presented here includes both offline modeling framework as well as the online serving infrastructure. The system is general and can be easily

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401331

Figure 1: Try searching for on LinkedIn mobile app.

applied in other search engines. Compared with our baseline, non-personalized Seq2Seq model, this work's main contributions include the following:

- We examined different strategies of incorporating structured member features into a Seq2Seq model and compared their performance in an industrial setting.
- With live production traffic, we showed that personalized Q.S significantly improves search experience compared with the non-personalized version.
- We also studied and presented results on how Q.S improved search experience for different job-seeking member segments in a professional social network setting.

2 RELATED WORK

Q.S has a long history of research (for example, Harman [5]). In this section, we will primarily focus on the personalization efforts in this space.

Early days of personalization efforts mostly focused on single-user behavior data. Term-co-occurrence based method, collaborative filtering method, and term-frequency based methods all found their applications in modeling local user profile data [4, 16]. The local user profile refers to data stored on a user's local machine, such as text documents, emails, and cached web browsing histories. The limitation of single-user data means that those approaches would suffer from data sparsity and could not learn shared features across different users. Later, Mei et al. [11] proposed a bipartite-graph based method to derive query suggestions. They were able to utilize all user data by pooling all data into the same graph. Their method achieved personalization by attaching a unique user identifier to each query to form a new pseudo query.

Those early efforts did not consider query semantics. In other words, query suggestions were derived by some form of association rules without considering the actual meaning of the query. However, query semantics are very useful. For example, Bouadjenek et al. [2] combined the use of the TF-IDF method and term semantic similarities on a single user's document data to derive the query suggestions. Jayanthi et al. [8] further used phrase semantics instead of term semantics to derive personalized query suggestions. Both showed that utilizing query semantics improved the quality of query suggestions.

Figure 2: Overview of the Seq2Seq model architecture.

In Sordoni et al. [14], the authors explored the use of hierarchical recurrent encoder-decoder for generating query suggestions. This method naturally models the query semantics and can generate unseen query suggestions. Their methodology is the most similar to ours. However, we have very different focuses: Sordoni et al. [14] were mostly interested in the context-awareness of the query suggestions, where context is defined by past search history, while our focuses are mainly on the personalization with user-features. Our approach is also battle-tested in a large scale industrial setting.

3 MODELING APPROACH

We used a Long Short-Term Memory (LSTM) [6] based Seq2Seq model. Our model follows Luong et al. [10] and is introduced briefly here.

3.1 Seq2Seq model

A Seq2Seq model consists of a decoder and an encoder. In our model, both the decoder and the encoder are recurrent neural networks (RNN) with two layers of stacked LSTM hidden units (Figure 2). The objective function we seek to optimize is (the logarithm of) the conditional probability:

$$;>6\,\%\tilde{}\mathrm{j}\mathrm{G}^{\mathrm{o}} = \sum_{8=1}{}\,;>6\,?^{1}\tilde{}_{8}\mathrm{j}\tilde{}_{1}\bullet\tilde{}_{2}\bullet\,"\,"\,"\,\bullet_{8-1}\bullet s^{\mathrm{o}}$$

where $G_1 \bullet G_2 \bullet \,"\,"\,"\,\bullet G$ are the word embeddings of the input source query, $\tilde{}_1 \bullet \tilde{}_2 \bullet \,"\,"\,"\,\bullet \tilde{}$ are the output words of the target sequence, and s is the hidden representation of the source query learned by the encoder. In addition, the model uses an attention mechanism [10]. Therefore, the objective function can be written as:

$$?^{1}\tilde{}_{8}\mathrm{j}\tilde{}_{1}\bullet\tilde{}_{2}\bullet\,"\,"\,"\,\bullet_{8-1}\bullet s^{\mathrm{o}} = \mathrm{B}>5\,\mathrm{C}<0,\mathrm{G}_{\mathrm{B}}\,\tilde{}^{\mathrm{o}}_{8}$$

where

$$\tilde{}_{8} = \mathrm{C}0=^{1},\ {}_{2}\,\text{»}2_{8};\ {}_{8}\text{¼}$$

, $_{\mathrm{B}}$ and, $_{2}$ are weight metrics; $_{8}$ is the RNN hidden unit, 2_{8} is the context vector (Figure 2). 2_{8} is calculated according to the global attention mechanism in Luong et al. [10].

3.2 Personalization strategy

Seq2Seq models are great at modeling unstructured text data. However, structured data features have proven extremely powerful in improving deep neural network (DNN) models [3]. Different strategies of incorporating structured data features into Seq2Seq models

Table 2: M.R.R. comparison of different models.

Model name	Lift w.r.t baseline [1]
baseline	-
add-vocab	3.97%
concat-100	2.91%
concat-10	1.73%

Figure 3: Two personalization strategies for the encoder.

Table 1: Example source and target queries.

Source	Target
talent acquisition specialist	recruiter
technical manager	java manager
nancial analyst	nancial analyst people in usa

exist, such as Jaech and Ostendorf [7], Mikolov and Zweig [12], and Johnson et al. [9]. Here, we explored two common practices of incorporating structured features: additional vocabulary and embedding concatenation (Figure 3). In the additional vocabulary strategy, the structured categorical feature acts as an additional word at the beginning of the source query. Its embedding will be trained together with other word embeddings. For the embedding concatenation strategy, the same categorical feature embedding is concatenated to each of the word embeddings of the source query.

The embedding concatenation strategy may seem to be more flexible since it allows the model to encode the personal feature at each position of the source query. However, as shown by Jaech and Ostendorf [7], this approach effectively only changes the bias term in the hidden unit updating equation. Therefore its flexibility is somewhat limited. On the other hand, the additional vocabulary strategy gives the model flexibility to adapt and attend to the additional personalized feature. As we will show later, the additional vocabulary strategy outperforms the embedding concatenation strategy.

4 EXPERIMENTS

4.1 Datasets

Training data. In our Seq2Seq model, the source query is a user input query, while the target sequence is a query suggestion. To collect training data, we mined English search query logs at LinkedIn using the following heuristics: 1). For two queries from the same user, if they happened within a small time frame, they are collected as candidate pairs; 2). Candidate pairs that share at least one non-stop word, or have a high occurrence count in the dataset remain in the training set. We remove all other candidate pairs, as well as query pairs that contain people's names.

The idea behind this data collection heuristic is that users often reformulate their search queries during a search attempt in order to improve the search results. We can utilize this kind of user-reformulation to train our Seq2Seq model. Table 1 shows a few examples of data points we collected from this procedure. In total, we have over 150 million training pairs. The majority of queries have less than five words.

Offline evaluation data. On the LinkedIn mobile app and the web app, we show a ranked list of query suggestions (Figure 1). An ideal model should rank the query suggestions that a user is most likely to click on the top of the list. Therefore, we collected the past clicks on query suggestions as the evaluation dataset.

4.2 Personalized feature improves Q.S. ranking

LinkedIn is a professional network, and most of its search traffic is profession-oriented. Therefore we chose a profession related feature from member profile data and incorporated it into the model using the personalization strategies mentioned earlier. By modeling users' professional background directly, we hope to provide more relevant query suggestions.

We trained models with different personalization strategies on the same dataset. For the embedding concatenation strategy, both 10 dimensions (concat-10) and 100 dimensions (concat-100) of feature embeddings were trained. Together with the additional vocabulary strategy (add-vocab), we evaluated three models on the same evaluation dataset. Specifically, we re-ranked the list of query suggestions in the evaluation dataset by scoring them using the trained models. The hypothesis is that if a model works better, it should rank a clicked suggestion higher. We calculated the mean reciprocal rank (M.R.R.) of all clicked query suggestions across the evaluation dataset and compare personalized models with the non-personalized baseline Seq2Seq model in Table 2.

The embedding concatenation strategy has a worse performance compared with the additional vocabulary strategy. As we mentioned earlier, this is most likely because embedding concatenation does not provide more model flexibility [7].

Personalization with a professional background feature significantly improved M.R.R. compared with the baseline model. It is worth noting that the position bias (the query suggestions on the top position are naturally more likely to be clicked) works in favor of the baseline model since it generated the query suggestions in the evaluation dataset. Even so, personalized models still outperform the baseline model.

4.3 Personalization improves search metrics

To further confirm personalization does improve query suggestion ranking in real-world traffic, we implemented the additional vocabulary model and tested it in a production environment. We measured whether the personalization could provide more relevant query suggestions using the click-through rate (CTR) both on the query suggestions (CTR_q) and on the overall search results (CTR_r). The hypothesis is that, if personalized query suggestions are more relevant than baseline suggestions, users will be more likely to engage with them. Indeed, our online A/B test showed a significant 5.62% (p-value 7.8×10^{-14}) increase on CTR_q. In addition,

[1] Absolute metric values are not presented due to corporate confidentiality policies.

Table 3: Online A/B test metrics on di erent groups of job seekers. Lift is w.r.t non-personalized query suggestions.

Metric	Job seeker group	Lift (p value)
search sessions	passive job seeker	+1.19% (2" 1 10 3)
	active job seeker	-0.49% (0" 1)
successful search sessions	passive job seeker	+1.24% (0" 02)
	active job seeker	-0.52% (0" 14)

CTR-r also improved 0.3% (p-value 0.02) compared with the non-personalized Seq2Seq model. This result shows that personalized query suggestions are more engaging and improve the overall user search experience.

At LinkedIn, we de ne a search session by the start of a new search query on the homepage or a time gap of user-inactivity. A successful session is one in which the user performed some meaningful action, such as saving a job or applying for a job. The number of successful search sessions is one fundamental business metric for LinkedIn. We classify users into active and passive job seekers based on their recent job-related activities, such as applying for a job. Moreover, we are interested in how changes in the search engine impact job seeker behaviors. Table 3 shows how personalization impacts di erent groups of job seekers.

Notably, personalization has a more signi cant impact on passive job seekers. We believe passive job seekers are more easily encouraged to follow the personalized query suggestions to start new searches because those suggestions are more relevant to their background and intent. For the same reason, those query suggestions can retrieve more attractive search results, leading to a better overall search experience. Active job seekers, on the other hand, may already have a good idea for what they want. The query suggestions provided by our model, therefore, has less impact on them.

5 ONLINE SERVING STRATEGY

A signi cant challenge for applying a DNN to production online inference is the large latency it incurs. However, in our particular application, the Q.S. service and the search retrieval and ranking process can be parallelized because they are independent of each other. We also added an in-memory cache to help reduce latency because the same input source query should always output the same target query suggestions for the same model. Figure 4 shows our online serving architecture. This strategy worked well: With tens of millions of requests served per day, we achieved an average latency of 23 ms and a 99^{C} percentile of 70 ms.

6 CONCLUSIONS

In this work, we presented a query suggestion framework based on recurrent neural networks. The framework models both structured user features as well as unstructured text data. In our o ine experiments, we showed that modeling both data gave a better performance than modeling text data alone. We further showed that treating the categorical user-feature as an additional vocabulary not only was straightforward to implement but also gave better performance compared with concatenating the user feature embedding to word embeddings.

Figure 4: Online serving architecture of Q.S. service.

A/B test on live tra c con rmed our o ine experiment observations. Personalization boosted core business metrics for LinkedIn search. Di erent user segments bene ted di erently from personalized query suggestions. We showed that passive job seekers are more likely to bene t from better query suggestions experience.

This framework is a production-grade, deep-learning based query suggestion framework. The online serving strategy presented in this work enabled us to make real-time inferences from deep learning models. Our system is general and can be applied to other search engines as well.

REFERENCES

[1] Hiteshwar Kumar Azad and Akshay Deepak. 2017. Query expansion techniques for information retrieval: A survey. Information Processing & Management 56, 5 (Aug 2017), 1698 1735.

[2] Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Johann Daigremont. 2011. Personalized social query expansion using social bookmarking systems. In SIGIR '11. ACM Press, New York, New York, USA, 1113.

[3] Heng-Tze Cheng, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, and Wei Chai. 2016. Wide & Deep Learning for Recommender Systems. In DLRS 2016. ACM Press, New York, New York, USA, 7 10.

[4] Paul Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. 2007. Personalized query expansion for the web. In SIGIR '07. New York, New York, USA.

[5] Donna Harman. 1988. Towards interactive query expansion. In SIGIR '88. ACM Press, New York, New York, USA, 321 331.

[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Computation (1997).

[7] Aaron Jaech and Mari Ostendorf. 2018. Low-Rank RNN Adaptation for Context-Aware Language Modeling. Transactions of the Association for Computational Linguistics 6 (Dec 2018), 497 510.

[8] J. Jayanthi, K. S. Jayakumar, and B. Akalya. 2011. Personalized Query Expansion based on phrases semantic similarity. In ICECT '11. IEEE, 273 277.

[9] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macdu Hughes, and Je rey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. Transactions of the Association for Computational Linguistics 5 (Dec 2017), 339 351.

[10] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. E ective Approaches to Attention-based Neural Machine Translation. In EMNLP '15 (Aug 2015), 1412 1421.

[11] Qiaozhu Mei, Dengyong Zhou, and Kenneth Church. 2008. Query suggestion using hitting time. In CIKM '08. ACM Press, New York, New York, USA, 469.

[12] Tomas Mikolov and Geo rey Zweig. 2012. Context dependent recurrent neural network language model. In SLT '12. IEEE, 234 239.

[13] Mandar Mitra, Amit Singhal, and Chris Buckley. 1998. Improving automatic query expansion. In SIGIR '98. ACM Press, New York, New York, USA, 206 214.

[14] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob G. Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder For Generative Context-Aware Query Suggestion. In CIKM '15 (Jul 2015), 553 562.

[15] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In NIPS '14 (Sep 2014).

[16] Zhengyu Zhu, Jingqiu Xu, Xiang Ren, Yunyan Tian, and Lipei Li. 2007. Query Expansion Based on a Personalized Web Search Model. In SKG 2007. IEEE, 128 133.