Weekly reading report

This week I read 4 papers about QAC.

[1] is addressing the issue that QAC system does not suggest queries for previously unseen text. Author applies recurrent neural network and it results in superior effectiveness for previously unseen queries. In [1], author lists some most-common approach for QAC such as MostPopularCompletion (MPC), and point out the weakness of being blind when queries are not present in the storage. This study, on the contrary to term-level QAC in [2] (cannot handle Out-of-Vocabulary words), is doing character-level QAC. [18] on the other hand, can only suggest queries after receiving a full query. [4] is able to suggest rare prefixes, but they are using n-gram suffix words with reranked results produced by neural network. A vanilla RNN has a vanishing gradient problem, to cope with this issue, author adopt Long short-term memory (LSTM) that augments RNN with a memory cell vector. To deal with the weakness of lack of semantic learning for words, the author incorporate word information into characters. Dropout layer is also used to prevent over-fitting. The paper also presents a beam-search algorithm where each iteration they keep only gamma candidates to be feed into the next round. Their experiment is following the setting in [4] and they conduct a MRR and partial-matching MRR metrics. They choose MPC and the work by Mitra and Craswell [5] to implement as baseline. And they obtain the word-embedded vector with the public-trained vector from Google News. The public AOL query log data set is used. They use training-validation and results show performance is as good as MPC for previously seen queries and outperform it by 43% in MRR for previously unseen queries.

Instead of suggesting the whole query, [2] uses a term-by-term approach which is beneficial when users are in a mobile environment and screen is limited. This paper summarizes related work other than popularity based model. For example, [6] suggests a context-sensitive model that considers the current browsing session. And also time-sensitive approaches which are useful in news search, and methods considering demographics factors. The paper also lists works based on user interactions, such as [7] which learns model from query log of Yandex. This paper defines a query-term graph to model likelihood of a sequence of query prefix. Their method is basically searching in the graph the continuing paths with product of

the probabilities of their edges are among the N highest, the procedure is able to prune candidate whose maximum probability is lower than the current candidate by depth-first search. Their experiment uses Yahoo query log, and their baseline is again most-popular completion. The metrics they are evaluating is basically the number of saved characters and terms from typing. They found that their approach performs better for rare query prefix but overshadowed by MPC for highly popular queries. Besides, they had a user survey on a series of measurements of user effort, such as the number of times they were activated, the ratio of accepted suggestions, the average position of accepted completion, the saved terms, the number of completions showed to users before finding the right one, as well as the number of saved characters.

I also read [8], mainly using Convolutional Latent Semantic Model and its property of finding query reformulations similarity by offset of vector. The paper mentions a few other Latent semantic models, such as Latent Semantic Analysis, Latent Dirichlet Allocation, and Neural network Semantic Hashing. They all help to infer more contextually relevant query suggestions, this benefits is good at understanding the actual user intent when only a few characters have been entered which will make MPC perform poorly. Their work shows a MRR improvement by 10% using dataset from Bing and AOL logs. Their CLSM model architecture adopt from [9] where each word is mapped to a feature vector corresponding to a trigram, then this feature is feed into convolutional layer to extract contextual features defined by its immediate neighbours with predetermined window size. They train the data based on succession query pair in user session in addition to clickthrough data. The validation data is constructed by splitting the query at a randomly selected position, for each prefix, a positive relevance judgement is made. They found models trained on session pairs perform better than those trained on clickthrough data, and supervised learning-to-rank models perform better than those trained with similarity features alone. This paper also observes many biases on dataset limitations such as available query logs are likely produced by QAC system also.

[9] is what [8] is based on, so I also read [9]. This paper adopts a convolutional deep structured semantic models which projects a context window to contextual feature, and it adds a max polling layer to extract the most salient local features. The word hashing feature vectors within a window are concatenated to form a

context window vector, and through convolutional layer, they are projected to a local contextual feature vector. Their experiments are reported using NDCG (Normalized Discounted Cumulative Gain), the stated method outperforms by a significant margin indicated by NDCG@1,3 and 10.

1 Reference

[1] Park, Dae Hoon, and Rikio Chiba. A Neural Language Model for Query Auto-Completion. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 11891192. SIGIR 17. New York, NY, USA: ACM, 2017. https://doi.org/10.1145/3077136.3080758.

[2] Term-by-Term Query Auto-Completion for Mobile Search by Sal Vargas.

[3] Szpektor, Idan, Aristides Gionis, and Yoelle Maarek. Improving Recommendation for Long-Tail Queries via Templates. In Proceedings of the 20th International Conference on World Wide Web, 4756. WWW 11. New York, NY, USA: ACM, 2011. https://doi.org/10.1145/1963405.1963416.

[4] Shokouhi, Milad. Learning to Personalize Query Auto-Completion. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, 103112. SIGIR 13. New York, NY, USA: ACM, 2013. https://doi.org/10.1145/2484028.2484076.

[5] Mitra, Bhaskar, and Nick Craswell. Query Auto-Completion for Rare Prefixes. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 17551758. CIKM 15. New York, NY, USA: ACM, 2015. https://doi.org/10.1145/2806416.2806599.

[6] Schmidt, Andreas, Johannes Hoffart, Dragan Milchevski, and Gerhard Weikum. Context-Sensitive Auto-Completion for Searching with Entities and Categories. In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, 10971100. SIGIR 16. Pisa, Italy: ACM, 2016. https://doi.org/10.1145/2911451.2911461.

[7] Kharitonov, Eugene, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. User Model-Based Metrics for Offline Query Suggestion Evaluation. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, 633642. SIGIR 13. New York, NY, USA: ACM, 2013. https://doi.org/10.1145/2484028.2484041.

[8] Mitra, Bhaskar. Exploring Session Context Using Distributed Representations of Queries and Reformulations. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 312. SIGIR 15. New York, NY, USA: ACM, 2015. https://doi.org/10.1145/2766462.2767702.
[9] Shen, Yelong, Xiaodong He, Jianfeng Gao, Li Deng, and Grgoire Mesnil. Learning Semantic Representations Using Convolutional Neural Networks for Web Search. In Proceedings of the 23rd International Conference on World Wide Web, 373374. WWW 14 Companion. New York, NY, USA: ACM, 2014. https://doi.org/10.1145/2567948.