Situational Context for Ranking in Personal Search

Hamed Zamani[†*] Michael Bendersky[‡] Xuanhui Wang[‡] Mingyang Zhang[‡]
[†]Center for Intelligent Information Retrieval, University of Massachusetts Amherst, MA 01003
[‡]Google Inc., Mountain View, CA 94043
zamani@cs.umass.edu {bemike, xuanhui, mingyang}@google.com

ABSTRACT

Modern search engines leverage a variety of sources, beyond the conventional query-document content similarity, to improve their ranking performance. Among them, query context has attracted attention in prior work. Previously, query context was mainly modeled by user search history, either long-term or short-term, to help the ranking of future queries. In this paper, we focus on *situational context*, i.e., the contextual features of the current search request that are independent from both query content and user history. As an example, situational context can depend on search request time and location. We propose two context-aware ranking models based on neural networks. The first model learns a low-dimensional deep representation from the combination of contextual features. The second model extends the first one by leveraging binarized contextual features in addition to the high-level abstractions learned using a deep network.

The existing context-aware ranking models are mainly based on search history, especially click data that can be gathered from the search engine logs. Although context-aware models have been widely explored in web search, their influence on search scenarios where click data is highly sparse is relatively unstudied. The focus of this paper, personal search (e.g., email search or on-device search), is one of such scenarios. We evaluate our models using the click data collected from one of the world's largest personal search engines. The experiments demonstrate that the proposed models significantly outperform the baselines which do not take context into account. These results indicate the importance of situational context for personal search, and open up a venue for further exploration of situational context in other search scenarios.

[*]Work done while at Google.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052648

Keywords

Personal search; email search; contextual information; query context; deep learning

1. INTRODUCTION

Many of the standard information retrieval models solely consider query content to compute <*query, document*> similarity scores. However, in many real-world search applications, more types of information, which can represent query context, are available. This context, in addition to the query content, can be leveraged to improve search quality.

Query context can be defined in various ways. Previous work [5, 31, 34, 36, 43, 45, 46] often refers to query context as the information from the previous queries and their corresponding clickthrough data (i.e., short- and/or long-term search history). However, the information about the previous queries is not always available, e.g., in the case of cold-start users or the first query in each session. Using search history as context can also be disrupted by search task switches.

Orthogonally to user search history, there are other types of contextual features, referred to as *situational context*, that can be leveraged for ranking. For instance, during search logs analysis, we often observed large variations of clickthrough rates (CTR) and click positions across countries, languages, and days of the week. These observations are supported by prior research [14, 30], and validate the following assumption:

User search behavior may depend on the situational context of a query — the properties of the current search request, independent from the query content.

The time of a search request and the location of the user while submitting the request are two examples of situational context which can be employed in ranking models. There are several other situational features which may influence user search behavior such as user device, browser, language, etc. Recent increase in the number of search requests from mobile devices intensifies the importance of situational context. Users tend to type very short and ambiguous queries on mobile devices; previous work demonstrates that employing context information can fill the gaps in query understanding in such scenarios [27].

Based on the aforementioned observations and assumption, we build two context-aware ranking models to make use of the situational context. Since situational features, such as location, can be highly sparse, we design our models based on neural networks, which have shown impressive

performance in many tasks which are dealing with sparse data [18, 33].

Our neural network models are based on a simple yet effective semantic matching network, which we extend to support contextual features. Our first context-aware model learns a low-dimensional dense context representation from a combination of sparse contextual features. The sparse features are first embedded into a lower-dimensional space to obtain close vector representations for similar feature values, and then modeled jointly using non-linear hidden layers to enhance feature interactions.

In the second context-aware model, we design a wide & deep network based on the following hypothesis: although we can learn a meaningful abstract representation for contextual features, we may lose some information due to high-level abstraction. In this model, we consider binarized contextual raw (sparse) features in addition to their dense representations learned via a deep architecture. The main idea behind using the binarized raw features (the wide side of the network) is modeling a memorization approach [10].

Various definitions of query context have been explored for web search [5, 31, 34, 36, 43, 45, 46] where clickthrough data plays a critical role. However, the influence of contextual features on other search scenarios, where click data is highly sparse, is relatively unknown. *Personal search* is one of these scenarios. Personal search is a well-known information retrieval (IR) task with many applications, such as email search [3, 9, 44], desktop search [15], and on-device search [28]. The main difference between personal search and web search is that users in personal search only have access to their own private documents. As a result, the vast majority of click models for web search, which learn the probability of a click from a large amount of click data per query-document pair are not applicable to personal search [3, 44]. This is one of the reasons that email search, as an example of personal search, still remains a difficult, time-consuming, and frustrating search scenario [9, 16]. In this paper, we focus on personal search as the application for our evaluation. To the best of our knowledge, this is the first attempt to study context-aware models for personal search.

The primary contributions of this paper can be summarized as follows. We first motivate the idea of using situational context for personal search in Section 3. We then formally define situational context, and propose two context-aware semantic matching models based on deep neural networks in Section 4. Finally, in Section 5, we conduct a thorough evaluation of our techniques using click data gathered from one of the world's largest personal search engines. We compare the two proposed context-aware semantic matching models to a semantic matching model, which does not take context into account [19]. We also use the output of our models as signals in a large scale learning to rank framework. An array of experiments clearly demonstrates the importance of employing situational context for ranking in personal search. The models incorporating situational context lead to significant improvements over several state-of-the-art baselines both as stand-alone matching functions, and as a part of a large scale learning to rank framework.

2. RELATED WORK

In this section, we first review prior work on personal search. We further study the use of context for improving ranking in different search scenarios. We finally provide an overview of recent work on deep learning models for various IR tasks.

2.1 Personal Search

Personal search refers to a search scenario which is over a collection of private documents of the user who initiates the search. Email search [9, 38], desktop search [15], and on-device search [28] are all examples of this kind. In personal search, each user has its own unique document corpus, which makes personal search different from web search. For instance, personal documents are often private information of users, thus developing explicit relevance judgments for personal search would be extremely expensive or even impossible. As a result, in personal search, designing ranking models based on clickthrough data as an indicator of implicit relevance judgment is essential.

The vast majority of learning methods based on click data which have been proposed for web search cannot be directly applied to personal search [44]. As documents in personal search environments are user-specific, clickthrough data based on the query-document pairs is extremely sparse. To conquer the sparsity, Bendersky et al. [3] proposed to use the clickthrough rate (CTR) of frequent n-grams extracted from the query-document pairs. We refer to this as the CTR memorization. In this paper, we develop a context-aware model on top of a semantic matching model using neural network and show that the new model can significantly outperform the CTR memorization baseline [3].

2.2 Context-Aware Ranking

Context-aware ranking has been widely explored in web search [5, 31, 34, 36, 43, 45, 46]. Most of the existing context-aware search models are based on the search history of whom submitted the query. Search history can be categorized to short- and long-term histories. Short-term search history includes the queries requested in the current search session and the corresponding clickthrough data. Shen et al. [39] proposed context-sensitive language models based on short-term search history and evaluated their models using TREC collections. Following their observations, a number of studies have focused on using short-term search history, especially their clickthrough data [34, 43, 45, 46]. Long-term search history refers to the historical information of a user's queries and is often used for personalization in web search [5, 12, 36]. While both short- and long-term search histories have been shown to be effective in identifying query intent and understanding users' information needs in web search, these types of query context cannot be used for cold-start users and/or queries with no session information (e.g., the first query of each session).

User-specific features are the other types of information that have been considered as contextual features. For instance, Kharitonov and Serdyukov [30] exploited demographic information of users (e.g., gender and age) for re-ranking documents in web search. Furthermore, Bennett et al. [4] estimated the location preference of a web page and used it to improve web search. The influence of many situational features and especially their joint effect on the ranking performance was relatively unstudied.

In addition to ranking in web search, there are several other applications that successfully make use of query context, such as query auto-completion [41], query suggestion [8], query classification [7], and query modification [12]. To the

best of our knowledge, contextual information has not been used and evaluated in personal search.

2.3 Deep Learning for IR

Neural network (and in particular, deep learning) approaches have shown impressive performance in many computer vision, natural language processing, and speech recognition tasks [33]. Recently, people started to study these approaches in various IR applications. For instance, distributional representation of words [37], also known as word embedding, has been employed to improve the performance of several IR tasks. Zamani and Croft [48] proposed a set of query expansion and pseudo-relevance feedback methods based on the semantic similarity of terms calculated via word embedding vectors. Diaz et al. [13] recently proposed to train word embedding vectors using the top retrieved documents of each query for query expansion. Word embedding has been also employed in other IR tasks, such as query classification [49], document classification [32], etc.

In addition to these approaches which use word embedding vectors as the input of their models, a number of studies designed and trained (deep) neural networks for a specific IR task, e.g., ad-hoc retrieval [19], question answering [11, 47], click models [6], etc. For instance, Borisov et al. [6] proposed a neural click model as an alternative to probabilistic graphical models. Huang et al. [23] proposed DSSM, a semantic matching model for web search based on click data. They developed a feed forward neural network with a word hashing phase to predict the click probability given a query string and a document title. C-DSSM [40] is another semantic matching model which is based on convolutional neural networks. Besides these representation-focused models, some interaction-focused matching models were also proposed. For example, DeepMatch [35] maps each text to a sequence of terms and trains a feed forward network for computing the matching score. Recently, Guo et al. [19] proposed a deep relevance matching model for ad-hoc retrieval based on feed forward networks. A number of these neural network-based matching techniques have only been demonstrated to be effective for a set of NLP tasks [19, 22].

In this paper, in order to model situational context for a given query, we develop and evaluate a deep neural network architecture. We also extend the architecture to use a wide & deep model [10].

3. MOTIVATION

In search engines, usually a search request has more properties than the actual content of the query. A number of these properties can be useful for improving search quality. For instance, the location of user when the query is submitted could be used for improving retrieval performance. As a more concrete example, for query "amazon", the user's information need can be the Amazon forest, or the Amazon retail website. Intuitively, a document related to the Amazon forest is more likely to be relevant to this query when the location of user is Brazil compared to US. Therefore, *location* is a property of query that can be potentially used for improving search quality.

Time of the search request is another query property which can be leveraged for ranking. Consider a short and ambiguous query like "reservation". If it is issued during working hours, it could be relevant to actions like *reserving network resources* or *booking a conference room*. If it is issued at night or over a weekend, "reservation" is more likely to be relevant to, e.g., an OpenTable restaurant reservation.

Device of the user is an additional query property that can potentially be used for improving the retrieval performance. Previous studies on web search [2, 20] showed that length and type of the queries came from mobile devices differ from those came from desktop computers. Thus the optimal ranking functions for different devices can be different.

The aforementioned intuitions have motivated us to study various properties of the current search request that are independent of the actual query content. We call these properties *situational context*. These situational features are at query-level and for a given query, they are the same for all candidate documents. The exact definition of the situational features that we used in our experiments are presented in Section 5.1. It should be noted that, since we, in this paper, are interested in studying the affect of situational features, we do not consider user's search histories, although, they have shown to be effective in web search [1, 24, 50]. To the best of our knowledge, there is no prior work on studying query context for the search scenarios where the click data is highly sparse, such as in personal search.[1]

4. METHODOLOGY

Our goal is to design a general methodology that can leverage a variety of situational contexts for ranking. Note that situational contextual features can be highly sparse. For instance, there are hundreds to thousands of different locations, but a search request is associated with only a single one. In addition, since there are multiple possible situational context types (time of day, day of the week, location, device, etc.), using all of their possible combinations may lead to a combinatorial explosion in the number of features.

With this in mind, deep neural networks have shown an impressive performance in various speech recognition and natural language processing tasks where the input data is extremely sparse [33]. Therefore, we explore neural network based approaches in this paper. Our models take *query content*, *situational context*, and *document content* as inputs, and output a score as the prediction of how relevant a document is for a given query.

4.1 Problem Formulation

Formally writing, let $\mathcal{Q} = \{Q_1, Q_2, \cdots, Q_N\}$ be a query set, where $Q_i = \{< \mathbf{q}_i, \mathbf{c}_i, \mathbf{d}_{i1}, l_{i1} >, \cdots, < \mathbf{q}_i, \mathbf{c}_i, \mathbf{d}_{ik}, l_{ik} >\}$ corresponds to the i^{th} query in the query set and contains the query features $\mathbf{q}_i$, contextual features $\mathbf{c}_i$, document features $\mathbf{d}_{ik}$ and the labels l_{ik} for each document. Following previous work [25, 44], we use clickthrough data as labels in this paper and thus l_{ik} is equal to 1 if $\mathbf{d}_{ik}$ is clicked for the query $\mathbf{q}_i$ and 0 otherwise.

Using $\mathcal{Q}$ or its subset as training data, our objective is to predict the label using all the query, contextual, and document features. We use the cross-entropy loss function as our training objective:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{k} l_{ij} \log p_{ij} + (1 - l_{ij}) \log (1 - p_{ij}), \quad (1)$$

[1]The specific characteristics of personal search and its main differences from web search are described in Section 2.1.

where p_{ij} denotes the predicted click probability (i.e., the output of the model) for the j^{th} document given the i^{th} query. In order to avoid over-fitting on the training data, we consider l_2 regularization term in our objective function.

A neural network model trained using our objective function is a *pointwise* ranking model. The output of such a model can be directly used to rank candidate documents of a given query. However, we are also interested in using it as a signal in a learning-to-rank framework, alongside many other ranking signals. For this reason, we limit ourself to a pointwise neural network model, but the described methods are general enough to extend to pairwise or listwise loss functions.

The focus of this paper is how to effectively leverage the contextual features $\mathbf{c}_i$ to improve personal search quality. In the following, we first start with a model that solely considers query and document features, i.e., purely semantic matching model (Section 4.2). We further extend this semantic matching model to incorporate situational context. To this end, we propose two network architectures, a deep one (Section 4.3) and a wide & deep one (Section 4.4).

4.2 Semantic Matching Model

According to [19, 23, 35], the problem of computing a relevance score between a query and a document (or two pieces of text, in general) can be cast to a text matching problem. In other words, the relevance score of a given query $\mathbf{q}$ and a document $\mathbf{d}$ can be calculated as:

$$\text{score}(\mathbf{q}, \mathbf{d}) = f(\Phi(\mathbf{q}), \Phi(\mathbf{d})) \tag{2}$$

where Φ denotes a function that maps a text to a vector representation and f denotes a matching function that computes the similarity of two computed representations. Existing semantic matching models provide different implementations of the functions Φ and f. For instance, Huang et al. [23] considered a feed forward neural network as Φ and the cosine similarity function as f.

In this paper, we consider a simple neural network-based semantic matching model. This model and its similar variations have been shown to be effective in various tasks, including ad-hoc retrieval [19] and web search [23]. The network topology of our semantic matching model is presented in Figure 1.

The input of this model is a query and a candidate document. In our model, queries and documents are represented as a list of n-grams (see Section 5.1 for more details). As shown in Figure 1, we first employ embedding layers on top of the input query and document to compute a dense representation for each of them. Note that the embedding layers differ from pre-trained word embedding vectors, which have been previously used in [13, 29, 48], in that our embeddings are directly learned from the click training data.

The goal of employing these embedding layers is to obtain low-dimensional (compared to the vocabulary size) dense vector representations for the input data (n-grams, in this case), in which close vectors demonstrate similar input. We can then use a stack of non-linear hidden layers to learn high-level abstractions of the input data. These hidden layers are fully-connected and the i^{th} node of each layer is calculated as:

$$h_i = \sigma(\mathbf{w}_i^T \cdot \mathbf{x} + b_i) \tag{3}$$

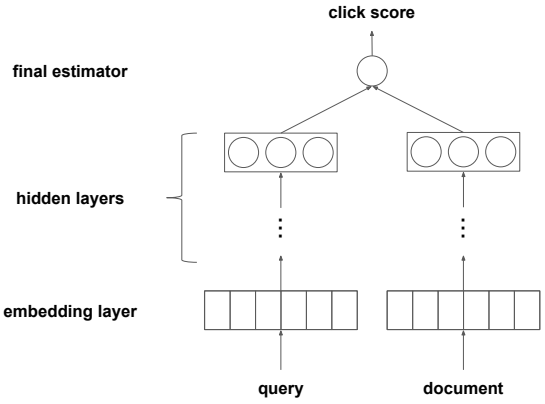


Figure 1: The network architecture of our semantic matching model.

where $\mathbf{w}_i$ and b_i denote the weight matrix and the bias value, respectively. $\mathbf{x}$ is the output vector from the previous layer and σ is the activation function. The rectifier linear unit (ReLU) activation function, which is one of the most common activation functions in the literature of deep learning [33], is used in our method:

$$\sigma(t) = \max(0, t).$$

At the output layer, we use a fully-connected single neuron with a sigmoid activation function

$$\delta(t) = \frac{1}{1 + e^{-t}}$$

to compute the matching score between the given query and document abstract representations. The score represents the click probability on the document given the query. This model is a representation-focused model for ranking. We refer to this semantic matching model as *DNN*, in the remainder of this paper.

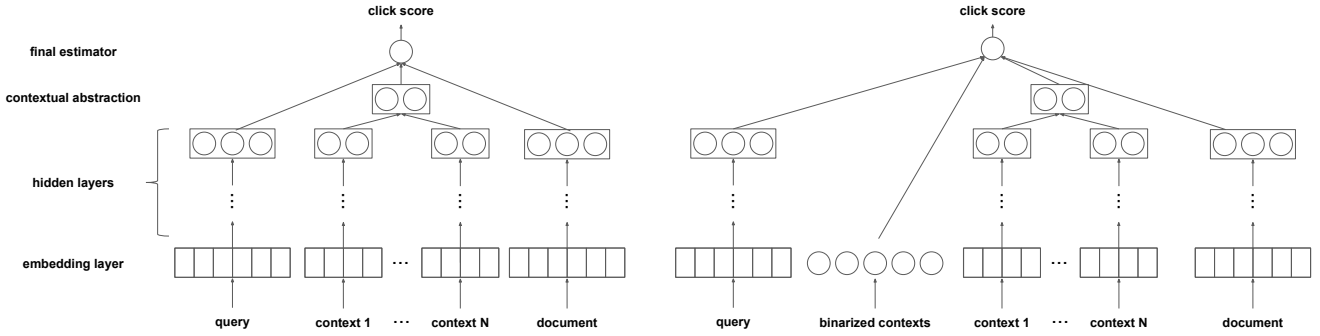
4.3 Context-Aware Deep Network Model

In order to incorporate contextual features into our semantic matching model, as shown in Figure 2a, we first use an embedding layer on top of each contextual feature. The reason for using embedding layers is to have close vector representations for the feature values that are similar to each other. For example, the representations of weekends should be intuitively close to each other and far from the weekday vectors. We then use a number of hidden layers on top of the embedding layers to learn deeper abstractions from these features.

To model the combination of all the contextual features, we use a non-linear hidden layer (i.e., the "contextual abstraction" layer in Figure 2a) which gets the concatenation of context representations as input. In other words, the i^{th} node of the combination abstraction layer is computed as:

$$c_i^D = \sigma(\sum_{k=1}^{n} \mathbf{w}_{ik}^T \cdot \hat{\mathbf{c}}_k + b_i) \tag{4}$$

where $\hat{\mathbf{c}}_k$ is the output vector of the previous layer corresponding to the k^{th} contextual feature and n represents the



(a) Context-DNN

(b) Context-WDNN

Figure 2: Network architectures of the Context-DNN and Context-WDNN models.

total number of contextual features. The model should learn the $\mathbf{w}_{ik}$ and b_i parameters. We again use ReLU as activation function in this layer and use c_i^D to denote the output (where superscript D stands for *deep* neural network).

The reason for using this hidden layer is that we can learn a meaningful representation from a combination of contextual features. For instance, weekends and holidays vary by country, and thus a combination of both location and time representations can lead to learning a more accurate contextual representation, which is used in the final estimation neuron. In our experiments, we found that this combination hidden layer is critical for ranking metric optimization (see Section 5.2). The final click probability is calculated based on the given query, document, and contextual features, similarly to Section 4.2. We refer to this matching model that takes both semantics and deep situational context representation into account as *Context-DNN*, in the remainder of this paper.

4.4 Context-Aware Wide and Deep Network Model

Although deep neural networks have been shown to be useful in learning abstract representations for a large number of features, some information might be over-generalized due to these high-level abstractions. Cheng et al. [10] recently proposed to use sparse features alongside the deep representation of features in order to model memorization. They showed that by using this approach, the performance of the recommender system used in Google Play (a commercial mobile app store) can be improved.

Inspired by their work on recommender systems, we propose a wide & deep neural network model for taking contextual features into account. As shown in Figure 2b, in addition to the deep network proposed in Section 4.3, we consider raw contextual features in a binarized format, which are extremely sparse. This can help us to model memorization approaches that are simple yet effective, when large amounts of training data are available.

Formally, we calculate the final click probability as follows:

$$\delta(\mathbf{w}_q^T \cdot \hat{\mathbf{q}} + \mathbf{w}_c^T \cdot \mathbf{c} + \mathbf{w}_{c^D}^T \cdot \hat{\mathbf{c}}^D + \mathbf{w}_d^T \cdot \hat{\mathbf{d}} + b) \qquad (5)$$

where $\hat{\mathbf{q}}$, $\hat{\mathbf{d}}$, and $\hat{\mathbf{c}}^D$ represent the final representation of query, document, and context between the hidden and output layers. $\mathbf{c}$ represents the binarized context features. δ is a sigmoid activation function. Parameters $\mathbf{w}$ and b are learned during the training phase. We refer to this matching model that takes both semantics, deep context representations and binarized sparse context features into account as *Context-WDNN*, in the remainder of this paper.

5. EVALUATION

We start this section by describing our data sources and the situational context features used in our experiments. We then evaluate each of the proposed models and investigate the impact of the situational context. We finally use the output of our models as a signal alongside other ranking signals in our final ranking phase, and explore how much improvement can be achieved by considering situational context in the final ranking model.

5.1 Data

We evaluate the proposed context-aware ranking models using the data gathered from the logs of one the world's largest personal search engines. The data is collected from email search logs (click data) of real users. In order to preserve the privacy of users, the data is anonymized and following the k-anonymity approach [42], our model can only access the query and document n-grams that are frequent enough in the whole collection. Since the inputs of our model are always sufficiently frequent, it gives us the possibility of learning meaningful dense representations for them.

The dataset contains approximately 20 million queries. For each query, there are four candidate documents available and the purpose is to rank these four documents for the given query. 90% of the queries are randomly selected for training, and the rest are used for evaluation.

In addition to the frequent query and document n-grams that are explained above, we also consider four situational features (two geographical and two temporal features), which are summarized in Table 1. It is notable that the language of each search query was automatically predicted based on the query content and the temporal features are all based on the *UTC* time, since we do not have access to the users' exact timezone (some countries may span multiple timezones). In Table 1, the two geographical features are both highly sparse.

Following previous work on web search [25, 26] and personal search [9, 44], we use click data as a judgment indicator

Table 1: The situational context features used in the experiments.

Type	Feature	Dimension	Description
Geographical	Country	~ 250	The country of the user at the time of the search request.
	Language*	~ 200	The language (including dialect, e.g., *US English*) of the search request.
Temporal	Week day	7	The day of the week of the search request.
	Hour	24	The time (hour) of the day of the search request.

* Since there is generally a strong correlation between user language and location, we categorize language as a geographical feature.

to train and evaluate our models. In our final ranking stage, we consider the position bias problem [26] similar to what is described in [44]. More details are provided in Section 5.3.

5.2 Context-Aware Model Evaluation

In this subsection, we rank candidate documents only based on the output of the proposed context-aware neural network models.

5.2.1 Experimental Setup

We implemented our neural network models using the open-source TensorFlow toolkit[2]. The neural network parameters were tuned over the training set as follows: we swept the learning rate between $\{0.0005, 0.005, 0.05, 0.1\}$ and the output dimension for each hidden layer (i.e., the number of neurons in each layer) between $\{2, 5, 10, 20, 30, 50\}$. The dimension of embedding layers was set to 100. The objective function is described in Section 4.1.

5.2.2 Evaluation Metric

We use mean reciprocal rank (MRR) to evaluate the models. In our personal search engine, only one of the retrieved documents can be clicked by the user, thus MRR is calculated as:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{6}$$

where Q and $rank_i$ respectively denote the evaluation query set and the rank of the clicked document in response to the i^{th} query.

In addition to MRR, we also use *success*@1 and *success*@3 as evaluation metrics. *success*@1 is equivalent to the precision of the top retrieved document, which represents the accuracy of the model in retrieving the correct (clicked) document at the first position. *success*@3 shows the accuracy of the system in *not* retrieving the correct document at the last position. In fact, *success*@1 = 100% indicates the *best* achievable ranking, and as our personal search engine returns at most 4 results, *success*@3 = 0 indicates the *worst* achievable ranking.

Statistically significant differences are determined using the two-tailed paired t-test computed at a 99% confidence level ($\alpha < 0.01$).

5.2.3 Results and Discussion

In this subsection, we empirically answer the following research questions:

- RQ1: Are neural network-based semantic matching models adequately suitable for our task?
- RQ2: Can contextual features be used individually to improve the retrieval performance?
- RQ3: Can we learn more accurate context representation with a combination of contextual information?
- RQ4: Do we lose some information by learning low-dimensional representations based on deep neural networks?

[2] `https://www.tensorflow.org/`

Table 2: Relative improvements achieved by the neural network-based semantic matching models compared to the CTR memorization baseline. Superscripts 0/1 indicate statistically significant improvements over the CTR memorization and the NN methods.

Method	MRR	success@1	success@3
NN	$+0.43\%^{0}$	$+0.90\%^{0}$	$+0.55\%^{0}$
DNN	$\mathbf{+1.76\%}^{01}$	$\mathbf{+3.46\%}^{01}$	$\mathbf{+2.05\%}^{01}$

To address RQ1, we consider the clickthrough rate (CTR) memorization method as a simple yet effective baseline, which was recently proposed in [3]. This method computes CTR for each `<query n-gram, document n-gram>` pair and ranks the candidate documents for a given query based on the CTR aggregation of their query and document n-grams pairs. This memorization baseline can perform well when a sufficient amount of training data is available, as in our case. There are a number of other approaches which use clickthrough data for web ranking (e.g., [17]). The main goal of this paper is to evaluate the effect of incorporating contextual features for personal search. Therefore, we do not consider other web-based approaches which rely on the availability of co-clicked `<query, url>` pairs, and thus cannot work well when the click data is sparse.

Table 2 reports the relative improvements achieved by two neural network-based semantic matching models compared with the CTR memorization baseline. In this table, NN refers to a simple semantic matching model (see Section 4.2) with only one embedding layer. DNN has a similar architecture with an additional non-linear hidden layer. According to Table 2, our semantic matching models significantly outperform the CTR memorization baseline, in terms of all evaluation metrics. Since our main goal is to use contextual information for improving the ranking quality, in the rest of the experiments, we keep DNN as our base semantic matching method and extend it using contextual information.

To address our second research question (RQ2), we extend DNN by adding situational features to the network. The network architecture would be similar to the Context-DNN model (see Figure 2a) but without the "contextual

Table 3: Relative improvements achieved by employing situational features compared to the DNN model. The superscript * indicates statistically significant improvements over the DNN model.

Method	MRR	success@1	success@3
DNN + Country	−0.05%	−0.16%	−0.00%
DNN + Language	−0.06%	−0.19%	−0.03%
DNN + Day	−0.01%	−0.08%	+0.03%
DNN + Hour	−0.02%	−0.05%	+0.04%
DNN + Geographical	+0.05%	+0.16%	+0.01%
DNN + Temporal	+0.07%	+0.22%	+0.03%
DNN + All	+0.15%	+0.35%	+0.06%
Context-DNN	+0.60%*	+0.70%*	+0.10%
Context-WDNN	**+0.69%***	**+0.93%***	**+0.15%**

abstraction" layer. Therefore, the output of each context representation would directly be fed to the final estimator neuron. We first consider only one contextual feature at a time, to see how they affect the results. The relative improvements of these models compared to DNN are reported in Table 3.

According to Table 3, individually, each contextual feature is not useful for ranking. The reason is that each individual feature is usually noisy and contains uncertainty. For example, our time features are based on the UTC timezone for all users, thus cannot give us useful information to improve ranking. As another example, language is estimated solely based on the query string. Since many languages share alphabets and common terms, a lot of queries could be miscategorized. Based on these uncertainty factors, which are common in real-world applications, we found that we cannot use individual contextual features to improve ranking.

Therefore, we propose the third research question (RQ3). The intuition behind it stems from the following question: can we capture meaningful information based on multiple situational context despite the uncertainty factors in each of them? To do so, we trained three models based on geographical, temporal, and all contextual features based on the same network topology (i.e., similar to Context-DNN without the "contextual abstraction" layer.). According to Table 3, we can get minor improvements by considering these contextual features together. Based on this observation, we propose to learn a more accurate situational context representation by adding a hidden layer to combine all context features together (see Figure 2a). The results show that Context-DNN significantly outperform DNN which does not consider situational features, as well as DNN + All which considers the situational features separately.

As shown in Table 3, Context-WDNN model is the best performing model, in terms of all evaluation metrics. The results achieved by Context-WDNN (see Figure 2b for the model's architecture) indicate that we lose some information by considering deep networks to learn low-dimensional abstractions for contextual features, and Context-WDNN, which adds raw contextual features to the Context-DNN model, improves the performance, in terms of all the evaluation metrics. Therefore, our answer to RQ4 is yes.

Learning curve. To have an insight into the amount of training data that we need to train the proposed neural network-based models, we consider our best performing network (i.e., Context-WDNN) and plot the MRR improvements achieved by different amounts of training data compared to 100K queries. The learning curve is plotted in Figure 3. According to this figure, more than four million queries is a reasonable amount of training data that we need to train this model.

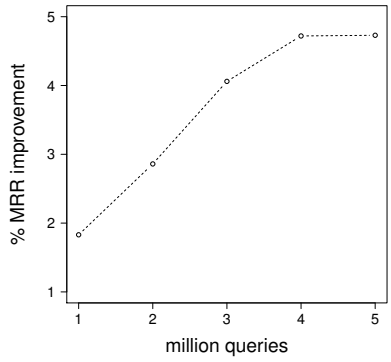


Figure 3: Learning curve for Context-WDNN in terms of relative improvements compared to the same model trained on 100K queries.

5.3 End-to-End Ranking Evaluation

It is a common practice today that production-grade search engines aggregate multiple signals in order to compute the final ranking score (see, e.g., [9]). In this set of experiments, we answer the following research question:

- RQ5: Can we improve the final ranking performance by using the output of our situational context models alongside many other ranking signals?

5.3.1 Experimental Setup

In the experiments, we used an implementation of an adaptive learning to rank framework based on multiple adaptive regression trees (MART) learning algorithm [21] as the final ranking model. This learning to rank framework considers the position bias correction in the training process, as suggested by some prior work [3, 44]

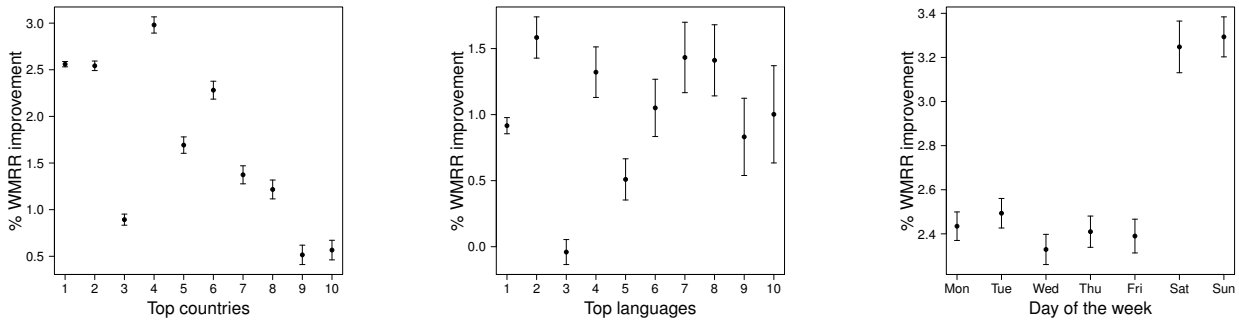
5.3.2 Evaluation Metric

As a target metric, we consider the weighted mean reciprocal rank (WMRR) proposed by Wang et al. [44]. WMRR takes the position bias into account, and is computed as follows:

$$WMRR = \frac{1}{\sum_{i=1}^{|Q|} w_i} \sum_{i=1}^{|Q|} w_i \frac{1}{rank_i}, \tag{7}$$

where w_i denotes the bias correction weight. We set these weights based on a result randomization study, as described in [44]. Effectively, w_i is inversely proportional to the probability of observing the click at the clicked position of a query due to position bias.

Statistically significant differences are determined using the two-tailed paired t-test, as described in the previous set of experiments (see Section 5.2.2).



(a) Top 10 countries (b) Top 10 languages (c) Different days of the week

Figure 4: Context-aware evaluation by country, language, and day of the week (US only) of the LTR + Context-WDNN model. Relative improvements compared to LTR, a model without contextual information, are reported.

Table 4: Relative improvements achieved by adding the output of each proposed model as a signal to our current personal ranking model compared to the performance of the ranking model without these signals (denoted LTR). Superscripts 0/1 indicate statistically significant improvements over LTR/LTR+DNN, respectively.

Method	WMRR
LTR + DNN	$+0.72\%^{0}$
LTR + Context-DNN	$+1.88\%^{01}$
LTR + Context-WDNN	$\mathbf{+1.90\%}^{01}$

5.3.3 Results and Discussion

In this set of experiments, we first consider the introduced learning to rank framework (LTR) with a number of standard email search signals that do not take the situational context into account (readers can refer to Carmel et al. [9] for an overview of such signals). We then add the output of each of the DNN, Context-DNN, and Context-WDNN models as a *signal* to the LTR framework. In fact, in these experiments, we investigate how much value will be added to the final ranking by considering the proposed models. Note that because of the sparsity of some contextual features, e.g., location and language, it is unlikely to improve the ranking performance by directly adding these features to LTR.

The relative improvement achieved by each method compared to LTR with non-contextual features is reported in Table 4. As shown in this table, by adding each of these signals, we can get significant improvements over the LTR baseline. Adding Context-DNN or Context-WDNN as a signal to LTR significantly outperforms the LTR + DNN model, which demonstrates the importance of the situational context in personal search ranking. Employing the output of Context-WDNN achieves the best performance, in terms of weighted MRR (WMRR).

5.3.4 Context-Aware Analysis

To analyze the results achieved by taking the situational context into account, we consider our best ranking model, i.e., LTR + Context-WDNN. We report relative improvements compared to the LTR baseline. In other words, the improvements achieved by a context-aware model compared to a model without any information about the context are reported. We first consider the top 10 countries (in terms of the number of requests from the countries in the test set) and plot the average and the standard deviation of the improvement achieved for each of these countries in Figure 4a. According to this figure, the average improvements per country is generally decreasing from the first to the tenth top county. The standard deviation is also increasing. Intuitively, this demonstrates that the proposed context-aware model has more impact for countries with more search requests, but provides a uniform improvement across countries.

Figure 4b presents the improvements per language for top 10 languages in terms of the number of requests from those languages. According to this figure, the standard deviation of improvements increases when we move from the first to the tenth language. This indicates that the impact on the least frequent languages varies more compared to the more common languages.

The improvements per day of the week are plotted in Figure 4c. In this figure, we only considered the queries that came from the United States, since we do not have access to the true timezone of the user. The observation here is that the improvements achieved on the weekends are similar to each other and are much higher than those obtained during the weekdays. It suggests that a LTR model without context is sub-optimal for weekends and our context-aware ranking models improve the search quality by modeling this context effectively in LTR framework.

The observations in context-aware result analysis demonstrate that the context-aware models can achieve varying improvements for different values of contextual features. This analysis can be further used to deploy a context-aware model for the user segments where they can achieve the highest improvements.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the problem of employing situational context in order to improve the ranking quality in personal search. We considered situational context as the properties of the current search request that are independent from both query content and user search history, such as time and location. As we observed a meaningful relation-

ship between user behavior and situational context features in search logs, we developed two supervised context-aware ranking models based on neural networks that can be trained using clickthrough data.

The first context-aware model considers a deep context representation learned from a combination of all the contextual features. Learning a dense representation based on all the features enables us to model cross-context information. For example, weekends and holidays may vary across countries, which can only be learned by considering time and location features in tandem.

The second model extends our first model based on the following hypothesis: we may lose some information by learning high-level abstraction of contextual features. Therefore, we proposed a wide & deep neural network which considers binarized contextual features in addition to their deep dense abstractions.

We evaluated our models using click data collected from one of the world's largest personal search engines. Our experiments demonstrated significant improvements over several state-of-the-art baselines and indicated the importance of situational context for personal search.

As a future research direction, other types of context, such as short- and long-term search history, or user profiles, can be studied in personal search scenarios. The importance of other situational features, e.g., user device, can also be explored in the future. We also intend to study pairwise and listwise context-aware networks (that extend the pointwise approach presented in this paper). Studying situational features in other search scenarios, such as web search, is another interesting direction that can be considered in the future.

7. 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 those of the authors and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*, pages 19–26, 2006.

[2] R. Baeza-Yates, G. Dupret, and J. Velasco. A study of mobile search queries in japan. In *WWW '07 Workshops*, 2007.

[3] M. Bendersky, X. Wang, D. Metzler, and M. Najork. Learning from user interactions in personal search via attribute parameterization. In *WSDM '17*, pages 791–799, 2017.

[4] P. N. Bennett, F. Radlinski, R. W. White, and E. Yilmaz. Inferring and using location metadata to personalize web search. In *SIGIR '11*, pages 135–144, 2011.

[5] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short- and long-term behavior on search personalization. In *SIGIR '12*, pages 185–194, 2012.

[6] A. Borisov, I. Markov, M. de Rijke, and P. Serdyukov. A neural click model for web search. In *WWW '16*, pages 531–541, 2016.

[7] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-aware query classification. In *SIGIR '09*, pages 3–10, 2009.

[8] 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 *KDD '08*, pages 875–883, 2008.

[9] D. Carmel, G. Halawi, L. Lewin-Eytan, Y. Maarek, and A. Raviv. Rank by time or by relevance?: Revisiting email search. In *CIKM '15*, pages 283–292, 2015.

[10] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah. Wide & deep learning for recommender systems. *CoRR*, abs/1606.07792, 2016.

[11] D. Cohen and W. B. Croft. End to end long short term memory networks for non-factoid question answering. In *ICTIR '16*, pages 143–146, 2016.

[12] W. B. Croft and X. Wei. Context-based topic models for query modification. Technical report, Center for Intelligent Information Retrieval, University of Massachusetts, 2005.

[13] F. Diaz, B. Mitra, and N. Craswell. Query expansion with locally-trained word embeddings. In *ACL '16*, pages 367–377, 2016.

[14] A. Drutsa, G. Gusev, and P. Serdyukov. Engagement periodicity in search engine usage: Analysis and its application to search quality evaluation. In *WSDM '15*, pages 27–36, 2015.

[15] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff i've seen: A system for personal information retrieval and re-use. In *SIGIR '03*, pages 72–79, 2003.

[16] D. Elsweiler, M. Harvey, and M. Hacker. Understanding re-finding behavior in naturalistic email interaction logs. In *SIGIR '11*, pages 35–44, 2011.

[17] J. Gao, K. Toutanova, and W.-t. Yih. Clickthrough-based latent semantic models for web search. In *SIGIR '11*, pages 675–684, 2011.

[18] Y. Goldberg. A primer on neural network models for natural language processing. *CoRR*, abs/1510.00726, 2015.

[19] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. A deep relevance matching model for ad-hoc retrieval. In *CIKM '16*, 2016.

[20] I. Guy. Searching by talking: Analysis of voice queries on mobile web search. In *SIGIR '16*, pages 35–44, 2016.

[21] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[22] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS '14*, pages 2042–2050. 2014.

[23] 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 *CIKM '13*, pages 2333–2338, 2013.

[24] D. Jiang, J. Pei, and H. Li. Mining search and browse logs for web search: A survey. *ACM Trans. Intell. Syst. Technol.*, 4(4):57:1–57:37, 2013.

[25] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, pages 133–142, 2002.

[26] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05*, pages 154–161, 2005.

[27] M. Kamvar and S. Baluja. The role of context in query input: Using contextual signals to complete queries on mobile devices. In *MobileHCI '07*, pages 405–412, 2007.

[28] M. Kamvar, M. Kellar, R. Patel, and Y. Xu. Computers and iphones and mobile phones, oh my!: A logs-based comparison of search users on different devices. In *WWW '09*, pages 801–810, 2009.

[29] T. Kenter and M. de Rijke. Short text similarity with word embeddings. In *CIKM '15*, pages 1411–1420, 2015.

[30] E. Kharitonov and P. Serdyukov. Demographic context in web search re-ranking. In *CIKM '12*, pages 2555–2558, 2012.

[31] W. Kong, R. Li, J. Luo, A. Zhang, Y. Chang, and J. Allan. Predicting search intent based on pre-search context. In *SIGIR '15*, pages 503–512, 2015.

[32] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *ICML '15*, pages 957–966, 2015.

[33] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[34] Z. Liao, D. Jiang, J. Pei, Y. Huang, E. Chen, H. Cao, and H. Li. A vlhmm approach to context-aware search. *ACM Trans. Web*, 7(4):22:1–22:38, 2013.

[35] Z. Lu and H. Li. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems 26*, NIPS '13, pages 1367–1375. 2013.

[36] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *WSDM '11*, pages 25–34, 2011.

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS '13*, pages 3111–3119. 2013.

[38] P. Ogilvie and J. Callan. Experiments with language models for known-item finding of e-mail messages. In *TREC '05*, 2005.

[39] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR '05*, pages 43–50, 2005.

[40] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. In *WWW '14 Companion*, pages 373–374, 2014.

[41] M. Shokouhi. Learning to personalize query auto-completion. In *SIGIR '13*, pages 103–112, 2013.

[42] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[43] Y. Ustinovskiy and P. Serdyukov. Personalization of web-search using short-term browsing context. In *CIKM '13*, pages 1979–1988, 2013.

[44] X. Wang, M. Bendersky, D. Metzler, and M. Najork. Learning to rank with selection bias in personal search. In *SIGIR '16*, pages 115–124, 2016.

[45] R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. In *CIKM '10*, pages 1009–1018, 2010.

[46] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In *SIGIR '10*, pages 451–458, 2010.

[47] L. Yang, Q. Ai, J. Guo, and W. B. Croft. aNMM: Ranking short answer texts with attention-based neural matching model. In *CIKM '16*, pages 287–296, 2016.

[48] H. Zamani and W. B. Croft. Embedding-based query language models. In *ICTIR '16*, pages 147–156, 2016.

[49] H. Zamani and W. B. Croft. Estimating embedding vectors for queries. In *ICTIR '16*, pages 123–132, 2016.

[50] G. Zhu and G. Mishne. Mining rich session context to improve web search. In *KDD '09*, pages 1037–1046, 2009.