Learning to Personalize Query Auto-Completion

Milad Shokouhi
Microsoft
Cambridge, United Kingdom
milads@microsoft.com

ABSTRACT

Query auto-completion (QAC) is one of the most prominent features of modern search engines. The list of query candidates is generated according to the *prefix* entered by the user in the search box and is updated on each new key stroke. Query prefixes tend to be short and ambiguous, and existing models mostly rely on the past popularity of *matching* candidates for ranking. However, the popularity of certain queries may vary drastically across different demographics and users. For instance, while *instagram* and *imdb* have comparable popularities overall and are both legitimate candidates to show for prefix *i*, the former is noticeably more popular among young female users, and the latter is more likely to be issued by men.

In this paper, we present a supervised framework for personalizing auto-completion ranking. We introduce a novel labelling strategy for generating offline training labels that can be used for learning personalized rankers. We compare the effectiveness of several user-specific and demographic-based features and show that among them, the user's long-term search history and location are the most effective for personalizing auto-completion rankers. We perform our experiments on the publicly available AOL query logs, and also on the larger-scale logs of Bing. The results suggest that supervised rankers enhanced by personalization features can significantly outperform the existing popularity-based baselines, in terms of mean reciprocal rank (MRR) by up to 9%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Query formulation, Web search

General Terms

Algorithms

Keywords

Query auto-completion, autosuggest, Personalized search

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

1. INTRODUCTION

Auto-completion is among the first services that the users interact with as they search and form their queries. Following each new character entered in the query box, search engines filter suggestions that match the updated *prefix*, and suggest the top-ranked candidates to the user. The first step (filtering) is often facilitated by using data structures such as prefix-trees (tries) that allow efficient lookups by prefix matching [Chaudhuri and Kaushik, 2009]. In the second step (ranking), those suggestions that match the prefix are ordered according to their expected *likelihood*. The likelihood values are often approximated with respect to aggregated *past* frequencies [Bar-Yossef and Kraus, 2011] although other approaches that rank suggestions according to their predicted *future* popularities have been also explored [Shokouhi and Radinsky, 2012].

In majority of previous work, the likelihood of QAC suggestions are computed globally and are considered to be the same for all users. Hence for a given prefix, all users are presented with the same set of suggestions. The two exceptions are the work by Bar-Yossef and Kraus [2011], and Weber and Castillo [2010]. Bar-Yossef and Kraus [2011] added a session bias parameter in auto-completion ranking by comparing candidates with the queries recently submitted by the user. However, the notion of likelihood for a query does not vary across users, or demographic groups, plus their work is not applicable on single-query sessions that account for no less than 50% of the search traffic [Jansen et al., 2007]. Weber and Castillo [2010] discussed the differences in query distributions across various demographics and briefly covered query completion by focusing on predicting the *second query term*. In essence, they build a conditional probabilistic model for common phrases based on a set of demographic features. Their model is based on simple aggregation over different demographics and does not address the sparsity issues as more features are added. Weber and Castillo [2010] do not consider any user-specific feature (as their focus is not on personalization), and do not report the results for more general scenarios where only the first few characters of queries are entered.

In this paper, we propose a novel supervised framework for learning to personalize auto-completion rankings. We are motivated by the previous studies that demonstrated that the query likelihoods vary drastically between different demographic groups [Weber and Castillo, 2010] and individual users [Teevan et al., 2011]. Inspired by these observations we develop several features based on users age, gender, location, short- and long-history for personalizing auto-completion

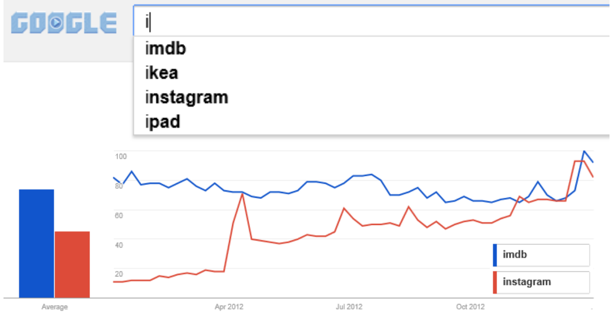


Figure 1: (Top) The default auto-completion candidates for prefix *i*, according to the US market version of `google.com`. The prefix was typed in private browsing mode and the snapshot was taken on Wed, Jan 16, 2013. (Bottom) The query frequencies of *instagram* and *imdb* according to Google Trends.

rankings. For instance, consider the auto-completion candidates returned by Google for prefix *i* in Figure 1 (top).[1] All four suggestions are popular (head) queries with comparable historical frequencies. In particular, the frequency distributions for *instagram*, and *imdb* are demonstrated in Figure 1 (bottom) according to Google Trends.[2] The depicted trends suggest similar likelihoods for both *imdb* and *instagram*, although the popularity of the latter is rising.

In general, in the absence of any information about the user, the ranking of QAC candidates in Figure 1 look reasonable – although models based on temporal query frequency trends [Shokouhi and Radinsky, 2012] may boost the position of *instagram*. The question we are addressing in this work, is how this ranking can be further improved if there are some additional information available about the user. Figure 2 (top) compares the likelihood of *instagram* and *imdb* among different demographics of users according to Yahoo! Clues.[3] At the bottom of Figure 2 the same analysis is repeated using the query logs of Bing search engine which we use as one of the testbeds in our experiments. The overall trends are remarkably similar; *instagram* is mostly popular among young female users below the age of 25. In contrast, the query *imdb* is issued more often by male users particularly those between the age of 25 to 44. Hence, going back to Figure 1, if we knew that the person issuing the query was an under-25 female user, perhaps the original order of QAC candidates could be improved by boosting *instagram*. Then again, if the previous query submitted by the user in the session was about *ipad covers* boosting *ipad* could be possibly better. We investigate how such additional information can be used in a supervised framework for personalizing auto-completion. To train personalized auto-completion rankers, we introduce a new strategy for generating training labels from previous queries in the logs. Our experiments on two large-scale query logs suggest that integrating demographic and personalized features can significantly improve the effectiveness of auto-completion.

The remainder of this paper is organized as follows; we continue by covering the related work in Section 2. Our new

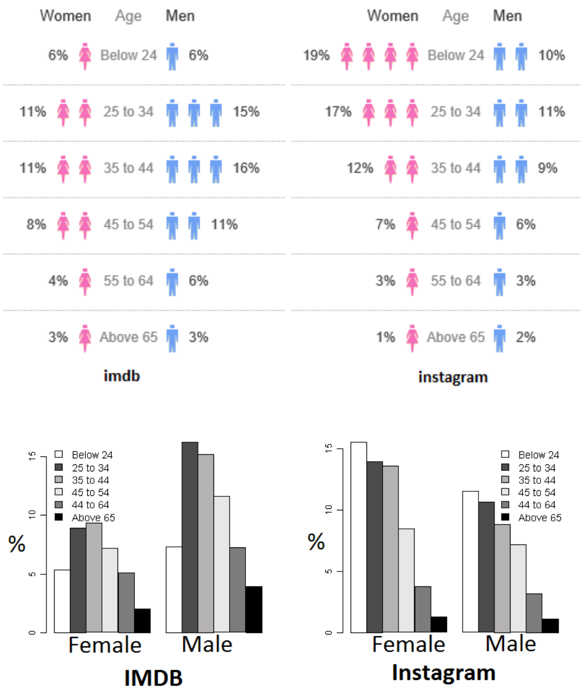


Figure 2: (Top) The likelihood of *instagram* and *imdb* in queries submitted by different demographics according to Yahoo! Clues. (Bottom) The likelihood of *instagram* and *imdb* in queries submitted by the logged-in users of Bing.

framework for personalized auto-completion is described in Section 3. Section 4 discusses our testbed data, and the features used for personalization. The evaluation results are presented in Section 5. Finally, we conclude in Section 6 and suggest a few directions for future work.

2. RELATED WORK

Query auto-completion. Auto-completion has been widely adopted in most modern text editors, browsers and search engines. In *predictive auto-completion* systems, the candidates are matched against the prefix on-the-fly using information retrieval and NLP techniques [Darragh et al., 1990; Grabski and Scheffer, 2004; Nandi and Jagadish, 2007]. For example, Grabski and Scheffer [2004] deployed an index-based retrieval algorithm and a cluster-based approach for their sentence-completion task. Bickel et al. [2005] learned a linearly interpolated n-gram model for sentence completion. Fan et al. [2010] proposed a generative model that incorporates the topical coherence of terms based on Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. White and Marchionini [2007] proposed a real-time query expansion model that generates new expansion terms as the user types in search box. The results suggested that their real-time query-expansion system helps users to form better queries for their information needs. Bhatia et al. [2011] mined frequently occurring phrases and n-grams from text collections and deployed them for generating and ranking auto-completion candidates for partial queries in the absence of search logs.

In *pre-computed auto-completion* systems, the list of match-

[1] The prefix was submitted to the US market version of `google.com` on January 16, 2013, in private browsing mode.

[2] `http://www.google.com/trends`

[3] `http://clues.yahoo.com`, discontinued in March 2013

ing candidates for each prefix are generated in advance and stored in efficient data structures such as prefix-trees (trie) for fast *lookups*. As the user types more characters, the list of candidates is updated by exact prefix matching although more relaxed lookups based on fuzzy matching have been also explored [Chaudhuri and Kaushik, 2009; Ji et al., 2009].

Once the matching candidates are filtered, they can be ranked according to different criteria. For instance, in an online store such as `amazon.com`, suggestions may be ordered according to price or review scores of products [Chaudhuri and Kaushik, 2009]. In web search scenarios, the common approach is to rank suggestions according their past popularity. Bar-Yossef and Kraus [2011] referred to this type of ranking as the *MostPopularCompletion* (MPC) model and argued that it can be regarded as an approximate maximum likelihood estimator. Given a search log of previous queries $\mathcal{Q}$, a prefix $\mathcal{P}$, and the list of query-completion candidates that match this prefix $\mathcal{C}(\mathcal{P})$, the MPC algorithm is essentially applying the following *Maximum Likelihood Estimation* for ranking:

$$MPC(\mathcal{P}) = \arg\max_{q \in \mathcal{C}(\mathcal{P})} w(q), \quad w(q) = \frac{f(q)}{\sum_{i \in \mathcal{Q}} f(i)} \qquad (1)$$

Here, $f(q)$ represents the past query frequency for q in the $\mathcal{Q}$ logs. Shokouhi and Radinsky [2012] later extended the MPC model and replaced the past frequency values $f(q)$ in Equation (1) with predicted frequency values $\hat{f}(q)$. They showed that the predicted values produced by applying time-series on query history are more effective for ranking auto-completion candidates. Strizhevskaya et al. [2012] also modelled the frequency trends of queries by time-series for improving the auto-completion ranking.

In the context of query auto-completion, the closest studies to ours are done by Bar-Yossef and Kraus [2011] and Weber and Castillo [2010]. The *NearCompletion* method [Bar-Yossef and Kraus, 2011] considers the user's recent queries as *context* and takes into account the similarity of QAC candidates with this context for ranking. Their hybrid model computes the final score of each candidate by linearly combining the popularity-based (MPC) and context-similarity scores. We go beyond session-based features and explore the effectiveness of considering users' age, gender, location and longer search history in auto-completion ranking. Further, in contrast to the NearCompletion model that uses a linear combination of two features for ranking, we propose a supervised framework for ranking, and define a novel objective function for optimizing it. It is also worth noting that while the NearCompletion approach is not applicable to single-query search sessions (more than 50% of traffic [Jansen et al., 2007]), our personalized model can be applied on all search queries. Weber and Castillo [2010] mostly focused on showing differences in query likelihoods across different demographics. They briefly discussed a special case of auto-completion for predicting the second term in a query based on an unsupervised probabilistic model generated according to phrase counts across different demographics. They did not discuss the effectiveness of individual features (e.g. age versus gender), and did not report any results for more general cases where only the first few characters of queries are available. Their work is still based on *aggregation* over different demographic groups and they do not address how such features can be combined with other user-specific features (e.g. session history) in a unified framework for ranking auto-completion candidates.

Query Suggestion. Query suggestion and auto-completion are closely related. Apart from differences in *matching* that are largely imposed by stricter latency constrains in auto-completion, the main distinction is in the type of user input – a query in query suggestion and a prefix in auto-completion. Query prefixes are by definition shorter than submitted queries and hence they are more ambiguous. Thus, the potential for disambiguation using personalized features is arguably higher in auto-completion. Nevertheless, the problems are sufficiently similar that covering some of the related work on query suggestions might be worthwhile.

Mei et al. [2008] generated query suggestions by running a random-walk on a bipartite click graph. They suggested that personalized subsets of this bipartite graph can be used for generating personalized suggestions. Song and He [2010] combined both click and skip information in random-walk for improving estimations on relatedness of queries. To remedy the data sparsity issues in click graphs, Cao et al. [2008] and Liao et al. [2011] first clustered the queries in the click graph into a smaller set of virtual *concepts*. In the second step, they matched the users' context captured based on their recent queries against these clusters for ranking query suggestions. Guo et al. [2011] proposed a similar two-step approach, in which the user's session context is matched against pre-generated topic models for ranking query suggestions. Song et al. [2011] re-ranked the original order of query suggestions promoting those that increase diversity and return documents from different topics.

Santos et al. [2012] extracted queries that frequently co-appeared in the same sessions to generate query suggestions. They took a learning-to-rank approach and evaluated the quality of suggestions with respect to their performance at ranking relevant documents. In a similar vein, Liu et al. [2012] learned to rank query suggestions based on their predicted performance focusing on difficult queries mainly. Ozertem et al. [2012] also presented a learning-to-rank framework for ranking query suggestions. They regarded the query co-occurrences in search logs as positive examples and trained a model based on lexical (e.g. edit-distance) and result set features (e.g. number of overlapping URLs). In the same manner Reda et al. [2012] combined several lexical and result set features for query suggestion on LinkedIn.[4] Song et al. [2012] built a term-transition graph from search logs and used it to learn a model for generating query suggestions by term replacement.

Personalized search. Our study is also related to a large body of previous work on search personalization. Noteworthy among them, Bennett et al. [2012b] investigated how short-term and long-term user behavior interact, and how can they be used for personalizing search results. They observed that long term history is particularly useful at the beginning of search session, while short-term history becomes more useful as the session evolves. Teevan et al. [2005] indexed all information copied or viewed by users over a limited period to form their profiles, and used that information to personalize (re-rank) their search results. Comparably Matthijs and Radlinski [2011] collected users browsing his-

[4] `http://www.linkedin.com`

Table 1: The process of assigning labels to auto-completion candidates offline. For a given query submitted by the user – in this case *australian open* – all prefixes are first extracted. For each prefix $\mathcal{P}$ the top-ranked candidates matched in the auto-completion trie are collected. The query which was eventually submitted by the user is considered as the only right (relevant) candidate and is assigned a positive label. The other candidates are all regarded as non-relevant and get zero labels. In the example below, the first line represents the prefix while c_1, c_2, c_3 and c_4 respectively denote the top-4 auto-completion candidates returned for the prefix. The relevant candidate in each list is specified by a checkmark (✓), and the Mean-Reciprocal-Rank (MRR) values computed with respect to these offline labels are presented in the last row.

$\mathcal{P}$	a	⇒	au	⇒	aus	···	australian open
c_1	amazon		autotrader		australia		australian open ✓
c_2	alaska airlines		autozone		austerity		australian open 2013
c_3	apple		audacity		australian open ✓		australian open tennis
c_4	aol		audible		australian shepherd		australian open 2012
MRR	0.00		0.00		0.33		1.00

tory by a browser add-on and used the language model of captured pages for personalizing search results.

Teevan et al. [2011] analysed the result re-visitation pattern of users and classified at least 15% of all clicks as *personal navigation* in which the user repeatedly searches for the same page. They showed that boosting the position of personal navigation pages in search results is effective for personalization. Xiang et al. [2010] captured the users' context according to their latest queries and used those contextual features for ranking the results of subsequent queries in the session. Cheng and Cantú-Paz [2010] used demographic-based and user-specific features for better click-prediction and personalizing the ranking of sponsored search results. Similarly, Kharitonov and Serdyukov [2012] used users age and gender for re-ranking and personalizing search results.

In the next section, we introduce a novel approach for personalizing query auto-completion. To the best of our knowledge, this is the first study that applies a supervised model for QAC ranking. We are also unaware of any other work that considers user-specific and demographic features for personalizing auto-completion in a unified framework.

3. PERSONALIZED AUTO-COMPLETION

Learning to personalize auto-completion ranking of query suggestions for prefixes, is analogous to learning to personalize search results for queries. In typical learning to rank frameworks in information retrieval [Liu, 2009], the training data consists of a set of *labeled* query-document pairs, and the goal is to learn a ranking model by optimizing a cost function that is expected to be correlated with user satisfaction. The same models can be applied for learning to rank auto-completion rankings. Here, the "query" is a prefix and the goal is learn a model for ranking query candidates ("documents"). The key missing piece is deciding on a strategy for assigning labels to auto-completion candidates for training.

Manual labelling of *relevance* for personalized search scenarios is notoriously difficult. Relevance assessors are often instructed to rate relevance according to the most likely intent(s). However as described earlier, the most likely intent may differ between users and demographic groups. To overcome this issue, Fox et al. [2005] introduced a new scheme for inferring personalized labels according to user implicit feedback which was later widely adopted for training personalized rankers [Bennett et al., 2011, 2012a; Collins-Thompson et al., 2011; Kharitonov and Serdyukov, 2012]. First, a set of unique *impressions* are sampled from the logs. Each impression consists of a unique user-ID, a time-stamp, the submitted query, and the set of results presented to the user along with information about implicit measures such as clicks and dwell time on those results. In the second step, documents that received a satisfied click (SAT clicks) are annotated with relevant labels and others are regarded as irrelevant. The definition of a SAT-Click could be subjective, but it is typically assumed that the last result click in the session, or clicks with longer than 30 seconds dwell time are SAT [Bennett et al., 2011; Fox et al., 2005].

We propose a similar labelling strategy for personalized auto-completion; we start by sampling a set of impressions from search logs. For each sampled impression, we assume that the query that was eventually submitted by the user is the only *right* (or the most relevant) suggestion that should have been suggested right after the first key-stroke and all the way until submission. With this assumption in mind, we decompose each sampled query into all prefixes that lead to it and for each case we obtain all query candidates that match in the auto-completion trie. In practice – and also for all experiments in this paper – we can restrict the list of candidates to the top-ranked auto-completion suggestions that are returned by a default context-free model such as MPC [Bar-Yossef and Kraus, 2011]. For each pair of prefix and auto-completion list constructed this way, we assign positive label to the query submitted by the user at the end (if it appears in the list) and zero label to others.

Table 1 provides an example: here we sampled an impression from search logs in which the user had submitted *australian open* as query. We split the query into prefixes and for each case obtain the top-ranked candidates from an auto-completion trie.[5] For each prefix, the only suggestion that is assigned a positive label is specified with a checkmark (✓). The last row contains the Mean-Reciprocal-Rank (MRR) value of each ranking computed according to the assigned labels. Note that for the same user-prefix pair, depending on the submitted query, the candidate labels may vary in different impressions. For example, if the same user issues

[5] The candidates in this example were obtained from `google.com` in private browsing mode on January 19, 2013.

amazon as query in another impression, with the exception of *amazon* all QAC candidates including *australian open* will be regarded as non-relevant for that impression.

Learning to rank. Once the training data is collected as described above, we can apply virtually any existing learning-to-rank algorithm for training a personalized auto-completion ranker. We chose Lambda-MART [Burges et al., 2011] as our learning algorithm. Lambda-MART is an extension of Lambda-Rank [Burges et al., 2006] based on boosted decision trees. It is one of the most effective state-of-the-art learning algorithms, and was chosen as the winner of the Yahoo! 2010 *Learning to Rank Challenge* (Track 1).[6] We used a fixed number of 200 trees across all experiments and tuned the learning parameters through standard training and validation on separate sets.

We consider the MostPopularCompletion (MPC) method [Bar-Yossef and Kraus, 2011] that ranks candidates according to their past popularity as our baseline. Other extensions of MPC such as the time-sensitive models [Shokouhi and Radinsky, 2012] are orthogonal to our technique and can be used alternatively without loss of generality. Any potential gains from temporal modelling is expected to also benefit our framework. We use the top-10 candidates returned by the MPC model as input to our ranker. Therefore in all experiments the rankers are compared against exactly the same set of candidates and gains and losses are solely due to personalized re-ranking. Given that there is only one relevant candidate per prefix in each impression, we use the Mean-Reciprocal-Rank of relevant candidates as our evaluation metric and report the average values over all our sampled impressions.

4. DATA & FEATURES

We conducted our experiments on two sets of query logs, one publicly available sampled from AOL search logs in 2006 [Pass et al., 2006], and one proprietary dataset consisting of queries sampled from the logs of Bing search engine. We respectively refer to these two datasets as AOL, and Bing testbeds hereafter.

AOL testbed. The queries in this dataset were sampled between 1 March, 2006 and 31 May, 2006. In total there are 16,946,938 queries submitted by 657,426 unique users. Each query has a time-stamp and we follow the common practice [Jansen et al., 2007] and group queries submitted within 30 minute window of each other to form sessions.

We used the queries submitted before 15 April 2006 as *background data* for generating the tries and forming the long-term history of users. The remaining data was split into two sets according to the user IDs. Users with even IDs were grouped together for training and validation, and those with odd IDs were used for testing.

Bing testbed. Our other experimental dataset consists of a sample of several million queries submitted by those Bing users who were *signed-in* with their Microsoft Live account when issuing their queries. The data was collected between 1 January, 2013 and 9 January, 2013. In total there are 196,190 unique users in this dataset, and each user-ID is associated with an age, gender and zip-code based on the user's profile information.[7] As in the previous dataset, we determine the session boundaries by applying the same 30 minute threshold. Sessions initiated before 7 January, 2013 were used as *background data* for generating the tries and forming the long-term history of users. The users in the remaining sessions were split based on their user-IDs – as described above – for training, validation and testing.

In both datasets, we filtered queries that appeared less than 10 times. A lot of these queries were rare, misspelled, and not popular enough to rank high in auto-completion lists anyway. The MPC auto-completion trie which we use as baseline and foundation for personalized re-ranking was constructed after this filtering and contained respectively $128,620$, and $699,862$ unique queries in the AOL and Bing datasets.

Next, we describe the user-specific and demographic-based features that we developed for personalizing auto-completion on these datasets.

User history features. We investigate the effectiveness of features developed based on both short-term and long-term search history of users for personalizing auto-completion. In addition to the raw historical frequency numbers, we also measured the n-gram similarity of auto-completion candidates with the past queries issued by the user. To generate the short-history features, only queries submitted previously in the same session are considered. For long-history, the entire search history of the user is considered. Similar features have been used in previous work for re-ranking search results [Bennett et al., 2012a; Teevan et al., 2011; Xiang et al., 2010] and query suggestions [Cao et al., 2008; Mei et al., 2008].

Demographic features. In the Bing dataset, we have access to users age, gender and zip-codes based on their Microsoft Live profile information. We split the users into five age groups {Below 20, 21-30, 31-40, 41-50, and above 50}. Suppose that for a given user-prefix pair in our testing set we would like to generate the age-specific ranking features for all matching auto-completion candidates; we first assign the user to one of the age groups above according to the profile information. Once the user's age group is determined, for each candidate, we count the number of times it has been issued as a query by all users that belong to the same age group in the *background data*. We follow the same procedure for computing our gender-specific features. We also generated similar location-specific features based on users zip-code information in their profiles. To reduce sparsity, we collapse the US zip-codes into 10 *regions* according to their first digits. Figure 3 depicts the regional groups formed this way in different colors.

MPC features. Our personalized auto-completion ranker is built on top of the MPC algorithm. That is, the matching candidates are first ranked according to their past popularity and the top-ranked ones – that otherwise would be presented directly to the user in the absence of personalization – will go through re-ranking. The original order before personalization provides rich insights about queries popularity that could be valuable to the ranker. Therefore, we use the position of each candidate in the original list, and also

[6] `http://learningtorankchallenge.yahoo.com`

[7] All user-IDs are annoymized in such way that the actual user-names could not be identified.

Table 2: The list of features used in our experiments for personalizing auto-completion. The aggregated features are computed in a cross-validation fashion. That is, to generate the features for candidates in the testing set, the numbers are aggregated over all (or subset of) users in the the training and validation sets.

Feature	Feature Group	Description
PrevQueryNgramSim	Short history	n-gram similarity with the previous query in the session ($n = 3$).
AvgSessionNgramSim	Short history	Average n-gram similarity with all previous queries in the session ($n = 3$).
LongHistoryFreq	Long history	The number of times a candidate is issued as query by the user in the past.
LongHistorySim	Long history	Average n-gram similarity with all previous queries in the user's search history.
SameAgeFrequency	Demographics	Candidate frequency over queries submitted by users in the same age group.
SameAgeLikelihood	Demographics	Candidate likelihood over queries submitted by users in the same age group.
SameGenderFrequency	Demographics	Candidate frequency over queries submitted by users in the same gender group.
SameGenderLikelihood	Demographics	Candidate likelihood over queries submitted by users in the same gender group.
SameRegionFrequency	Demographics	Candidate frequency over queries submitted by users in the same region group.
SameRegionLikelihood	Demographics	Candidate likelihood over queries submitted by users in the same region group.
SameOriginalPosition	MPC	The position of candidate in the MPC ranked list.
SameOriginalScore	MPC	The score of candidate in the MPC ranked list computed based on past popularity.

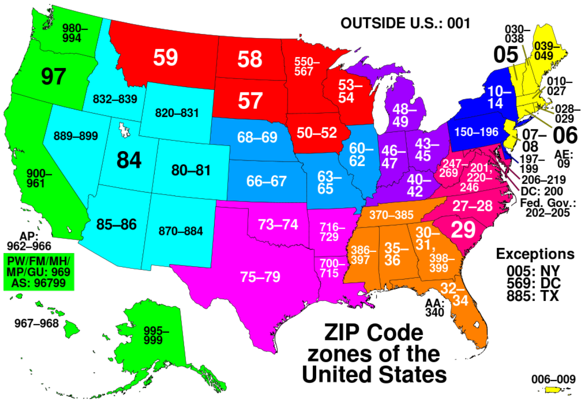


Figure 3: The zip code zones in the United States (Source: Wikipedia). In our experiments, we collapsed all the zip-codes that start with the same digit together. The colour codes above correspond to these collapsed regions.

its overall past popularity (total frequency across all users in the other sets) as two additional ranking features.

More details about the features used in our re-ranking experiments are provided in Table 2.

5. EXPERIMENTS

We begin our analysis by investigating effectiveness of each feature group and subgroup listed in Table 2 separately. After providing a number of re-ranking examples for each case we report the final results from a ranker that is trained by all features. Given the propitiatory nature of our Bing dataset, we do not report the absolute MRR values on that dataset and instead the relative wins and losses against the no-reranking baseline are presented.

In all our experiments, the list of candidates are generated from the top-10 queries ranked by MPC [Bar-Yossef and Kraus, 2011]. Instances in which the *right* query does not appear among the top-10 candidates are removed from the analysis. Note that this does not change any of the conclusions as the MRR is going to be zero before and after personalization (re-ranking) for such cases.

Table 3: The effectiveness of auto-completion personalization according to the user's short session history in terms of MRR. All differences are detected to be statistically significant by the t-test ($p < 0.01$).

Testbed	Baseline	Personalized	MRR
		(Short, MPC)	(Gain/Loss)
AOL	0.666	0.679	+1.95%
Bing	-	-	+0.91%

Short history features. Table 3 contains the result of auto-completion personalization based on features generated from users short-term search history (session). Here, we compare the lexical similarity of candidates with the previous queries submitted by the user in the same session and use that for re-ranking. We always keep the MPC features (default score and position) in our rankers, as they are the most effective features in general, and allow the rankers to learn a safe backoff strategy.

The NearCompletion model [Bar-Yossef and Kraus, 2011] also relied on the user's session history for re-ranking. While we trained our ranker based on decision-trees, the NearCompletion model is based on a linear combination of MPC and session-similarity scores. In addition, NearCompletion expands candidates and previous queries and map them into a higher dimensional vector space for computing the lexical similarities, while we simply rely on n-gram matching. Despite these differences, the numbers for Personalized ranker in Table 3 can be regarded as reference points for the performance of session-based re-ranking techniques such as NearCompletion. The MRR numbers show about 2% improvement on the AOL dataset, and just less than 1% improvement on the Bing dataset. The example provided below is based on a session taken from our testing subset of the AOL dataset. Here, the user (52822) had already issued the following queries in the session, and has typed *d* in the search box as prefix.

```
52822   ryans pet supplies    2006-05-24 19:13:49
52822   dell computer         2006-05-24 20:04:46
52822   circut city           2006-05-24 20:05:20
```

The default order of auto-completion candidates matching

Table 4: The effectiveness of auto-completion personalization according to the user's long history in terms of MRR. All differences are detected to be statistically significant by the t-test ($p < 0.01$).

Testbed	Baseline	Personalized (Long, MPC)	MRR (Gain/Loss)
AOL	0.666	0.696	+4.45%
Bing	-	-	+5.57%

d based on their MPC scores is respectively: *dictionary*, *driving directions*, *deal or no deal*, *delta airlines* and *dell*. The personalized auto-completion ranker boosts the ranking of dell by 4 positions and places it at position one, while keeping the original order for the rest of candidates. Here, the personalization model has picked up on the high lexical similarity between *dell* (candidate), and *dell computer* (2nd last query in the session) for re-ranking.

Long history features. The re-ranking results based on the long history features are reported in Table 4. The first thing to notice compared to the previous experiment based on short-term features is that the MRR gains are higher. Bennett et al. [2012a] compared the impact of long-term and short-term user behaviour features on personalizing search results and found the former to be more effective early in the session when the intents are more ambiguous and relatively less exploratory. Given that by definition prefixes are more ambiguous than queries and also given that more than 50% of sessions in both our datasets have only single queries, the trends observed here are to be expected.

As an example, consider the long search history of user (46669) in the AOL dataset. Note that for brevity, the entire history is not presented here. According to the AOL data, the same user started typing *n* on 2006-05-31 at 08:54:22. The MPC ranking of matching candidates based on past popularity would be: *nascar*, *netflix*, *nick.com*, *nascar.com*, *nextel*, *northwest airlines* and so forth. After personalization however, the top-3 candidates were respectively: *netflix*, *nascar*, and *northwest airlines*. The other candidates were ranked lower in their respective original order. The personalization model realizes that *netflix* has appeared twice in user's search history before and boosts it to position one. This is analogous to the Personal-Navigation model for search result personalization [Teevan et al., 2011]. It is also interesting to note that *northwest airlines* has moved from position six to three after personalization due to its high n-gram similarities with some of the previous queries in user's search history (e.g. *united airlines* and *american airlines*).

```
46669  netflix            2006-03-05 17:31:56
46669  greentortoise      2006-03-05 17:42:14
46669  united airlines    2006-03-08 12:16:25
46669  american airlines  2006-03-08 12:40:44
46669  bank one           2006-03-08 16:51:50
46669  google             2006-03-09 22:23:57
46669  british airways    2006-03-10 08:56:23
46669  netflix            2006-04-24 20:01:20
46669  apple              2006-04-26 09:58:20
```

Table 5: The effectiveness of auto-completion personalization according to the user's age group in terms of MRR. The users age groups are obtained from their profiles in the Bing dataset. The MRR gain is detected as statistically significant by the t-test ($p < 0.01$).

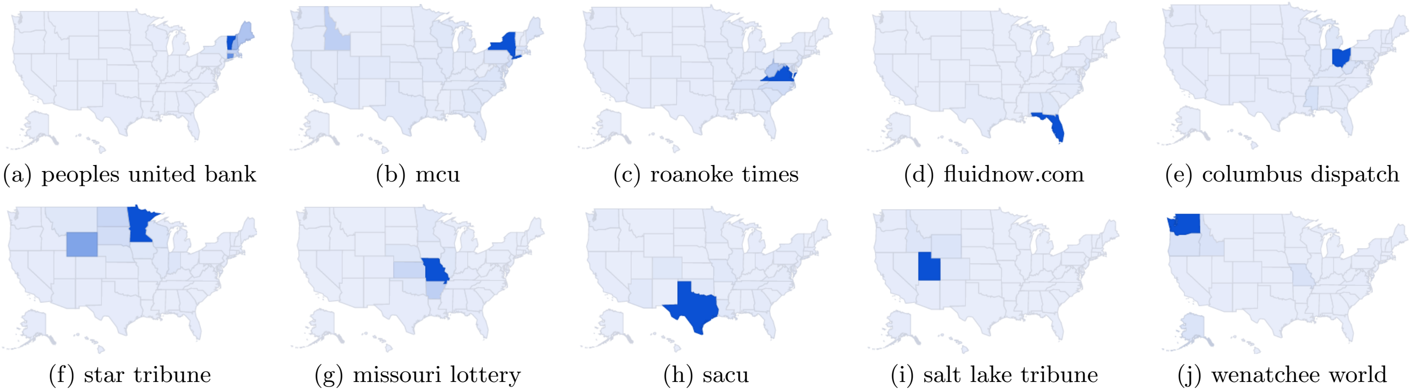
Testbed	Baseline	Personalized (Age, MPC)	MRR (Gain/Loss)
Bing	-	-	+3.80%

Table 6: The biggest movers in personalized auto-completion rankings when the ranker is trained by age features. Each column includes the candidates that were boosted most frequently in the personalized auto-completion rankings for users of the specified age groups.

Below 20	**21-30**	**31-40**
taylor swift	piers morgan	bank of america
justin bieber	richard nixon	worldstarhiphop
deviantart	weather	alex jones
full house	beyonce	indeed
harry styles	movies	national weather service
41-50		**Above 50**
national cathedral		mapquest
target		fedex tracking
chase		florida lottery
microsoft		pogo
traductor google		jigsaw puzzles

Age features. Table 5 contains the results of personalized auto-completion by a ranker trained based on age-specific and MPC features. The AOL dataset does not have any information about users age groups but users in the Bing dataset were *signed-in* when issuing their queries, and their age information were collected from their profiles at the time. The results show statistically significant improvements in MRR when the user's age information is used for personalization. To better understand the type of auto-completion suggestions that are boosted for each age group, we have provided the list of *big movers* in Table 6. For each age group, we extracted queries that were promoted most frequently in personalized auto-completion rankings. As expected, the under-20 group list is dominated by teen's favorite stars, and celebrities. Most of the top movers in the other groups are also intuitive; online banking and stores for users between 30-50, and online games for those above 50 are noteworthy examples.

Gender features. The QAC personalization results based on the gender features are included in Table 7. The personalization gains are comparable to those achieved by using the gender features. Once again we have listed the top movers in Table 8. These are the suggestions that were boosted by personalization and their positions in the auto-completion rankings differed most across the two gender groups before and after personalization. For male users the list was dominated by the names of popular pornographic websites, and car related suggestions. For females, job seeking websites, kids games and shopping related suggestions were among big movers.



(a) peoples united bank (b) mcu (c) roanoke times (d) fluidnow.com (e) columbus dispatch

(f) star tribune (g) missouri lottery (h) sacu (i) salt lake tribune (j) wenatchee world

Figure 4: The top movers in each region. These are queries that their average positions in rankings with and without personalization differ the most in each region. The regions are specified by collapsing the first zip-code digits and the users in each region are grouped accordingly. The counters (a)–(j) respectively refer to regions in Figure 3. Each map shows the distribution of query popularity across different US states according to Google Trends, and the colors range between light blue (rare) and dark blue (popular).

Table 7: The effectiveness of auto-completion personalization according to the user's gender group in terms of MRR. The users age groups are obtained from their profiles in the Bing dataset. The MRR gain is detected as statistically significant by the t-test ($p < 0.01$).

Testbed	Baseline	Personalized	MRR
		(Gender, MPC)	(Gain/Loss)
Bing	-	-	+3.59%

Table 8: The biggest movers in personalized auto-completion rankings when the ranker is trained by gender features. Each column includes the candidates that were boosted most frequently in the personalized auto-completion rankings for users of the specified gender groups.

Male	imdb, *pornographic-related*[8], drudge, autotrader, usaa
Female	indeed.com, poptropica, daily mail, victoria secret

Region features. We also investigate the effectiveness of using user's location for auto-completion personalization. According to the results in Table 9, the *location-aware* ranker shows statistically significant improvement over the baseline, and the location features show the highest gain among all demographic-based features.

The top movers in each region are displayed in Figure 4. These are queries that their average positions in rankings with and without personalization differ the most in each region. The regions are determined by collapsing the users zip-codes by their first digits and correspond to the color codes illustrated in Figure 3. The counters (a)–(j) in Figure 4 respectively refer to regions in Figure 3. Each map shows the distribution of query popularity across different US states, and the colors range between light blue (rate) and dark blue (popular). While our experiments are conducted on the Bing dataset, we show the popularity distribution of big movers according to Google Trends so that they can be verified externally. The query popularity trends in the Bing dataset are remarkably similar. For instance, 83% of users searching for *peoples united bank* were from region 0, while 9.5% of them were from region 1, which are comparable statistics with those in Figure 4 (a). Similarly, 89% of users search for *mcu* in our Bing dataset are from region 1, which is consistent with distributions from Google Trends in Figure 4(b).

Table 9: The effectiveness of auto-completion personalization according to the user's region group in terms of MRR. The users region groups are obtained from their profiles in the Bing dataset. The MRR gain is detected as statistically significant by the t-test ($p < 0.01$).

Testbed	Baseline	Personalized	MRR
		(Region, MPC)	(Gain/Loss)
Bing	-	-	+4.58%

Table 10: The effectiveness of auto-completion personalization according to all user-specific and demographic-based features in terms of MRR. Note that there are no demographic information available in the AOL dataset. The MRR gain is detected as statistically significant by the t-test ($p < 0.01$).

Testbed	Baseline	Personalized	MRR
		(All)	(Gain/Loss)
AOL	0.666	0.709	+6.45%
Bing	-	-	+9.42%

Overall, the list of top movers is mainly dominated by queries with significant local intent particularly, regional news agencies (e.g. *columbus dispatch*, *star tribune*, *salt lake tribune*, *wentachee world*) and various local financial agencies (e.g. *peoples united bank* and *mcu*).

All features. So far, we demonstrated that each of our user-specific and demographic-based feature groups can contribute to significant MRR gains when used individually for

for personalization. Here, we report the results of a ranker which is trained with combinations of all these features. The results are presented in Table 10 and as expected, the MRR gains when using all features are substantially higher than all individual feature groups. The best performing individual feature group was long history that achieved +5.57% MRR improvements on the Bing testbed (+4.45% on AOL), followed by the location features that improved MRR by 4.58%. On the Bing dataset, using all user-specific and demographic features together, the MRR gains can be almost doubled to reach 9.42%. On the AOL dataset, using both short- and long-history features increases the gains to +6.45.

We also performed an A/B testing evaluation on the live traffic of a commercial search engine. We performed this evaluation for 16 days in January 2013, over approximately 3.1 million users. While further details cannot be shared due to their sensitivity, we can confirm that MRR and other user-engagement metrics were significantly improved by the personalized ranker.

6. CONCLUSIONS

We proposed a new approach for learning to personalize auto-completion rankings. While previous auto-completion models rely on aggregated statistics across all (or demographic groups of) users, we showed that user-specific and demographic-based features can be used together under the same framework. We introduced a novel strategy for extracting training labels from previous logs and showed that it can be used for training auto-completion rankers. We also compared the effectiveness of various user-specific and demographic features and showed that certain demographic features such as location are more effective than others for personalization, and as expected, adding more features based on users demographics and search history leads to further boost in personalization effectiveness.

There are several directions for future work; while we considered our labels to be binary, it would be interesting to investigate how multi-graded labels (perhaps based on user's interactions with landing pages) may change the results. Along similar lines, our model can be extended by allowing more than one relevant candidate per ranking if they are closely related. For instance, if a sampled impression has *facebook* as query, both *facebook* and *facebook.com* may be regarded as relevant candidates for prefix *f*.

In addition, investigating the impact of personalization on different types of suggestions (e.g. informational, navigational) may reveal interesting insights about those segments that are more likely to benefit from personalization. Last but not least, we sampled our training impressions at random and that may introduce a slight bias towards longer queries as the match more prefixes. Other sampling strategies for generating training data may lead to more balanced training and potentially bigger gains.

Acknowledgements

The author is grateful to Sebastian Blohm for proof reading the first draft of this work, and also to anonymous reviewers for their insightful comments.

References

Z. Bar-Yossef and N. Kraus. Context-sensitive query auto-completion. In *Proc. WWW*, pages 107–116, Hyderabad, India, 2011.

P. N. Bennett, F. Radlinski, R. W. White, and E. Yilmaz. Inferring and using location metadata to personalize web search. In *Proc. SIGIR*, pages 135–144, Beijing, China, 2011. ISBN 978-1-4503-0757-4.

P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short- and long-term behavior on search personalization. In *Proc. SIGIR*, pages 185–194, Portland, OR, 2012a. ISBN 978-1-4503-1472-5.

P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short- and long-term behavior on search personalization. In *Proc. SIGIR*, SIGIR '12, pages 185–194, Portland, OR, 2012b. ISBN 978-1-4503-1472-5.

S. Bhatia, D. Majumdar, and P. Mitra. Query suggestions in the absence of query logs. In *Proc. SIGIR*, pages 795–804, Beijing, China, 2011. ISBN 978-1-4503-0757-4.

S. Bickel, P. Haider, and T. Scheffer. Learning to complete sentences. In *Proc. ECML*, volume 3720 of *Lecture Notes in Computer Science*, pages 497–504. Springer, 2005. ISBN 3-540-29243-8.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *Proc. NIPS*, pages 193–200, Vancouver, BC, 2006. MIT Press.

C. Burges, K. Svore, P. Bennett, A. Pastusiak, and Q. Wu. Learning to rank using an ensemble of lambda-gradient models. *Journal of Machine Learning Research - Proceedings Track*, 14:25–35, 2011.

H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In *Proc. SIGKDD*, pages 875–883, Las Vegas, NV, 2008. ISBN 978-1-60558-193-4.

S. Chaudhuri and R. Kaushik. Extending autocompletion to tolerate errors. In *Proc. SIGMOD*, pages 707–718, Providence, Rhode Island, USA, 2009.

H. Cheng and E. Cantú-Paz. Personalized click prediction in sponsored search. In *Proc. WSDM*, pages 351–360, New York, NY, 2010. ISBN 978-1-60558-889-6.

K. Collins-Thompson, P. N. Bennett, R. W. White, S. de la Chica, and D. Sontag. Personalizing web search results by reading level. In *Proc. CIKM*, pages 403–412, Glasgow, UK, 2011. ISBN 978-1-4503-0717-8.

J. J. Darragh, I. H. Witten, and M. L. James. The reactive keyboard: A predictive typing aid. *Computer*, 23:41–49, November 1990.

J. Fan, H. Wu, G. Li, and L. Zhou. Suggesting topic-based query terms as you type. In *Proc. APWEB*, pages 61–67, Washington, DC, 2010.

S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23 (2):147–168, Apr. 2005. ISSN 1046-8188.

K. Grabski and T. Scheffer. Sentence completion. In *Proc. SIGIR*, pages 433–439, Sheffield, United Kingdom, 2004.

J. Guo, X. Cheng, G. Xu, and X. Zhu. Intent-aware query similarity. In *Proc. CIKM*, pages 259–268, Glasgow, UK, 2011. ISBN 978-1-4503-0717-8.

B. J. Jansen, A. H. Spink, C. Blakely, and S. Koshman. Defining a session on web search engines. *Journal of the American Society for Information Science and Technology*, 58(6):862–871, 2007.

S. Ji, G. Li, C. Li, and J. Feng. Efficient interactive fuzzy keyword search. In *Proc. WWW*, pages 371–380, Madrid, Spain, 2009.

E. Kharitonov and P. Serdyukov. Demographic context in web search re-ranking. In *Proc. CIKM*, pages 2555–2558, 2012. ISBN 978-1-4503-1156-4.

Z. Liao, D. Jiang, E. Chen, J. Pei, H. Cao, and H. Li. Mining concept sequences from large-scale search logs for context-aware query suggestion. *ACM Transactions on Intelligent Systems and Technology*, 3(1):17:1–17:40, Oct. 2011. ISSN 2157-6904.

T.-Y. Liu. Learning to rank for information retrieval. *Foundation and Trends in Information Retrieval*, 3(3):225–331, Mar. 2009. ISSN 1554-0669.

Y. Liu, R. Song, Y. Chen, J.-Y. Nie, and J.-R. Wen. Adaptive query suggestion for difficult queries. In *Proc. SIGIR*, pages 15–24, Portland, OR, 2012. ISBN 978-1-4503-1472-5.

N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *Proc. WSDM*, pages 25–34, 2011. ISBN 978-1-4503-0493-1.

Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *Proc. CIKM*, pages 469–478, Napa Valley, CA, 2008. ISBN 978-1-59593-991-3.

A. Nandi and H. V. Jagadish. Effective phrase prediction. In *Proc. VLDB*, pages 219–230, Vienna, Austria, 2007.

U. Ozertem, O. Chapelle, P. Donmez, and E. Velipasaoglu. Learning to suggest: a machine learning framework for ranking query suggestions. In *Proc. SIGIR*, pages 25–34, Portland, OR, 2012. ISBN 978-1-4503-1472-5.

G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *Proc. InfoScale*, New York, NY, USA, 2006. ACM. ISBN 1-59593-428-6.

A. Reda, Y. Park, M. Tiwari, C. Posse, and S. Shah. Metaphor: a system for related search recommendations. In *Proc. CIKM*, pages 664–673, Maui, HI, 2012. ISBN 978-1-4503-1156-4.

R. Santos, C. Macdonald, and I. Ounis. Learning to rank query suggestions for adhoc and diversity search. *Information Retrieval*, pages 1–23, 2012. ISSN 1386-4564.

M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *Proc. SIGIR*, pages 601–610, Portland, Oregon, USA, 2012. ISBN 978-1-4503-1472-5.

Y. Song and L.-w. He. Optimal rare query suggestion with implicit user feedback. In *Proc. WWW*, pages 901–910, Raleigh, NC, 2010. ISBN 978-1-60558-799-8.

Y. Song, D. Zhou, and L.-w. He. Post-ranking query suggestion by diversifying search results. In *Proc. SIGIR*, pages 815–824, 2011. ISBN 978-1-4503-0757-4.

Y. Song, D. Zhou, and L.-w. He. Query suggestion by constructing term-transition graphs. In *Proc. WSDM*, WSDM '12, pages 353–362, 2012. ISBN 978-1-4503-0747-5.

A. Strizhevskaya, A. Baytin, I. Galinskaya, and P. Serdyukov. Actualization of query suggestions using query logs. In *Proc. WWW*, pages 611–612, Lyon, France, 2012. ISBN 978-1-4503-1230-1.

J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. SIGIR*, SIGIR '05, Salvador, Brazil, 2005. ISBN 1-59593-034-5.

J. Teevan, D. J. Liebling, and G. Ravichandran Geetha. Understanding and predicting personal navigation. In *Proc. WSDM*, pages 85–94, 2011. ISBN 978-1-4503-0493-1.

I. Weber and C. Castillo. The demographics of web search. In *Proc. SIGIR*, pages 523–530, Geneva, Switzerland, 2010. ISBN 978-1-4503-0153-4.

R. W. White and G. Marchionini. Examining the effectiveness of real-time query expansion. *Inf. Process. Manage.*, 43:685–704, May 2007.

B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In *Proc. SIGIR*, pages 451–458, 2010. ISBN 978-1-4503-0153-4.