Measuring Usefulness of Context for Context-Aware Ranking

Andrey Kustarev, Yury Ustinovsky, Pavel Serdyukov
Yandex
Leo Tolstoy st. 16, Moscow, Russia
kustarev@yandex-team.ru, yuraust@yandex-team.ru, pavser@yandex-team.ru

Abstract

Most of major search engines develop different types of personalisation of search results. Personalisation includes deriving user's long-term preferences, query disambiguation etc. User sessions provide very powerful tool commonly used for these problems. In this paper we focus on personalisation based on context-aware reranking. We implement a machine learning framework to approach this problem and study importance of different types of features. We stress that features concerning temporal and context relatedness of queries along with features relied on user's actions are most important and play crucial role for this type of personalisation.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Search Process

General Terms: Algorithms, Experimentation.

Keywords: Context-Aware Ranking, Learning to Rank, Query History

1. INTRODUCTION

Personalisation of search results and context-sensitive ranking are challenging problems for developing search engines. Most of known approaches to such kind of problems rely on click-through logs — a rich source of implicit feedback from user. Some of them rely on the session context of the current ambiguous query. Shen *et al.*[1] were one of the first who proposed a method incorporating preceding queries and their clickthroughs. They built several context-sensitive language models relying on user's query history and texts of snippets or titles of clicked documents. Xiang *et al.* [2] have incorporated learning to rank approach into Shen's initial idea. For every query q they store all queries and clickthrough data from the same session. Then they learn a model using click-based and text-based features extracted from whole session and verify it on both human labeled and user click data. Since query log is important instrument in personalisation and context detection it is important to extract logical sessions from the raw query logs. In this setting one assumes that any two queries are either related by similar information need, or not related at all. In paper [3] Boldi *et al.* solve the problem of segmentation of raw query log into sets of related queries. To any pair of queries they assign a weight, measuring their context proximity. These weights are learned on manually labeled data and from a set of features.

In this work we address the problem of context-aware ranking and consider the paper [2] as the groundwork. We implicitly extract user's preferences and context of current query q from the single previous query q_0 in the same logical session and user's actions on the search engine retrieval page for query q_0. More precisely we learn a function $f(q, d; q_0)$, which predicts a click on a document d for a query q relying on these data. Our data collection consist of pairs of queries from the same logical session, which went through classifier from [3].

Our main contribution are 1) employment of logical sessions for construction of the test and training sets; 2) implementation of query-relatedness features in context-aware ranking problem. We measure importance of each feature and demonstrate that features based on query relatedness are among the most important.

2. LEARNING TO RANK FRAMEWORK

Most of known personalisation approaches are based on reranking approach (see [2], [4], [5]), which is very popular due to simplicity of its implementation and transparency of ranking process. In general this approach reorders the set of documents retrieved from search engine relying on some new ranking function (in our case user's preferences oriented function). All these methods consist of 3 stages: dataset preparation, learning to rank process, reranking process. Further we describe each of them in our setting.

Each sample in our learning set corresponds to a triple: query q; document d retrieved for a query q; a query q_0 preceding q in the same logical session. It contains the following data: 1) target function $c(q, d)$ — indicates whether document d was clicked for query q; 2) feature set based on query q, document d and user's actions on q_0.

We emphasize, that pairs of context-related queries in this set were picked out throughout 24-hours on the basis of classifier (see [3]), which says whether two queries have the same information need. This process allows us to throw out from learning set unconnected pairs of consecutive queries. We believe that this purification of raw query logs leads to improvement in quality of training set and simplifies learning process. At the same time it makes our baseline stronger than in [2], as long as the more strongly related queries are in the session the less important it should be to distinguish them by usefulness for the current query. However, as we further demonstrate, it still appears to be quite beneficial.

For learning a context-oriented ranking function we have

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

implemented three types of features listed below (whole list of features is contained in Table 2). Our dataset is based on the dataset implemented in [2] with the following differences: (1) text relevance features are computed relying only on snippet and title of document d and not on its URL (since almost all our queries are in Russian); (2) query-relatedness features are added. The point is that these query-relatedness features are expected to be vital for quality of ranking. All our features (excluding the position of the document in the original ranking) could be divided into three sets (here $q_0 \backslash q$, $q \backslash q_0$ are sets of removed and added words resp.):

F1. User's actions on the SERP of query q_0

F2. Text relevance between d and q_0, q, $q_0 \backslash q$, $q \backslash q_0$

F3. Relation between queries q and q_0

Given the training set we learn a gradient boosting decision tree model. This model approximates target function — $c(q, d)$, minimizing mean square error. The function $f(q, d)$ we have learned generates some new ranking $R_1(d)$ of top 10 document retrieved for a query q. We combine the original ranking of search engine $R_0(d)$ and this new ranking following the method proposed in [2]. Namely, we rerank top 10 documents according to scoring function

$$\text{score}(d) = \alpha \frac{1}{R_0(d)} + (1 - \alpha) \frac{1}{R_1(d)},$$

where $R_0(d)$ and $R_1(d)$ are ranks of document d and $\alpha \in [0; 1]$ is a parameter tuned on the validation set.

3. EVALUATION

We have collected raw clickthrough data from Yandex search engine. Pairs of context-related queries were extracted from logs, so each pair is a consecutive pair in a logical session. All in all we have obtained 14K pairs of temporally ordered, related queries q_0, q. Taking top10 documents for each second query q we get 114K samples in training set: documents d shown for the current query q with click data on both the previous query in the session q_0 and the current query q. On these data we run gradient boosting decision tree model to learn a function predicting a click $c(q, d)$.

We use 10-fold cross validation and compute *performance* of each of 3 types of features listed above. Under performance we understand improvement of click metric in comparison with the original ranking. We use the following click metrics: mean click position MCP, mean first click position FCP, mean reciprocal rank of clicks MRR. As a baseline we consider a ranking function learned on *user action* and *text relevance* features, since it models ranking function in paper [2]. We do not implement cosine distances between queries and document, since they are correlated with Jaccard distances and give no additional signal for learning. Relative improvements in comparison with original ranking are reported in Table 1. Here F1, F2, F3 are feature sets listed above, with * we denote significant difference ($p < 0.01$) between methods and the baseline.

Descriptions of all features are given in Table 2. Features are sorted according to their contribution (weighted sum of the underlying binary features) in the final ranking function $f(q, d)$. Evaluation shows that query-relatedness features along with user-action features are among the most important. It is an interesting observation that employment of text relevance features does not increase algorithm's performance significantly, especially considering that authors of [2] do not provide any information about contribution of different types of features to the final ranking model. Improvements are comparable with improvements in other papers on personalisation (see [5],[2]).

Table 1: Feature sets and their performance

Feature sets	MCP	FCP	MRR
F1 + F2 (bl.)	-0.27%	-1.10%	0.15%
F1 + F3	*-0.58%	*-1.62%	*0.47%
F2 + F3	-0.03%	-0.12%	0%
F1 + F2 + F3	*-0.72%	*-1.98%	*0.62%

Table 2: Feature ranks

Rk.	Set	Feature description
1	–	original rank of d
2	F3	number of clicks between q_0 and q
3	F3	time between q_0 and q
4	F3	num. of common URLs on SERP's for q, q_0
5	F2	Jaccard dst. between title of d and q
6	F3	number of words in $q \cap q_0$
7	F1	d was clicked for q_0
8	F1	document d was skiped for q_0
9	F3	number of words in $q \backslash q_0$
10	F3	fraction of common words in q and q_0
11	F3	fraction of common words in q
12	F2	Jaccard dst. between title of d $q \backslash q_0$
13	F2	Jaccard dst. between title of q_0
14	F2	Jaccard dst. between snippet of d and $q \backslash q_0$
15	F2	Jaccard dst. between snippet of d and $q_0 \backslash q$
16	F2	Jaccard dst. between snippet of d and q_0
17	F3	number of words in $q_0 \backslash q$
18	F2	Jaccard dst. between snippet of d and q
19	F3	time between last action on q_0 and q
20	F3	fraction of common words in q_0
21	F2	Jaccard dst. between title of d and $q_0 \backslash q$

4. CONCLUSIONS

In this paper we study machine learning approach to the problem of context-aware ranking. We describe a method of construction learning sets from clickthrough data and show that employment of query-relatedness features significantly improves quality of reranking algorithm. In the future we plan to incorporate long-term user's preferences into the described context-based reranking approach.

5. REFERENCES

[1] X. Shen, Context-Sensitive Information Retrieval Using Implicit Feedback, *proc. of SIGIR*, 2005.

[2] Biao Xiang et al, Context-Aware Ranking in Web Search, *proc. of SIGIR*, 2010.

[3] P. Boldi et al, The Query-flow Graph: Model and Applications, *proc. of CIKM*, 609–617, 2008.

[4] D. Jiang, K. Wai-Ting Leung, W. Ng, Context-Aware Search Personalization with Concept Preference, *proc. of CIKM*, 563–572, 2011.

[5] K. Collins-Thompson et al, Personalizing Web Search Results by Reading Level, *proc. of CIKM*, 2011.