Search Retargeting using Directed Query Embeddings

Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Narayan Bhamidipati
Yahoo Labs, Sunnyvale, CA, USA
{mihajlo, nemanja, vladan, narayanb}@yahoo-inc.com

ABSTRACT

Determining user audience for online ad campaigns is a critical problem to companies competing in online advertising space. One of the most popular strategies is search retargeting, which involves targeting users that issued search queries related to advertiser's core business, commonly specified by advertisers themselves. However, advertisers often fail to include many relevant queries, which results in suboptimal campaigns and negatively impacts revenue for both advertisers and publishers. To address this issue, we use recently proposed neural language models to learn low-dimensional, distributed query embeddings, which can be used to expand query lists with related queries through simple nearest neighbor searches in the embedding space. Experiments on real-world data set strongly suggest benefits of the approach.

1. INTRODUCTION

Web environment provides ad publishers with means to understand user behavior in great detail by analyzing users' search queries, ad clicks, purchases, and other signals. This brings the ability to choose which ad from a set of alternatives to show to users based on their behavioral patterns, called ad targeting [3]. Techniques to match users with ads range from using machine learning models to retargeting with rules. Interestingly, even though targeting using machine learning made a lot of headway in the recent years, retargeting still dominates the industry, as it is intuitive, easy to implement, and works well in practice.

Search retargeting (SRT) is such a technique, that takes a list of keywords and returns a list of users that searched for any keyword from the list. Given their domain knowledge, advertisers have a rough idea of which keywords are suited for their campaigns in terms of maximizing user response. However, oftentimes they fail to include many relevant, non-obvious keywords, leading to lower-coverage campaigns with suboptimal performance, which is an issue commonly observed in practice. To mitigate this problem, publishers expand keyword lists to include related queries, with the aim of enlarging segments without sacrificing campaign quality.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742774.

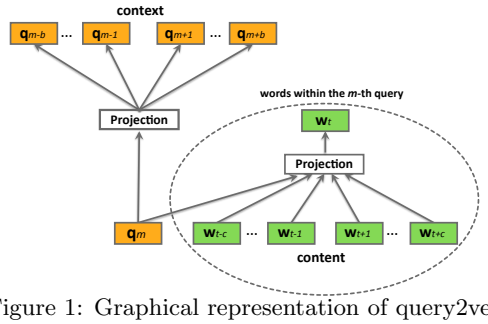

Figure 1: Graphical representation of query2vec

The problem of query expansion is not new and can be found in related applications, such as sponsored search, where a common approach is to remove or add terms to the original query. Another popular method is Query Flow Graph (QFG) [1], shown to obtain good performance in practice.

2. PROPOSED APPROACH

We describe a novel approach to query expansion, motivated by advances in distributed language models in natural language processing (NLP) [4]. In the context of NLP, these models are able to learn word representations in a low-dimensional space using a context in which the words appear, such that semantically related words are close in the embedding space [4]. Extending language models to find representations of queries, as opposed to words, brings unique challenges different from those found in NLP. Contrary to everyday language where words and sentences are clearly defined, in web search there is no "sentence" or "word context". Nevertheless, we argue that such notions can be mapped to web search space, and describe how to employ state-of-the-art language models for query expansion task.

We propose *query2vec* (Fig. 1), a two-level architecture [2] where upper layer models temporal context of query sequences using skip-gram [4] and bottom layer models word sequences within a query using continuous bag-of-words [4]. Given set $\mathcal{S}$ of S search sessions, where, w.l.o.g., session $s = (q_1, \ldots, q_M) \in \mathcal{S}$ is an uninterrupted sequence of M queries (analogous to a sentence in NLP; we use 30-min inactivity as a session boundary), and q_m comprises T words, $q_m = (w_{m1}, \ldots, w_{mT})$, objective is to maximize log-likelihood,

$$\mathcal{L} = \sum_{s \in \mathcal{S}} \Big(\sum_{q_m \in s} \alpha_m \sum_{w_{mt} \in q_m} \log \mathbb{P}(w_{mt} | w_{m,t-c} : w_{m,t+c}, q_m) + \sum_{q_m \in s} \big(\alpha_m \log \mathbb{P}(q_m | w_{m1} : w_{mT}) + \sum_{-b \le i \le b, i \neq 0} \log \mathbb{P}(q_{m+i} | q_m) \big) \Big), \quad (1)$$

Table 1: Yield rate improvement on the conversion task

Algorithm	YR improvement (%)
word2vec	-2.306
QFG	-1.294
query2vec	0.579
ad-query2vec	0.794
directed ad-query2vec	**0.911**

where α weights specify a trade-off between minimization of log-likelihood of query sequences (i.e., context) and of word sequences (i.e., content), and b and c are context widths for queries and words, respectively. We set $\alpha_m = 1/\log(1+K_m)$, where K_m is the m^{th} query frequency, such that tail queries rely more on content and head queries more on context.

Probability $\mathbb{P}(q_{m+i}|q_m)$ of observing a neighboring query based on the current query is defined using the soft-max,

$$\mathbb{P}(q_{m+i}|q_m) = \frac{\exp(\mathbf{v}_{q_m}^\top \mathbf{v}'_{q_{m+i}})}{\sum_{q=1}^{N} \exp(\mathbf{v}_{q_m}^\top \mathbf{v}'_q)}, \quad (2)$$

where $\mathbf{v}_q$ and $\mathbf{v}'_q$ are input and output representations of q. Probability of observing a content word depends on its surrounding words and the query that the word belongs to,

$$\mathbb{P}(w_{mt}|w_{m,t-c}:w_{m,t+c}, q_m) = \frac{\exp(\bar{\mathbf{v}}^\top \mathbf{v}'_{w_{mt}})}{\sum_{w=1}^{W} \exp(\bar{\mathbf{v}}^\top \mathbf{v}'_w)}, \quad (3)$$

where $\mathbf{v}'_{w_{mt}}$ is the output vector of w_{mt}, and $\bar{\mathbf{v}}$ is an average of input vector representations of surrounding words and query q_m. Probability $\mathbb{P}(q_m|w_{m1}:w_{mT})$ is defined similarly.

Learning representations of ad clicks: Search engine logs often include ad click info which can be used as an additional context to improve the quality of query expansions. Thus, to produce more commercially-focused expansions, we propose to also learn vectors for ad clicks in the upper layer, inserting them in the query sequences after queries that occurred prior to ad click (called *ad-query2vec* model). In addition, to capture temporal aspects of search we propose to use directed language model, where as context we only use past queries. The change was made only in the upper layer, allowing us to learn query embeddings capable of predicting future ad clicks (called *directed ad-query2vec* model).

3. EXPERIMENTS

We considered expansions produced by the following methods: 1) query2vec; 2) ad-query2vec; 3) directed ad-query2vec; 4) QFG [1]; and 5) word2vec [4], where we used publicly available word vectors and generated query vectors by summing the vectors of query words (excluding stopwords). More than 45 million query vectors were trained using one of the largest search data set reported so far, comprising 12 billion sessions collected on US website of Yahoo Search. Ad clicks were represented at the level of a category, using an in-house, two-level taxonomy (e.g., "travel/hotels", "retail/home").

Conversion prediction: Given an original set of keywords specified by 47 advertisers, we evaluated how well the keywords target users that eventually bought a product, as compared to the keywords supplemented by keywords produced by the expansion methods (each keyword was expanded with 10 nearest neighbors). Similarly to [3], we performed offline evaluation using historical activity logs. Given user search and purchase data in the month following the training period, we measured success in terms of yield rate, calculated as $\text{YR} = \frac{\#\text{conversions}}{\#\text{qualifications}}$. We used number of queries that resulted in users' qualification in the denominator, and number of purchases that occurred after the qualifying query in the numerator. Table 1 shows average YR improvements over targeting using the original keywords.

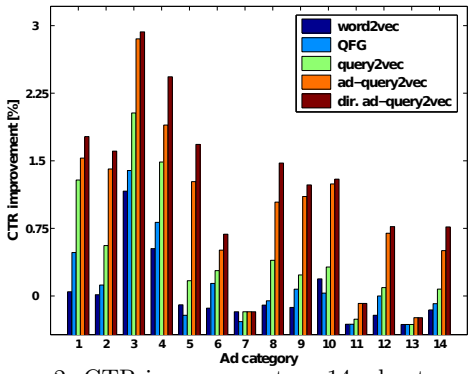

Figure 2: CTR improvement on 14 ad categories

Click prediction: Further analysis was done on 14 top-level interest categories, for which the publisher (i.e., Yahoo) maintains a standard set of keywords to help on-board new advertisers. For these interest categories we used word2vec, query2vec, and QFG to expand the keyword lists with 10 nearest neighbors. In addition, for ad-query2vec and directed ad-query2vec we generated keyword lists from scratch, by finding 2,000 nearest query vectors for ad vectors associated with these categories. Evaluation was done based on the click-through rate (CTR) computed for each ad category separately, $\text{CTR} = \frac{\#\text{clicks}}{\#\text{impressions}}$. In particular, we counted the number of clicks on ads from a certain category that occurred after a query from that category's list, and divided by the total number of shown ads from that category. Figure 2 illustrates improvements over the standard keyword set.

We can see that directed ad-query2vec achieved the best average YR improvement of 0.91% and the best average CTR improvement of 1.12% while increasing user coverage by 13% on average (results not shown), and that proposed techniques outperformed QFG by a large margin. In addition, we see that all query2vec methods, specifically tailored for queries, performed better than word2vec. Moreover, incorporating ad clicks and search direction showed further performance boosts. The results strongly suggest benefits of the proposed approaches for query expansion in SRT[1].

4. REFERENCES

[1] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: Model and applications. In *CIKM*, pages 609–618, 2008.

[2] N. Djuric, H. Wu, V. Radosavljevic, M. Grbovic, and N. Bhamidipati. Hierarchical neural language models for joint representation of streaming documents and their content. In *WWW*, 2015.

[3] M. Grbovic and S. Vucetic. Generating ad targeting rules using sparse principal component analysis with constraints. In *WWW*, pages 283–284, 2014.

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

[1]See youtu.be/uwNPJfRNe3A for a demo of the method.