Embedding-based Query Language Models

Hamed Zamani
Center for Intelligent Information Retrieval
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003
zamani@cs.umass.edu

W. Bruce Croft
Center for Intelligent Information Retrieval
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003
croft@cs.umass.edu

ABSTRACT

Word embeddings, which are low-dimensional vector representations of vocabulary terms that capture the semantic similarity between them, have recently been shown to achieve impressive performance in many natural language processing tasks. The use of word embeddings in information retrieval, however, has only begun to be studied. In this paper, we explore the use of word embeddings to enhance the accuracy of query language models in the ad-hoc retrieval task. To this end, we propose to use word embeddings to incorporate and weight terms that do not occur in the query, but are semantically related to the query terms. We describe two embedding-based query expansion models with different assumptions. Since pseudo-relevance feedback methods that use the top retrieved documents to update the original query model are well-known to be effective, we also develop an embedding-based relevance model, an extension of the effective and robust relevance model approach. In these models, we transform the similarity values obtained by the widely-used cosine similarity with a sigmoid function to have more discriminative semantic similarity values. We evaluate our proposed methods using three TREC newswire and web collections. The experimental results demonstrate that the embedding-based methods significantly outperform competitive baselines in most cases. The embedding-based methods are also shown to be more robust than the baselines.

CCS Concepts

•Information systems → Query representation; Query reformulation; Language models;

Keywords

Word embedding; language models; pseudo-relevance feedback; query expansion

ICTIR '16, September 12-16, 2016, Newark, DE, USA

DOI: http://dx.doi.org/10.1145/2970398.2970405

1. INTRODUCTION

Capturing the semantic similarities between vocabulary terms have been an interesting and challenging issue in natural language processing (NLP) and information retrieval (IR) for some time. Many approaches have been proposed for finding semantically similar words, such as latent semantic indexing [7] and the information content-based method [23]. Recent developments in distributed semantic representations, also called word embedding, have been shown to be highly effective in many NLP tasks, such as word analogy [19] and named-entity recognition [8]. Word embedding techniques assign each term a low-dimensional (compared to the vocabulary size) vector in a "semantic vector space". In this space, close vectors are supposed to demonstrate high semantic or syntactic similarity between the corresponding words. Word2vec [19] and GloVe [21] are examples of successful implementations of word embeddings that respectively use neural networks and matrix factorization to learn embedding vectors.

Although word embeddings have shown significant improvements in many NLP tasks, there is less known about the use of word embedding vectors to improve retrieval performance. In this paper, we focus on the vocabulary mismatch problem, i.e., the mismatch of different vocabulary terms with the same concept. This is a fundamental IR problem, since users often use different words to describe a concept in the queries than those that authors of documents use to describe the same concept [28]. Therefore, it will cause poor retrieval performance. We address the vocabulary mismatch problem in the language modeling framework for ad-hoc information retrieval by focusing on the semantic similarity between terms. Recently, the word embedding techniques have been employed to improve the accuracy of document language models in [10, 32]. In contrast, in this paper, we focus on estimating accurate language models for queries based on word embedding vectors, which leads to more efficient and effective methods. In addition to the terms that appear in the query, we incorporate and weight the words that do not occur in the query, but are semantically similar to the query terms. To do so, we propose two query expansion models with different simplifying assumptions. The first model assumes that given each term w, all query terms are conditionally independent. The second model assumes that the semantic similarity between each pair of vocabulary terms is independent of the queries.

A well-known and effective technique in ad-hoc information retrieval to address the vocabulary mismatch problem is pseudo-relevance feedback (PRF) [15, 18, 24, 28, 29]. PRF

assumes that a small set of top-retrieved documents is relevant to the query, and thus a number of relevant terms can be selected from this set of feedback documents (also called pseudo-relevant documents) to be added to the query model. In this paper, we extend the relevance model approach [15], one of the most effective and robust PRF methods, by considering the semantic similarity between the terms, in addition to the term matching similarity.

Furthermore, we observe that the semantic similarity scores of the vocabulary terms in the embedding semantic space are not sufficiently discriminative for our task, and potentially for many other IR tasks. Previous work on word embedding considers typical similarity functions, e.g., the cosine similarity or the Euclidean distance functions, that are shown to be very effective in detecting nearest words in terms of their semantic similarity [12, 17]. In fact, these metrics are sufficient when the order of words in terms of their semantic similarity to a given word is needed. In contrast, the similarity values are not sufficiently discriminative to be used in IR models. Thus, we propose transforming the similarity values using the well-known sigmoid function.

We evaluate the proposed methods using three standard TREC collections: Associated Press (AP), the TREC Robust Track 2004 collection, and the TREC Terabyte Track 2004-2006 collection (GOV2). The first two collections contain high-quality news articles, while the last one is a large web collection. The results indicate that the proposed methods outperform competitive baselines, in all collections. The MAP improvements in most cases are statistically significant. The proposed methods also perform quite well in improving the precision of top retrieved documents. We also show that the proposed methods are more robust than the baselines. In addition, the experimental results suggest that the sigmoid transformation of embedding similarities can significantly improve the performance.

2. RELATED WORK

In this section, we first briefly review previous work on query expansion and pseudo-relevance feedback. Then, we introduce the applications of word embeddings in information retrieval.

2.1 Query Expansion

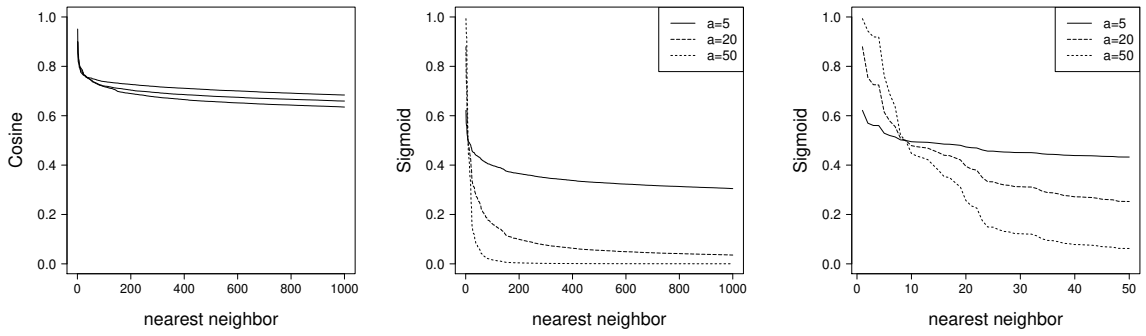
Query expansion is the process of adding relevant terms to a query to improve the retrieval effectiveness. There are a number of query expansion methods based on linguistic resources, such as WordNet, but they have not substantially improved the retrieval performance [26]. Although a number of data-driven query expansion methods, such as [3], can improve the average retrieval performance, they are shown to be unstable across queries [4, 6]. According to Xu and Croft [28], the aforementioned query expansion techniques are based on global analysis. Global analysis often relies on external resources or document collections. On the other hand, local analysis expands queries using the documents that are related to them, like top-retrieved documents. Query expansion via pseudo-relevance feedback is a common technique used to improve retrieval effectiveness in many retrieval models [15, 18, 24, 29]. Relevance models [15], mixture model, and divergence minimization model [29] are the first PRF methods proposed for the language modeling framework. Since then, several other methods have been proposed, but relevance models are shown to be still among the state-of-the-art PRF methods and perform more robustly than many other methods [18]. Recently, Montazeralghaem et al. [20] stated that PRF models can be improved by taking the semantic similarity between feedback terms and the query into account. There are also a number of learning-based PRF methods. Although some of these methods have promising performance, they are not related to our work and are not considered in this paper. In this paper, the first two proposed query expansion models are based on global analysis, while the third model is a combination of local and global analysis.

2.2 Word Embeddings for IR

Unlike NLP, where word embeddings have been successfully employed in several tasks, word embedding techniques in IR are still relatively unstudied. Recently, Zheng and Callan [30] proposed a supervised embedding-based technique to reweight terms for the IR models, e.g., BM25. They learned term weights using the distributed semantic representations. Clinchant and Perronnin [5] proposed Fisher Vector (FV), a document representation framework based on continuous word embeddings, which aggregates the non-linear mapping of word vectors into a document-level representation. Although FV outperforms latent semantic indexing (LSI) [7] in ad-hoc retrieval, it does not perform better than popular IR frameworks, such as TF-IDF and the divergence from randomness retrieval model.

A number of methods have focused on computing the semantic similarity between two documents. Le and Mikolov [16] proposed Paragraph Vector (PV), a method to compute an embedding vector for each sentence, paragraph, or document. Since queries are not available during the training time of embedding vectors, PV cannot always be employed for computing the embedding vectors of queries. Kusner et al. [13] recently proposed word mover's distance (WMD), a distance function between two documents, which measures the minimum traveling distance from the embedded words of one document to another one. WMD achieved good performance in the document classification task. A supervised method for computing the semantic similarities between short texts was also proposed by Kenter and de Rijke [12]. Zhou et al. [31] recently proposed an embedding-based method for question retrieval in the community question answering systems.

BWESG [27], a bilingual word embedding method, has recently been developed and applied to information retrieval. This model learns bilingual embedding vectors from document-aligned comparable corpora. Regarding the well-defined structure of language modeling framework in information retrieval, a number of methods have been proposed to improve the performance of language models using the word embedding vectors. For instance, Ganguly et al. [10] considered the semantic similarities between vocabulary terms to smooth document language models. Recently, Zuccon et al. [32] proposed to employ word embeddings within the well-known translation model for IR. Their proposed method achieves comparable performance to the mutual information-based translation language models [11]. The main idea behind the methods proposed in [10] and [32] is similar. Both methods consider semantic similarity in computing the probability of terms in the documents. Recently, ALMasri et al. [2] proposed a heuristic query expansion method based on word embedding similarities. Their method is a term-by-term ex-



(a) cosine similarity (b) sigmoid - top 1000 words (c) sigmoid - top 50 words

Figure 1: The cosine similarity function (the left plot) and its sigmoid transformation (the two right plots) of nearest neighbors (in descending order in terms of their similarity value). For the sigmoid function, the parameter c is set to 0.8.

pansion instead of expanding the whole query. Sordoni et al. [25] also proposed a query expansion method based on concept embedding. Their method is a supervised learning approach with the quantum entropy loss function that uses click-through data. More recently, Diaz et al. [9] proposed locally-trained word embeddings for query expansion.

The main difference between this paper and the existing work is that we focus on estimating accurate query language models which efficiently outperforms the embedding-based document language models [10].

3. EMBEDDING-BASED QUERY MODELS

In the language modeling framework [22], estimating accurate query language models plays a key role in achieving high retrieval accuracy. Maximum Likelihood Estimation (MLE) is a simple yet effective method for estimating query language models. Several techniques, such as pseudo-relevance feedback (PRF), have been proposed to estimate more accurate query language models. The goal of this paper is to estimate accurate query language models by employing word embedding vectors to address the vocabulary mismatch problem in information retrieval. In this section, we first introduce the idea behind the word embedding techniques whose purpose is to capture the semantic similarity between vocabulary terms. We propose to use the sigmoid function over the well-known similarity metrics to achieve more discriminative similarity values between terms. We further propose to employ word embedding vectors to estimate query language models via query expansion in a weighted manner. We afterward extend the idea of relevance models [15], one of the most effective and robust PRF methods [18], by leveraging word embedding vectors.

3.1 Word Embedding

Word embedding techniques learn a low-dimensional vector (compared to the vocabulary size) for each vocabulary term in which the similarity between the word vectors can show the semantic as well as the syntactic similarities between the corresponding words. This is why word embeddings are also called distributed semantic representations. Note that the word embedding methods are categorized as unsupervised learning algorithms, since they only need a large amount of raw textual data in their training phase. A popular method to compute these word embedding vectors is neural network-based language models. For instance, Mikolov et al. [19] introduced word2vec, an embedding method that learns word vectors via a neural network with a single hidden layer. Another successful trend in learning semantic word representations is employing global matrix factorization over the word-word matrices. GloVe [21] is an example of such methods.

Word embeddings have been shown to be extremely useful in many NLP tasks, such as word analogy [19] and named-entity recognition [8]. In these tasks, the semantic similarity between terms are often computed using the cosine similarity of the word embedding vectors. Figure 1a plots the cosine similarity[1] of the top 1000 similar words to three random words.[2] As shown in this figure, these three curves have similar behaviours. An interesting observation in this figure is that there are no substantial differences (e.g., two times more) between the similarity of the most similar term and the 1000^{th} similar term to a given term w, while the 1000^{th} word is unlikely to have any semantic similarity with w. In other words, the similarity values are not discriminative enough. On the other hand, as extensively explored in various NLP tasks, the order of words in terms of their semantic similarity to a given term is accurate, especially for the very close terms. Therefore, we need a monotone mapping function to transform the similarity scores achieved by the popular similarity metrics, such as the cosine similarity.

Intuitively, there will be a small number of words that are semantically similar to a given word w. Hence, the similarity of most of words with w should have a value close to zero. Therefore, we propose using the well-known sigmoid function to transform the similarity scores coming from the cosine similarity function. The sigmoid function is a non-linear mapping function that maps values in $[-\infty, +\infty]$ to the $[0, 1]$ interval. This function, which has been used in many machine learning algorithms, such as the logistic regression and neural networks, can be calculated as:

$$S(x) = \frac{1}{1 + e^{-a(x-c)}} \tag{1}$$

where x denotes the input of the sigmoid function with two free parameters a and c. Figure 1b plots the cosine similar-

[1]In this paper, we linearly transform the cosine similarity values to the $[0, 1]$ interval.

[2]In Figure 1, we plot the cosine similarity over the vectors learned by the GloVe method. The other popular similarity metrics, such as the Euclidean distance and the KL-divergence, also suffer from the same problem.

ity values of the top 1000 similar words to a random word that are transformed using the sigmoid function with different values of a. By increasing the value of a, the similarity scores drop more quickly. To have a better understanding of the sigmoid behaviour, we also plot the same curves for only the top 50 words in Figure 1c. As shown in this figure, when a is set to a large number, the similarity scores drop quickly and more close values are assigned to the top terms (e.g., top 10 terms). The parameters a and c make the sigmoid function a flexible transformation for similarity scores, that can be employed in different scenarios. For instance, when there are a few semantically similar words to a word w, but there is not much distinction between them, in terms of their similarity to w, the sigmoid transformation can give close weights to the very similar terms, in addition to dropping very fast. Another advantage of transforming the similarity scores using the sigmoid function is that the results are always in the $[0, 1]$ interval, and thus all distance functions with unbounded values can be used for computing the similarity between embedding vectors. These behaviours of the sigmoid function make it suitable for our task.[3]

3.2 Embedding-based Query Expansion

Similar to most language modeling-based methods, we focus on unigram language models. Therefore, for each vocabulary term w, we should estimate $p(w|\theta_Q)$ where θ_Q denotes the language model of the query Q. In this subsection, we propose two novel estimations for the query language models by making use of semantic similarity between the terms coming from the similarity between the word embedding vectors. These two estimations have different simplifying assumptions. The first method considers that there is a conditional independence assumption between the query terms; while the second method assumes that the semantic similarity between two given terms is independent of the query. Interestingly, as we will see in the following, the first assumption leads to an expansion method based on multiplicative similarity, i.e., the expanded terms should be similar to all query terms. On the other hand, the second assumption leads to an additive similarity (a mixture model).

3.2.1 Conditional Independence of Query Terms

To estimate $p(w|\theta_Q)$, we first consider the Bayes rule as follows:

$$p(w|\theta_Q) = \frac{p(\theta_Q|w)p(w)}{p(Q)} \propto p(\theta_Q|w)p(w) \tag{2}$$

In the above equation, we ignore $p(Q)$ since it is independent of the term w. We assume that query terms are independent of each other, but we keep their dependence on w. Therefore, we can estimate the query language model as follows:

$$p(w|\theta_Q) \propto p(q_1, q_2, \ldots, q_k|w)p(w) = p(w)\prod_{i=1}^{k} p(q_i|w) \tag{3}$$

where $q_1, q_2, \ldots, q_k$ are the query terms and k is the query length. To compute $p(q_i|w)$, we consider the word embedding similarities which can capture the semantic similarity between vocabulary terms. To do so, we compute this probability as follows:

$$p(q_i|w) = \frac{\delta(q_i, w)}{\sum_{w' \in V} \delta(w', w)} \tag{4}$$

where δ and V denote the similarity function (e.g., the sigmoid function over the cosine similarity) and the vocabulary set, respectively. Considering the law of total probability, we compute $p(w)$ using the following equation:[4]

$$p(w) = \sum_{w' \in V} p(w, w') \propto \sum_{w' \in V} \delta(w, w') \tag{5}$$

The generated language model θ_Q can be linearly interpolated with the maximum likelihood estimation of the query language model with the coefficient of α.

3.2.2 Query-Independent Term Similarities

To estimate the query language model θ_Q, we first use the law of total probability as follows:

$$p(w|\theta_Q) = \sum_{w' \in V} p(w, w'|\theta_Q) = \sum_{w' \in V} p(w|w', \theta_Q)p(w'|\theta_Q) \tag{6}$$

In the above calculations, we use the Bayes rule. We assume that the semantic similarity between the terms is independent of the query language model. Therefore, $p(w|w', \theta_Q)$ can be computed as follows:

$$p(w|w', \theta_Q) = p(w|w') = \frac{\delta(w, w')}{\sum_{w'' \in V} \delta(w'', w')} \tag{7}$$

where δ denotes the semantic similarity between two given terms and can be calculated by transforming the cosine similarity values using the sigmoid function.

Considering the maximum likelihood estimation, we can rewrite Equation 6 as follows:

$$p(w|\theta_Q) \propto \sum_{w' \in Q} \frac{\delta(w, w')}{\sum_{w'' \in V} \delta(w'', w')} \times \frac{c(w', Q)}{|Q|} \tag{8}$$

where $c(w', Q)$ and $|Q|$ denote the count of term w' in the query and the query length, respectively.

Similar to the previous embedding-based estimation of query models, the generated query language model can be linearly interpolated with the maximum likelihood estimation of the query language model with a coefficient of α.

3.3 Embedding-based Relevance Model

Pseudo-relevance feedback has been shown to be highly effective in improving the retrieval performance [15, 18, 29]. In PRF, it is assumed that the top-retrieved documents are relevant to the query, and thus they can be used to improve the query language model accuracy. In this subsection, we propose an embedding-based relevance model, a method inspired by the relevance model approach proposed by Lavrenko and Croft [15], which has been shown to be one of the most effective and robust PRF methods [18].

[3] We have examined other transformation functions, such as the softmax and the power functions, but the sigmoid transformation provides more reasonable similarity scores, which leads to better performance. For the sake of space, we only introduce the best transformation function that we found.

[4] Intuitively, a word with higher semantic similarity with all the other words will achieve higher probability. In other words, general terms are supposed to have high probabilities according to this definition, which is also consistent with the other definitions for $p(w)$, such as the the frequency ratio of the word w in a large corpus.

Table 1: Collections statistics.

ID	collection	queries (title only)	#docs	doc length	#qrels
AP	Associated Press 88-89	TREC 1-3 Ad-Hoc Track, topics 51-200	165k	287	15,838
Robust	TREC Disks 4 & 5 minus Congressional Record	TREC 2004 Robust Track, topics 301-450 & 601-700	528k	254	17,412
GOV2	2004 crawl of .gov domains	TREC 2004-2006 Terabyte Track, topics 701-850	25,205k	648	26,917

We compute the feedback language model as follows:

$$p(w|\theta_F) \propto \sum_{D \in F} p(w, Q, D) = \sum_{D \in F} p(Q|w, D)p(w|D)p(D) \tag{9}$$

where θ_F and F respectively denote the feedback language model and the set of feedback documents, i.e., the top-retrieved documents. To compute $p(Q|w, D)$, we consider both term matching and semantic similarities. This probability can be computed via a linear interpolation with the coefficient of β:

$$p(Q|w, D) = \beta\ p_{tm}(Q|w, D) + (1 - \beta)\ p_{sem}(Q|w, D) \tag{10}$$

where p_{tm} and p_{sem} denote the probabilities coming from the term matching similarities and the semantic similarities, respectively. Similar to RM3 [1, 15], $p_{tm}(Q|w, D)$ can be estimated by considering the independence assumption of terms (query terms and w) as follows:

$$p_{tm}(Q|w, D) = \prod_{i=1}^{k} p(q_i|D) \tag{11}$$

where q_i denotes the i^{th} term of the query Q with the length of k. To compute $p_{sem}(Q|w, D)$, we assume that query terms are independent of each other, but we keep their dependence to the term w and document D. Therefore, we can calculate this probability as follows:

$$p_{sem}(Q|w, D) = \prod_{i=1}^{k} p_{sem}(q_i|w, D) \triangleq \prod_{i=1}^{k} \frac{\delta(q_i, w)c(q_i, D)}{Z} \tag{12}$$

where δ computes the semantic similarity between two given terms and $c(q_i, D)$ is the count of term q_i in the document D. Z is a normalization factor, which is only needed to be a summation over the terms appeared in the document D (instead of all vocabulary terms), and thus it is not computationally expensive.

Similar to RM3, we compute $p(w|D)$ (see Equation (9)) using the maximum likelihood estimation (MLE) smoothed by a reference language model.[5] We assume that there is no prior knowledge available about the relevance score of pseudo-relevant documents, and thus we calculate $p(D)$ using a uniform distribution. It is notable that the proposed embedding-based relevance model satisfies the "semantic effect" constraint recently proposed by Montazeralghaem et al. [20]. The updated query language model can be calculated using the linear interpolation of the original query language model with the computed feedback language model:

$$p(w|\theta_Q^*) = \alpha\ p(w|\theta_Q) + (1 - \alpha)\ p(w|\theta_F) \tag{13}$$

[5]We also considered embedding-based semantic similarities in computing this probability using the law of total probability ($p(w|D) = \sum_{w' \in V} p(w, w'|D)$), but there is no significant improvement over the MLE estimation in our experiments. Therefore, we keep it as simple as possible.

Note that the original query language model can be computed using either the maximum likelihood estimation, or one of the embedding based query language models proposed in Section 3.2.

It should be noted that since the retrieval models, such as the KL-divergence retrieval model, compute the similarity between the query and documents with respect to the terms that appeared in the query, it is reasonable that the updated query model has limited number of terms with non-zero weights. Hence, in the proposed embedding-based relevance model as well as in the two proposed query expansion models (see Section 3.2), we select the top m terms in the updated language model with highest probabilities to be added to the query.

4. EXPERIMENTS

4.1 Experimental Setup

To evaluate the proposed methods, we used three standard TREC collections: AP (Associated Press 1988-1989), Robust (TREC Robust Track 2004 collection), and GOV2 (TREC Terabyte Track 2004-2006 collection). The first two collections contain high-quality news articles and the last one is a large web collection. The statistics of these collections are reported in Table 1. We considered the title of topics as queries in our experiments. The standard INQUERY stopword list was used in all experiments, and no stemming was performed.

We employed the KL-divergence retrieval model [14] with the Dirichlet prior smoothing method. All experiments were carried out using the Galago toolkit[6]. In all the experiments, we used the word embeddings extracted using the GloVe method [21]. The word embeddings were extracted from a 6 billion token collection (the Wikipedia dump 2014 plus the Gigawords 5).[7] Note that we considered only the queries where the embedding vectors of all query terms are available. The employed embedding vectors contain all query terms for 146 (out of 150), 241 (out of 250), and 147 (out of 150) queries in AP, Robust, and GOV2, respectively. Note that to have a fair evaluation, we also used these queries for the baselines methods.

4.1.1 Parameters Setting

In all the experiments, the Dirichlet prior smoothing parameter μ was set to Galago's default value of 1500. In the experiments related to the pseudo-relevance feedback, the number of feedback documents was set to the typical value of 10. In all the experiments (except in those that explicitly mentioned), the parameters α (the linear interpolation coefficient), m (the number of terms added to the queries), and β (see Section 3.3) were set using 2-fold cross-validation to optimize MAP over the queries of each collection. We sweeped the parameters α and β between $\{0.1, \ldots, 0.9\}$.

[6]http://www.lemurproject.org/galago.php

[7]Statistics of this collection is reported in Table 4.

Table 2: Comparing the proposed embedding-based query expansion methods with the baselines. The superscript 1/2/3/4 denotes that the MAP improvements over MLE/GLM/VEXP/AWE are statistically significant. The highest value in each row is marked in bold.

Dataset	Metric	MLE	GLM	VEXP	AWE	EQE1	EQE2
AP	MAP	0.2236	0.2254	0.2338	0.2304	0.2388^{1234}	$\mathbf{0.2391}^{1234}$
	P@5	0.4260	0.4369	0.4412	0.4356	0.4397	**0.4466**
	P@10	0.4014	0.4051	0.4038	0.4058	**0.4075**	0.4014
	RI	–	0.10	0.18	0.14	**0.32**	**0.32**
Robust	MAP	0.2190	0.2244	0.2253	0.2224	$\mathbf{0.2292}^{124}$	0.2257^{1}
	P@5	0.4606	0.4523	0.4722	0.4680	**0.4739**	0.4622
	P@10	0.3979	0.3929	0.4133	0.4066	0.4162	**0.4183**
	RI	–	0.22	0.17	0.14	**0.30**	0.22
GOV2	MAP	0.2696	0.2684	0.2687	0.2657	$\mathbf{0.2745}^{1234}$	0.2727^{4}
	P@5	0.5592	0.5537	0.5932	0.5537	**0.5959**	0.5810
	P@10	0.5531	0.5483	0.5537	0.5503	**0.5660**	0.5517
	RI	–	-0.14	0.10	-0.18	**0.20**	0.08

The value of the parameter m and the number of feedback documents (for PRF experiments) were also selected from $\{10, 20, ..., 100\}$. The sigmoid parameters (a and c in Equation (1)) were chosen using cross-validation from $\{5, 10, ..., 50\}$ and $\{0.7, 0.75, \ldots, 0.9\}$, respectively. The free hyper-parameters of the baselines were also set using the same procedure. In all experiments unless explicitly mentioned, the dimension of embedding vectors was set to 200.

4.1.2 Evaluation Metrics

Mean Average Precision (MAP) of the top-ranked 1000 documents is selected as the main evaluation metric to evaluate the retrieval effectiveness. Furthermore, we also consider the precision of the top 5 and 10 retrieved documents (P@5 and P@10). Statistically significant differences of performances are determined using the two-tailed paired t-test computed at a 95% confidence level based on the average precision per query.

To evaluate the robustness of methods, we consider the robustness index (RI) [6] which is defined as $\frac{N_+ - N_-}{|Q|}$, where $|Q|$ denotes the number of queries, and N_+/N_- shows the number of queries improved/decreased by each method compared to the maximum likelihood estimation baseline.[8] The RI value is in the $[-1, 1]$ interval and the higher RI values determine more robust methods.

4.2 Results and Discussion

In this subsection, we first evaluate the two proposed embedding-based query expansion models. We further evaluate the proposed methods in the PRF scenario. Then, we experiment the sensitivity of the proposed methods to the word embedding vectors. At the end, we briefly analyze the behaviour of the sigmoid function in our task.

4.2.1 Embedding-based Query Expansion Models

To evaluate the two proposed embedding-based query expansion models (EQE1 and EQE2), we consider four baselines: (1) the standard maximum likelihood estimation of the query model (MLE), (2) the embedding-based document language model smoothing method (GLM)[9] [10] (which is also similar to [32]), (3) a heuristic-based query expansion method based on word embeddings (VEXP) [2], and (4) a query expansion method based on the similarity of vocabulary term vectors and the average embedding vector of all query terms (AWE). Although the AWE baseline was not previously used for the query expansion purpose, the idea of averaging vectors of query/sentence terms has been previously used in a number of previous work [16, 27, 30]. Note that we do not use supervised approaches, such as [30, 25], as well as the methods that use external resources, such as knowledge graphs, as baseline.

The results achieved by the proposed methods (EQE1 and EQE2) and the baselines are reported in Table 2. According to this table, both proposed methods outperform all the baselines in all collections, in terms of MAP, P@5, P@10, and RI. The t-test shows that the MAP improvements of EQE1 compared to all the baselines are always significant, except compared to VEXP in the Robust collection. Although in some cases EQE2 outperforms EQE1, we can generally claim that in most cases EQE1 has superior performance. Note that EQE1 is based on multiplicative similarity, while EQE2 is based on additive similarity. Multiplicative similarity means that the expanded terms should be close to all query terms.

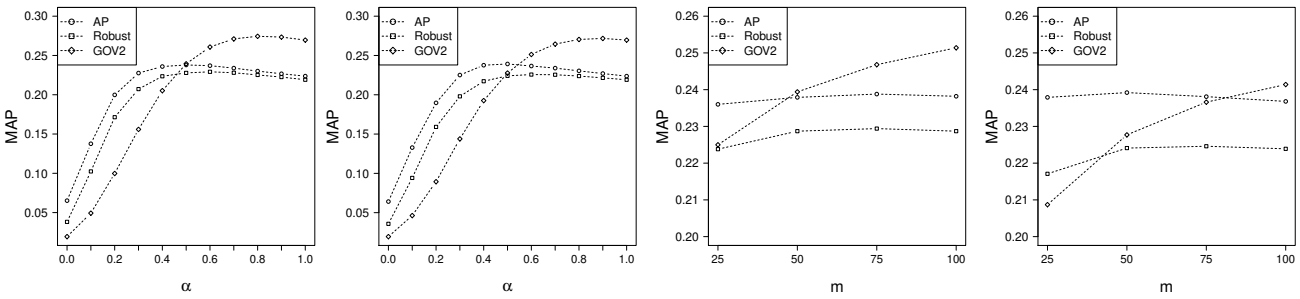
As shown in Table 2, the improvements over the MLE baseline in AP and Robust are higher than those in the GOV2 collection. The reason could be related to the corpus used for extracting the embedding vectors. This corpus mostly contains formal texts. Therefore, the context of the employed word embedding vectors is more similar to the context of the newswire collections rather than the GOV2 collection.[10] It should be noted that the results achieved by EQE1 and EQE2 can be further improved. For instance, we show in Section 4.2.3 that by increasing the dimension of word embedding vectors, the results will be improved.

In the next set of experiments, we study the sensitivity of the proposed methods (EQE1 and EQE2) to the two parameters α and m (see Section 3.2), in terms of MAP. We set α and m to 0.5 and 50, respectively. In each step, we sweep one of these parameters and fix the other one. Figures 2a and 2b respectively plot the sensitivity of EQE1 and EQE2 to the parameter α, the coefficient of interpolating the original query language model with the generated embedding-based

[8] To avoid the influence of very small average precision differences in the RI values, we only consider the improvements/losses higher than 10% (relatively).

[9] Note that all other methods use Dirichlet prior smoothing, but GLM is a linear interpolation smoothing method with constant coefficients.

[10] The results achieved by the embedding vectors trained on other corpora are reported in Section 4.2.3.



(a) EQE1 (b) EQE2 (c) EQE1 (d) EQE2

Figure 2: Sensitivity of EQE1 and EQE2 to the interpolation coefficient (α) and the term count (m), in terms of MAP.

Table 3: Evaluating the proposed methods in the pseudo-relevance feedback scenario. The superscript 1/2 denotes that the MAP improvements over MLE/RM3 are statistically significant. The highest value in each row is marked in bold.

Dataset	Metric	MLE	MLE+RM1 (RM3)	EQE1+RM1	EQE2+RM1	MLE+ERM	EQE1+ERM	EQE2+ERM
AP	MAP	0.2236	0.3051	0.3118^{12}	0.3115^{12}	0.3102^{12}	$\mathbf{0.3178}^{12}$	0.3140^{12}
	P@5	0.4260	0.4644	0.4808	0.4795	0.4699	**0.4822**	0.4644
	P@10	0.4014	0.4500	0.4500	0.4452	0.4521	**0.4568**	0.4479
	RI	–	0.47	0.45	0.41	**0.52**	0.47	**0.52**
Robust	MAP	0.2190	0.2677	0.2712^{12}	0.2710^{12}	0.2711^{12}	0.2731^{12}	$\mathbf{0.2750}^{12}$
	P@5	0.4606	0.4581	0.4747	0.4722	0.4639	**0.4797**	0.4730
	P@10	0.3979	0.4191	0.4241	0.4295	0.4241	0.4307	**0.4369**
	RI	–	0.31	**0.39**	0.35	0.31	0.32	0.36
GOV2	MAP	0.2696	0.2938	0.2987^{12}	0.2922^{1}	0.3005^{12}	$\mathbf{0.3012}^{12}$	0.2957^{1}
	P@5	0.5592	0.5592	0.5687	0.5673	0.5823	**0.5850**	0.5782
	P@10	0.5531	0.5599	0.5816	0.5714	0.5830	**0.5844**	0.5782
	RI	–	0.15	**0.22**	0.14	**0.22**	0.20	0.20

query language model. According to these figures, the performance of both methods when the original query language model is not considered (i.e., $\alpha = 0$) is quite low. This indicates that the generated embedding-based language model needs to be interpolated with the original language model. The reason is that we do not consider the query terms in the generated language model and these terms play key roles in the retrieval effectiveness. According to these two figures, the behaviours of both methods are similar to each other. Interestingly, the curves corresponding to AP and Robust (the two newswire collections) have similar behaviours. The results show that the best value for the parameter α in these two collections is 0.5. In contrast, in the GOV2 collection, the original language model needs to get higher weight in the linear interpolation. The best α values for EQE1 and EQE2 in GOV2 are 0.8 and 0.9, respectively.

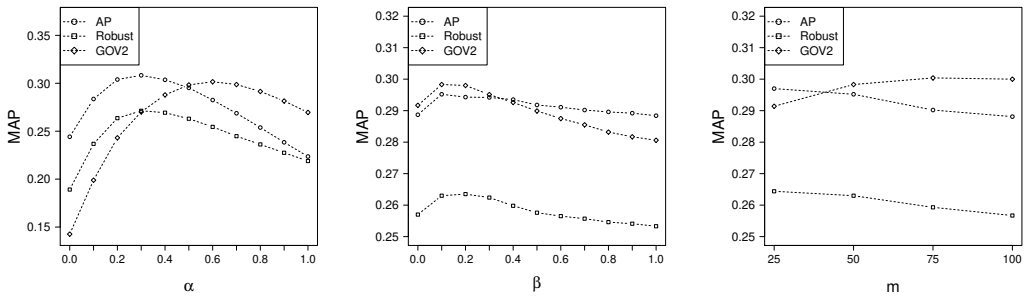
Figures 2c and 2d respectively show the sensitivity of EQE1 and EQE2 to the parameter m (term count). According to these figures, by varying the term count parameter, the retrieval performance in AP and Robust does not change dramatically, compared to GOV2. In the GOV2 collection, by increasing the number of terms, the performance is improved. While in AP and Robust, 50 or 75 are the best values for the parameter m.

To summarize, Figure 2 shows that α and m are collection-dependent parameters, and thus proper values should be chosen for them with respect to the test collection. In addition, the similar behaviours of curves corresponding to AP and Robust show that the retrieval performances in similar collections behave similarly. Thus, these parameters can be tuned in one collection and be set for another similar one.

4.2.2 *Embedding-based Pseudo-Relevance Feedback*

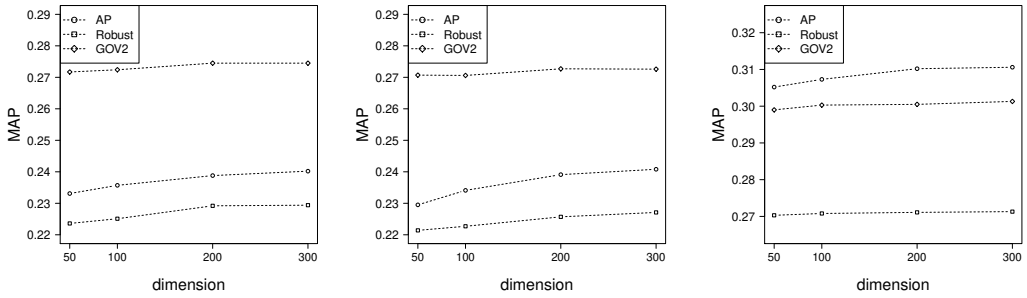
In this subsection, we evaluate the proposed embedding-based query language models in the pseudo-relevance feedback scenario. In these experiments, we consider two baselines: (1) the maximum likelihood estimation without feedback (MLE), and (2) the relevance model with the i.i.d. sampling assumption (i.e., RM3) [1, 15], a state-of-the-art PRF method that has been shown to perform well in various collections [18]. There are several other PRF methods that also perform well. Since the proposed ERM model is an extension of the RM3 model, we only consider it as the baseline, which is the most similar method to the proposed one.

The results are reported in Table 3. According to this table, RM3 significantly outperforms MLE in all collections in terms of MAP. This shows the effectiveness of pseudo-relevance feedback in information retrieval. As known, RM3 is the linear interpolation of MLE with RM1 [1]. More details about the RM1 feedback model can be found in [15]. In the first set of experiments, we employ EQE1 and EQE2 instead of MLE in the RM3 calculations. In other words, we linearly interpolate EQE1/EQE2 with RM1. As reported in Table 3, EQE1+RM1 outperforms RM3 in all collections, in terms of MAP, P@5, and P@10 (both methods achieve the same P@10 value in the AP collection). Except in two cases (MAP in GOV2 and P@10 in AP), EQE2+RM1 also outperforms RM3 in all collections, in terms of MAP, P@5, and P@10. In GOV2, RM3 performs better than EQE2+RM1, in terms of MAP. The reason is that EQE2 achieves lower P@10 compared to MLE in this collection (see Table 2) and since the feedback terms are extracted from the top retrieved documents, it leads to more accurate feedback models in



(a) (b) (c)

Figure 3: Sensitivity of MLE+ERM to the parameters α, β, and m (term count), in terms of MAP.



(a) EQE1 (b) EQE2 (c) ERM

Figure 4: Sensitivity of EQE1, EQE2, and ERM (MLE+ERM) to the dimension size of embedding vectors, in terms of MAP.

RM3. It is worth noting that RM3 does not perform well in terms of improving the precision of the top-retrieved documents in the GOV2 collection. In contrast, by employing EQE1/EQE2 instead of MLE in RM3, P@5 and P@10 values are substantially improved. The robustness index metric demonstrates that EQE1+RM1 is more robust than RM3 in the Robust and GOV2 collections. In AP, RM3 is slightly more robust than EQE1+RM1.

To evaluate the proposed ERM feedback method, we linearly interpolate it with MLE, EQE1, and EQE2. According to Table 3, MLE+ERM outperforms RM3 (MLE+RM1), in terms of MAP, P@5, P@10, and RI, in all collections (the same RI value achieved in the Robust collection). The MAP improvements of MLE+ERM over RM3 are always statistically significant. Similar to EQE1+RM1 and EQE2+RM1, MLE+ERM also achieves high P@5 and P@10 values compared to RM3, especially in GOV2. This shows the importance of capturing semantic similarities for the PRF task. Among all the considered feedback methods, EQE1+ERM outperforms all the other methods in AP and GOV2, in terms of MAP, P@5, and P@10. In Robust, EQE2+ERM performs well, in terms of MAP and P@10. This shows the effectiveness of the proposed embedding-based models for query reformulation. Note that as reported in [18], the RM3 method is a very robust PRF method, and the experiments show that RM3 is less robust than MLE+ERM, EQE1+ERM, and EQE2+ERM. This indicates the robustness of the proposed ERM method.

The next set of experiments focuses on studying the sensitivity of the proposed ERM (MLE+ERM) method to the three hyper-parameters α, β, and m. In all these experiments, we set α, β, and m to 0.5, 0.1, and 50. Then, in each step, we sweep one of these parameters. The results are shown in Figure 3. According to the plots in this figure, the behaviour of MLE+ERM in AP and Robust are similar to each other since they are both newswire collections with similar characteristics. Figure 3a plots the MAP values achieved by MLE+ERM with different α values. This parameter controls the influence of the original query model (MLE) in the final query language model. Based on this figure, the best value for the parameter α in both AP and Robust collections is 0.3. This means that higher weight should be given to the feedback language model generated by ERM, compared to MLE. In contrast, the best α value in GOV2 is 0.6. The reason is that although the P@10 values achieved by MLE in GOV2 are higher than those achieved in AP and Robust, the feedback language models in AP and Robust are more accurate than those in the GOV2 collection. An interesting observation in this figure is that the MAP values achieved in the Robust and GOV2 collections when $\alpha = 1$ are higher than those when $\alpha = 0$. In other words, the generated language model by ERM itself (without interpolating with MLE) is a better representation for the query than the original query language model.

Figure 3b plots the sensitivity of the proposed MLE+ERM method to the parameter β. According to this figure, the performance does not change drastically, when we sweep the value of the parameter β. The best MAP values are achieved when β is set to 0.1 (in AP and GOV2) or 0.2 (in Robust). As mentioned in Section 3.3, the parameter β controls the influence of term matching similarity vs. semantic similarity. Therefore, the results show that the term matching similarities are still more important than the semantic similarities in the relevance feedback.

According to Figure 3c, in newswire collections, we can have a very good query representation with a small number of words added to the query. In fact, by increasing the number of feedback terms, the performance slightly decreases. In the GOV2 collection, more feedback terms are needed to be added to the query, which is also similar to the query expan-

Table 4: The corpora used for training the embedding vectors.

ID	corpus	#tokens	#vocab.
Wiki	Wikipedia 2004 & Gigawords 5	6b	400k
Web 42b	Web crawl	42b	1.9m
Web 840b	Web crawl	840b	2.2m

Table 5: The MAP values achieved by EQE1, EQE2, and ERM (MLE+ERM) with different corpora for training the embedding vectors (dimension = 300).

Dataset	Method	Wiki	Web 42b	Web 840b
AP (146 queries)	EQE1	0.2402	0.2356	0.2362
	EQE2	0.2408	0.2352	0.2400
	ERM	0.3106	0.3094	0.3081
Robust (240 queries)	EQE1	0.2294	0.2255	0.2273
	EQE2	0.2271	0.2237	0.2266
	ERM	0.2713	0.2705	0.2683
GOV2 (146 queries)	EQE1	0.2745	0.2729	0.2767
	EQE2	0.2726	0.2713	0.2743
	ERM	0.3013	0.2989	0.3021

sion experiments (see Figure 2). The reason could be related to the characteristics of these collections and the amount of noise terms that the documents of each collection contain.

4.2.3 Sensitivity to the Embedding Vectors

In this section, we study the sensitivity of the proposed methods to the employed embedding vectors. We first analyze how sensitive the proposed methods are to the dimension of embedding vectors. Then, we study the performance of the proposed methods when the corpus that embedding vectors are trained on changes.

The plots in Figure 4 demonstrate the performance of EQE1, EQE2 and ERM (MLE+ERM) with respect to changes in the dimension of embedding vectors. All vectors are extracted from the same corpus (Wikipedia 2004 + Gigawords 5), with the same configuration. As shown in this figure, by increasing the dimension of embedding vectors, the performance of both EQE1 and EQE2 methods are increased in the AP and Robust collections. In contrast, the performance of these methods is not stable in the GOV2 collection. Note that these performance changes are minor (non-significant). According to Figure 4, the performance of ERM is not highly sensitive to the dimension of embedding vectors, especially in the Robust and GOV2 collections. In AP, by increasing the dimension of embedding vectors, the ERM performance is slightly improved, but these improvements are not statistically significant.

To analyze the robustness of the proposed methods to the choices made in training the word embedding vectors, we consider three external corpora: Wiki, Web 42b, and Web 840b. The Wiki corpus mostly contains articles with formal language; while the other two corpora are two web collections containing 42 and 840 billion tokens. The statistics of these corpora are reported in Table 4.[11] The dimension of embedding vectors extracted from these corpora is 300. We report the MAP of queries that all the embedding vector sets contain all of the query terms. The results are reported in Table 5. According to this table, the results are robust to the corpus that is used for training the word embedding vectors. There is no significant differences between the values obtained by employing different corpora for learning the embedding vectors. In the GOV2 collection, the Web 840b corpus seems to be slightly better than the other ones. Despite the large gap between the size of Wiki and the other two corpora, the results achieved by Wiki are higher than those obtained by the other ones, in the newswire collections.

[11]Embedding data is available at http://nlp.stanford.edu/projects/glove/.

Table 6: The MAP values achieved by EQE1, EQE2, and ERM (MLE+ERM) with and without the sigmoid transformation for computing the similarity of embedding vectors. The superscript * indicates significant differences.

Dataset	Method	EQE1	EQE2	ERM
AP	Cosine	0.2293	0.2366	0.3038
	Sigmoid	**0.2388***	**0.2391**	**0.3102***
Robust	Cosine	0.2247	0.2233	0.2677
	Sigmoid	**0.2292***	**0.2257**	**0.2711***
GOV2	Cosine	0.2709	0.2654	0.2971
	Sigmoid	**0.2745***	**0.2727***	**0.3005**

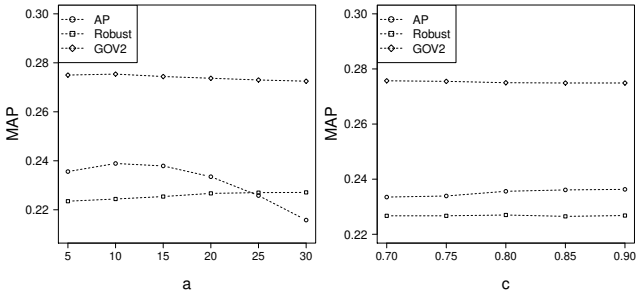


Figure 5: Performance of EQE1 with respect to the changes in the sigmoid parameter values.

4.2.4 Analysis of the Sigmoid Function

In this subsection, we study the behaviour of the proposed sigmoid transformation of the cosine similarity scores. To do this, we compare the proposed methods with two different similarity functions: the cosine similarity and its sigmoid transformation. The results are reported in Table 6.[12] According to this table, the results achieved by employing the sigmoid function are always higher than those obtained by the cosine similarity function. These improvements are statistically significant, in most cases.

To analyze the behaviour of the proposed EQE1 method[13] to the changes in the values of the sigmoid parameters (see Equation (1)), we fix one of these parameters ($a = 10$ and $c = 0.8$) and sweep the other one. The results are shown in Figure 5. According to these plots, the performance of EQE1 is sensitive to the value of parameter a. Therefore, a proper value for this parameter should be selected based on the retrieval collection. The best values for the parameter a are 10 for AP and GOV2, and 30 for Robust. Conversely, the performance is not very sensitive to the value of c. Note that we varied the value of c between $[0.7, 0.9]$. Based on our observations in Figure 1a, the selected interval is reasonable.

[12]For the sake of space, we just report the MAP values.
[13]The other proposed methods also behave similarly.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we first proposed two novel query expansion methods to estimate accurate query language models based on word embeddings. We further proposed the embedding-based relevance model, a pseudo-relevance feedback method based on word embeddings. The proposed methods use the semantic similarity of terms computed based on the similarity between the embedding vectors corresponding to the given terms. To obtain discriminative similarity scores that can be used in our methods, we transformed the cosine similarity scores by the sigmoid function.

We evaluated the proposed methods using three standard TREC newswire and web collections. The results indicated not only that the proposed methods significantly outperform the baselines in nearly all cases, but also they were shown to be more robust than the baselines. Studying the sensitivity of the proposed methods to the hyper-parameters showed that, in most cases, each proposed method behaves similarly in both newswire collections. The results also suggest that the sigmoid transformation of embedding similarities can significantly outperform the cosine similarity function.

Instead of employing the sigmoid function, an interesting future direction would be modifying the learning process of embedding vectors to produce discriminative similarity values that can be employed in many IR tasks. In addition, the theoretical analysis of employing sigmoid function for this purpose could be a possible future work. We also intend to study the use of word embeddings in other aspects of IR.

6. ACKNOWLEDGEMENTS

The authors thank Jiafeng Guo and Faegheh Hasibi for their invaluable comments. This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #CNS-0934322. 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.

7. REFERENCES

[1] N. Abdul-jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, D. Metzler, M. D. Smucker, T. Strohman, H. Turtle, and C. Wade. UMass at TREC 2004: Novelty and HARD. In *TREC '04*, 2004.

[2] M. ALMasri, C. Berrut, and J.-P. Chevallet. A Comparison of Deep Learning Based Query Expansion with Pseudo-Relevance Feedback and Mutual Information. In *ECIR '16*, pages 709–715, 2016.

[3] J. Bai, J.-Y. Nie, G. Cao, and H. Bouchard. Using Query Contexts in Information Retrieval. In *SIGIR '07*, pages 15–22, 2007.

[4] C. Carpineto and G. Romano. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, 2012.

[5] S. Clinchant and F. Perronnin. Aggregating Continuous Word Embeddings for Information Retrieval. In *CVSC@ACL '13*, pages 100–109, 2013.

[6] K. Collins-Thompson. Reducing the Risk of Query Expansion via Robust Constrained Optimization. In *CIKM '09*, pages 837–846, 2009.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[8] P. Dhillon, D. P. Foster, and L. H. Ungar. Multi-View Learning of Word Embeddings via CCA. In *NIPS '11*, pages 199–207, 2011.

[9] F. Diaz, B. Mitra, and N. Craswell. Query Expansion with Locally-Trained Word Embeddings. In *ACL '16*, 2016.

[10] D. Ganguly, D. Roy, M. Mitra, and G. J. Jones. Word Embedding Based Generalized Language Model for Information Retrieval. In *SIGIR '15*, pages 795–798, 2015.

[11] M. Karimzadehgan and C. Zhai. Estimation of Statistical Translation Models Based on Mutual Information for Ad Hoc Information Retrieval. In *SIGIR '10*, pages 323–330, 2010.

[12] T. Kenter and M. de Rijke. Short Text Similarity with Word Embeddings. In *CIKM '15*, pages 1411–1420, 2015.

[13] 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.

[14] J. Lafferty and C. Zhai. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *SIGIR '01*, pages 111–119, 2001.

[15] V. Lavrenko and W. B. Croft. Relevance Based Language Models. In *SIGIR '01*, pages 120–127, 2001.

[16] Q. V. Le and T. Mikolov. Distributed Representations of Sentences and Documents. In *ICML '14*, pages 1188–1196, 2014.

[17] O. Levy, Y. Goldberg, and I. Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *TACL*, 3:211–225, 2015.

[18] Y. Lv and C. Zhai. A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback. In *CIKM '09*, pages 1895–1898, 2009.

[19] 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.

[20] A. Montazeralghaem, H. Zamani, and A. Shakery. Axiomatic Analysis for Improving the Log-Logistic Feedback Model. In *SIGIR '16*, pages 765–768, 2016.

[21] J. Pennington, R. Socher, and C. Manning. GloVe: Global Vectors for Word Representation. In *EMNLP '14*, pages 1532–1543, 2014.

[22] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR '98*, pages 275–281, 1998.

[23] P. Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *IJCAI '95*, pages 448–453, 1995.

[24] J. J. Rocchio. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. 1971.

[25] A. Sordoni, Y. Bengio, and J.-Y. Nie. Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization. In *AAAI '14*, pages 1586–1592, 2014.

[26] E. M. Voorhees. Query Expansion Using Lexical-semantic Relations. In *SIGIR '94*, pages 61–69, 1994.

[27] I. Vulić and M.-F. Moens. Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. In *SIGIR '15*, pages 363–372, 2015.

[28] J. Xu and W. B. Croft. Query Expansion Using Local and Global Document Analysis. In *SIGIR '96*, pages 4–11, 1996.

[29] C. Zhai and J. Lafferty. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *CIKM '01*, pages 403–410, 2001.

[30] G. Zheng and J. Callan. Learning to Reweight Terms with Distributed Representations. In *SIGIR '15*, pages 575–584, 2015.

[31] G. Zhou, T. He, J. Zhao, and P. Hu. Learning Continuous Word Embedding with Metadata for Question Retrieval in Community Question Answering. In *ACL '15*, pages 250–259, 2015.

[32] G. Zuccon, B. Koopman, P. Bruza, and L. Azzopardi. Integrating and Evaluating Neural Word Embeddings in Information Retrieval. In *ADCS '15*, pages 12:1–12:8, 2015.