Semantic Matching by Non-Linear Word Transportation for Information Retrieval

Jiafeng Guo†, Yixing Fan†, Qingyao Ai‡, W. Bruce Croft‡
†CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
‡Center for Intelligent Information Retrieval, University of Massachusetts Amherst, MA, USA
guojiafeng@ict.ac.cn, fanyixing@software.ict.ac.cn, {aiqy,croft}@cs.umass.edu

ABSTRACT

A common limitation of many information retrieval (IR) models is that relevance scores are solely based on exact (i.e., syntactic) matching of words in queries and documents under the simple Bag-of-Words (BoW) representation. This not only leads to the well-known vocabulary mismatch problem, but also does not allow semantically related words to contribute to the relevance score. Recent advances in word embedding have shown that semantic representations for words can be efficiently learned by distributional models. A natural generalization is then to represent both queries and documents as Bag-of-Word-Embeddings (BoWE), which provides a better foundation for semantic matching than BoW. Based on this representation, we introduce a novel retrieval model by viewing the matching between queries and documents as a non-linear word transportation (NWT) problem. With this formulation, we define the capacity and profit of a transportation model designed for the IR task. We show that this transportation problem can be efficiently solved via pruning and indexing strategies. Experimental results on several representative benchmark datasets show that our model can outperform many state-of-the-art retrieval models as well as recently introduced word embedding-based models. We also conducted extensive experiments to analyze the effect of different settings on our semantic matching model.

Keywords

Word Embedding, Retrieval Model, Word Transportation

1. INTRODUCTION

Developing effective retrieval models is a central challenge in information retrieval (IR) research. Many different retrieval models have been proposed over the past decades, such as vector space models and probabilistic models [9]. Despite the differences in their theoretical foundations, most existing models are based on the Bag-of-Words (BoW) representation of queries and documents, where each word denotes a distinct dimension of a semantic space. This leads to a common limitation of many retrieval models that relevance scores are solely based on exact (i.e., syntactic) matching of words in queries and documents. Since it is unlikely that the authors of relevant documents always use exactly the same words as a user does in a query, retrieval models based on exact matching could face a significant *vocabulary mismatch* problem. Moreover, with the underlying independence assumption between words, these retrieval models do not allow semantically related words in a document to match the corresponding query words and contribute to the relevance score, leading to non-optimal retrieval performance.

Several techniques have been developed to support semantic matching in IR, such as query expansion [21, 20], latent models [10, 31] and translation models [3, 14]. Query expansion based on pseudo-relevance feedback (PRF) has been shown to improve search results in some cases, but is prone to the problem of query drift [6]. Latent models such as latent semantic indexing [10] and probabilistic topic models [31] circumvent this problem by representing both documents and queries in a latent space. However, these approaches alone often do not improve the empirical performance of traditional IR models due to the loss of many detailed matching signals over words [2]. Translation models attempt to estimate how likely a given document can be "translated" into a query by leveraging word dependency. The key difficulty of these models is how to formalize and estimate the translation probabilities between words [19, 15].

Recent advances in the natural language processing (NLP) community have shown that semantical representations of words can be efficiently acquired by distributional models [23, 26]. These representations, referred to as "word embeddings" or "word vectors", have been shown to improve the performance of a variety of NLP tasks [7]. These successes inspire us to consider IR based on a more general representation, namely Bag-of-Word-Embeddings (BoWE), where both query and document are represented as a bag of word vectors learned a priori to capture the semantic relations between them. There have been some recent attempts in IR to use BoWE representations. For example, Vulic et al. [30] constructed query and document representations by the weighted sum of word vectors for monlingual and bilingual retrieval. Clinchant et al. [5] generated document representations using the Fisher kernel framework. However, by representing documents as compact vectors, these methods suffer the same problem as latent models for IR, and cannot outperform traditional exact matching based retrieval models. Ganguly et al. [12] proposed a generalized

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA
© 2016 ACM. ISBN 978-1-4503-4073-1/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2983323.2983768

language model (GLM) based on word embeddings which showed improved performance compared with the language model (LM) approach. In fact, the GLM is an embedding based translation model linearly combined with a traditional language model. A similar idea was proposed in [33] where word embeddings were explored in a translation language model framework.

In this paper, we introduce a novel semantic matching based retrieval model based on the BoWE representation. We view the semantic matching between queries and documents as a non-linear word transportation (NWT) problem with fixed document capacity but non-fixed query capacity. We define the specific document word capacity and transportation profit based on some IR assumptions and constraints. We provide insight on the optimal solution of the non-linear transportation problem, and show the optimal solution can be efficiently approximated with neighborhood pruning and indexing strategies. We also discuss the connections and differences between the proposed model and several well-known semantic matching based retrieval models. Our model is inspired by success of the *Word Mover's Distance* (WMD) for document classification. We show why the non-linear formulation can better fit the IR problem than the original WMD.

We evaluate the effectiveness of our NWT model based on three benchmark retrieval datasets. The empirical results show that our model can outperform state-of-the-art retrieval models as well as recently introduced word embedding based models. We also conduct extensive experiments to study the effect of different settings on our NWT model, including word embedding variations, indexed neighbor size, and different objective formulations.

The major contributions of the paper includes:

1. We propose a new transportation view of semantic matching for IR, which is highly interpretable and well distinguished from existing retrieval models.

2. We introduce specific forms of capacity and profit in the transportation problem with respect to IR assumptions and constraints, and develop efficient algorithms for practical computation.

3. We conduct extensive experiments to demonstrate the effectiveness of the new model by comparing with state-of-the-art retrieval models with detailed analysis.

2. RELATED WORK

In this section, we briefly review three research areas related to our work: semantic matching in IR, word embedding based retrieval models and transportation problems.

2.1 Semantic Matching in IR

Many semantic matching techniques have been proposed to tackle the vocabulary mismatch problem between queries and documents, including query expansion, latent models and translation models.

In automatic query expansion, a query is expanded using words or phrases with similar meaning to those in the query and the chances of matching words in relevant documents are therefore increased. Methods in this line split into two major classes: global methods [21] and local methods [20]. Global methods expand or reformulate query words by analyzing the word co-occurrences from the corpus being searched or using hand-crafted thesaurus. Local methods, on the other hand, adjust a query based on the top ranked documents retrieved by the original query. Although query expansion using local analysis has shown to improve search results in some cases, it is prone to the problem of query drift [6].

Latent models such as latent semantic indexing [10] have also been proposed for dealing with the vocabulary mismatch problem. The idea is to represent both the document and query in a latent space of reduced dimensionality, and to compute the retrieval score based on these representations. However, these approaches alone often do not improve the empirical performance of traditional IR models due to the loss of many detailed matching signals over words [2]. It is thus necessary to combine such latent models with standard IR approaches to observe effectiveness improvements. For example, Wei et al. [31] proposed an LDA-based document model within the language modeling framework.

Translation models [3, 14] incorporate word relationships into language modeling approaches by viewing the matching from documents to queries as machine translation. A key difficulty in translation models is how to formalize and estimate the translation probability. For example, Berger et al. [3] proposed two models of document-query translation, i.e., a mixture model and a binomial model, based on different document-query translation processes. The probability of translating a document word to a query word is estimated based on synthetic training data. Jin et al. [14] considered a document title as a possible query, and use the title-document pairs to train the translation model. Karimzadehgan et al. [15] addressed this estimation problem based on normalized mutual information between words, which is less computationally expensive and has better coverage of query words than the synthetic query method of estimation [3].

2.2 Word Embedding and Embedding based Retrieval Models

Embedding words as vectors in a relatively low-dimensional space goes back several decades in linguistics [10]. The learned word embeddings, which can better capture semantic and syntactic information of words, can be used as basic features in a variety of applications, such as named entity recognition, question answering, disambiguation, and parsing [7]. While word embeddings have been proven to be useful in a variety of NLP tasks in recent years, their potential in IR needs to be further explored.

Recently, there have been some attempts to use word embeddings for retrieval tasks. For exmaple, Vulic et al. [30] studied both monolingual and bilingual retrieval models using word embeddings. In their work, they represent both query and document as compact vectors by simple (weighted) sum of the word embeddings. The proposed model does not improve on the traditional language model in the monolingual retrieval task. The best results are obtained by the combination of the word embedding based method and a unigram language model. Recently, Ganguly et al. [12] proposed a word embedding based generalized language model (GLM) for IR. In their work, words in a query are assumed to be generated independently in three possible ways, i.e., direct term sampling, transformation via document sampling, and transformation via collection sampling. The final GLM is a combination of the three events. The empirical results show that GLM can perform better than the traditional language model. However, it is not difficult to show that GLM is inherently an embedding based translation model linearly combined with a traditional language model, since the direct term sampling is a standard language model. Similarly, Zuccon et al. [33] leveraged word embeddings for the estimation of translation probability between words, and combined the neural translation language model with collection background probabilities using the Dirichlet smoothing strategy.

All the existing work shows that employing word embedding for IR can improve retrieval effectiveness. However, this is achieved by linearly combining an embedding based model with traditional retrieval models.

2.3 Transportation Problems

The transportation problem [13] is a typical linear programming problem that has been extensively studied in mathematics and economics, and widely applied in urban planning and civil engineering. Beyond these original applications, Rubner et al. [28] introduced the transportation problem into computer vision to derive a distance metric between two distributions, namely *Earth Mover's Distance* (EMD), which has been used successfully in many applications such as image retrieval, gesture recognition, and multimedia search [25]. Following the idea of EMD, Kusner et al. [18] introduced Word Mover's Distance (WDM) by formulating a transportation problem between two documents based on word embedding representations, and demonstrated its effectiveness in document classification. Our work is inspired by WMD. However, we show that the formulation of the transportation problem in IR is significantly different from that in document classification.

3. BoWE REPRESENTATION FOR IR

Many traditional IR models are based on the BoW representation, where both query and document are represented as a bag (multiset) of words, and each word is treated as independent from others. We can view each word as a "one-hot" vector, which represents a distinct dimension of a semantic space, and both query and document are represented as a point in such a mutually orthogonal word space. Obviously, this simplified representation cannot properly capture the semantic similarity between words, e.g., "car" and "auto", which leads to the vocabulary mismatch problem and non-optimal retrieval performance.

Rather than the sparse one-hot representation of words, a better way to capture the semantic relations between words is to map each word to a low-dimensional continuous vector where similar words are close to each other [23, 26]. In this way, we can extend the BoW representation in IR into a more general setting, namely Bag-of-Word-Embedding (BoWE), by dropping the word independence assumption.

Formally, suppose we are given a word embedding matrix $W \in \mathbb{R}^{K\times|V|}$ for a finite size vocabulary of $|V|$ words, where the i-th column, $w_i \in \mathbb{R}^K$, denotes the embedding of the i-th word in the K-dimensional space. We assume both query and document are represented as a bag (multiset) of word vectors. We denote the document as $D = \{(w_1^{(d)}, tf_1), \dots, (w_m^{(d)}, tf_m)\}$ where tf_i denotes the frequency of i-th word in the document, and similarly the query as $Q = \{(w_1^{(q)}, qtf_1), \dots, (w_n^{(q)}, qtf_n)\}$ where qtf_j denotes the frequency of j-th word in the query. Obviously, this is a basic formulation of BoWE by simply using frequency as the weighting scheme. Other word weighting schemes could be used in BoWE, e.g., tf-idf.

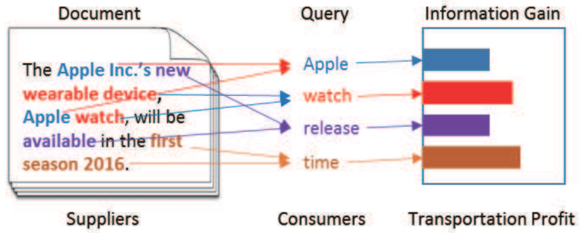
Document
Query
Information Gain
The Apple Inc.'s new wearable device, Apple watch, will be available in the first season 2016.
Apple
watch
release
time
Suppliers
Consumers
Transportation Profit

Figure 1: Semantic Matching as Word Transportation.

With word embeddings as the basic building blocks, the BoWE provides a richer representation of queries and documents and thus a good foundation for developing the following semantic matching based retrieval models.

4. SEMANTIC MATCHING AS NON-LINEAR WORD TRANSPORTATION

Information retrieval is the activity of obtaining documents relevant to the information need of a query from a data collection, where the relevance is mainly revealed by the semantic matching between documents and queries. Intuitively, if a document contains more words that can match the query words, either exactly or semantically, it might be more relevant. Inspired by the success of the WMD in computing semantic distance between documents from a transportation view, we propose to formulate the sematic matching between documents and queries as a transportation problem by viewing document words as "suppliers", query words as "consumers" and information as "product" respectively, as illustrated in Figure 1.

Specifically, we have the following assumptions for the transportation problem in the IR scenario: (1) Each document word has fixed information capacity based on its occurrences, while each query word has unlimited capacity to accommodate as much relevant information as possible from the document; (2) The information gain (i.e., perceived semantic relevance information) of transporting (i.e., matching) a document word to a query word decides the transportation "profit"; (3) The total profit on each query word should obey the law of diminishing marginal returns, in the sense that the gain of transporting document words to one query word would decrease with the total amount. Note that in assumption (1) we do not assume fixed capacity on query words due to the vague nature of query intent. It is difficult to know or define how much information each query word needs a priori. Assumption (3) is inspired by previous findings in exact matching based retrieval models, where term weighting is usually defined to have a damping effect [27]. Finally, the target of IR is to find the documents that can bring the maximum net returns for a given query.

Based on the above idea, we formulate the semantic matching between document and query in IR as the following non-linear word transportation problem. Given a query and a document with BoWE representations, one aims to find a set of optimal flows $F = \{f_{ij}\}$ that satisfy

$$\begin{aligned} \max \quad & \sum_{j \in Q} \log \sum_{i \in D} f_{ij} r_{ij} && (1) \\ \text{subject to :} \quad & f_{ij} \geq 0 \qquad \forall i \in D, \forall j \in Q \\ & \sum_{j \in Q} f_{ij} = c_i \quad \forall i \in D \end{aligned}$$

where f_{ij} denotes how much capacity of the i-th document word flows to the j-th query word, r_{ij} denotes the corresponding transportation profit, and c_i denotes the information capacity of the i-th document word. The remaining problem is how to define the document word capacity and transportation profit.

Document Word Capacity. A straightforward choice for the information capacity of a document word is its frequency in the document. However, since documents usually have varied lengths in IR, this will make the transportation solution strongly biased towards long documents. It is, therefore, natural to perform length normalization to make the total capacity of each document as a unit constant. In this way, we have the document word capacity defined as

$$c_i = \frac{tf_i}{|D|}$$

However, such simple normalization would over-estimate the importance of word occurrences, leading to high capacity for words occurring in short documents. A standard way to solve this problem in language modeling is to introduce the smoothing factor. Here we adopt the well-known Bayesian smoothing using Dirichlet priors, and define the document word capacity as

$$c_i = \frac{tf_i + \mu \frac{cf_i}{|C|}}{|D| + \mu} \qquad (2)$$

where cf_i denotes the corpus frequency of the i-th document word, $|C|$ denotes the total size of the corpus, and μ is the smoothing parameter. In this way, each document can be viewed as a distribution over the entire vocabulary with the total capacity to be unit.

Transportation Profit. Transportation profit refers to the information gain when we transport (i.e., match) a document word to a query word. A straightforward idea is to define it as the semantic closeness between two words. A widely adopted measure for the semantic closeness between word embeddings is the cosine similarity [23, 26].

$$r_{ij} = \widehat{cos}(w_i^{(d)}, w_j^{(q)}) = \max\left(cos(w_i^{(d)}, w_j^{(q)}), 0\right) \quad \forall i \in D, \forall j \in Q$$

Here we use the truncated cosine similarity $\widehat{cos}(w_i^{(d)}, w_j^{(q)})$ to avoid negative profit, which is inappropriate in a transportation problem. This also means that if two words are too distant in the semantic space, the transportation profit between them could be neglected.

However, simply using cosine similarity as the transportation profit would over-emphasize the importance of semantic matching. For example, the cosine similarity between "salmon" and "fish" is about 0.72 in Google-released word vectors. Given a query "salmon", a document containing two "fish" words would easily accumulate higher profit than the document containing one "salmon" word and some other word. This is not appropriate in IR since matching a query word exactly should always contribute more to the relevance score than matching a semantically related word multiple times [11]. One way to solve this problem is to introduce a *matching risk* parameter to control the profit gap between exact matching and semantic matching.

$$r_{ij} = \widehat{cos}(w_i^{(d)}, w_j^{(q)})^{\alpha}$$

where α is the risk parameter. The higher the risk parameter, the less profit the transportation can bring. However, an arbitrarily high risk parameter would make the transportation model degrade to the exact matching case, losing the power of semantic matching in IR.

In our analysis, we find that the risk parameter should not be a fixed value but query word dependent. If the query word is discriminative, like "salmon", the risk of matching semantically related words like "fish" should be high. On the other hand, if the query word is more general, like "war", there is less risk in matching semantically related document words like "conflict" and "warfare". Since the inverted document frequency (IDF) is a strong signal of the discriminative power of a word, here we simply define the risk parameter as a function of IDF and obtain the following profit definition.

$$r_{ij} = \widehat{cos}(w_i^{(d)}, w_j^{(q)})^{g(idf_j)} \qquad \forall i \in D, \forall j \in Q$$

where

$$g(idf_j) = idf_j + b, \quad idf_j = \frac{N - df_j + 0.5}{df_j + 0.5}$$

Here df_j denotes the document frequency of the j-th query word, N denotes the total number of documents in the corpus, and b is a free parameter denoting the default offset of the risk.

Model Summary. We have introduced the specific definition of our non-linear word transportation model. We can see that the model is highly interpretable and can capture both the exact and semantic matching signals in a unified framework rather than a simple linear combination as in previous work [30, 12]. In fact, our problem formulation provides a very general view of semantic matching in IR, in the sense that one may design a number of model variations under this view. For instance, in our work, we use the logarithm function to model the diminishing return effect. One may employ other functions with damping effect to achieve the same purpose, e.g., sigmoid functions. For the document word capacity, we use a Bayesian smoothing function to ensure the total capacity of each document is fixed and to avoid bias towards either long or short documents. Obviously, other weighting and smoothing schemes appropriate for IR could be utilized here. The transportation profit could also be defined flexibly. For example, one may employ Gaussian kernel functions to compute the similarity based on word embeddings. Many other features could be involved to define the matching risk, e,g., Park et al. [24] proposed a set of features to find key concepts in queries (i.e., high matching risk part) and use the non-key parts in a translation model for retrieval. We will leave the exploration of these variations in our future work.

4.1 Efficient Solution

The objective of the non-linear word transportation problem is a concave maximization problem which can be directly solved by convex optimization approaches. However, according to the definition of the word capacity, the supplier is actually the entire vocabulary. If we view the word transportation problem as a network flow problem on a bipartite graph, we will have $|V|$ nodes on the document side and $|Q|$ nodes on the query side, and the total edge number is $|V| \times |Q|$[1]. This will make the computational cost prohibitive. In fact, users would not attempt to match all the words in a document to query words when judging relevance. They usually only perceive highly related document words as useful matching signals for the query. Inspired by this, we propose to significantly reduce the computational complexity by pruning the document nodes and corresponding edges if the document words are too distant from the query words. Such a pruning method has also been proposed in previous work [25] to produce fast and robust transportation solutions.

To achieve this, we first construct a k-nearest-neighbor indexing for each word in the vocabulary in an offline step. Specifically, for each word, we rank all the other words in the vocabulary according to the transportation profit. Only the top k ranked words are kept and indexed for fast access. In online computing, given a query and a candidate document, the k-nearest-neighbors of each query word are aggregated to form a pruned representation of the document, with the capacity defined by the document specific weighting as shown in Equation (2). We then solve the non-linear transportation problem between the pruned document and the query. As we can see, the total document node number would be less than $k|Q|$ and the edge number would be less than $k|Q|^2$ in the bipartite graph, where $|Q|$ is typically very small in IR and k is a predefined parameter (ranging from tens to hundreds in our work) to ensure most closely related words can be captured. This leads to a very efficient solution for the transportation problem that would be practical for online ranking.

5. MODEL DISCUSSION

In this section, we will discuss the properties of our model solution, as well as the connections and differences between the proposed model and several existing retrieval models.

5.1 Insight on Model Solution

We first analyze the solution of our non-linear transportation problem to get a better understanding how our model generates the relevance scores. To simplify, we first look at the linear problem by removing the logarithm from the objective function. As we can see, the formulation turns into a relaxed transportation problem since we only have the constraints on the supplier side. The optimization of this relaxed transportation problem is straightforward. The optimal solution is for each document word to move all its capacity to the query word with the largest transportation profit. An intuitive explanation of this optimal solution is semantic word alignment, where each document word is aligned to the best matching query word. In this way, the solution is similar to other alignment approaches such as in machine translation [4] or semantic text similarity [16]. However, the limitation of the linear formulation is that there is no preference between the matching of multiple query words and the matching of a single query word, making it not appropriate for IR [11].

By adding logarithm into the objective function, we actually introduce a damping effect on the total transportation profit of each query word. In this way, the optimal solution is no longer the simple word alignment, since the gain of transporting a document word to a query word will decrease with the accumulated profit, preventing over-matching on a single query word. An alternative view is that the non-linear formulation will try to interpret all the query words as well as possible. Therefore, a document that can interpret more distinct query words would be assigned a higher score, which is an expected property for IR models [11].

Overall, the word alignment effect due to the relaxation of constraints on the query side and the marginal diminishing effect due to the non-linear formulation are two key properties of our model, making it unique and useful for IR.

5.2 Relationship with Sematic Matching based Retrieval models

Statistical Translation models. Although the transportation between words bears a similarity to the translation process, there are clear differences between our model and statistical translation models for IR. The translation models for IR are usually formalized under the probabilistic framework, where the relevance score is defined by the probability that the query would have been generated as a translation of the document. Instead our model formalizes semantic matching as a transportation problem, and the relevance score is obtained by optimizing a non-linear objective. In our model, there is no requirement for the transportation profit to be a probability, which brings more flexibility and the potential to involve multiple features in estimation.

Query Expansion. Our model shares some similar ideas with global analysis based query expansion methods but works in different ways. In global methods, word association relationships are derived from the entire corpus, and the neighbors of query words are used to enrich the representation of the original query for exact matching. In our work, word representations are learned from co-occurrence patterns with advanced embedding technology, and the neighbors of query words are identified in the documents as the information suppliers for transportation. The idea of query expansion methods with local analysis is different from ours. In these local methods, the word relationships are obtained from a query dependent local context, i.e., the top ranked results retrieved by the original query. In this sense, query expansion methods with local analysis are orthogonal to our work, and could be used as an extension to enrich the query representation in our transportation framework.

Latent Models. The major difference between previous latent models and our model lies in the representation of the document. Previous latent models represent the document as a compact vector, thus losing many detailed matching signals over words. Our model represents the document as a bag of word embeddings, and both exact and semantic matching signals over words can be captured during the transportation process.

5.3 Relationship with Word Mover's Distance

Our model is inspired by the WMD proposed by Kusner et al. [18]. However, the model formulations are significantly different due to the differences of the problems addressed.

WMD aims to measure the dissimilarity between text documents. It formulates the distance as a minimum cost that

[1] In fact, the number of valid edges would be much smaller than $|V| \times |Q|$ since all the edges with zero transportation profit (due to truncated cosine similarity) can be removed.

one document need to take to exactly match the other, which is a standard linear transportation problem. Since it models a pair of objects with the same type (i.e., documents), symmetric constraints are applied on both the supplier and consumer sides. The obtained WMD is a symmetric metric to ensure that the distance from document A to document B is the same as the distance from document B to document A. Obviously, with WMD, the most similar document for any document is itself.

In contrast to WMD, our work aims to model the semantic relevance between queries and documents in IR. Semantic relevance is mainly revealed by the similarity signals. Therefore, we formulate the problem as a maximization type transportation problem based on profits (i.e., similarity signals). In this transportation problem, queries and documents are two different types of objects. Queries are usually short and vague in intent, while documents are usually long and clear in meaning. Therefore, we relax the constraints on the query side to allow each query word to accommodate as much relevant information as possible. By our definition, the obtained semantic relevance will not be a distance metric, which means it does not satisfy the identity of indiscernibles property. This is very reasonable in IR since the most relevant document to a given query should not be the query itself (i.e., ranking exactly the same query string as the most relevant result is pointless in search).

6. EXPERIMENTS

In this section, we conduct empirical experiments to test the effectiveness of our proposed retrieval model. We first introduce the experimental settings, including the datasets, word embeddings, evaluation measures and baseline methods. We then compare our model with state-of-the-art retrieval models. Finally, we analyze the effect of different settings on the proposed model.

6.1 Experimental Settings

Datasets. To conduct experiments, we use three TREC collection, i.e., Robust04, GOV2, and ClueWeb-09-Cat-B. The details of these collections are provided in Table 1. As we can see, they represent different sizes and genres of heterogeneous text collections. Robust04 is a small news dataset. Its topics are collected from TREC Robust Track 2004. Both GOV2 and ClueWeb-09-Cat-B are large Web collections, where ClueWeb-09-Cat-B is filtered to the set of documents with spam scores in the 60^{th} percentile, using the Waterloo Fusion spam scores [8]. The GOV2 topics are accumulated over TREC Terabyte Tracks 2004, 2005, and 2006. The Clue-Web-09-Cat-B topics are accumulated from TREC Web Tracks 2009, 2010, and 2011. For all the datasets, we made use of both the title and the description of each TREC topic in our experiments. The retrieval experiments described in this section are implemented using the Galago Search Engine[2]. During indexing and retrieval, both documents and query words are stemmed using the Krovetz stemmer [17]. Stopword removal is performed on query words during retrieval using the INQUERY stop list.

Word embeddings. We adopt both corpus-specific and corpus-independent word embeddings in our experiments. For corpus-specific embeddings, we train word vectors using both the Continuous Bag-of-Words (CBOW) Model and the Skip-Gram (SG) model [23] on Robust04, GOV2, and Clueweb-09-Cat-B collections, respectively. Specifically, we used 10 as the context window size and used 10 negative samples and a subsampling of frequent words with sampling threshold of 10^{-4} as suggested by Word2Vec[4]. Each corpus was pre-processed by removing HTML tags and stemming. We also discarded from the vocabulary all words that occurred less than 10 times in the corpus, which resulted in a vocabulary of size 0.1M, 1.9M, and 4.1M on the Robust04, GOV2, and Clueweb-09-Cat-B collections, respectively. For corpus-independent embeddings, we make use of two publicly available sets of word vectors. The first one is the pre-trained 300-dimensional vectors released by Google, which is trained on a Google News corpus of about 100 billion words[3]. The second set is the pre-trained 300-dimensional Glove word embeddings, which is trained on Wikipedia and Gigaword of about 6 billion tokens[4]. For corpus-independent embeddings, we tried both original word vectors and stemmed word vectors[5], and found there is no significant performance difference between these two settings.

Table 1: Statistics of the TREC collections used in this study. The ClueWeb-09-Cat-B collection has been filtered to the set of documents in the 60^{th} percentile of spam scores.

	Robust04	GOV2	ClueWeb-09-Cat-B
Vocabulary	0.6M	35M	38M
Document Count	0.5M	25M	34M
Collection Length	252M	22B	26B
Query Count	250	150	150

One thing we need to address in using word embeddings for semantic matching is the out-of-vocabulary (OOV) words. One way to deal with these OOV words is to simply ignore them. However, if these words are within the queries, important semantic information may be lost. For example, the numbers or rare words in a query, which are typically not learned in word embedding, often convey very specific information for matching. Therefore, following the practice in previous work [16], we map all the OOV words to random vectors, while remembering which OOV word maps to which random vector. In this way, if the same OOV word is observed in both query and document, the transportation takes place and contributes to the final score. Otherwise, no transportation will be conducted over the OOV words.

Evaluation measures. Given the limited number of queries for each collection, we conduct 5-fold cross-validation to minimize over-fitting without reducing the number of learning instances. Topics for each collection are randomly divided into 5 folds. The parameters for each model are tuned on 4-of-5 folds. The final fold in each case is used to evaluate the optimal parameters. This process is repeated 5 times, once for each fold. Mean average precision (MAP) is the optimized metric for all retrieval models. Throughout this paper each displayed evaluation statistic is the average of the five fold-level evaluation values. For evaluation, The top-ranked $1,000$ documents are compared using the mean average precision (MAP), normalized discounted cu-

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

[3] https://code.google.com/p/word2vec/

[4] http://nlp.stanford.edu/projects/glove/

[5] For stemmed word vectors, we applied Krovetz stemmer over the words in Google or Glove vocabulary, and add word vectors whose words have the same stemmed form.

mulative gain at rank 20 (nDCG@20), and precision at rank 20 (P@20). Statistical differences between models are computed using the Fisher randomization test [29] ($\alpha = 0.05$).

Baseline methods. We adopt both exact matching based and semantic matching based models for comparison. Models based on exact matching are as follows.

QL: Query likelihood model based on Dirichlet smoothing [32] is one of the best performing language models.

BM25: The BM25 formula [27] is another highly effective retrieval model that represents the classical probabilistic retrieval model.

SDM: SDM is a state-of-the-art language model addressing term dependence using Markov random fields [22].

For the semantic matching based retrieval models, we consider the following approaches.

RM3: One of the representative PRF models under language modeling framework is the Relevance Model (RM) [20]. Relevant expansion words are extracted and used in combination with the original query (the RM3 variant).

LM+LDA: For latent models, we adopt the LDA-based document model within the language modeling framework introduced in [31]. The LDA model is trained on the corresponding collection using 300 topics for fair comparison.

LM+WE-VS: A linear combination of word embedding based retrieval model and unigram language model was introduced in [30]. The embedding based ranking function is simply the cosine similarity between the document and query representations, which are constructed by aggregating the corresponding word vectors with weights or not.

WE-GLM: A word embedding based translation model for IR was proposed in [12], where query words are assumed to be generated in three ways, i.e., direct term sampling, transformation via document sampling, and transformation via collection sampling.

NWT: We refer to our proposed non-linear word transportation model as NWT.

Note that the parameters for each model are tuned with the 5-fold cross-validation method mentioned above using the typical ranges suggested by the original papers [32, 27, 1, 22, 20, 31, 30, 12]. For our NWT model, there are three free parameters, i.e., smoothing parameter μ in the document word capacity, the offset b in the risk function and the neighbor size k in offline indexing. We tune μ ($100 < u < 2000$) and b ($0 < b < 3$) using the same cross validation process while k ($20 < k < 200$) is pre-defined empirically according to our analysis in Section 6.3.2. For the LDA-based and word embedding-based retrieval models, we adopt a re-ranking strategy for efficient computation. An initial retrieval is performed using the QL model to obtain the top $2,000$ ranked documents. We then use the LDA-based model and word embedding-based models to re-rank these top results. The top-ranked $1,000$ documents are then used for comparison.

6.2 Retrieval Performance and Analysis

This section presents the performance results of different retrieval models over the three datasets. A summary of results is displayed in Table 2. As compared with the simple exact matching based models, including QL and BM25, our NWT model can outperform them on nearly every dataset under different evaluation metrics. Taking the Robust04 collection as an example, the relative improvement of our model over the QL model under MAP is about 8.3% and 8.9% using titles and descriptions, respectively. The results indicate that by properly involving semantic matching information, one can improve the retrieval performance consistently. We can also find that the improvements on descriptions are usually larger than that on titles across different collections, indicating that semantic matching is more beneficial for longer queries with richer content.

SDM is a special exact matching based model which takes into account the word dependency (n-grams) in ranking, making it a very strong baseline. Although it is not fair to directly compare our model with SDM since our NWT model is based on BoWE (unigrams), we use SDM in the comparison to see how close a unigram based semantic matching model can be to a state-of-the-art baseline. Interestingly, our NWT model not only achieves similar performance in many cases, but also improves on SDM on some datasets, e.g. NWT outperforms SDM in terms of all the metrics on Robust04 and on ClueWeb-09-Cat-B using topic descriptions. These results further demonstrate the effectiveness of our semantic matching based model.

When considering the semantic matching based baselines, we first see that RM3, the pseudo relevance feedback model, is in general an effective model. However, while RM3 can improve the performance of the whole ranking list (i.e., MAP) as compared with the simple exact matching based models, it may hurt the performance of the top ranked results (i.e., NDCG@20 and P@20) on some Web collections. When comparing RM3 with our NWT model, we can see that our model can outperform RM3 consistently except on the Robust04 collection using topic titles. We note that the improvements of RM3 and NWT over simple exact matching based models are much more significant on the Robust04 collection than the other two collections. It shows that semantic matching is more effective on clean news data than on noisy Web data.

As for the latent model, here we only report the results for the LM+LDA model on the Robust04 collection due to the prohibitive training time of LDA on the other two collections. However, for the other two collections we can take LM+WE-VS as a proxy for LM+LDA since they both describe the document using a compact representation and LM+WE-VS has been shown comparable to or more effective than LM+LDA in both our work and previous work [30]. By using the topic model as an additional language model estimation, LM+LDA can obtain better results than simple QL model on the Robust04 collection, but the improvements are very limited especially on topic descriptions. Comparing with the NWT model, we can see that our model can consistently outperform LM+LDA model under all the metrics.

Finally, we take a look at the existing word embedding based models. Overall, we can see that very limited improvements can be obtained by both LM+WE-VS and WE-GLM models as compared with simple exact matching based models, and our NWT model can outperform both models consistently over the three collections under all the evaluation metrics. For LM+WE-VS, the results demonstrate that using a compact representation of documents may lose many detailed semantic matching signals, leading to limited performance improvements. GLM, as an embedding based translation model, tries to explain each query word from all the document words within a probabilistic framework. In NWT, query words are mainly explained by the most related document words via transportation without any strict

Table 2: Comparison of different retrieval models over the Robust-04, GOV2, and Clueweb-09-Cat-B collections. Significant improvement or degradation with respect to NWT is indicated (+/-) (p-$value \leq 0.05$).

Robust-04 collection

		Topic titles			Topic descriptions		
Model Type	Model Name	MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
Exact Matching Baselines	QL	0.253$^-$	0.415$^-$	0.369$^-$	0.246$^-$	0.391$^-$	0.334$^-$
	BM25	0.255$^-$	0.418	0.370	0.241$^-$	0.399$^-$	0.337$^-$
	SDM	0.263	0.423	0.375	0.261	0.409	0.349
Semantic Matching Baselines	RM3	0.295$^+$	0.423	0.375	0.264	0.387$^-$	0.345
	LM+LDA	0.258$^-$	0.421	0.374	0.247$^-$	0.392$^-$	0.336$^-$
	LM+WE-VS	0.255$^-$	0.417$^-$	0.370$^-$	0.253$^-$	0.401$^-$	0.341$^-$
	WE-GLM	0.255$^-$	0.417	0.371	0.252$^-$	0.400$^-$	0.340$^-$
Our Approach	NWT	0.274	0.426	0.380	0.268	0.413	0.353

GOV2 collection

		Topic titles			Topic descriptions		
Model Type	Model Name	MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
Exact Matching Baselines	QL	0.295$^-$	0.409$^-$	0.510$^-$	0.249$^-$	0.371$^-$	0.470$^-$
	BM25	0.295	0.421	0.523	0.256$^-$	0.394	0.483
	SDM	0.319$^+$	0.441$^+$	0.549$^+$	0.275	0.411	0.512$^+$
Semantic Matching Baselines	RM3	0.301	0.395$^-$	0.512	0.263$^-$	0.372$^-$	0.476
	LM+WE-VS	0.295$^-$	0.408$^-$	0.509$^-$	0.254$^-$	0.382$^-$	0.474$^-$
	WE-GLM	0.299$^-$	0.411$^-$	0.513	0.253$^-$	0.384$^-$	0.478
Our Approach	NWT	0.304	0.422	0.524	0.274	0.404	0.492

Clueweb-09-Cat-B collection

		Topic titles			Topic descriptions		
Model Type	Model Name	MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
Exact Matching Baselines	QL	0.100$^-$	0.224$^-$	0.328	0.075$^-$	0.183$^-$	0.234$^-$
	BM25	0.101	0.225	0.326	0.080	0.196	0.255
	SDM	0.109	0.242	0.351	0.079	0.193	0.243
Semantic Matching Baselines	RM3	0.103	0.224	0.323	0.074	0.182$^-$	0.230$^-$
	LM+WE-VS	0.101$^-$	0.225$^-$	0.331	0.075$^-$	0.187$^-$	0.240$^-$
	WE-GLM	0.102$^-$	0.228$^-$	0.335	0.075$^-$	0.187$^-$	0.241$^-$
Our Approach	NWT	0.107	0.236	0.341	0.080	0.204	0.264

probabilistic constraints. Comparing GLM with NWT, the results show that the alignment affect and the flexibility in model definition in NWT can bring more benefits for semantic matching in IR.

6.2.1 *Case Studies*

We further conducted some case studies to analyze the limitation of our model. We analyze the queries on which there is a performance drop in our NWT model as compared with baseline methods. There are two major findings.

Firstly, the performance of our NWT model sometimes decreases on queries with named entities. For example, given the query "brazil america relation", the NWT model will try to match words like "argentina" and "spain" for "brazil", and words like "europe" and "africa" for "america" since these words are some of the nearest neighbors of the query words. Matches such as these may hurt the performance since a non-relevant document talks about spain and europe relation might be ranked highly. Similar cases can also be found on queries with person names, such as "rick warren". Since there are more queries containing such named entities on Gov2 and Clueweb-09-Cat-B (around 30%) than on Robust04 (about 19%), this explains why the improvements of the NWT model would be smaller on these two collections. Based on the above observation, we tried one simple experiment by treating named entity words as OOV words so that only exact matching can be conducted over these words. We find notable improvements on GOV2 and Clueweb-09-Cat-B collections, e.g., NWT can achieve 0.305 and 0.276 in terms of MAP on GOV2 with topic titles and descriptions, respectively. In fact, other than treating named entities as OOV words, a better way to solve this problem might be to use more features (e.g., word type) beyond IDF to define the risk parameter in the transportation profit. We will leave this for the future study.

Secondly, our NWT model may not work well on some queries containing short and ambiguous acronyms. This kind of acronym is more popular in the TREC Web Track queries on Clueweb-09-Cat-B collection. For example, given the query "Find information on taking the SAT college entrance exam", the closest words for the query word "SAT" are "fri", "tue" and "wed", which is clearly not related to the exam meaning of the word. One way to solve this problem may be to take the contexts of the query word into account to decide the transportation profit between document words and query words.

6.3 Analysis on NWT model

We conducted extensive experiments to study the effect of different settings on our model. Through these experiments, we try to gain a better understanding of our model.

6.3.1 *Impact of Word Embeddings*

Since our model is based on the BoWE representation, the quality of word embeddings is definitely an important factor. We studied the effect of different types of word em-

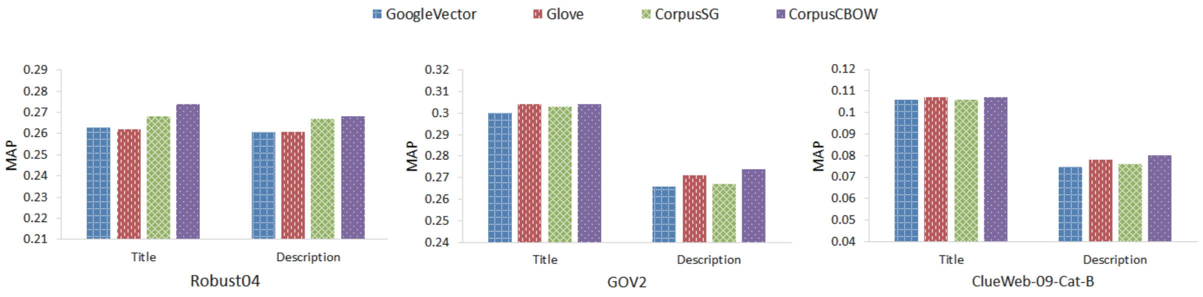

Figure 2: Comparison of different word embeddings in NWT over the three collections under MAP.

beddings on the retrieval performance. We report the MAP results over the three collections using different word embeddings, both corpus-specific (denoted as CorpusSG and CorpusCBOW) and corpus-independent (denoted as GoogleVector and Glove), in Figure 2. From the results we can see that, in general the models based on corpus-specific word embeddings are better than that using corpus-indenpendent word embeddings. The difference is more obvious on the smaller Robust04 collection than the other two collections. A possible reason is that the word embeddings learned over a specific collection can better capture the meaning of the words in that corpus, leading to more corpus-specific neighbors as compared with those learned from an independent dataset. For example, the top 3 nearest neighbors of "smuggle" are "contraband", "illegal" and "counterfeit" in learned embeddings from Robust04 collection, and "traffick", "launder" and "sneak" in Google vectors[6]. Such corpus-specific neighbors in turn may better capture the semantic matching signals, and bring more benefits to the retrieval performance. As for the two word embedding methods, CBOW seems to work better than SG, indicating that CBOW can learn higher quality word vectors than SG on these collections.

We further study the effect of embedding dimensionality on the retrieval performance. Here we report the performance results on the Robust04 collection using word embeddings trained by CBOW model with 50, 100, 300, and 500 dimensions, respectively. As shown in Table 3, the performance first increases and then slightly drops with the increase of dimensionality. As we know, word embeddings of different dimensionality provide different levels of granularity of semantic similarity; they may also require different amounts of training data. With lower dimensionality, the similarity between word embeddings might be coarse and hurt the semantic matching performance. However, with larger dimensionality, one may need more data to train reliable word embeddings. Our results suggest that 300 dimensions is sufficient for learning word embeddings effective for semantic matching on the Robust04 collection.

6.3.2 *Impact of Indexed Neighbor Size*

As mentioned previously, we index the nearest neighbors of words for efficient solution of the transportation problem. Here we study the effect of neighbor size on the retrieval performance. Due to the space limitation, we plot the performance variations of our model under MAP over the Robust04 collection with respect to the indexed neighbor size in Figure 3. Similar trends can be found on the other collections. As we can see, the performance increases rapidly with the indexed neighbor size, and then keeps stable after a certain point. The results show that with a small neighbor size, the performance improvement is limited since very few semantic matching signals can be leveraged. When the neighbor size is large enough, we can obtain quite stable results, e.g., we did not observe obvious performance drop even when the neighbor size reaches 200. A possible reason is that the transportation profits become smaller for lower ranked neighbors and thus the matching over these words will make little contribution to the relevance scores.

Table 3: Performance comparison of NWT over different dimensionality of word embeddings trained by CBOW on the Robust04 collection.

Topic	Embedding	MAP	NDCG@20	P@20
Titles	CBOW-50d	0.255	0.409	0.360
	CBOW-100d	0.265	0.420	0.371
	CBOW-300d	0.274	0.426	0.380
	CBOW-500d	0.272	0.426	0.379
Descriptions	CBOW-50d	0.246	0.394	0.336
	CBOW-100d	0.256	0.400	0.342
	CBOW-300d	0.268	0.413	0.353
	CBOW-500d	0.265	0.411	0.352

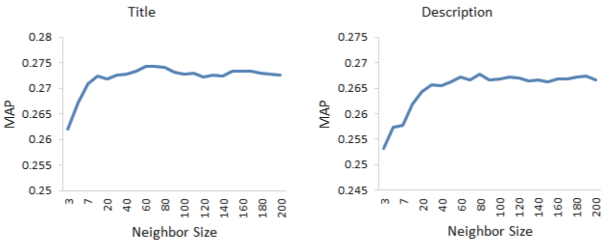

Figure 3: Performance variation of NWT with respect to the indexed neighbor size on the Robust04 collection under MAP.

6.3.3 *Linear vs. Non-Linear*

In our work, we formulate the semantic matching between documents and queries as a non-linear word transportation problem. Here we test whether the non-linearity is a necessary part for semantic matching. We consider two linear formulations. One is formulated by simply dropping the logarithm in the objective function of NWT. In this way,

[6] Note that only words can be found on Robust04 after stemming are considered as neighbors.

Table 4: Performance comparison between linear and non-linear models on the Robust04 collection.

	Models	MAP	NDCG@20	P@20
Linear	WMD	0.040	0.062	0.059
	RWT	0.079	0.143	0.129
Non-linear	NWT	0.274	0.426	0.380

we obtain a relaxed word transportation (RWT) formulation as there is only capacity constraints on the document side. The other is the WMD model [18], which defines a distance metric based on a standard word transportation formulation and has shown superior performance in document classification. The performance results on Robust04 using topic titles are shown in Table 4. As we can see, the retrieval performances of both linear models are poor. RWT works better than WMD, demonstrating that the relaxation of the constraints on query side is reasonable and beneficial in IR. NWT can significantly outperform RWT, indicating that the introduction of the non-linearity to model the diminishing marginal returns of transportation profit is crucial for semantic matching based IR.

7. CONCLUSIONS

In this paper, we introduce a new view of semantic matching for IR via a non-linear word transportation framework. We show that there are three important factors in the model that make it different from existing semantic matching based models and effective for IR: (1) Transportation based on the BoWE representation enables our model to capture detailed semantic matching signals between document words and query words. (2) The word alignment effect due to the relaxation of constraints and the marginal diminishing effect due to the non-linear formulation can better model the semantic matching process. (3) The flexibility in model definition (e.g., word capacity and transportation profit) enables the design of models dedicated to the IR task. Extensive experimental results demonstrate the effectiveness of the new retrieval model compared with state-of-the-art retrieval models. For future work, we plan to explore different model variations within the transportation framework.

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the 973 Program of China under Grant No. 2014CB340401 and 2013CB329606, in part by the National Natural Science Foundation of China under Grant No. 61232010, 61472401, 61425016, and 61203298, and in part by the Youth Innovation Promotion Association CAS under Grant No. 20144310 and 2016102.

9. REFERENCES

[1] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM TOIS*, 20(4):357–389, 2002.

[2] A. Atreya and C. Elkan. Latent semantic indexing (LSI) fails for TREC collections. *ACM SIGKDD Explorations Newsletter*, 12(2):5–10, 2011.

[3] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *SIGIR*, pages 222–229. ACM, 1999.

[4] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[5] S. Clinchant and F. Perronnin. Aggregating continuous word embeddings for information retrieval. In *ACL*, page 100, 2013.

[6] K. Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. In *CIKM*, pages 837–846. ACM, 2009.

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[8] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

[9] W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.

[10] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[11] H. Fang and C. Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *SIGIR*, pages 115–122. ACM, 2006.

[12] D. Ganguly, D. Roy, M. Mitra, and G. J. Jones. Word embedding based generalized language model for information retrieval. In *SIGIR*, pages 795–798. ACM, 2015.

[13] F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *J. Math. phys*, 20(2):224–230, 1941.

[14] R. Jin, A. G. Hauptmann, and C. X. Zhai. Language model for information retrieval. In *SIGIR*, pages 42–48. ACM, 2002.

[15] M. Karimzadehgan and C. Zhai. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *SIGIR*, pages 323–330. ACM, 2010.

[16] T. Kenter and M. de Rijke. Short text similarity with word embeddings. In *CIKM*, pages 1411–1420. ACM, 2015.

[17] R. Krovetz. Viewing morphology as an inference process. In *SIGIR*, pages 191–202. ACM, 1993.

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

[19] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR*, pages 111–119. ACM, 2001.

[20] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR*, pages 120–127. ACM, 2001.

[21] M. E. Lesk. Word-word associations in document retrieval systems. *American documentation*, 20(1):27–38, 1969.

[22] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *SIGIR*, pages 472–479. ACM, 2005.

[23] 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.

[24] J. H. Park and W. B. Croft. Using key concepts in a translation model for retrieval. In *SIGIR*, pages 927–930. ACM, 2015.

[25] O. Pele and M. Werman. Fast and robust Earth Mover's Distances. In *Computer vision, 2009 IEEE 12th international conference on*, pages 460–467. IEEE, 2009.

[26] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[27] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *SIGIR*, pages 232–241. ACM, 1994.

[28] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[29] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM*, pages 623–632. ACM, 2007.

[30] I. Vulić and M.-F. Moens. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *SIGIR*, pages 363–372. ACM, 2015.

[31] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185. ACM, 2006.

[32] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, pages 334–342. ACM, 2001.

[33] G. Zuccon, B. Koopman, P. Bruza, and L. Azzopardi. Integrating and evaluating neural word embeddings in information retrieval. In *ADCS*, pages 12:1–12:8. ACM, 2015.