SCIENCE CHINA
Information Sciences

• LETTER •

November 2016, Vol. 59 119101:1–119101:3
doi: 10.1007/s11432-016-5531-y

A greedy selection approach for query suggestion diversification in search systems

Fei CAI[1,2]*, Honghui CHEN[1] & Zhen SHU[1]

[1]*Science and Technology on Information Systems Engineering Laboratory,*
National University of Defense Technology, Changsha 410073*, China;*
[2]*University of Amsterdam, Amsterdam* 1098XH*, The Netherlands*

Received October 8, 2015; accepted November 24, 2015; published online May 24, 2016

Citation Cai F, Chen H H, Shu Z. A greedy selection approach for query suggestion diversification in search systems. Sci China Inf Sci, 2016, 59(11): 119101, doi: 10.1007/s11432-016-5531-y

Dear editor,

Query suggestion is a widely known and well embedded mechanism in modern search systems to facilitate the task of formulating queries. The goal is to accurately predict user's intended query with only few keystrokes (i.e., typed prefix), thereby helping the user formulate a satisfactory query with less effort and avoid misspelling it as well. Suggested candidates are typically generated based on previous queries that have been submitted to the search systems. Such related approaches have attracted increasing interests from the research community with proposals that take personalization [1, 2], time-awareness [3, 4] and user behaviors into account in a query suggestion framework.

However, limited by the length of query suggestion list, i.e., the number of returned query candidates by the search engine, e.g., 8 for Bing and 4 for Google, user's satisfaction cannot be guaranteed, especially for short prefixes with a large population of ambiguous and redundant query candidates. Therefore, without additional information to disambiguate user's search intent, the search engine has to focus on how best to produce a set of most potential and diversified query candidates that covers different interpretations to maximize the probability of satisfying the general population of user typing the prefix. By doing so, the possibility that any user typing the same prefix may find at least one satisfactory query candidate to their particular information need is maximized. Hence, how to capture user's search intent and reduce the redundancy of query completions for precisely predicting the query when typing the prefix is challenging.

In this letter, we propose a greedy query selection (GQS) approach which aims to return the correct query early in the candidate list and reduce the redundancy of the list as well, where the search intents related to the current search popularity and implicitly expressed by previous search behaviors are considered. We identify a query's search intents using the clickthrough data via categorizing its clicked URLs with the ODP taxonomy[1)], which is a topical hierarchy structure for URL categorization.

Methodology. We first illustrate the problem of query suggestion diversification. Given prefix p of the last query q_T in a search session consisting of T queries: $\{q_1, q_2, \ldots, q_T\}$, an initial query ranking list R_I produced for this prefix p with size $|R_I| = k_I$, a probability distribution of search intents i for the prefix $P(i|p, C)$ giv-

*Corresponding author (email: f.cai@uva.nl)

The authors declare that they have no conflict of interest.

1) http://www.dmoz.org.

en the search context C consisting of a sequence of preceding queries before q_T: $\{q_1, q_2, \ldots, q_{T-1}\}$, and the satisfaction values of the query candidates $S_v(q_c|p, i, C)$ matching the search intent i. We try to find a reranked list of query candidates R_R with $|R_R| = k_R$, such that $k_R = k_I$, which maximizes

$$P(R_R|p, C) = \sum_i P(i|p, C)(1 - \prod_{q_c \in R_R}(1 - S_v(q_c|p, i, C))).$$

This objective can be achieved by iteratively filling R_R with one query $q^\star \in R_I \backslash R_R$ each time until $|R_R| = k_R$ by the criterion:

$$\arg \max_{q_c \in R_I \backslash R_R} \sum_i P(q_c|p, i, C) \prod_{q_s \in R_R}(1 - P(i|p, q_s, C)).$$

We assume the probability that a query candidate q_c meets the search intent, i.e., $P(q_c|p, i, C)$, can be attributed to the search popularity or expressed by the closeness to previous queries in the session context C, with λ ($0 \leqslant \lambda \leqslant 1$) controlling the tradeoff between these two parts, thus

$$\begin{aligned} & P(q_c|p, i, C) \\ & \quad = \lambda P(q_c|p) + (1-\lambda) P(q_c|i, C) \\ & \quad = \lambda P(q_c|p) + (1-\lambda) \prod_{q_t \in C} P(q_c|i, q_t), \end{aligned} \tag{1}$$

where we further assume that queries are independent to each other. The search intent related to search popularity $P(q_c|p)$ can be directly estimated from the search logs by

$$P(q_c|p) = \frac{f(q_c)}{\sum_{q \in R_I} f(q)}, \tag{2}$$

with $f(q)$ indicating the frequency of query $q \in R_I$. The probability $P(q_c|i, q_t)$ in (1) can be directly estimated by the normalized distance between q_c and q_p given a specific intent i, weighted by the temporal intervals:

$$P(q_c|i, q_t) = \omega_t \times \left(1 - \frac{|q_c(i) - q_t(i)|}{\mathrm{dis}(q_c, q_t)}\right), \tag{3}$$

where $\mathrm{dis}(q_c, q_t)$ returns the 2-norm distance between q_c and q_t, and ω_t is a normalized decay brought by the temporal interval between q_t and q_c (or q_T) to make $\sum \omega_t = 1$, as temporally close queries in a search session are apt to share common search intents. We compute ω_t as $\omega_t \leftarrow \mathrm{norm}(f^{\mathrm{TD}(q_t)-1})$, where f is a decay factor and $\mathrm{TD}(q_t)$ refers to the time interval, e.g., $\mathrm{TD}(q_t) = 1$ for the last query q_{T-1} in the context C. The queries, e.g., q_c and q_t, can be represented by a distribution over aspects returned by matrix factorization. Hence these probabilities can be computed offline before ranking.

The probability $P(i|p, q_s, C)$ indicates to what degree the selected query candidate $q_s \in R_R$ meets the search intent, which can be learnt from the search logs. Following the assumption mentioned in [5] that the intents are independent to the typed prefix, we simplify $P(i|p, q_s, C)$ as

$$P(i|p, q_s, C) = P(i|q_s, C), \tag{4}$$

indicating the probability that a query candidate matches the search intent is dominated by the closeness to the intent of preceding queries in the session. Finally, we have

$$P(i|p, q_s, C) \propto \prod_{q_t \in C} P(q_s|i, q_t), \tag{5}$$

as $P(i|q_s, q_t) \propto P(q_s|i, q_t)$, where $P(q_s|i, q_t)$ can be equally derived as $P(q_c|i, q_t)$ in (1). By doing so, we can gradually inject one query to the list R_R at one time until the size $|R_R|$ is matched.

In practice, we initialize R_R, i.e., we fix a query candidate to start with, which is achieved by $R_R \leftarrow q_*$, where q_* is a candidate, receiving the highest score as follows:

$$\mathrm{Score}(q_c) = \gamma \cdot \mathrm{MPC}(q_c) + (1-\gamma) \cdot \mathrm{Sem}(q_c),$$

where $\mathrm{MPC}(q_c)$ depends on the popularity and $\mathrm{Sem}(q_c)$ relies on the semantic similarity to search context measured by word2vec [6]. As $\mathrm{MPC}(q_c)$ and $\mathrm{Sem}(q_c)$ use different units and scales, they need to be standardized before being combined. We standardize $\mathrm{MPC}(q_c)$ as

$$\mathrm{MPC}(q_c) \leftarrow \frac{f(q_c) - \mu_T}{\sigma_T}, \tag{6}$$

where μ_T and σ_T are the mean and standard deviation of popularity of queries in R_I. Similarly, we have

$$\mathrm{Sem}(q_c) \leftarrow \frac{\mathrm{Sem}(q_c) - \mu_s}{\sigma_s}, \tag{7}$$

where μ_s and σ_s are the mean and standard deviation of similarity scores of queries in R_I.

Next, we infer multi-aspect relevance for a query from clickthrough data using ODP. In detail, this scenario consists of two major steps to build a query-aspect matrix. The first step involves constructing the clickthrough data from the log. By doing so, we get a list of all clicked URLs for each unique query. The second step involves labelling these URLs using ODP. After that, we infer the aspects of a query by aggregating all aspects from its clicked URLs. By doing so, we can use a multi-aspect relevance vector corresponding to an aspect relevance label.

Finally, we replace the zeros in the query-aspect matrix which is built for the cases that no direct relationships between query q and aspect a are inferred using ODP. We use Bayesian Probabilistic Matrix Factorization (BPMF) [7] to derive the distribution of queries over all aspects instead. BPM-F can be directly applied to the original query-aspect matrix to generate an approximation matrix, which assigns a non-zero value to each entry in the original query-aspect matrix to overcome the problem of sparseness and zero-probabilities.

Results and discussion. We compare the performance of various models tested on the AOL and MSN datasets including three baselines, i.e., M-PC [2], MMR [8] as well as MPC-R [4], and report the results in Table 1.

Table 1 Performance of various models tested on the AOL and MSN datasets. The results produced by the best performer in each column are boldfaced and those results generated by the best baseline are underlined.

Dataset	Method	MRR	α-nDCG@5	α-nDCG@10
AOL	MPC	.6205	.6906	.7662
	MMR	.6223	.6984	.7713
	MPC-R	.6351	.6957	.7702
	GQS	**.6418**	**.7211**	**.7983**
MSN	MPC	.6341	.7056	.7723
	MMR	.6378	.7102	.7794
	MPC-R	.6408	.7083	.7756
	GQS	**.6670**	**.7371**	**.8117**

Clearly, as shown in Table 1, on the AOL dataset, GQS performs the best among these four methods, both in terms of MRR and α-nDCG. Compared to the MMR model, GQS presents near 3.1% improvement in terms of MRR and around 3.5% improvement in terms of α-nDCG@10. In addition, all the improvements are statistically significant using t-test at level $p = .05$. Compared to MPC-R, GQS shows more than 1% improvement in terms of MRR. However, the MRR improvements of GQS over MPC-R are not significant. In contrast, the improvements in terms of α-nDCG@5 and α-nDCG@10 are both significant at level $p = .05$ using t-test. It could be attributed to that, for some cases, the redundant queries below the final submitted query are removed, which makes no sense to improving MRR scores but helps boost the α-nDCG@10 scores.

In contrast, on the MSN dataset, as seen in Table 1, different from the observations on the AOL dataset, our GQS model shows a bit more improvement over MMR. Particularly, the GQS model reports around 4.5% improvement in terms of MRR and 4.1% improvement in terms of α-nDCG@10 over the MMR model. Both these improvements are statistically significant using t-test at level $p = .01$. However, fewer improvements are made by the GQS model over the MMR model in terms of α-nDCG@5, near 3.5%. It could be explained by the fact that limited redundant query candidates can be found in the top 5 of the candidate list, however, relatively more redundant candidates can be found in the top 10. Compared to the MPC-R model, our GQS model shows a significant improvement at level $\alpha = .05$ in terms of MRR.

Conclusion and future work. In this letter, we propose a greedy query selection (GQS) model to address the query completion diversification task, using the ODP taxonomy to identify aspects of queries. For future work, it would be interesting to further collect users' long term search histories so as to boost the performance.

Acknowledgements This research was partially supported by Innovation Foundation of NUDT for Postgraduate (Grant No. B130503).

References

1 Shokouhi M. Learning to personalize query auto-completion. In: Proceedings of 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, 2013. 103–112

2 Bar-Yossef Z, Kraus N. Context-sensitive query auto-completion. In: Proceedings of 20th International World Wide Web Conference, Hyderabad, 2011. 107–116

3 Shokouhi M, Radinsky K. Time-sensitive query auto-completion. In: Proceedings of 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, 2012. 601–610

4 Whiting S, Jose J M. Recent and robust query auto-completion. In: Proceedings of 23rd International World Wide Web Conference, Seoul, 2014. 971–982

5 Bennett P N, Svore K, Dumais S T. Classification-enhanced ranking. In: Proceedings of 19th International World Wide Web Conference, Raleigh, 2010. 111–120

6 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of Advances in Neural Information Processing Systems, Lake Tahoe, 2013. 3111–3119

7 Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov Chain Monte Carlo. In: Proceedings of 25th International Conference on Machine Learning, Helsinki, 2008. 880–887

8 Carbonell J, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, 1998. 335–336