Exploring Session Context using Distributed Representations of Queries and Reformulations

Bhaskar Mitra
Microsoft
Cambridge, UK
bmitra@microsoft.com

ABSTRACT

Search logs contain examples of frequently occurring patterns of user reformulations of queries. Intuitively, the reformulation "san francisco" → "san francisco 49ers" is semantically similar to "detroit" → "detroit lions". Likewise, "london" → "things to do in london" and "new york" → "new york tourist attractions" can also be considered similar transitions in intent. The reformulation "movies" → "new movies" and "york" → "new york", however, are clearly different despite the lexical similarities in the two reformulations. In this paper, we study the distributed representation of queries learnt by deep neural network models, such as the *Convolutional Latent Semantic Model*, and show that they can be used to represent *query reformulations* as vectors. These reformulation vectors exhibit favourable properties such as mapping semantically and syntactically similar query changes closer in the embedding space. Our work is motivated by the success of continuous space language models in capturing relationships between words and their meanings using offset vectors. We demonstrate a way to extend the same intuition to represent query reformulations.

Furthermore, we show that the distributed representations of queries and reformulations are both useful for modelling session context for query prediction tasks, such as for query auto-completion (QAC) ranking. Our empirical study demonstrates that short-term (session) history context features based on these two representations improves the mean reciprocal rank (MRR) for the QAC ranking task by more than 10% over a supervised ranker baseline. Our results also show that by using features based on both these representations together we achieve a better performance, than either of them individually.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

Keywords

Query auto-completion; Deep learning; Contextual search

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.

DOI: http://dx.doi.org/10.1145/2766462.2767702

1. INTRODUCTION

Latent semantic models, such as Latent Semantic Analysis (LSA) [10], Latent Dirichlet Allocation (LDA) [3], Bi-Lingual Topic Model (BLTM) [12] and more recently neural network based models such as Semantic Hashing [34] and Convolutional Latent Semantic Model (CLSM) [36], have been successfully applied to various information retrieval (IR) tasks. By representing queries and documents as low-dimensional vectors in a semantic space these models provide a mechanism to study higher order relationships between these query and document entities. In this paper, we explore ways to model short-term (session) history of Web search users for retrieving contextually more relevant query suggestions using the vector space framework.

As users type their query into the search box, search engines provide ranked lists of query suggestions based on the current prefix at the given point in time. When only a few characters have been entered the search engine has little understanding of the actual user intent and the generic suggestions provided by a non-contextual *query auto-completion* (QAC) system typically perform poorly [1]. The high ambiguity associated with short prefixes makes QAC a particularly interesting candidate for leveraging any additional information available about the user's current task. The same study also showed that 49% of Web searches are preceded by a different search which can be used to gain additional insights into the user's current information need.

The majority of previous work [7, 35] on using short-term user history for search personalization has been focused on modelling the topical relevance of the candidate results (documents or query suggestions) to the previous queries and viewed documents in the same search session. Using such *implicit feedback* has been shown to be a very attractive strategy for improving retrieval performance when the user intent is ambiguous. For example, knowing that the user's previous query was "guardians of the galaxy" can help to inform a QAC system to promote the query "imdb" in ranking over "instagram" when the user has just typed "i" in the search box. Query reformulation behaviours within search sessions have also been studied but are mostly limited to taxonomy based classifications [20, 28] and models based on syntactic changes [14]. A quick study of a sample of Bing's search engine logs reveal that users frequently search for "san francisco 49ers" and "san francisco weather" immediately after searching for "san francisco". Similarly, the query "detroit" is often followed by the queries "detroit lions" and "detroit weather". Intuitively, "san francisco" → "san francisco 49ers" represents a similar shift in user's intent as "detroit" → "detroit lions". We can see many such frequently occurring patterns of reformulations in large scale search logs. Modelling these reformulations using lexical matching alone is difficult. For example, we understand that "movies" → "new movies" is not the same intent shift as "york" → "new york" even though in both cases the same term was added to

both the queries by the user. On the other hand, "london" → "things to do in london" and "new york" → "new york tourist attractions" are semantically similar although the two reformulations involve the addition of completely disjoint sets of new terms to the queries.

In text processing, Mikolov et al. [31] demonstrated that the distributed representation of words learnt by continuous space language models are surprisingly good at capturing syntactic and semantic relationships between the words. Simple algebraic operations on the word vectors have been shown to produce intuitive results. For example, $vector(\text{“king”}) - vector(\text{“man”}) + vector(\text{“woman”})$ results in a vector that is in close proximity to the $vector(\text{“queen”})$. In Section 3, we will show that the embeddings learnt by the *Convolutional Latent Semantic Model* (CLSM) [36] exhibit similar favourable properties and hence provide an intuitive mechanism to represent *query reformulations* as the offsets between the query vectors.

Our empirical study, described in Section 4, demonstrate that the vector representations of queries and reformulations can be useful for capturing session context for the retrieval of query suggestions. The CLSM models are trained to map queries (and documents) with similar intents to the same neighbourhood in the semantic space. Therefore they are suitable for measuring the topical similarity between candidate suggestions and the user's recent queries. In addition, our experiments show that the vector representation of the reformulation, from the user's previous query to the candidate suggestion, can also be a useful signal for predicting the relevance of the suggestion. We present our results in Section 5 that demonstrate that session context features based on these vector representations can significantly improve the QAC ranking over the supervised ranking baseline proposed by Shokouhi [37].

The main contributions of this paper are,

- Demonstrating that *query reformulations* can be represented as low-dimensional vectors which map syntactically and semantically similar query changes close together in the embedding space. We believe that this is the first time that a distributed representation for query reformulations has been proposed and studied in the literature.
- Using features based on the distributed representations of queries and reformulations to improve upon a supervised ranking baseline for session context-aware QAC ranking. Our experiments on the large-scale query logs of the Bing search engine and the publicly available AOL query logs [33] show that these features can improve the *Mean Reciprocal Rank* (MRR) by more than 10% on these testbeds.
- Demonstrating that convolutional latent semantic models trained on *session query pairs* perform significantly better for the contextual QAC ranking task compared to the CLSM model trained on clicked query-document pairs.

Next, we review related work that are relevant to this study.

2. RELATED WORK

Latent semantic models for Web search. Latent semantic models have received significant attention in IR. Latent Semantic Analysis (LSA) [10], Probabilistic LSA (PLSA) [18] and Latent Dirichlet Allocation (LDA) [3] are some of the well-known models proposed to represent queries and documents in low-dimensional space for semantic matching. Unlike these models, which commonly use unsupervised learning, Gao et al. [12] trained Bi-Lingual Topic Models (BLTM) and linear Discriminative Projection Models (DPM) on *clickthrough* data, consisting of query and clicked documents. Salakhutdinov and Hinton [34] used auto-encoders to show that deep learning can be useful for extracting hierarchical semantic structures from queries and documents.

More recently, Huang et al. [21] discriminatively trained a Deep Structured Semantic Model (DSSM) on clickthrough data to maximize the conditional likelihood of the clicked documents for the corresponding queries. By training on query-document pairs they generate a pair of models for projecting the queries and the documents, respectively, to the same embedding space. Their experiments demonstrated better retrieval performance over other existing semantic models by directly optimizing for the document ranking task. They also proposed a *word hashing* technique for dealing with large vocabularies that are commonly associated with Web corpora. The first layer of their model maps the high-dimensional term vectors corresponding to the input queries and documents into a lower-dimensional letter based n-gram vectors, while the subsequent layers learn a non-linear projection of this n-gram vectors to a low-dimensional semantic space. The cosine similarity between the vectors of a query and a document represents their mutual relevance.

While the DSSM treats an input query as a raw term vector or a *bag-of-words*, Shen et al. [36] added a convolutional-pooling structure to the network architecture in the *Convolutional Latent Semantic Model* (CLSM) to capture richer contextual structures in the input text. The CLSM model has been shown to perform better than DSSM and many other existing state-of-the-art techniques on various information retrieval tasks such as Web document ranking and contextual entity search [13].

An examination of the CLSM model outputs reveals syntactic and semantic regularities in the distributed representation of queries. The regularities are akin to the ones reported by Mikolov et al. [31] about the embeddings learnt by continuous space language models, where simple vector offsets between words were found to capture semantic and syntactic inter-word relationships. Mikolov et al. [29] further proposed models that can be trained on large scale datasets and extended the vector representations to phrases [30]. Unlike these continuous space language models [30, 31], CLSM can project multi-word variable length queries into the embedding space. In this paper, we study the vector offset technique in the context of the CLSM outputs.

Query auto-completion. Most modern browsers, search engines, text editors and command shells implement some form of an auto-completion feature to aid users in faster text entry. In Web search, *pre-computed auto-completion* systems are popular, where the suggestions are typically filtered by exact prefix matching from a pre-selected set of candidates and ranked according to past popularity. Ranking suggestions by past frequency is commonly referred to as the *MostPopularCompletion* (MPC) model and can be regarded as a maximum likelihood approximator [1]. Given a prefix $\mathcal{P}$ and all queries $\mathcal{Q}$ from the search logs,

$$MPC(\mathcal{P}) = \underset{\bar{q} \in completions(\mathcal{P})}{\arg\max} P(\bar{q}), \quad P(\bar{q}) = \frac{Freq(\bar{q})}{\sum_{q_i \in \mathcal{Q}} Freq(q_i)} \tag{1}$$

Weber and Castillo [39] and Shokouhi [37] showed how query distributions change across different user demographics and argued that QAC systems based on personalization features can significantly outperform popularity-based baselines. Ranking suggestions based on temporal context has also been explored [38, 41].

The two QAC related studies most relevant to our work have been done by Shokouhi [37] and Kharitonov et al. [24]. To capture short-term context, Shokouhi [37] relied on letter n-gram matches between the previous queries and the candidates, and trained a supervised ranking model for combining them with MPC and other non-

contextual and user demographic features. Kharitonov et al. [24] proposed a unified framework for contextualizing and diversifying the ranking of QAC suggestions. Their empirical evaluations show that by considering the user's previous query alone more than 96% of the improvements can be achieved, as compared to additionally considering the document examination history and diversification context. Given the previous query, their proposed model computes the expected probability of a given completion as follows,

$$P(q_1|q_0) = P(c=0|q_0)P(q_1) + P(c=1|q_0)P(q_1|c=1,q_0) \quad (2)$$

Where c is an indicator variable whose value is 1 if the user continues the current task, and 0 otherwise. The two primary components of the above equation are $P(q_1)$ and $P(q_1|c=1,q_0)$, which correspond to the probability of observing the query q_1 globally and in the context of the query q_0, respectively, in the query logs.

For our evaluation, we implement the supervised ranking framework proposed by Shokouhi and include the n-gram similarity, the query frequency and the query pairwise frequency features among others as described in Section 4. We believe that the baseline used in our experiment is comparable with any state-of-the-art baselines described in the literature for QAC ranking.

Session context. In Web search, Bennett et al. [2] investigated the impact of short-term and long-term user behaviour on relevance prediction, and showed that short-term user history becomes more important as the session progresses. Li et al. [25] evaluated DSSM and convolutional-DSSM for modelling session context for Web search. Besides the primary IR task, QAC as opposed to Web ranking, our work differs from this study by going beyond computing the topical similarity using the existing models and explicitly modelling query reformulations as vectors. We also show the benefits of optimizing a CLSM model directly for capturing session context by training on session query pairs.

Yan et al. [44] proposed an approach that maps queries and clicks to latent search intents represented using Open Directory Project[1] categories for making context-aware query recommendations. Cao et al. [7] and Liao et al. [27] have explored session context using latent concept clusters from click-through bipartite graphs, while Guo et al. [15] represented the user's previous queries using a regularized topic model. Zhang et al. [45] proposed a task-centric click model for characterizing user behaviour within a single search session. Cao et al. [8] learnt a variable length *Hidden Markov Model* from large scale search logs, whereas Boldi et al. [4] studied *random walks* on query-flow graphs for improved recommendations.

Lastly, previous studies on the relationships between neighbouring queries from a search session have been mostly focused on categorizing the reformulations based on broad manually defined taxonomies (e.g., *generalization*, *specialization*, *error correction* and *parallel move*) [5] or understanding the user goals behind common actions (e.g., *addition*, *removal* or *substitution* of terms) [19]. Motivated by the broad manually identified reformulation categories Xiang et al. [43] and Jiang et al. [23] designed simple features for supervised retrieval models. Finally, Guan et al. [14] use reinforcement learning for modifying term weights in response to the observed modifications made to the query by the user.

While clearly using session context for Web search is a well-studied topic, context-sensitive query auto-completion has been discussed less thoroughly in the literature. Also, to the best of our knowledge this is the first time that an explicit vector representation of query reformulations has been proposed and studied.

[1]http://www.dmoz.org/

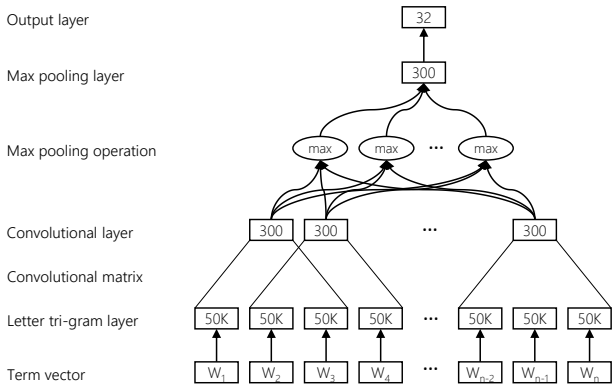

Figure 1: Architecture of the convolutional latent semantic model (CLSM). The model has an input layer that performs the *word hashing*, a convolutional layer, a max pooling layer, and an output layer that produces the final semantic vector representation of the query.

3. MODELS

Architecture. We adopt the CLSM architecture proposed by Shen et al. [36] for our study. As illustrated in Figure 1, the CLSM model is a deep neural network with a convolutional-pooling structure that projects a variable-length text into a fixed-length real-valued vector.

Each word in the input text (e.g., query) is first *hashed* to a letter trigram vector as described by Huang et al. [21]. The *word hashing* step, for example, maps the word "top" to a feature vector with non-zero values corresponding to the trigrams "#to", "top" and "op#", where "#" denotes the word boundary. It has been pointed out in previous work [21] that the trigram hashing technique is a robust strategy for dealing with misspellings and morphological variants of words in the data. It also scales much better for large vocabularies compared to the *one-hot*[2] representation where each unique word is associated with a separate identifier. The number of unique words in a Web corpus can be extremely large, whereas the number of distinct letter trigrams tends to be much more manageable (limited to the top 50K for this study). These factors makes the trigram hashing strategy particularly useful for Web search.

Next, for each word the convolutional layer extracts contextual features based on its immediate neighbours as defined by a predetermined window size. A max pooling layer combines the output of the convolutional layer into a fixed-length feature vector. The max pooling layer in turn is connected to the output layer which produces the final vector representation of the whole query. Unless specified otherwise, for all models in this paper the window size for the convolutional layer is set to three and the dimensions of the output vector to 32.

Training. The training data for the CLSM models consists of source-target text pairs. The original DSSM [21] and CLSM [36] models were trained on *clickthrough data* which consists of pairs of queries and document titles, corresponding to clicked results.

In addition to clickthrough data, we also train the CLSM models on sampled pairs of queries from search logs that were observed in succession during user sessions. In the rest of this paper, we refer to this as the *session pairs* dataset. For a pair of observed queries $\{q_1, q_2\}$, if the dataset includes both the ordering $\{q_1, q_2\}$ and $\{q_2, q_1\}$

[2]http://en.wikipedia.org/wiki/One-hot

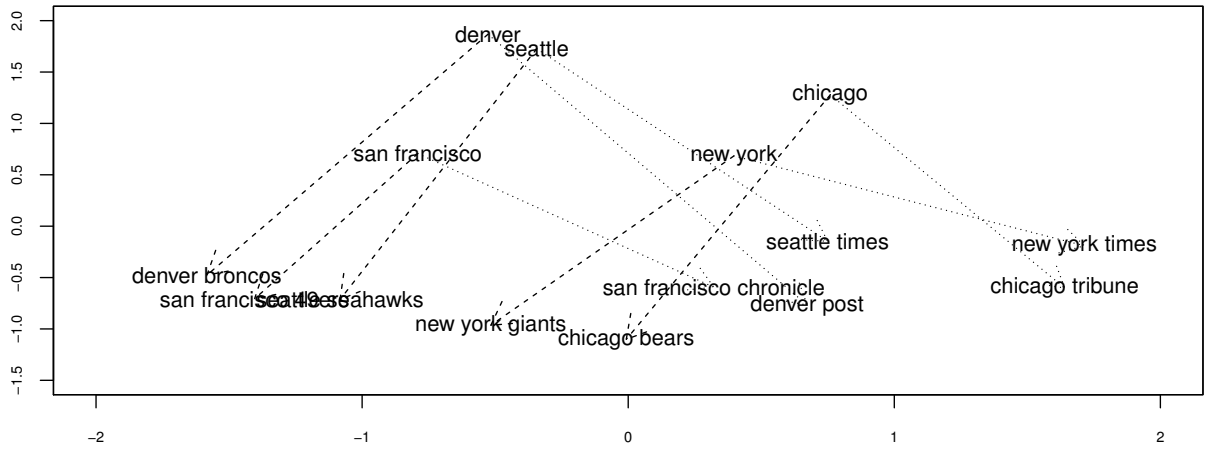

Figure 2: A two-dimensional PCA projection of the 32 dimensional CLSM output vectors shows how intuitively similar intent transitions, represented by the directed edges, are automatically modelled in the embedding space. The CLSM model used for this illustration is trained on the symmetric session pairs dataset.

then we refer to it as the *symmetric* session pairs dataset, otherwise as *asymmetric*. The symmetric session pairs data is further randomly sub-sampled by half to keep the count of the training pairs in both the datasets comparable.

The *session pairs* datasets are extracted from the exact same user sessions from which the *clickthrough* data is generated. While this does not imply that the actual count of training pairs in these two types of datasets are equal, it does make the comparison more meaningful as it assumes the same amount of raw log data is examined for training both the types of models. In practice, however, we did observe the data sizes to be comparable across all three datasets during this study.

All the CLSM models in this study are trained using mini-batch based stochastic gradient descent, as described by Shen et al. [36]. Each mini-batch consists of 1024 training samples (source-target pairs) and for each positive pair 100 negative targets are randomly sampled from the data for that source that were not originally paired.

Distributed Representations. The CLSM models project the queries to an embedding space with fixed number of dimensions. The semantic similarity between two queries q_1 and q_2 in this semantic space is defined by,

$$Sim(q_1, q_2) = cosine(y_1, y_2) = \frac{y_1^\intercal y_2}{\|y_1\|\|y_2\|} \tag{3}$$

where y_1 and y_2 are the CLSM vector outputs corresponding to the two queries, respectively. A close examination of the CLSM output vectors reveal that the learnt distributed representations hold useful information about inter-query relationships. Figure 2 illustrates how the offset vectors between pairs of queries, represented by the directed edges, are directionally similar in the embedding space for similar intent transitions. This matches the observations made by Mikolov et al. [29] on continuous space language models for text processing, and gives us an intuitively understandable representation of query reformulations as their offset vectors in the embedding space. More specifically, we define the *reformulation* from query q_1 to q_2 as,

$$Ref(q_1, q_2) = \hat{y_2} - \hat{y_1} = \frac{y_2}{\|y_2\|} - \frac{y_1}{\|y_1\|} \tag{4}$$

where y_1 and y_2 are the CLSM vector embeddings of the two queries, respectively. This explicit vector representation provides a framework for studying frequently occurring query reformulation patterns. To illustrate this, we randomly sample approximately 65K pairs of queries that were observed in succession in Bing's logs. For each pair, we compute the offset vector using a CLSM model. We then run a simple k-means clustering (k = 100) and examine the top clusters. Example reformulations from five of the biggest clusters are shown in Table 1.

A further study of these reformulation vectors can reveal important insights about user behaviour, such as the popularity of certain reformulation patterns. For example, we randomly sampled 100,000 adjacent pairs of queries from Bing's logs that were observed in search sessions. Our analysis show that there are more pairs similar to the *narrowing* reformulation "new york" $\rightarrow$ "things to do in new york" in the sampled set, than its inverse. Similarly, the misspelling "fcebook" followed by "facebook" is a more commonly observed pattern than the other way around, as illustrated in Figure 3.

Next, we list qualitative examples in Table 2 to demonstrate the predictive aspect of these reformulation vectors. Similar to the analogy based test proposed by Mikolov et al. [29], these examples show that we can obtain intuitively understandable results by performing simple algebraic operations in the embedding space. For example, we compute the vector sum of the projections (normalized to their unit norm) of the queries "new york" and "newspaper".

$$y_{target} = \hat{y}_{newyork} + \hat{y}_{newspaper} = \frac{y_{newyork}}{\|y_{newyork}\|} + \frac{y_{newspaper}}{\|y_{newspaper}\|} \tag{5}$$

Then from a fixed set of candidates we find the query whose embedding has the highest cosine similarity with y_{target}. For our analysis we picked the top one million most popular queries from one day of Bing's logs as the candidate set. In this query set, the closest query vector to y_{target} corresponds to the query "new york times". Similarly, the nearest neighbour search for $vector($"how old is gwen stefani"$) - vector($"gwen stefani"$) + vector($"meghan trainor"$)$ yields a vector close to $vector($"how old is meghan trainor"$)$. These examples show that the vector representation captures simple syntactic as well as semantic relationships. We intentionally also include some examples where the nearest neighbour search yields unex-

Table 2: Examples of simple syntactic and semantic relationships in the query embedding space. The nearest neighbour search is performed on a candidate set of one million most popular queries from one day of Bing's logs.

Query vector	Nearest neighbour
$vector(\text{“chicago”}) + vector(\text{“newspaper”})$	$vector(\text{“chicago suntimes”})$
$vector(\text{“new york”}) + vector(\text{“newspaper”})$	$vector(\text{“new york times”})$
$vector(\text{“san francisco”}) + vector(\text{“newspaper”})$	$vector(\text{“la times”})$
$vector(\text{“beyonce”}) + vector(\text{“pictures”})$	$vector(\text{“beyonce images”})$
$vector(\text{“beyonce”}) + vector(\text{“videos”})$	$vector(\text{“beyonce videos”})$
$vector(\text{“beyonce”}) + vector(\text{“net worth”})$	$vector(\text{“jaden smith net worth”})$
$vector(\text{“www.facebook.com”}) - vector(\text{“facebook”}) + vector(\text{“twitter”})$	$vector(\text{“www.twitter.com”})$
$vector(\text{“www.facebook.com”}) - vector(\text{“facebook”}) + vector(\text{“gmail”})$	$vector(\text{“www.googlemail.com”})$
$vector(\text{“www.facebook.com”}) - vector(\text{“facebook”}) + vector(\text{“hotmail”})$	$vector(\text{“www.hotmail.xom”})$
$vector(\text{“how tall is tom cruise”}) - vector(\text{“tom cruise”}) + vector(\text{“tom selleck”})$	$vector(\text{“how tall is tom selleck”})$
$vector(\text{“how old is gwen stefani”}) - vector(\text{“gwen stefani”}) + vector(\text{“meghan trainor”})$	$vector(\text{“how old is meghan trainor”})$
$vector(\text{“how old is gwen stefani”}) - vector(\text{“gwen stefani”}) + vector(\text{“ariana grande”})$	$vector(\text{“how old is ariana grande 2014”})$
$vector(\text{“university of washington”}) - vector(\text{“seattle”}) + vector(\text{“chicago”})$	$vector(\text{“chicago state university”})$
$vector(\text{“university of washington”}) - vector(\text{“seattle”}) + vector(\text{“denver”})$	$vector(\text{“university of colorado”})$
$vector(\text{“university of washington”}) - vector(\text{“seattle”}) + vector(\text{“detroit”})$	$vector(\text{“northern illinois university”})$

pected results (e.g., $vector(\text{“beyonce”}) + vector(\text{“net worth”})$) to highlight that these predictions are often noisy.

4. EXPERIMENT SETUP

Our empirical evaluations are based on the learning to rank framework proposed by Shokouhi [37] for personalized query autocompletions. In this setup, we learn a supervised ranking model based on training data generated from implicit user feedback. The output of the CLSM models, described in the previous section, are used to generate additional features for this supervised ranking model. The baseline ranking model (henceforth referred to simply as the *baseline model*) contains both the non-contextual and the (non-CLSM based) contextual features. We compare all models using the mean reciprocal rank (MRR) metric, and the study is repeated on two different testbeds to further confirm the validity of the results.

4.1 Testbeds

We conduct our experiments on a large scale search query dataset sampled from the logs of the Bing search engine. We also reproduce our results using the publicly available AOL query logs [33]. In the rest of this paper we refer to these two datasets as the *Bing testbed* and the *AOL testbed*, respectively.

Bing testbed. Bing's logs contain a record of all the queries submitted by its users associated with the corresponding anonymized user IDs, timestamps and any clicked Web results[3] (the URL and the displayed title). We sampled queries from these logs for the duration of the last week of October, 2014 and use this as the *background data*, for computing the feature values and training the CLSM models. From the first week of November, we sampled 175,392 queries from two consecutive days for training the supervised ranking models, and from the following two individual days we sampled 79,000 queries for validation and 74,663 queries for testing, respectively.

AOL testbed. This dataset contains queries sampled between 1 March, 2006 and 31 May, 2006. For each query, the data includes an anonymized user ID and a timestamp. If a result was clicked then the rank of the clicked item and the domain portion of its URL are also included. In aggregate, the data contains 16,946,938 query submissions and 36,389,567 document clicks by 657,426 users.

We consider all queries before 1 May, 2006 as the *background data*. All queries from the next two weeks of data are used for training the supervised ranking models, and the remaining two sets, consisting of one week of data each, is used for validation and testing, respectively.

To have a separation of users in training and test datasets, on both the testbeds we use only the users with even user IDs for training and validation, and those with odd numbered user IDs for testing. Also, in all the datasets the queries are lower-cased and the punctuations are removed.

4.2 Learning to rank

To generate the training, the validation and the test sets we sample query impressions from the corresponding portions of the logs. For each query impression, a prefix is generated by splitting the query at a randomly selected position[4]. For each prefix a positive relevance judgment is assigned to the suggestion candidate that matches the final submitted query and all the others are labelled as irrelevant.

The training data collected in the above process consists of labelled *prefix-query* pairs. With respect to the choice of learning-to-rank algorithms, we chose LambdaMART [42], a boosted tree version of LambdaRank [6], that won the Yahoo! Learning to Rank Challenge (2010) [9] and is considered as one of the state-of-the-art learning algorithms. We train 500 trees across all our experiments with the same set of fixed parameters tuned using standard training and validation on separate sets.

We consider the top 10 million most popular queries in the background data as the pre-computed list of suggestion candidates and filter out all the impressions where the final submitted query is not present in this list. For each impression in the training, the validation and the test sets we retain a maximum of 20 suggestion candidates - the submitted query as the positive candidate and 19 other most frequently observed queries from the background data that starts with the same prefix, as the negative examples. Furthermore, for each impression up to 10 previous queries from the same session are

[3]For impressions with multiple clicked results we consider only the last clicked document.

[4]The prefixes in our study are strictly shorter than the original query and limited to no more than 30 characters in length.

Table 1: k-means clustering of 65K in-session query pairs observed in search logs. Examples from five of the top ten biggest clusters shown here. The first and the second clusters contain examples where the follow up query is a different formulation of the exact same intent. The third and the fourth clusters contain examples of *narrowing* intent, in particular the fourth cluster contains reformulations where the additional specification is based on location disambiguation. Finally, the last cluster contains examples of intent jumps across tasks.

soundcloud	→	www.soundcloud.com
coasthills coop	→	www.coasthills.coop
american express	→	www.barclaycardus.com login
duke energy bill pay	→	www.duke-energy.com pay my bill
cool math games	→	www.coolmath.com
majesty shih tzu	→	what is a majesty shih tzu
hard drive dock	→	what is a hard drive dock
lugia in leaf green	→	where is lugia in leaf green
red river log jam	→	what is th red river log jam
prowl	→	what does prowl mean
rottweiler	→	rottweiler facebook
sundry	→	sundry expense
elections	→	florida governor race 2014
pleurisy	→	pleurisy shoulder pain
elections	→	2014 rowan county election results
cna classes	→	cna classes in lexington tennessee
container services inc	→	container services ringgold ga
enclosed trailers for sale	→	enclosed trailers for sale north carolina
firewood for sale	→	firewood for sale in asheboro nc
us senate race in colorado	→	us senate race in georgia
siol	→	facebook
cowboy bebop	→	facebook
mr doob	→	google
great west 100 west 29th	→	facebook
avatar dragons	→	youtube

made available for computing the session context features. Similar to other previous work [11, 22] we define the end of a session by a 30 minute window of user inactivity.

For our final evaluation we report the Mean Reciprocal Rank of the submitted query averaged over all sampled impressions on each of the two testbeds.

4.3 Features

The baseline contextual and non-contextual features, as well as the features based on the CLSM outputs are described in this section.

Non-contextual features. The *MostPopularCompletion* (MPC) model is one of the baselines for our study. We also use the output of this model as a feature for the supervised ranking model. Other non-contextual features include the *prefix length* (in characters), the *suggestion length* (in both characters and words), the *vowels to alphabets* ratio in the suggestion and a boolean feature indicating whether the *suggestion contains numeric* characters.

N-gram similarity features. We compute the character n-gram similarity (n=3) between the suggestion candidate and the previous queries from the same user session. This is an implementation of the *short history* features described by Shokouhi [37]. A maximum of 10 previous queries are considered.

Pairwise frequency feature. From the background data, we generate the top 10 million most popular adjacent pairs of queries observed in search sessions. For a given impression, the previous query and the suggestion candidate pair is matched against this dataset and the corresponding frequency count is used as the feature value. If no matches are found, then the feature value is set to zero.

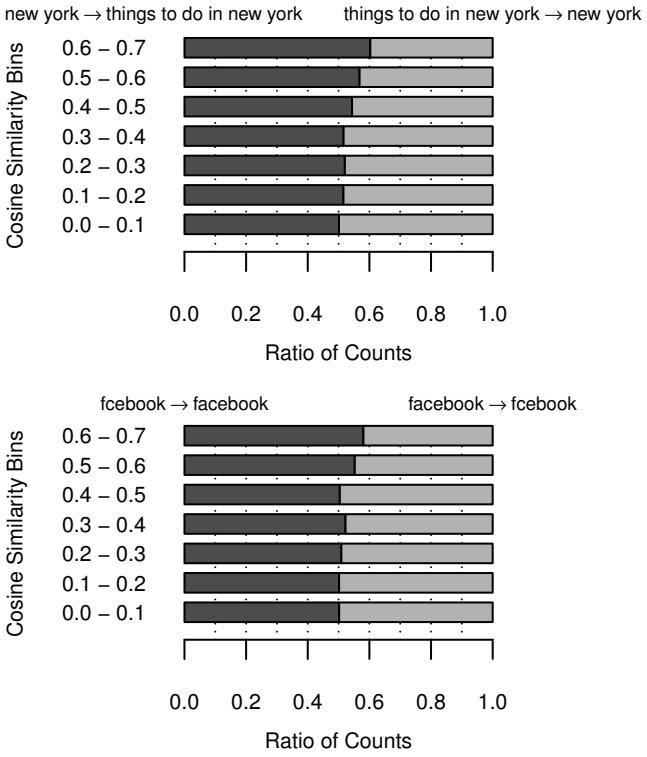

Figure 3: Visualization of the cosine similarity scores of a given reformulation with respect to a set of 100,000 other reformulations randomly sampled from Bing's logs. The similarity scores are binned and the ratio of the counts are shown above. The counts corresponding to bins with cosine similarity greater than 0.7 were too small, hence excluded.

CLSM topical similarity features. The CLSM models are trained as described in Section 3 using the background portion of the data on each testbed. The cosine similarity between the CLSM vectors corresponding to the suggestion candidate and a maximum of previous 10 queries from the same session are computed and used as 10 distinct features in the QAC ranking model.

Training on the session query pairs data produces a pair of *pre-post* CLSM models. When trained on the asymmetric data, the pre-model is used for projecting the user's previous queries and the $post$-model is used for projecting the suggestion candidates for the cosine similarity computation. For the symmetric data however, both the pre- and the $post$- models are equivalent, and hence we use only the pre- model in our experiments.

The AOL logs contains only the domain portion of the clicked results. Hence we are unable to get the corresponding document titles. Therefore we only train the session pairs based CLSM models on this testbed and report those results in this paper.

CLSM reformulation features. We compute the n-dimensional (n=32) vector representation of the reformulation from the previous query to the suggestion candidate. The raw values from this vector are used as n distinct features into the supervised ranking model.

Table 3: Comparison of QAC ranking models trained with CLSM based features against the MPC model and the supervised baseline ranker model. All the reported MRR improvements are statistically significant by the t-test ($p < 0.01$) over the MPC baseline and the baseline model. Additionally, corresponding to each of the different CLSM models, the ranking model containing both the *similarity* and the *reformulation* features shows statistically significant ($p < 0.01$) improvements in MRR over the model containing only the *similarity* features on both the testbeds. The three highest MRR improvements per testbed are shown in bold below.

	Bing	AOL	
Models	% Improv.	MRR	% Improv.
Baselines			
MostPopularCompletion	-	0.5110	-
Baseline Model	+48.6	0.7983	+56.2
CLSM (query-document pairs)			
All features	+55.9	-	-
Reformulation features	+54.3	-	-
Similarity features	+55.3	-	-
CLSM (Asymmetric session query pairs)			
All features	**+58.0**	0.8775	**+71.7**
Reformulation features	**+57.4**	0.8747	**+71.2**
Similarity features	+54.2	0.8580	+67.9
CLSM (Symmetric session query pairs)			
All features	**+59.0**	0.8801	**+72.2**
Reformulation features	+57.2	0.8744	+71.1
Similarity features	+55.8	0.8636	+69.0

For both the session pair based models, the pre- model is used for projecting the suggestion candidates, as well as the previous query.

5. RESULTS

Table 3 compares the results of training the supervised QAC ranking model with the different CLSM based session context features. Due to the proprietary nature of Bing's data, we report only relative improvements of each of the models over the MPC baseline for this testbed. On the AOL testbed, however, we report both the absolute MRR values and the relative improvements for all the models.

On both the testbeds, the *baseline model* which also contains session context features (the n-gram similarity and the pairwise frequency) shows a large improvement over the *MPC* baseline, which is expected. All the models trained with the CLSM based contextual features show further statistically significant improvements over the baseline model. Both the CLSM models trained on *session pairs* perform better than the models trained on *clickthrough* data, with the model trained on the *symmetric* session pairs performing slightly better overall. Table 5 lists examples of cases from one of the test sets where the ranking model with the CLSM based contextual features perform better compared to both the baselines.

The supervised ranking models trained with both the CLSM based similarity features and the CLSM based reformulation features perform better than the corresponding models trained with the similarity features alone. The improvements are statistically significant and demonstrate the additional information provided by the reformulation features to the ranking model over the CLSM based similarity features. The reformulation features perform particularly superior when the CLSM model has been trained on the *session pairs* dataset.

Table 4: Comparison of QAC ranking models with CLSM similarity features computed considering different maximum number of previous queries in the same session. The results show that most of the improvements from short-term history similarity features can be achieved by considering just the immediately previous query.

	Bing	AOL	
Models	% Improv.	MRR	% Improv.
Baselines			
MostPopularCompletion	-	0.5110	-
Baseline Model	+48.6	0.7983	+56.2
CLSM (Symmetric session query pairs)			
Previous 1 query	+55.2	0.8631	+68.9
Previous 3 queries	+56.1	0.8639	+69.1
Previous 5 queries	+56.1	0.8642	+69.1
Previous 10 queries	+55.8	0.8636	+69.0

Table 4 shows the impact of considering different number of previous queries in the session for computing the CLSM based similarity features. The results indicate that considering the previous query alone achieves most of the improvements observed from these similarity features.

We also compare the improvements from the different models based on the length of the input prefixes. Bar-Yossef and Kraus [1] have previously reported that non-contextual QAC systems generally perform poorly when the user has typed only a few characters due to the obvious ambiguity in user intent. Figure 4 illustrates this behaviour on the AOL testbed. Both the supervised ranking models, the baseline and the model with the CLSM features, show significantly large improvements over the MPC baseline on short prefixes. After the user has typed a few more characters in the search box, the set of suggestion candidates reduce significantly and the performance of the MPC model improves. Therefore the improvements on the longer prefixes are smaller for both the supervised ranking models. The supervised ranking model with the CLSM features, however, show statistically significant better MRR compared to both the MPC baseline and the supervised baseline ranking model on all the prefix length based segments. Finally, Figure 5 shows that better MRR can be achieved by training the CLSM model with a higher number of output dimensions.

6. DISCUSSION

In the previous section we demonstrated significant improvements in the query auto-completion ranking task using the CLSM based session context features. We now discuss potential implications of these vector representations on session modelling and list some of the assumptions and limitations of the evaluation framework used in this study.

Implications for session modelling. The distributed representation of queries and query reformulations provides an interesting framework for thinking about sessions and task context. The sequence of queries (and documents) in a search session can be considered as a directed path in the embedding space. What are the common attributes shared by these *session paths*? What properties of these paths vary depending on the type of the user task or information need? These are examples of research questions that may be interesting to study under the distributed representation framework. Hassan et al. [16], for example, studied long search sessions and compared user behaviours when the user is *struggling* in their

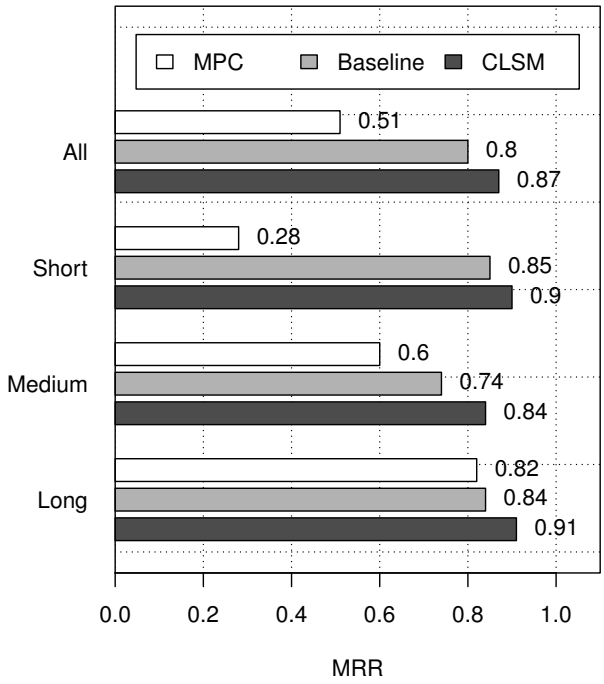

Figure 4: Comparison of the MPC model, the baseline ranker model and the experimental ranker model with the CLSM based features (the CLSM model considered here is trained on *symmetric session pairs* with *all features*) across different prefix lengths on the AOL testbed. Prefixes less than 4 characters are considered as *short*, 4 to 10 characters as *medium*, and greater than 10 characters as *long*. Both the supervised ranking models contain contextual features (CLSM based or otherwise) and hence show large improvements on the short prefixes where the ambiguity is maximum. Across all prefix lengths the model with CLSM based features out-perform the baseline ranking model. All reported differences in MRR are statistically significant by the t-test ($p < 0.01$).

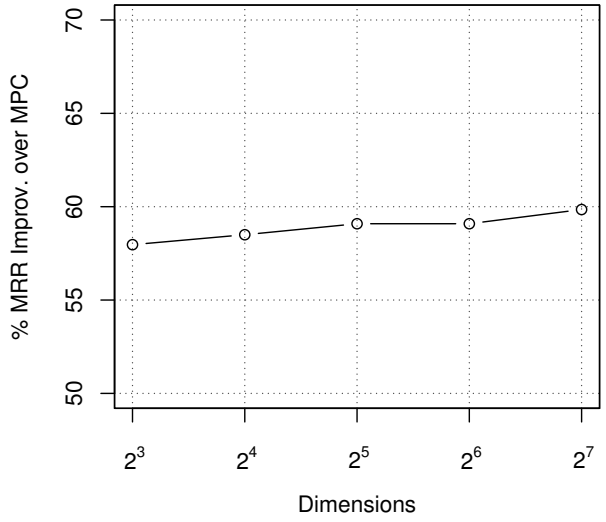

Figure 5: Evaluation of the impact of training the CLSM models with different number of dimensions. Except for the pair of CLSM models trained with 32 and 64 dimensions, all other reported differences in MRR are statistically significant by the t-test ($p < 0.01$).

information task to when they are *exploring*. Features based on the CLSM projections of queries and documents, such as the types of user reformulations in the session and the similarity between submitted queries and viewed documents, can be explored to improve the prediction accuracy for such session classification tasks.

In this paper we have examined individual query reformulations. Studying reformulation chains may teach us further about how user intents evolve during a session and support the design of future models for *session search*. For example, White and Huang [40] have explored the value of *search trails*, over the origins and the destinations. While we have only examined the representation of queries and reformulations in this paper, CLSM also allows for documents to be represented in the same embedding space. A unified study of queries, reformulations and viewed (searched or browsed) documents using the vector representation framework is an area for future work.

In the query change retrieval model (QCM) proposed by Guan et al. [14], we can explore using the reformulation vectors for representing the user agent's actions. Similarly, we may be able to gain further insights by conducting a similar study as Hollink et al. [19] by examining query changes under the vector representation framework.

Generating a distributed representation of users based on their search and other online activities is also an interesting problem. Other potential directions for future studies using the vector framework includes examining how query reformulations differ based on the search expertise of the user and the kind of device the search is performed on.

Assumptions and limitations. We have based our empirical study on the supervised ranking framework proposed by Shokouhi [37]. In doing so, we inherit some of the assumptions in the designs of that framework. Firstly, we assume that the user has a predetermined query in mind for input and would be satisfied if it appears in the QAC suggestions list. However Hofmann et al. [17] have shown that due to the high examination bias towards top-ranked results, sub-optimal QAC ranking can negatively affect the quality of the query submitted by the user. As most popular Web search engines implement some form of an auto-completion feature, it is likely that those QAC systems influenced the actual query observed in the logs. We ignore this effect in the generation of our training and test sets.

The generation of the prefixes also assumes that each query was typed *completely* by the user in a strictly left-to-right progression and the user is equally likely to examine and engage with the QAC system after each character is typed. In practice, however, users are often aided in the query formulation process (partially or completely) by various features of the search engine, such as QAC or related query recommendations. Users also often correct already entered text during the query formulation process. In these cases the generation of all possible prefixes from the submitted query does not accurately reflect the actual prefixes typed by the user.

Li et al. [26] and Mitra et al. [32] have also shown that user engagement with QAC varies with different factors such as whether the user is at a word boundary or the distance of the next character to be typed on the keyboard. This suggests that prefixes should be sampled with different importance depending on the likelihood that the user would examine the QAC suggestions for that prefix. Li et al. [26] proposed a two-dimensional click model for QAC, demonstrating that in the presence of keystroke level logging of QAC sessions the click model can be used to filter out prefix impressions

Table 5: Examples from the win-loss analysis on one of the test sets. For a given prefix and the previous query from the same user session, the top ranked suggestion by the different models are shown below. The actual submitted query is denoted by the checkmark (✓). The CLSM features include both the similarity and the reformulation features and the CLSM model is trained on the symmetric session pairs dataset.

Previous query	the fighter	airline tickets	pace university
Prefix	amer	amer	amer
MPC model	american express	american express	american express
Supervised Baseline model	american express	american express	american girl
Supervised Model with CLSM Features	american psycho movie ✓	american airlines ✓	american university ✓

Previous query	usairways	2007 toyota yaris	master of philosophy jobs
Prefix	us	us	us
MPC model	us elections 2014 predictions	us elections 2014 predictions	us elections 2014 predictions
Supervised Baseline model	usps.com	usaa	usps.com
Supervised Model with CLSM Features	usairways.com ✓	used cars ✓	usa jobs ✓

with low expected probability of examination. However, as the testbeds we consider for this study do not all have the keystroke level granularity of records, we do not pursue this line of experimentation.

Lastly, Shokouhi [37] generates all the possible prefixes of each query in the log data. This results in an obvious over-representation of long prefixes in the generated datasets. To avoid this issue we extract a single prefix per query by splitting at a random position within the query.

Despite the different underlying assumptions, the framework proposed by Shokouhi [37] provides a reasonable setup to learn a baseline context-aware ranking model for QAC, and hence we adopt it for this study.

7. CONCLUSION

We have demonstrated that the distributed representation of queries by the convolutional latent semantic models holds useful information about inter-query relationships. The reformulation vectors exhibit regularities that makes them interesting for modelling session context for query suggestion tasks. Our experiments show that using features based on the reformulation vectors improves MRR for QAC ranking over using features based on the query vectors alone. The best improvements, however, are achieved by the combination of features based on both these vector representations. We have also demonstrated that training the latent semantic models on session query pairs produces further improvements over the model trained on query-document pairs. While the biggest improvements are observed on short prefixes, the ranking model containing the CLSM based features perform better than the supervised ranking baseline on all the prefix length based segments. We have also studied the effects of considering different number of previous queries within the session for context and the number of dimensions used to represent the query and reformulation vectors on the model performance. While we evaluate these models on the query auto-completion ranking task, the features we described in this paper may also be useful for generating context sensitive related query recommendations and query rewriting. Furthermore, by projecting documents to this same embedding space, future studies may be able to extend these contextual features to document ranking in Web search.

Lastly, the reformulation vectors provide an interesting framework for studying sessions and intent progressions. We anticipate that these distributed representations of queries, documents and reformulations will become more frequently used as tools for future studies on search personalization and session search.

Acknowledgements. The author is grateful to Nick Craswell, Milad Shokouhi, Filip Radlinski, Vanessa Murdock, Xiaodong He, Jianfeng Gao, Piotr Mirowski, Peter Bailey, David Hawking, Katja Hofmann, Panagiotis Tigkas and Daniel Voinea for their insightful feedback, questions and discussions around related areas.

References

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

[2] 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, 2012.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3: 993–1022, 2003.

[4] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *Proceedings of the 2009 workshop on Web Search Click Data*, pages 56–63. ACM, 2009.

[5] P. Boldi, F. Bonchi, C. Castillo, and S. Vigna. Query reformulation mining: models, patterns, and applications. *Information retrieval*, 14(3):257–289, 2011.

[6] C. Burges, R. Ragno, and Q. Le. Learning to rank with non-smooth cost functions. In *Proc. NIPS*, 2006.

[7] 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, 2008.

[8] H. Cao, D. Jiang, J. Pei, E. Chen, and H. Li. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *Proc. WWW*, pages 191–200. ACM, 2009.

[9] O. Chapelle, Y. Chang, and T.-Y. Liu. The yahoo! learning to rank challenge, 2010. *URL http://learningtorankchallenge. yahoo. com*.

[10] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[11] D. Downey, S. Dumais, D. Liebling, and E. Horvitz. Understanding the relationship between searchers' queries and information goals. In *Proc. CIKM*, pages 449–458. ACM, 2008.

[12] J. Gao, K. Toutanova, and W.-t. Yih. Clickthrough-based latent semantic models for web search. In *Proc. SIGIR*, pages 675–684. ACM, 2011.

[13] J. Gao, P. Pantel, M. Gamon, X. He, L. Deng, and Y. Shen. Modeling interestingness with deep neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2014.

[14] D. Guan, S. Zhang, and H. Yang. Utilizing query change for session search. In *Proc. SIGIR*, pages 453–462. ACM, 2013.

[15] J. Guo, X. Cheng, G. Xu, and X. Zhu. Intent-aware query similarity. In *Proc. CIKM*, pages 259–268, 2011.

[16] A. Hassan, R. W. White, S. T. Dumais, and Y.-M. Wang. Struggling or exploring?: disambiguating long search sessions. In *Proc. WSDM*, pages 53–62. ACM, 2014.

[17] K. Hofmann, B. Mitra, F. Radlinski, and M. Shokouhi. An eye-tracking study of user interactions with query auto completion. In *Proc. CIKM*, pages 549–558. ACM, 2014.

[18] T. Hofmann. Probabilistic latent semantic indexing. In *Proc. SIGIR*, pages 50–57. ACM, 1999.

[19] V. Hollink, J. He, and A. de Vries. Explaining query modifications. In *Advances in Information Retrieval*, pages 1–12. Springer, 2012.

[20] J. Huang and E. N. Efthimiadis. Analyzing and evaluating query reformulation strategies in web search logs. In *Proc. CIKM*, pages 77–86. ACM, 2009.

[21] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proc. CIKM*, pages 2333–2338. ACM, 2013.

[22] 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.

[23] J.-Y. Jiang, Y.-Y. Ke, P.-Y. Chien, and P.-J. Cheng. Learning user reformulation behavior for query auto-completion. In *Proc. SIGIR*, pages 445–454. ACM, 2014.

[24] E. Kharitonov, C. Macdonald, P. Serdyukov, and I. Ounis. Intent models for contextualising and diversifying query suggestions. In *Proc. CIKM*, pages 2303–2308. ACM, 2013.

[25] X. Li, C. Guo, W. Chu, Y.-Y. Wang, and J. Shavlik. Deep learning powered in-session contextual ranking using clickthrough data. In *Proc. NIPS*, 2014.

[26] Y. Li, A. Dong, H. Wang, H. Deng, Y. Chang, and C. Zhai. A two-dimensional click model for query auto-completion. In *Proc. SIGIR*, pages 455–464. ACM, 2014.

[27] 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 Trans. on Intelligent Systems and Technology*, 3(1):17:1–17:40, Oct. 2011.

[28] C. Liu, J. Gwizdka, J. Liu, T. Xu, and N. J. Belkin. Analysis and evaluation of query reformulations in different task types. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–9, 2010.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*, pages 3111–3119, 2013.

[31] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. Citeseer, 2013.

[32] B. Mitra, M. Shokouhi, F. Radlinski, and K. Hofmann. On user interactions with query auto-completion. In *Proc. SIGIR*, pages 1055–1058, 2014.

[33] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *Proc. InfoScale*. ACM, 2006. ISBN 1-59593-428-6.

[34] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7): 969–978, 2009.

[35] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proc. SIGIR*, pages 43–50. ACM, 2005.

[36] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proc. WWW*, pages 373–374. International World Wide Web Conferences Steering Committee, 2014.

[37] M. Shokouhi. Learning to personalize query auto-completion. In *Proc. SIGIR*, pages 103–112, 2013.

[38] M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *Proc. SIGIR*, pages 601–610, 2012.

[39] I. Weber and C. Castillo. The demographics of web search. In *Proc. SIGIR*, pages 523–530, 2010.

[40] R. W. White and J. Huang. Assessing the scenic route: measuring the value of search trails in web logs. In *Proc. SIGIR*, pages 587–594. ACM, 2010.

[41] S. Whiting and J. M. Jose. Recent and robust query auto-completion. In *Proc. WWW*, pages 971–982. International World Wide Web Conferences Steering Committee, 2014.

[42] Q. Wu, C. Burges, K. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Journal of Information Retrieval*, 13:254–270, 2009.

[43] 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.

[44] X. Yan, J. Guo, and X. Cheng. Context-aware query recommendation by learning high-order relation in query logs. In *Proc. CIKM*, pages 2073–2076. ACM, 2011.

[45] Y. Zhang, W. Chen, D. Wang, and Q. Yang. User-click modeling for understanding and predicting search-behavior. In *Proc. SIGKDD*, pages 1388–1396. ACM, 2011.