Search Personalization with Embeddings

Thanh Vu[1], Dat Quoc Nguyen[2](✉), Mark Johnson[2], Dawei Song[1,3], and Alistair Willis[1]

[1] The Open University, Milton Keynes, UK
{thanh.vu,dawei.song,alistair.willis}@open.ac.uk
[2] Department of Computing, Macquarie University, Sydney, Australia
dat.nguyen@students.mq.edu.au, mark.johnson@mq.edu.au
[3] Tianjin University, Tianjin, People's Republic of China

Abstract. Recent research has shown that the performance of search personalization depends on the richness of user profiles which normally represent the user's topical interests. In this paper, we propose a new embedding approach to learning user profiles, where users are embedded on a topical interest space. We then directly utilize the user profiles for search personalization. Experiments on query logs from a major commercial web search engine demonstrate that our embedding approach improves the performance of the search engine and also achieves better search performance than other strong baselines.

1 Introduction

Users' personal data, such as a user's historic interaction with the search engine (e.g., submitted queries, clicked documents), have been shown useful to personalize search results to the users' information need [1,15]. Crucial to effective search personalization is the construction of user profiles to represent individual users' interests [1,3,6,7,12]. A common approach is to use main topics discussed in the user's clicked documents [1,6,12,15], which can be obtained by using a human generated ontology [1,15] or using a topic modeling technique [6,12].

However, using the user profile to directly personalize a search has been not very successful with a *minor* improvement [6,12] or even *deteriorate* the search performance [5]. The reason is that each user profile is normally built using only the user's relevant documents (e.g., clicked documents), ignoring user interest-dependent information related to input queries. Alternatively, the user profile is utilized as a feature of a multi-feature learning-to-rank (L2R) framework [1, 13–15]. In this case, apart from the user profile, dozens of other features has been proposed as the input of an L2R algorithm [1]. Despite being successful in improving search quality, the contribution of the user profile is not very clear.

To handle these problems, in this paper, we propose a new *embedding* approach to constructing a user profile, using both the user's input queries and relevant documents. We represent each user profile using two projection matrices and a user embedding. The two projection matrices is to identify the user interest-dependent aspects of input queries and relevant documents while the

© Springer International Publishing AG 2017
J.M. Jose et al. (Eds.): ECIR 2017, LNCS 10193, pp. 598–604, 2017.
DOI: 10.1007/978-3-319-56608-5_54

user embedding is to capture the relationship between the queries and documents in this user interest-dependent subspace. We then *directly* utilize the user profile to re-rank the search results returned by a commercial search engine. Experiments on the query logs of a commercial web search engine demonstrate that modeling user profile with embeddings helps to significantly improve the performance of the search engine and also achieve better results than other comparative baselines [1,11,14] do.

2 Our Approach

We start with our new embedding approach to building user profiles in Sect. 2.1, using pre-learned document embeddings and query embeddings. We then detail the processes of using an unsupervised topic model (i.e., Latent Dirichlet Allocation (LDA) [2]) to learn document embeddings and query embeddings in Sects. 2.2 and 2.3, respectively. We finally use the user profiles to personalize the search results returned by a commercial search engine in Sect. 2.4.

2.1 Building User Profiles with Embeddings

Let $\mathcal{Q}$ denote the set of queries, $\mathcal{U}$ be the set of users, and $\mathcal{D}$ be the set of documents. Let (q, u, d) represent a triple $(\mathsf{query}, \mathsf{user}, \mathsf{document})$. The query $q \in \mathcal{Q}$, user $u \in \mathcal{U}$ and document $d \in \mathcal{D}$ are represented by vector embeddings $\boldsymbol{v}_q$, $\boldsymbol{v}_u$ and $\boldsymbol{v}_d \in \mathbb{R}^k$, respectively.

Our goal is to select a *score function* f such that the implausibility value $f(q, u, d)$ of a correct triple (q, u, d) (i.e. d is a relevant document of u given q) is *smaller* than the implausibility value $f(q', u', d')$ of an incorrect triple (q', u', d') (i.e. d' is not a relevant document of u' given q'). Inspired by embedding models of entities and relationships in knowledge bases [9,10], the score function f is defined as follows:

$$f(q, u, d) = \|\mathbf{W}_{u,1}\boldsymbol{v}_q + \boldsymbol{v}_u - \mathbf{W}_{u,2}\boldsymbol{v}_d\|_{\ell_{1/2}} \tag{1}$$

here we represent the profile for the user u by two matrices $\mathbf{W}_{u,1}$ and $\mathbf{W}_{u,2} \in \mathbb{R}^{k\times k}$ and a vector embedding $\boldsymbol{v}_u$, which represents the user's topical interests. Specifically, we use the interest-specific matrices $\mathbf{W}_{u,1}$ and $\mathbf{W}_{u,2}$ to identify the interest-dependent aspects of both query q and document d, and use vector $\boldsymbol{v}_u$ to describe the relationship between q and d in this interest-dependent subspace.

In this paper, $\boldsymbol{v}_d$ and $\boldsymbol{v}_q$ are pre-determined by employing the LDA topic model [2], which are detailed in next Sects. 2.2 and 2.3. Our model parameters are only the user embeddings $\boldsymbol{v}_u$ and matrices $\mathbf{W}_{u,1}$ and $\mathbf{W}_{u,2}$. To learn these user embeddings and matrices, we minimize the margin-based objective function:

$$\mathcal{L} = \sum_{\substack{(q,u,d)\in\mathcal{G} \\ (q',u,d')\in\mathcal{G}'_{(q,u,d)}}} \max\left(0, \gamma + f(q, u, d) - f(q', u, d')\right) \tag{2}$$

where γ is the margin hyper-parameter, $\mathcal{G}$ is the training set that contains only correct triples, and $\mathcal{G}'_{(q,u,d)}$ is the set of incorrect triples generated by corrupting the correct triple (q, u, d) (i.e. replacing the relevant document/query d/q in (q, u, d) by irrelevant documents/queries d'/q'). We use Stochastic Gradient Descent (SGD) to minimize $\mathcal{L}$, and impose the following constraints during training: $\|\boldsymbol{v}_u\|_2 \leqslant 1$, $\|\mathbf{W}_{u,1}\boldsymbol{v}_q\|_2 \leqslant 1$ and $\|\mathbf{W}_{u,2}\boldsymbol{v}_d\|_2 \leqslant 1$. First, we initialize user matrices as identity matrices and then fix them to only learn the randomly initialized user embeddings. Then in the next step, we fine-tune the user embeddings and user matrices together. In all experiments shown in Sect. 3, we train for 200 epochs during each two optimization step.

2.2 Using LDA to Learn Document Embeddings

In this paper, we model document embeddings by using topics extracted from relevant documents. We use LDA [2] to *automatically* learn k topics from the relevant document collection. After training an LDA model to calculate the probability distribution over topics for each document, we use the topic proportion vector of each document as its document embedding. Specifically, the zth element $(z = 1, 2, ..., k)$ of the vector embedding for document d is: $\boldsymbol{v}_{d,z} = \mathrm{P}(z \mid d)$ where $\mathrm{P}(z \mid d)$ is the probability of the topic z given the document d.

2.3 Modeling Search Queries with Embeddings

We also represent each query as a probability distribution $\boldsymbol{v}_q$ over topics, i.e. the zth element of the vector embedding for query q is defined as: $\boldsymbol{v}_{q,z} = \mathrm{P}(z \mid q)$ where $\mathrm{P}(z \mid q)$ is the probability of the topic z given the query q. Following [1,14], we define $\mathrm{P}(z \mid q)$ as a mixture of LDA topic probabilities of z given documents related to q. Let $\mathcal{D}_q = \{d_1, d_2, ..., d_n\}$ be the set of top n ranked documents returned for a query q (in the experiments we select $n = 10$). We define $\mathrm{P}(z \mid q)$ as follows:

$$\mathrm{P}(z \mid q) = \sum\nolimits_{i=1}^{n} \lambda_i \mathrm{P}(z \mid d_i) \tag{3}$$

where $\lambda_i = \dfrac{\delta^{i-1}}{\sum_{j=1}^{n} \delta^{j-1}}$ is the exponential decay function of i which is the rank of d_i in D_q. And δ is the decay hyper-parameter $(0 < \delta < 1)$. The decay function is to specify the fact that a higher ranked document is more relevant to user in term of the lexical matching (i.e. we set the larger mixture weights to higher ranked documents).

2.4 Personalizing Search Results

We utilize the user profiles (i.e., the learned user embeddings and matrices) to re-rank the original list of documents produced by a commercial search engine as follows: (1) We download the top n ranked documents given the input query

q. We denote a downloaded document as d. (2) For each document d we apply the trained LDA model to infer the topic distribution $\boldsymbol{v}_d$. We then model the query q as a topic distribution $\boldsymbol{v}_q$ as in Sect. 2.3. (3) For each triple (q, u, d), we calculate the implausibility value $f(q, u, d)$ as defined in Eq. 1. We then sort the values in the ascending order to achieve a new ranked list.

3 Experimental Methodology

Dataset: We evaluate our new approach using the search results returned by a commercial search engine. We use a dataset of query logs of 106 anonymous users in 15 days from 01 July 2012 to 15 July 2012. A log entity contains a user identifier, a query, top-10 URLs ranked by the search engine, and clicked URLs along with the user's dwell time. We also download the content documents of these URLs for training LDA [2] to learn document and query embeddings (Sects. 2.2 and 2.3).

Bennett *et al.* [1] indicate that short-term (i.e. session) profiles achieved better search performance than the longer-term profiles. Short-term profiles are usually constructed using the user's search interactions within a search session and used to personalize the search within the session [1]. To identify a search session, we use 30 min of user inactivity to demarcate the session boundary. In our experiments, we build short-term profiles and utilize the profiles to personalize the returned results. Specifically, we uniformly separate the last log entries within search sessions into a *test set* and a *validation set*. The remainder of log entities within search sessions are used for *training* (e.g. to learn user embeddings and matrices in our approach).

Evaluation Methodology: We use the SAT criteria detailed in [4] to identify whether a clicked URL is relevant from the query logs (i.e., a SAT click). That is either a click with a dwell time of at least 30 s or the last result click in a search session. We assign a positive (relevant) label to a returned URL if it is a SAT click. The remainder of the top-10 URLs is assigned negative (irrelevant) labels. We use the rank positions of the positive labeled URLs as the ground truth to evaluate the search performance before and after re-ranking. We also apply a simple pre-processing on these datasets as follows. At first, we remove the queries whose positive label set is empty from the dataset. After that, we discard the domain-related queries (e.g. Facebook, Youtube). To this end, the training set consists of 5,658 correct triples. The test and validation sets contain 1,210 and 1,184 correct triples, respectively. Table 1 presents the dataset statistics after pre-processing.

Table 1. Basic statistics of the dataset after pre-processing

#days	#users	#distinct queries	#SAT clicks	#sessions	#distinct documents
15	106	6,632	8,052	2,394	33,591

Evaluation Metrics: We use two standard evaluation metrics in document ranking [1,8]: *mean reciprocal rank* (**MRR**) and *precision* (**P@1**). For each metric, the higher value indicates the better ranking performance.

Baselines: We employ three comparative baselines with the same experimental setup: (1) **SE**: The original rank from the search engine (2) **CI**: We promote returned documents previously clicked by the user. This baseline is similar to the personalized navigation method in Teevan *et al.* [11]. (3) **SP**: The search personalization method using the short-term profile [1,14]. These are very comparative baselines given that they start with the ranking provided by the major search engine and add other signals (e.g., clicked documents) to get a better ranking performance [1,11].

Hyper-Parameter Tuning: We perform a grid search to select optimal hyper-parameters on the validation set. We train the LDA model[1] using only the relevant documents (i.e., SAT clicks) extracted from the query logs, with the number of topics (i.e. the number of vector dimensions) $k \in \{50, 100, 200\}$. We then apply the trained LDA model to infer document embeddings and query embeddings for all documents and queries. We then choose either the ℓ_1 or ℓ_2 norm in the score function f, and select SGD learning rate $\eta \in \{0.001, 0.005, 0.01\}$, the margin hyper-parameter $\gamma \in \{1, 3, 5\}$ and the decay hyper-parameter $\delta \in \{0.7, 0.8, 0.9\}$. The highest MRR on the validation set is obtained when using $k = 200$, ℓ_1 in f, $\eta = 0.005$, $\gamma = 5$, and $\delta = 0.8$.

4 Experimental Results

Table 2 shows the performances of the baselines and our proposed method. Using the previously clicked documents *CI* helps to significantly improve the search performance ($p < 0.05$ with the *paired t-test*) with the relative improvements of about $7^{+}\%$ in both MRR and P@1 metrics. With the use of short-term profiles as a feature of a learning-to-rank framework, *SP* [1,14] improves the MRR score over the original rank significantly ($p < 0.01$) and achieves a better performance than *CI*'s.

Table 2. Overall performances of the methods in the test set. **Our method**$_{-W}$ denotes the simplified version of our method. The subscripts denote the relative improvement over the baseline *SE*.

Metric	SE	CI [11]	SP [1,14]	Our method	Our method$_{-W}$
MRR	0.559	$0.597_{+6.9\%}$	$0.631_{+12.9\%}$	$\mathbf{0.656}_{+17.3\%}$	$0.645_{+15.4\%}$
P@1	0.385	$0.416_{+8.1\%}$	$0.452_{+17.4\%}$	$\mathbf{0.501}_{+30.3\%}$	$0.481_{+24.9\%}$

By directly learning user profiles and applying them to re-rank the search results, our embedding approach achieves the highest performance of search

[1] We use the LDA implementation in Mallet toolkit: http://mallet.cs.umass.edu/.

personalization. Specifically, our MRR score is significantly ($p < 0.05$) higher than that of *SP* (with the relative improvement of 4% over SP). Likewise, the P@1 score obtained by our approach is significantly higher than that of the baseline *SP* ($p < 0.01$) with the relative improvement of 11%.

In Table 2, we also present the performances of a simplified version of our embedding approach where we fix the user matrices as identity matrices and then only learn the user embeddings. Table 2 shows that our simplified version achieves second highest scores compared to all others.[2] Specifically, our simplified version obtains significantly higher P@1 score (with $p < 0.05$) than *SP*.

5 Conclusions

In this paper, we propose a new embedding approach to building user profiles. We model each user profile using a user embedding together with two user matrices. The user embedding and matrices are then learned using LDA-based vector embeddings of the user's relevant documents and submitted queries. Applying it to web search, we use the profile to re-rank search results returned by a commercial web search engine. Our experimental results show that the proposed method can stably and significantly improve the ranking quality.

Acknowledgments. The first two authors contributed equally to this work. Dat Quoc Nguyen is supported by an Australian Government Research Training Program Scholarship and a NICTA NRPA Top-Up Scholarship.

References

1. Bennett, P.N., White, R.W., Chu, W., Dumais, S.T., Bailey, P., Borisyuk, F., Cui, X.: Modeling the impact of short-and long-term behavior on search personalization. In: SIGIR (2012)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
3. Cheng, Z., Jialie, S., Hoi, S.C.: On effective personalized music retrieval by exploring online user behaviors. In: SIGIR (2016)
4. Fox, S., Karnawat, K., Mydland, M., Dumais, S., White, T.: Evaluating implicit measures to improve web search. ACM Trans. Inf. Syst. **23**(2), 147–168 (2005)
5. Harvey, M., Carman, M.J., Ruthven, I., Crestani, F.: Bayesian latent variable models for collaborative item rating prediction. In: CIKM (2011)
6. Harvey, M., Crestani, F., Carman, M.J.: Building user profiles from topic models for personalised search. In: CIKM (2013)
7. Liu, X.: Modeling users' dynamic preference for personalized recommendation. In: IJCAI (2015)
8. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)

[2] Our approach obtains significantly higher P@1 score ($p < 0.05$) than our simplified version with 4% relative improvement.

9. Nguyen, D.Q., Sirts, K., Qu, L., Johnson, M.: Neighborhood mixture model for knowledge base completion. In: CoNLL (2016)
10. Nguyen, D.Q., Sirts, K., Qu, L., Johnson, M.: STransE: a novel embedding model of entities and relationships in knowledge bases. In: NAACL-HLT (2016)
11. Teevan, J., Liebling, D.J., Ravichandran Geetha, G.: Understanding and predicting personal navigation. In: WSDM (2011)
12. Vu, T., Song, D., Willis, A., Tran, S.N., Li, J.: Improving search personalisation with dynamic group formation. In: SIGIR (2014)
13. Vu, T., Willis, A., Kruschwitz, U., Song, D.: Personalised query suggestion for intranet search with temporal user profiling. In: CHIIR (2017)
14. Vu, T., Willis, A., Tran, S.N., Song, D.: Temporal latent topic user profiles for search personalisation. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 605–616. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16354-3_67
15. White, R.W., Chu, W., Hassan, A., He, X., Song, Y., Wang, H.: Enhancing personalized search by mining and modeling task behavior. In: WWW (2013)