A Kernel Approach to Addressing Term Mismatch

Jun Xu
Microsoft Research Asia
No. 49 Zhichun Road
Beijing, China 100190
junxu@microsoft.com

Wei Wu
Dept. of Probability& Statistics
Peking University
Beijing, China 100871
wuwei19850318@hotmail.com

Hang Li, Gu Xu
Microsoft Research Asia
No. 49 Zhichun Road
Beijing, China 100190
{hangli,guxu}@microsoft.com

ABSTRACT

This paper addresses the problem of dealing with term mismatch in web search using 'blending'. In blending, the input query as well as queries similar to it are used to retrieve documents, the ranking results of documents with respect to the queries are combined to generate a new ranking list. We propose a principled approach to blending, using a kernel method and click-through data. Our approach consists of three elements: a way of calculating query similarity using click-through data, a mixture model for combination of rankings using relevance, query similarity, and document similarity scores, and an algorithm for learning the weights of blending model based on the kernel method. Large scale experiments on web search and enterprise search data sets show that our approach can effectively solve term mismatch problem and significantly outperform the baseline methods of query expansion and heuristic blending.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

term mismatch, blending, kernel methods, click-through data

1. INTRODUCTION

Term mismatch is one of the most critical challenges for web search. That is, the document and the query may be relevant but they do not share terms. To tackle the term mismatch problem, many approaches have been proposed in traditional IR and web search. For example, query expansion or pseudo relevance feedback [1] reformulates the query by adding related terms or weighting terms and conducts retrieval and ranking with the reformulated query. Blending is another approach [2] which first finds similar queries for the input query from an offline query repository, and then uses the input query and its similar queries to retrieve documents from an offline document index, and finally combines and re-ranks the documents to create a new ranking list (cf., Figure 1).

We propose a principled approach for blending to tackle term mismatch problem. In our approach, we employ (1) a way of calculating query similarities using click-through data, (2) a mixture model for blending, and (3) a kernel method for learning the weights of the blending model.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

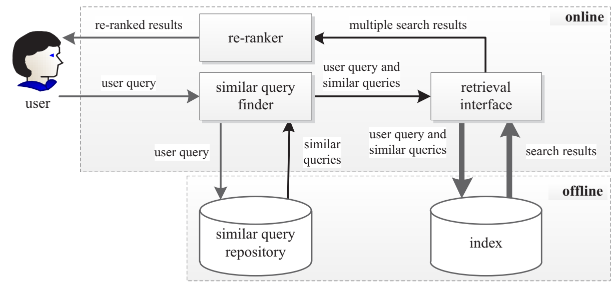

Figure 1: System overview of blending.

The kernel method exploits a positive semi-definite kernel which measures the similarity between two query-document pairs as product of query similarity, document similarity, and the relevance scores of the query-document pairs. The output of the kernel method is exactly the optimal blending model with respect to the training data. Our method also includes an implementation of the kernel method using Ranking SVM technique and click-through data.

2. OUR APPROACH

We exploit a mixture model for blending:

$$\begin{aligned} f(q,d) &= \sum_{i=1}^{N} \alpha_i r(q,d) k_Q(q,q_i) k_D(d,d_i) r(q_i,d_i) \\ &= r(q,d) \times \sum_{i=1}^{N} \alpha_i k_Q(q,q_i) k_D(d,d_i) r(q_i,d_i), \end{aligned} \tag{1}$$

where $0 \leq k_Q(\cdot,\cdot) \leq 1$ denotes query similarity, $0 \leq k_D(\cdot,\cdot) \leq 1$ denotes document similarity, α_i denotes weights, $r(\cdot,\cdot) > 0$ denotes basic ranking model, and N denotes number of training query-document pairs. The training data is concerned with the input query and its similar queries.

The blending model (1) determines the ranking score of query q and document d, not only based on query q and document d themselves, but also based on similar queries q' and their associated documents d'. The rationale behind is that even the ranking score of q and d is not reliable (in an extreme case, d cannot be retrieved with q), one can still use the ranking scores of their similar queries to smooth the score and make it reliable. All the ranking scores are linearly combined, and weighted by query similarity k_Q, document similarity k_D, and weight $\{\alpha_i\}$.

We specifically define $k_Q(q,q')$ as Pearson correlation coefficient

Table 1: Dataset statistics.

	Web search	Enterprise search
# of judged queries	8,294	2,864
# of judged query-URL pairs	1,715,844	282,130
# of impressions in click-through	490,085,192	17,383,935
# of unique queries in click-through	14,977,647	2,368,640
# of unique URLs in click-through	30,166,304	2,419,866
# of clicks in click-through	2,605,404,156	4,996,027

between the clicked URLs of two queries:

$$k_Q(q,q') = \frac{\sum_{i=1}^{n}(u_i-\bar{u})(v_i-\bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i-\bar{u})^2}\sqrt{\sum_{i=1}^{n}(v_i-\bar{v})^2}}, \tag{2}$$

where u_i and v_i denote numbers of clicks on URL i by query q and q' respectively, $\bar{u}$ and $\bar{v}$ denote average numbers of clicks of q and q' respectively, and n denotes total number of clicked URLs by q and q'. The underlying intuition is that if two queries convey the same search intent, they tend to have many co-clicked URLs.

We employ a kernel method to learn the weights $\{\alpha_i\}$. Suppose we are given training data $S = \{(q_i,d_i),t_i\}_{i=1}^{N}$, where t_i is the relevance rank of document d_i with respect to query q_i. We select the optimal blending model $f(q,d)$ by solving the following problem:

$$\min_{f\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} l(f(q_i,d_i),t_i) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2, \tag{3}$$

where $\mathcal{H}$ is a reproducing kernel Hilbert space, $\|\cdot\|_{\mathcal{H}}$ denotes a regularization on space $\mathcal{H}$, $l(\cdot,\cdot)$ is the loss function, and $\lambda > 0$ is coefficient. Suppose the RKHS $\mathcal{H}$ is generated by a positive semi-definite kernel $k : (\mathcal{Q}\times\mathcal{D})\times(\mathcal{Q}\times\mathcal{D}) \to \mathbb{R}$ defined as

$$k((q,d),(q',d')) = r(q,d)k_Q(q,q')k_D(d,d')r(q',d'), \tag{4}$$

According to the Representer Theorem of kernel methods [4], the learned optimal blending model is exactly the one given by Eq. (1).

As a specific implementation, we choose BM25 as the basic ranker $r(q,d)$. Document similarity $k_D(d,d')$ is simply defined as cosine similarity between the titles and URLs of two documents d and d'. Following the proposal in [3], we generate pairwise training instances from click-through data, and we use Ranking SVM technique [3] to train the model. More details about our method can be found in [5].

3. EXPERIMENTS

We used two large scale datasets from a web search engine and an enterprise search engine running in an IT company. The two datasets consist of query-URL pairs and their relevance judgments. The relevance judgments can be "Perfect", "Excellent", "Good", "Fair", or "Bad". We also collected large scale click-through data from both search engines. Table 1 gives the dataset statistics.

Our experimental results show that by using Eq. (2), one can really find similar queries that represent the same search intents. Table 2 shows some examples.

We considered the following baseline methods: BM25, BM25 plus pseudo relevance feedback (using title of top 1 retrieved document by $r(\cdot,\cdot)$ and denoted as "Query Expansion (PRF)"), and BM25 plus a number of approximations of query expansion. Among the approximated query expansion methods, " Query Expansion (Tit)" uses the title of the most clicked document; "Query Expansion (SimQry)" uses the most similar query; "Query Expansion (SimQryTit)" uses title of most clicked document of similar query; and "Query Expansion (Heu)" first uses the most clicked title; if the there is no clicked document, then it uses the title of most clicked document of similar query. We also considered two blending methods as baseline. First, we consider an additive model:

Table 2: Similar queries extracted from click-through data.

input query	similar queries
wallmart	wall mart, walmart, wal mart, walmarts
ironman	iron man, ironman movie, irnman, www.iron man.com
knives	knifes, knives.com, knife outlet, knife
aircraft for sale	aircraft sales, airplanes for sale, used airplanes for sale, used planes for sale
ucsd	ucsd.edu, uc san diego, uscd, university of california san diego

Table 3: Ranking accuracies.

	MAP	NDCG@1	NDCG@3	NDCG@5
Web search data				
Our Approach	**0.1219**	**0.2480**	**0.2587**	**0.2716**
Blending (Mul)	0.1181	0.2295	0.2519	0.2665
Blending (Add)	0.1039	0.2273	0.2396	0.2512
Query Expansion (Heu)	0.0957	0.1832	0.2115	0.2282
Query Expansion (Tit)	0.0963	0.1797	0.2061	0.2237
Query Expansion (SimQry)	0.0961	0.1796	0.2060	0.2237
Query Expansion (SimQryTit)	0.0980	0.1786	0.2064	0.2304
Query Expansion (PRF)	0.0799	0.1539	0.1704	0.1831
BM25	0.0908	0.1728	0.2019	0.2180
Enterprise search data				
Our Approach	**0.3122**	**0.4780**	**0.5065**	**0.5295**
Blending (Mul)	0.3046	0.4636	0.4910	0.5102
Blending (Add)	0.3020	0.4543	0.4914	0.5005
Query Expansion (Heu)	0.3015	0.4392	0.4842	0.5070
Query Expansion (Tit)	0.2955	0.4076	0.4712	0.4958
Query Expansion (SimQry)	0.2975	0.4325	0.4781	0.5011
Query Expansion (SimQryTit)	0.2983	0.4336	0.4826	0.5052
Query Expansion (PRF)	0.2867	0.4007	0.4555	0.4829
BM25	0.2745	0.4246	0.4531	0.4741

$$f(q,d) = r(q,d) + \sum_{i=1}^{N} \alpha_i k_Q(q,q_i)k_D(d,d_i)r(q_i,d_i),$$

which is denoted as "Blending (Add)". Furthermore, we denote the multiplication model in Eq. (1) as "Blending (Mul)". The weights in both models are determined uniformly. As for evaluation measures, we used MAP and NDCG at the positions of 1, 3, and 5.

Table 3 gives the experimental results on web search data and enterprise search data. We can see that our method outperforms the baseline methods for dealing with term mismatch. We conducted significant tests (t-test) on the improvements. The results show that the improvements are all statistically significant (p-value < 0.05).

4. REFERENCES

[1] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999.

[2] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. The effect multiple query representations on information retrieval system performance. In *SIGIR '93*, pages 339–346, 1993.

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

[4] B. Schölkopf and A. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[5] W. Wu, J. Xu, H. Li, and S. Oyama. Asymmetric Kernel Learning *Microsoft Technical Report,* MSR-TR-2010-85, 2010.