# Building Rich User Search Queries Profiles

Elif Aktolga[1], Alpa Jain[2], and Emre Velipasaoglu[3,⋆]

[1] CIIR, UMass Amherst, Amherst MA 01003, USA
elif@cs.umass.edu
[2] Twitter, Inc., San Francisco, CA 94103, USA
alpajn@gmail.com
[3] Magnet Systems, Inc., Palo Alto, CA 94301, USA
emre.velipasaoglu@magnet.com

**Abstract.** It is well-known that for a variety of search tasks involving queries more relevant results can be presented if they are *personalized* according to a user's interests and search behavior. This can be achieved with user-dependent, rich web search queries profiles. These are typically built as part of a specific search personalization task so that it is unclear which characteristics of queries are most effective for modeling the user-query relationship *in general*. In this paper, we explore various approaches for explicitly modeling this user-query relationship independently of other search components. Our models employ generative models in layers in a prediction task. The results show that the best signals for modeling the user-query relationship come from the given query's terms and entities together with information from related entities and terms, yielding a relative improvement of up to 24.5% in MRR and Success over the baseline methods.

**Keywords:** User Profiles, Personalization, Named Entities.

## 1 Introduction

Commercial search engines answer millions of queries on a daily basis while serving a variety of user intents. To enable users to better formulate their search tasks, search engines offer query suggestion tools which auto-complete a user's partial query or suggest follow-up queries after the query is typed. A naive approach is to provide a static list of suggestions to *all* users for *all* search intents. This ignores information available from each user's unique search queries profile, which is required to achieve better personalization.

A critical challenge with building user profiles is testing their effectiveness. Typically, user profiles are employed as part of search personalization tasks such as search result reranking or query suggestion, and the effect of personalization is observed through the applied task. This way, it is not obvious which characteristics of queries are most effective for explicitly modeling the user-query relationship in general for any search personalization task. Therefore, in this

⋆ All the authors were at Yahoo! Labs when this work was done.

S. Carberry et al. (Eds.): UMAP 2013, LNCS 7899, pp. 254–266, 2013.
© Springer-Verlag Berlin Heidelberg 2013

paper we aim at *building rich user search queries profiles and measuring their quality in isolation of these other components.* This is framed by means of the following *prediction task*: given users' past queries, can we effectively predict the most likely user to have issued a given present/future query?

For this task we solely utilize users' queries obtained through historical query log data without soliciting any explicit input from users and without employing users' clicks on documents in search results.

Prior research has dealt with personalization for search results or building user profiles based on their search results interaction, i.e., clicked urls, content of the visited web pages [3], [4], [8], [13], [18]. These works primarily study users' intents over time for better *document ranking* [1], [4], [9], [11], [12], [15], [17], [19], [20] or they research query suggestion utilizing document click through [3], [8], [13], [16]. Building deep profiles from terse queries poses many interesting challenges: (1) we have a *cold-start problem* [17], [18], [20], i.e., information about the user needs to be gathered first before queries or results can be personalized; (2) a substantial fraction of queries does not yield to personalization (navigational queries like 'facebook', 'bank of america', etc.) or purely numeric queries; (3) search queries are terse and often ambiguous. Understanding how to effectively build user profiles from search queries can benefit many personalization tasks including query suggestion, search result reranking, and federation optimization.

Our proposed approach uses personalizable information from users' past queries. Our contribution consists of a generative model that employs related terms and entities to overcome the sparsity of queries in user profiles. We present an extensive evaluation suite to measure the personalization potential of our proposed algorithms over a large query log from a major search engine. Our experiments show that the best performing approach yields a relative improvement of up to 24.5% in MRR and Success over the baseline methods.

## 2 Related Work

Personalization is a well researched area that aims at better addressing search intents of users for tasks such as document (re)ranking or document retrieval [1], [4], [9], [11], [12], [15], [17], [19], [20]. The value of personalizing web search results has widely been studied [4], [12], [19]: Teevan et al. [19] analyze query intents of users and discover that there are noticeable variations in search intents for the same query and in the interpretation of these intents. Further, the difficulty of resolving abbreviated and ambiguous queries is identified. Teevan et al. show that by employing personalization, the results can be efficiently reranked to improve retrieval effectiveness. Mei and Church [12] study personalization through backoff, i.e., including information from similar users or from a user's group for estimating the likelihood of a document click for the given user. Morris et al., Teevan et al., and Xue et al. study several further techniques based on user groups [14], [21], [22]. Dou et al. [4] analyze when and how personalization is useful by using click-based and query-based personalization techniques. Luxenburger et al. study language modeling approaches for selectively personalizing queries only when it is required [10].

Some prior work has specifically focused on the personalization of query suggestions and query completion suggestions [3], [5], [8], [13]. All these techniques utilize click through data on documents for estimating user preferences instead of query history-based user profiles for personalization, whose construction we deal with in this paper.

There has also been a lot of prior work in applying user profiles for improved web search ranking: Sugiyama et al. employ two user profiles in combination for every user; one that represents the user's long term browsing history, and another one that represents her current browsing history [18]. Teevan et al. also study long-term and short-term interests of users in their profiles for better document ranking [20].

Unlike prior personalization related work, our paper studies user profiling for a more general situation: how can we build rich user profiles from queries as a basis for various personalization tasks? This is a substantial difference from prior personalization approaches that are geared towards optimizing a specific task such as document ranking only.

## 3 Modeling Search Behavior

In this section, we consider a variety of natural ways of designing user profiles for search queries. Specifically, we study *phrase-level* models that exploit the query terms and a *syntax-level* model that exploits the query syntax. We build these models in layers from users' queries so that gradually more evidence is included in the profiles. All the models estimate the likelihood of a given query to have been issued by a certain user.

### 3.1 Phrase-Level Models

These models include phrase-level information from a user's queries in the user profile. The simplest phrase-level model is **Query Terms (T)**, defined as follows:

$$P_T(u|q) = \sum_{t \in T(q)} P(u|t) \tag{1}$$

where $t \in T(q)$ denotes the set of unigram terms in $q$. The estimation of this probability is carried out as described in the generalized form in Section 3.3. The next model **Entities (E)** (referred to in the equations below as $P_E(u|q)$) includes tagged entity phrases as detected by our named entity recognizer. The model is mathematically the same as Equation 1; the only difference is that $t \in T(q)$ refers to the set of unigram terms *and* entity phrases (e,g,. 'pizza hut', 'toyota' etc.) in the query.

The following model **Entities and their categories (EC)** expands the **Entities** model further, by including the named entity categories that were located by our named entity recognizer:

$$P_{EC}(u|q) = \alpha \cdot \sum_{e \in E(q)} P(u|e) + (1 - \alpha) \cdot P_E(u|q) \tag{2}$$

where $e \in E(q)$ refers to the set of entity categories (such as 'person', 'product' etc.) in the query. Again, refer to Section 3.3 for the estimation of the probabilities. We interpolate $\alpha$ with Dirichlet smoothing by setting

$$\alpha = \frac{|E(q)|}{|E(q)| + \mu} \tag{3}$$

where $\mu$ is tuned by means of a parameter sweep on the training data. This means that the more entities are found in $q$, the heavier is the emphasis on the entity categories part, and otherwise the terms and entities will be weighted stronger in the model.

Our final phrase-level model is **Related Terms and Entities (RelTE)**, for which we first define how to estimate the likelihood of a certain user $u$ to issue similar phrases:

$$P(u|\mathcal{S}) = \sum_{s \in Sim(t)} P(u|s) \cdot P(s|t) \tag{4}$$

where $\mathcal{S} = \{s | s \in Sim(t), t \in T(q)\}$ are similar phrases. Specifically, $t$ are again terms and entities as in the **Entities** model above, and $Sim(t)$ are related terms and entities for $t$. To locate these, we use a term to term co-occurrence mined dictionary from a 1-year query log. On average, every term has 4-5 related terms and entities. Each entry $\langle s, t \rangle$ has an associated co-occurrence score $P(s|t) = \frac{tf(s,t)}{\sum tf(t)}$, describing the relatedness of the term or entity phrase $s$ to $t$. $P(u|s)$ on the other hand denotes how likely user $u$ is to use this related term or entity in his profile. We combine $P(u|\mathcal{S})$ together with **Entities and their categories** from Equation 2 into the following:

$$P_{RelTE}(u|q) = (1 - \gamma) \cdot P_{EC}(u|q) + \gamma \cdot P(u|\mathcal{S}) \tag{5}$$

where $\gamma$ is again tuned similarly to $\alpha$ in Equation 3, depending on the availability of related terms and entities for $q$:

$$\gamma = \frac{|Sim(t)|}{|Sim(t)| + \kappa} \tag{6}$$

We tune $\kappa$ with a parameter sweep on the training data.

### 3.2 Syntax-Level Model

The phrases for this model are derived by analyzing a user's queries with a dependency parser. For this, we observe dependency parse label sequences (DPLS), which allow us to extract phrases that are not necessarily sequential n-grams in the query. Figure 1 shows example parses for two queries. By observing 16 different DPLS, we can extract the following phrases from the query *chest x-ray showed evidence of peptic disease*:

– chest x-ray showed peptic disease (`nn nsubj amod`)
– chest x-ray showed evidence (`nn nsubj dobj`)
– x-ray showed peptic disease (`nsubj amod`)
– x-ray showed evidence (`nsubj dobj`)
– x-ray showed evidence disease (`nsubj dobj prep`)

Note that while extracting these phrases, we only consider nodes that are directly connected to the affected edge (see Figure 1). The resulting phrases generalize or specify the query in different ways. For comparison, our named entity tagger only recognized the phrase 'x-ray' in this query. From the other query *wood fired portable pizza oven* we can extract the following phrases:

– wood fired portable oven (`nsubj amod`)
– wood fired portable pizza oven (`nsubj amod nn`)
– wood fired oven (`nsubj dobj`)
– wood fired pizza oven (`nsubj nn`)

Again, the only named entity phrase that was detected is 'oven'. While in these examples the quality of the extracted sub phrases is good, this is not always the case for other queries. Therefore, we apply dependency parsing to queries having at least 3 terms to guarantee good sub phrase quality [7]. For queries with fewer than 3 terms, the approach defaults to one of the phrase-level models.

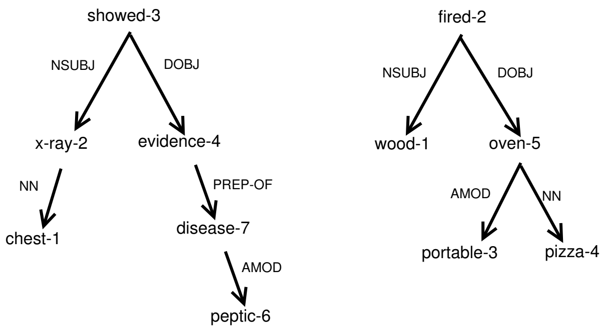


**Fig. 1.** Dependency parse trees for the two queries *chest x-ray showed evidence of peptic disease* (left) and *wood fired portable pizza oven* (right)

For this model – **Syntactic phrases (DP)** – we first estimate the likelihood of query $q$ to have been issued by user $u$ by means of the sub phrases $\mathcal{D}$ of $q$ as follows:

$$P(u|\mathcal{D}) = \sum_{d \in Dep(q)} P(u|d) \cdot P(d|q) \tag{7}$$

where $\mathcal{D} = \{d | d \in Dep(q)\}$ is the set of all sub phrases of $q$ extracted through observing various DPLS in $q$. Each sub phrase $d$ is extracted through exactly one dependency parse label sequence $dp$ from the query $q$. The first factor $P(u|d)$ in

Equation 7 denotes the likelihood of user $u$ issuing the given sub phrase $d$, which we calculate by observing the maximum likelihood of the DPLS $dp \in d$ in this user's profile. The estimation of these probabilities is carried out as described in the generalized form in Section 3.3. By utilizing DPLS here we avoid the problem of sub phrase sparsity in user profiles.

The second factor $P(d|q)$ in Equation 7 refers to the importance or quality of this extracted sub phrase $d$ given the query $q$. This is estimated by observing the maximum likelihood of this sub phrase being generated as an exact match within all other sub phrases for $q$. Hence, sub phrases generated as exact matches within others through several DPLS have higher importance, whereas low-quality phrases occurring less frequently are demoted.

We then combine $P(u|\mathcal{D})$ together with the **Entities Model and their categories** from Equation 2 into **Syntactic phrases (DP)** as follows:

$$P_{DP}(u|q) = (1 - \delta) \cdot P_{EC}(u|q) + \delta \cdot P(u|\mathcal{D}) \tag{8}$$

where $P_{EC}(u|q)$ was defined in Equation 2. $\delta$ is again tuned similarly to $\alpha$ in Equation 3, depending on the number of sub phrases that were extracted for $q$:

$$\delta = \frac{|Dep(q)|}{|Dep(q)| + \tau} \tag{9}$$

Again, we tune $\tau$ with a parameter sweep on the training data.

### 3.3 Estimation and Smoothing

**Model Estimation.** All our models use conditional probabilities of the form $P(u|x)$. These are estimated as follows, using the Bayes' Rule:

$$P(u|x) = \frac{P(x|u) \cdot P(u)}{P(x)} \propto P(x|u) \cdot P(u) \tag{10}$$

where $P(x)$ is a constant denoting the prior likelihood of an item $x$, which is the same across all users and can therefore be safely omitted to yield rank-equivalent results. We obtain the probabilities $P(x|u)$ and $P(u)$ directly from the learned user profile of user $u$ through maximum likelihood estimations.

**Smoothing.** We smooth the probabilities $P_{\text{MLE}}(x|u)$ for the models **Query Terms**, **Entities** and **Entities and their categories** for considering *missing items* in a user's profile. For simplicity, we apply Add-1 Smoothing [2] for this so that a *missing term* $x$ is assumed to occur only once rather than not at all in the user's profile, and the probabilities for other items in the user profile are adjusted accordingly. This helps with the user profile sparsity problem.

## 4 Experimental Setup

### 4.1 Dataset and Statistics

We utilize queries that were collected in a query log of the Yahoo! search engine over a period of 2.5 months in 2011. Each query is associated with a user. We ran a series of filters to eliminate adult, robot, or garbled queries. Further, we prepared the data as follows:

1. **Filtering navigational queries:** Queries that have been identified as navigational are not suitable for personalization because the user's purpose is merely to reach a particular web page. We filter navigational queries by means of a dictionary with 1,000,000 entries.
2. **Which users to prefer for user profile learning and query prediction:** For our experiments we build user profiles and evaluate their quality by predicting the owners of unseen 'future' queries. We choose a qualitative set of users from the query log according to two criteria: (1) users with the highest number of unique queries; (2) users with the highest number of unique entity categories in their queries.
3. **Choosing users for user profile learning and query prediction:** From the two approaches to ranking users, we choose the top $k = 100$ users with their queries, yielding close to 37,000 queries for the experiments. The datasets are referred to as numQ (number of unique queries) and numE (number of unique entity categories).
4. **Train/test split:** In order to split the data into train and test we choose a *time-based split*: information from the first 2 months of the query log for the $k$ users is used for model learning and construction, whereas the last 2 weeks are the basis for query prediction and testing. For model construction, we consider all the (unfiltered) queries of the $k$ users.
5. **Prediction Queries:** For the experiments we use the test queries obtained from the time-based split of the query log described in step 4. We filter these queries to only retain those that have been issued by at least 2 unique users. This is to ensure an evaluation over a qualitative query set. The resulting queries are used for the prediction experiments.
6. **Experiments:** We do 5-fold cross-validation on this test query set by dividing it randomly into 5 roughly equal, non-overlapping splits. We report the average results over the 5 validation splits in the paper.

As named-entity tagger we use a production quality tagger at Yahoo! that identified 38 different named entity categories in the entire query log. Purely numeric queries such as phone or tracking numbers are filtered since they are unsuitable for personalization.

For the syntax-level models we utilize the Stanford Dependency Parser [6] for extracting sub phrases with 16 different dependency parse label sequences (DPLS). A quick analysis revealed that diverse DPLS are very well represented in users' profiles in the whole query log.

### 4.2 Evaluation Approach

Our objective in this paper is to model the user-query relationship independently of other search components: given users' past queries, can we effectively predict the most likely user to have issued a given present/future query? There can be several users that have actually issued a query. For each query we determine the set of correct users $C_q = \{c_1, \ldots, c_n\}$ as follows: each user $c_i$ in the *test time split of the query log after time* $\mathcal{T}$ having issued the query $q$ is regarded as a correct user for $q$. This guarantees that all correct users are unseen during the profile building or *training* stage, which uses the time split of the query log *before* time $\mathcal{T}$.

We want to evaluate whether at least one correct user was found until rank $n$. This can be measured with Success:

$$\text{Success@n} = I_{\text{correct}}(q, n) \tag{11}$$

where $I_{\text{correct}}(q, n)$ is an indicator random variable equaling 1 if at least one user $c_i \in C_q$ for $q$ is present until rank $n$, and 0 otherwise. In our experiments, we report Average Success@n across all test queries. Then, we would like to know what the rank of the first correct user is, which can be addressed with (average) MRR:

$$\text{MRR@n} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_{c_i}} \tag{12}$$

where $\text{rank}_i$ is the rank of the first correct user $c_i$ for $q$. Users are considered until rank $n$. Finally, we must evaluate how many of all correct users were found until rank $n$. We utilize Recall for this:

$$\text{Recall@n} = \frac{|\{\text{correct users until rank } n\}|}{|C_q|} \tag{13}$$

where the numerator denotes the number of correct users found for $q$ until rank $n$, and the denominator refers to the total count of correct users $|C_q|$. Again, we report Average Recall@n over all test queries in our results.

## 5 Experimental Results

### 5.1 Effectiveness of Search Query Profiles

The following experiments have been performed by means of 5-fold-cross validation as explained in Section 4. In Table 1 we have the results for the prediction experiments for the numQ data set. In this data set the users have the highest number of unique queries in their user profiles. We compare all the methods that we introduced in Section 3 by observing MRR and Success at different ranks. In the MRR results, we can see significant incremental changes from **T** to **E** and to **EC** over all MRR ranks as more information from the user profiles is employed. It is not surprising that adding in entity phrases in addition to terms yields a larger gain (**T** $\rightarrow$ **E**) than including named entity category information in addition to terms and entities (**E** $\rightarrow$ **EC**). Related terms and entities (**RelTE**) result in another major boost over all ranks, yielding a maximum relative improvement of 24.5% over **T** at rank 1 for both measures (0.1868). This is statistically significant over all the other methods at the same ranks.

**Table 1.** Averaged cross-validation results on the numQ dataset for the baseline method (**T**), Entity Model (**E**), Entities and their categories Model (**EC**), Related Entities Model (**RelTE**), Syntactic Phrases Model (**DP**), and the Syntactic Phrases Model falling back to RelTE (**DP+RelTE**). Bold entries are statistically significant over all non-bold ones at same ranks with p-value $< 0.03$ using the paired two-sided t-test. Starred entries are statistically significant to immediately adjacent entries to the left.

| **MRR@rank** | **T** | **E** | **EC** | **RelTE** | **DP** | **DP+RelTE** |
|---|---|---|---|---|---|---|
| **1** | 0.1501 | 0.1715* | 0.1771* | **0.1868*** | 0.1771 | 0.1779 |
| **5** | 0.2096 | 0.2254* | 0.2389* | **0.2502*** | 0.2386 | 0.2403* |
| **10** | 0.2231 | 0.2363* | 0.2534* | **0.2642*** | 0.2532 | 0.2548* |
| **20** | 0.2316 | 0.2424* | 0.2617* | **0.2720*** | 0.2617 | 0.2632* |
| **Success@rank** | **T** | **E** | **EC** | **RelTE** | **DP** | **DP+RelTE** |
| **1** | 0.1501 | 0.1715* | 0.1771* | **0.1868*** | 0.1771 | 0.1779 |
| **5** | 0.3170 | 0.3166 | 0.3490* | **0.3658*** | 0.3473 | 0.3577* |
| **10** | 0.4185 | 0.3978 | 0.4584* | **0.4706*** | 0.4575 | 0.4656* |
| **20** | 0.5402 | 0.4891 | 0.5788* | 0.5836 | 0.5815 | 0.5844 |

**Table 2.** Averaged cross-validation results on the numE dataset for the same methods as in Table 1. The same statistical significance test is performed with p-value $< 0.05$.

| **MRR@rank** | **T** | **E** | **EC** | **RelTE** | **DP** | **DP+RelTE** |
|---|---|---|---|---|---|---|
| **1** | 0.1440 | 0.1661* | 0.1687 | **0.1739*** | 0.1688 | 0.1718 |
| **5** | 0.1960 | 0.2151* | 0.2245* | **0.2306*** | 0.2243 | 0.2274* |
| **10** | 0.2083 | 0.2246* | 0.2364* | **0.2423*** | 0.2360 | 0.2389* |
| **20** | 0.2172 | 0.2318* | 0.2452* | **0.2508*** | 0.2448 | 0.2476 |
| **Success@rank** | **T** | **E** | **EC** | **RelTE** | **DP** | **DP+RelTE** |
| **1** | 0.1440 | 0.1661* | 0.1687 | **0.1739*** | 0.1688 | 0.1718 |
| **5** | 0.2916 | 0.3034* | 0.3217* | **0.3297*** | 0.3208 | 0.3250 |
| **10** | 0.3849 | 0.3753 | 0.4134* | **0.4196*** | 0.4103 | 0.4134 |
| **20** | 0.5145 | 0.4808 | 0.5439* | 0.5443 | 0.5428 | 0.5407 |

However the **DP** and **DP+RelTE** models do not improve the rankings of the users further. In the Success results there is a similar trend as with MRR. **RelTE** remains the most successful model except for rank 20 at which we observe no significant difference between the models.

Table 2 shows the same results for the numE data set with slightly lower numbers over all. In this dataset the users have the highest number of unique entity categories in their profiles. This aspect makes the prediction task more difficult since some entity categories may be unseen in the user profile.

The next two Figures 2 and 3 show Recall at different ranks for all the methods. This allows us to understand how good the methods are at predicting *all* the correct users for a query. One observation in these results is that **E** gets worse in recall with higher ranks. We can clearly see the usefulness of including named entity category information here, which is stronger pronounced than in the MRR and Success results. **RelTE** has the highest Recall at all ranks except for rank 50: here **EC** dominates.

In order to see how the models perform on a user by user basis, we plotted the personalization potential in Figures 4 and 5 for the best performing three models. Using the same results as before, these graphs show for what fraction of

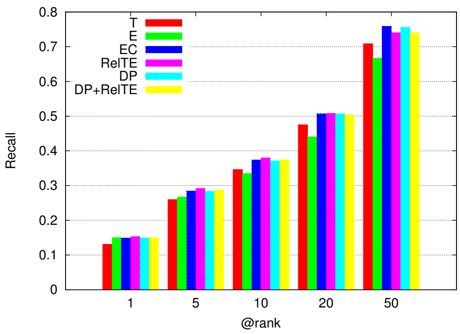


**Fig. 2.** Recall for all models on the data set numE at different ranks

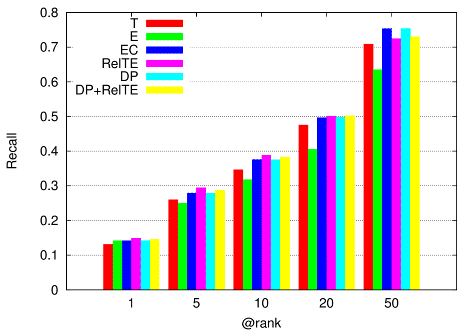


**Fig. 3.** Recall for all models on the data set numQ at different ranks

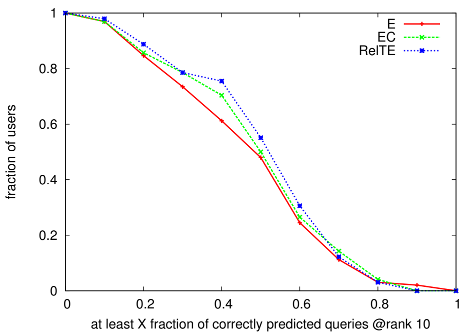


**Fig. 4.** Personalization Potential for data set numE displaying the fraction of users for which at least a certain fraction of queries were correctly predicted at rank 10

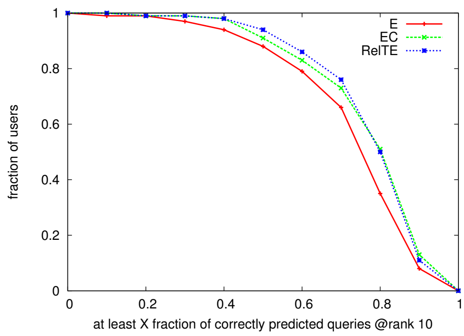


**Fig. 5.** Personalization Potential for data set numQ displaying the fraction of users for which at least a certain fraction of queries were correctly predicted at rank 10

the users at least a certain fraction of queries were predicted correctly at rank 10. A higher curve signifies that more queries were predicted correctly for more users. The graphs show that there is larger personalization potential with the numQ data set, which again confirms that the numE data set is more challenging. But in both graphs **RelTE** performs best, followed by **EC**, and then **E**. Only in the 0.8 - 1.0 range on the x axis in Figure 4 we can see a small inconsistency: it is harder for the better models to predict at least 80% of the queries correctly for a small fraction of users than it is for **E**. This is interesting and hints to entity phrases and terms being a reliable source of information when we want to achieve high recall for a single user – as opposed to achieving high recall for a query across several users (which is what the results in Figures 2 and 3 showed).

### 5.2 Qualitative Analysis

When analyzing the results further, we observe a roughly equal size of queries for which performance changes in terms of the Success measure for the models **DP** and **EC**. Although there are changes in the scores, the syntax level models do not discriminate well enough between users to significantly change the order of the rankings. This may be due to the strong representation of various dependency parse label sequences in users' profiles. Table 3 shows some example queries together with their DPLS. Queries in the upper part improving performance tend to be longer and have more DPLS. In the lower part we have queries that get worse. Often, for such queries none of the 16 rules we have can be applied. We experimented by expanding our rule set but this yielded very small performance gains. We conclude that for user profiling syntax level models may only be applicable to very long and well formed queries. In our current test data set fewer than 21% queries (for numQ; and 27% for numE) have at least 3 terms, so this candidate set is not ideal for testing syntax level models. DPLS are well-represented in user profiles, but the majority of the queries is short. This analysis gives us some intuition about where these approaches could be useful.

**Table 3.** The upper part of the table shows two queries for which performance improves for **DP** over **EC**, and two queries with decreasing performance in the lower part

| Query | DPLS |
|---|---|
| consumer reports best coffee maker | `nsubj nn`: consumer reports coffee maker |
| | `nsubj dobj`: consumer reports maker |
| 1980's south african president | `nsubj nn`: 1980 president african president |
| | `nsubj amod nn`: 1980 president south president african president |
| | `nsubj amod`: 1980 president south president |
| bristol motor speedway | – |
| it's a mad mad mad mad world | `nsubj amod`: it world mad world |

## 6 Conclusions

User profiles are embedded in a variety of search personalization tasks in order to present more relevant results to users. To understand the user-query relationship, we study user search queries profiling models in isolation of other components. For this, we utilize named entities detected in the query, the corresponding entity categories, and related terms and entities in our models. Further, we analyze the syntactic structure of longer queries in syntax-level models. Our experimental results reveal that the best performing model employs related terms and entities in addition to the query's own terms and entities.

The next step is to utilize these findings for user modeling in search personalization tasks such as providing better query suggestions: given a user and a query, which query is most likely to be issued next? For this, we can modify our user profile learning model by just flipping the conditional probability around: instead of estimating the owner of a query, we would estimate the likelihood of a query for a given user. Of course, this model could be further extended to include other components like information from users' clicks on documents.

**Acknowledgments.** This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

## References

1. Cao, B., Sun, J.-T., Xiang, E.W., Hu, D.H., Yang, Q., Chen, Z.: PQC: Personalized Query Classification. In: CIKM (2009)
2. Chen, S.F., Goodman, J.: An Empirical Study of Smoothing Techniques for Language Modeling. In: ACL (1996)
3. Chen, Y., Zhang, Y.-Q.: A Personalised Query Suggestion Agent based on Query-Concept Bipartite Graphs and Concept Relation Trees. Int. J. Adv. Intell. Paradigms 1, 398–417 (2009)
4. Dou, Z., Song, R., Wen, J.-R.: A Large-scale Evaluation and Analysis of Personalized Search Strategies. In: WWW (2007)
5. Kamvar, M., Baluja, S.: The Role of Context in Query Input: Using Contextual Signals to Complete Queries on Mobile Devices. In: MobileHCI (2007)
6. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: ACL (2003)
7. Kumaran, G., Allan, J.: Effective and Efficient User Interaction for Long Queries. In: SIGIR (2008)
8. Leung, K.-T., Ng, W., Lee, D.L.: Personalized Concept-Based Clustering of Search Engine Queries. IEEE Transactions on Knowledge and Data Engineering 20(11), 1505–1518 (2008)
9. Lu, J., Callan, J.: User Modeling for Full-Text Federated Search in Peer-to-Peer Networks. In: SIGIR (2006)
10. Luxenburger, J., Elbassuoni, S., Weikum, G.: Matching Task Profiles and User Needs in Personalized Web Search. In: CIKM (2008)
11. Matthijs, N., Radlinski, F.: Personalizing Web Search using Long Term Browsing History. In: WSDM (2011)
12. Mei, Q., Church, K.: Entropy of Search Logs: How Hard is Search? With Personalization? With Backoff? In: WSDM (2008)
13. Mei, Q., Zhou, D., Church, K.: Query Suggestion Using Hitting Time. In: CIKM (2008)
14. Morris, M.R., Teevan, J., Bush, S.: Enhancing Collaborative Web Search with Personalization: Groupization, Smart Splitting, and Group Hit-Highlighting. In: CSCW (2008)
15. Shmueli-Scheuer, M., Roitman, H., Carmel, D., Mass, Y., Konopnicki, D.: Extracting User Profiles from Large Scale Data. In: MDAC (2010)
16. Sontag, D., Collins-Thompson, K., Bennett, P.N., White, R.W., Dumais, S., Billerbeck, B.: Probabilistic Models for Personalizing Web Search. In: WSDM (2012)
17. Stamou, S., Ntoulas, A.: Search Personalization through Query and Page Topical Analysis. User Modeling and User-Adapted Interaction (February 2009)
18. Sugiyama, K., Hatano, K., Yoshikawa, M.: Adaptive Web Search Based on User Profile Constructed without Any Effort from Users. In: WWW (2004)

19. Teevan, J., Dumais, S.T., Horvitz, E.: Beyond the Commons: Investigating the Value of Personalizing Web Search. In: Proc. Workshop on New Technologies for Personalized Information Access (2005)
20. Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing Search via Automated Analysis of Interests and Activities. In: SIGIR (2005)
21. Teevan, J., Morris, M.R., Bush, S.: Discovering and Using Groups to Improve Personalized Search. In: WSDM (2009)
22. Xue, G.-R., Han, J., Yu, Y., Yang, Q.: User Language Model for Collaborative Personalized Search. ACM Trans. Inf. Syst. (March 2009)