Word Embedding based Approaches for Information Retrieval

Dwaipayan Roy
Indian Statistical Institute, Kolkata
203, B.T. Road
Kolkata, India - 700108
dwaipayan_r@isical.ac.in

Word Embedding, Information Retrieval

1. INTRODUCTION

The interest on using word embedding has expanded in various areas of text processing in recent years following the introduction of the word2vec model by Mikolov et al. (2013) and Pennington et al. (2014). The word embedding models use a large amount of text to create low dimensional representations of words capturing relationships between words without any external supervision. The resultant representation is shown to replicate many linguistic regularities such as, *semantic similarity* between terms, *conceptual composition* of terms, *laws of analogy* of terms. These features can be used for Information Retrieval (IR) where, the retrieval functions primarily depend on statistical co-occurrences. For keyword based retrieval systems, often there is the problem of vocabulary mismatch. For example, given the information need ‘vehicle industry in Germany’, relevant documents might not get retrieved due to the presence of ‘automobile’ in place of ‘vehicle’. Documents with ‘volkswagen’ might not get the suitable importance due to the vocabulary mismatch. Researches has been going on to overcome the problem of vocabulary mismatch problem utilising word embedding. In this paper, some of the approaches that use word embedding for better retrieval is presented. Empirical experiments have shown the positive contribution of the embedding informations in text retrievals. The rest of the paper is organized as follows. In Section 2, a baseline retrieval model, that uses word embedding, is presented. Following that, in Section 3, word embedding based query expansion methods are elaborated. The empirical evidence of superiority of both type of models over the state-of-the-art retrieval models are shown after the presentation of the corresponding models. The paper is concluded with some future directions of studies in Section 4.

2. BASELINE RETRIEVALS

In this section, a novel baseline retrieval method is presented. Namely, the Generalized Language Model (GLM), a baseline retrieval model that uses word embedding for better retrieval performance, is presented with experimental results in Section 2.1.

2.1. Generalized Language Model (Ganguly et al. 2015)

This model is intended to address the vocabulary mismatch problem for retrieval. Given a query $Q = \{q_1, q_2, \dots, q_n\}$, documents are returned as a list, ordered by the posterior probability $P(d|Q)$ which is estimated using the prior probabilities $P(q|d)$ and $P(q|C)$, that is the maximum likelihood estimated probabilities of generating a query term q from the document d and the collection C respectively, using frequency statistics (See Equation 1). Keeping the semantic meaning, what if a noise channel changes the word q to q' in the vocabulary while processing? In that case, Equation 1 will not consider the presence of those q' in the documents. To overcome this problem, a term transformation based model is proposed (see Equation 2). In this model, the prior probability is estimated using three term transformation events (See Figure 1 for graphical representation).

$$P(d|Q) = \prod_{q \in Q} \lambda \frac{tf(q, d)}{|d|} + (1 - \lambda) \frac{cf(q)}{|Coll|} \quad (1)$$

1. **Direct term sampling:** Standard language model term sampling.
2. **Transformation via Document Sampling:** Sampling a term q' from d which is then transformed

Figure 1: Schematics of generating a query term q in GLM. GLM degenerates to LM when $\alpha = \beta = 0$.

to q by a noisy channel.

$$\begin{aligned} P(q, q'|d)P(q') &= P(q|q', d)P(q'|d) \\ &= \frac{\text{sim}(q, q')}{\sum_{t \in d} \text{sim}(q, t)} \frac{tf(q', d)}{|d|} \end{aligned}$$

3. **Transformation via Collection Sampling:** Sampling a term q' from collection which is then transformed to q by a noisy channel.

$$\begin{aligned} P(q, q'|C)P(q') &= P(q|q', C)P(q'|C) \\ &= \frac{\text{sim}(q, q')}{\sum_{t \in N_q} \text{sim}(q, t)} \frac{cf(q')}{|coll|} \end{aligned}$$

Here, $\text{sim}(a, b)$ represents the cosine similarity of the embeddings of the term a and b . The prior probability of the transformed term q' , $P(q')$ is kept uniform for all terms. Together, considering these three term transformation event, the retrieval function is stated in Equation 2.

$$\begin{aligned} P(d|Q) &= \prod_{q \in Q} \lambda \frac{tf(q, d)}{|d|} + \alpha \sum_{q' \in d} P(q, q'|d)P(q') \\ &\quad + \beta \sum_{q' \in N_q} P(q, q'|C)P(q') + (1 - \alpha - \beta - \lambda) \frac{cf(q)}{|Coll|} \end{aligned} \quad (2)$$

Experiments on TREC collections show that this method yields significantly better performance than standard LM based baseline models as well as an LDA smoothed language model (See Table 1).

3. QUERY EXPANSION

The semantic similarity information can also be used for query expansion. In this section, two query expansion methods are demonstrated that use word embeddings. The first work, in Section 3.1, is a nearest neighbour based approach that expands a given query by terms which are similar to the query terms in the embedded space. In Section 3.2, the semantic similarity is applied in conjunction with relevance model. Specifically, RM3 (Jaleel et al. 2004), the mixture model of Relevance

Table 1: Comparative performance of LM, LDA and GLM on the TREC query sets.

Topic Set	Method	Metrics		
		MAP	GMAP	Recall
TREC-6	LM	0.2148	0.0761	0.4778
	LDA-LM	0.2192	0.0790	0.5333
	GLM	0.2287	0.0956	0.5020
TREC-7	LM	0.1771	0.0706	0.4867
	LDA-LM	0.1631	0.0693	0.4854
	GLM	0.1958	0.0867	0.5021
TREC-8	LM	0.2357	0.1316	0.5895
	LDA-LM	0.2428	0.1471	0.5833
	GLM	0.2503	0.1492	0.6246
Robust	LM	0.2555	0.1290	0.7715
	LDA-LM	0.2623	0.1712	0.8005
	GLM	0.2864	0.1656	0.7967

based Language Model (RLM) (Lavrenko and Croft 2001) and the query likelihood model, is expanded to include semantic informations in the model. In the rest of the paper, RM3 and RLM are used interchangeably to indicate to the same underlying procedure.

3.1. Nearest Neighbour based Query Expansion (Roy et al. 2016b)

In this section, we will describe two QE methods that use the embeddings of individual query terms. The first method is a kNN based QE method that makes use of the semantically similar terms to expand a query. Unlike pseudo relevance feedback (PRF) based QE methods, this approach does not require an initial round of retrieval. The second approach we tried is a straightforward variation of the first approach that uses word embeddings in conjunction with a set of pseudo relevant documents in an ad-hoc way.

Pre-retrieval kNN based approach

Let the given query Q be $q_1, \dots, q_m$. In this approach, we define the set C of candidate expansion terms by finding terms which are semantically similar to the q_i s, following Equation 3.

$$C = NN(\vec{Q}) = \bigcup_{q \in Q} NN(\vec{q}) \quad (3)$$

For a term t to be chosen as the nearest-neighbours of Q , it should have high mean inner product similarity between t and all terms of Q , following Equation 4.

$$\text{Sim}(t, Q) = \frac{1}{|Q|} \sum_{q \in Q} \vec{t} \cdot \vec{q} \quad (4)$$

Post-retrieval kNN based approach

In the next approach, a set of pseudo-relevant documents (PRD) are used to restrict the search domain for the

Collection	Baseline	Pre-ret	Post-ret
TREC6	0.2303	0.2406*	0.2393
TREC7	0.1750	0.1806	0.1806*
TREC8	0.2373	0.2535*	0.2531*
ROBUST	0.2651	0.2842*	0.2885*
WT10G	0.1454	0.1718*	0.1709*

Table 2: MAP for baseline retrieval and various QE strategies. A * in the Pre-ret and Post-ret columns denotes a significant improvement over the baseline. Significance testing has been performed using paired t-test with 95% confidence.

candidate expansion terms. Instead of searching for nearest neighbours within the entire vocabulary of the document collection, only those terms are considered that occur within PRD. The rest of the procedure for obtaining the expanded query is the same as it's pre-retrieval counterpart.

In Table 2, the empirical results of the methods are presented. As a baseline method, language model with linear smoothing is used. It can be seen that the semantic similarity is useful even when they are applied in an ad-hoc way. Note that, the performance of both the methods (pre and post-retrieval) are similar. The reason behind this behaviour is due to the fact that the embedding model is learned considering the entire vocabulary. Even if the search space is minimized, most of the terms having similar context as that of the query terms, are retrieved during the retrieval.

3.2. Word Embedding based Relevance Model (Roy et al. 2016a)

As shown in Section 3.1 that simple approaches using word embedding for retrieval can results in significantly better performance. Applying embedding in a systematic approach, in this section, a novel word embedding based relevance model is presented. Relevance model hypothesises that both a query and its relevant documents are sampled from a hidden relevance model R , which needs to be estimated. The query $Q = \{q_1, \dots, q_k\}$ serves as the only evidence about R . In absence of relevance judgement, after performing retrieval, top ranked documents can be considered as set of (pseudo-) relevant documents. Thus, the probability density function that generates R , or the probability of sampling a term w from this model, denoted by $P(w|R)$, is approximated by $P(w|Q)$. This conditional probability can be expressed as the joint probability of observing the term w along with the query words. The probability $P(w|Q)$ is then estimated in the following way: The term w is sampled conditionally, together with $q_1, \dots, q_k$ from the same distribution, underlying a top ranked document. Thus the probability estimation of $P(w|R)$ will follow Equation 5, where M is the set of (pseudo-)relevant documents. Finally, the terms with high $P(w|R)$ are chosen as

possible expansion terms and re-retrieval is performed.

$$P(w|R) = \sum_{D \in M} P(w|D) \prod_{q \in Q} P(q|D) \quad (5)$$

(Jaleel et al. 2004) showed that RLM feedback model works significantly better when applied in conjunction with the query likelihood model, following Equation 6.

$$P'(w|R) = \mu P(w|R) + (1 - \mu) P(w|Q) \quad (6)$$

Kernel Density Estimation based Relevance Model

Given a set of observed points, kernel density estimation (KDE) estimates the distribution function that generates the data. Mathematically, given n observed data points and a kernel function $K()$, Equation 7 represents the estimated probability distribution function where h is the bandwidth parameter and α_i is the weight of the local kernel function around i^{th} data point, where $\sum \alpha_i = 1$. For unknown data point x , $f(x, \alpha)$ will be high if x is similar to all observed data points x_i ($i = 1 \dots n$) on an average.

$$f(x, \alpha) = \frac{1}{nh} \sum_{i=1}^n \alpha_i K\left(\frac{x - x_i}{h}\right) \quad (7)$$

In order to go beyond binary-term-independence based co-occurrence statistics, a word embedding based scheme is used. Following this scheme, if a word w and a query term q_i are semantically related, they will occur in similar contexts; consequently, the vector embeddings of w and q_i will be in close proximity. Now with the query terms embedded as vectors, the probability density function $P(w|R)$ can be estimated with KDE. To see how this happens, imagine the existence of a continuous probability density function $f(w)$ from which the discrete probabilities $P(w|R)$ of the RLM (see Equations 5) are sampled. The shape of this *relevance density function* is controlled by a set of pivot points consisting of the query term embeddings. The key idea is illustrated in Figure 2, which shows a sample query with three terms. Note that the embedding of the query terms as vectors makes it possible to define the notion of distances between them. This enables the relevance model probability distribution function to be visualized as a function pivoted around the query vectors projected on a one dimensional line, as shown in Figure 2a.

For a query consisting of k terms, $Q = \{q_1, \dots, q_k\}$, instead of k pivot points, as in the one dimensional KDE, we now consider kM points, where M is the number of feedback documents. Each pivot point represents a query term $q_i, i = 1, \dots, k$ occurring in a document $D_j, j = 1, \dots, M$, the points being shown in a grid layout in Figure 2c. The x-axis corresponds to the query term vectors similar to one dimensional KDE (Figures 2a and 2b). The y-axis, in this case, represents the normalized term frequency of query terms in respective feedback documents. The shape of the density function, which is

Figure 2: Relevance model density estimation with 1 and 2 dimensional KDE.

now a mixture of two dimensional Gaussians, is shown as contour lines. It can be seen from Figure 2c that the value of the density function is maximum at the pivot points themselves and gradually decreases with increasing neighbourhood size. This is shown with different shades of grey in Figure 2c, where a darker shade denotes a higher value for the density function and a lighter one denotes a smaller value. Formally speaking, the kernel functions around the data points are bivariate normal distributions. A pivot point $\mathbf{x}_{ij} = (q_i, D_j)$ in this two dimensional space encapsulates the word vector for q_i and its normalized term frequency in feedback document D_j , i.e. $P(q_i|D_j)$. For simplicity, we define the covariance matrix Σ as a diagonal matrix with equal covariance σ for both the dimensions (notice in Figure 2c that the contours are circles instead of ellipses).

$$\begin{aligned} f(\mathbf{x}, \alpha) &= \frac{1}{kMh} \sum_{i=1}^k \sum_{j=1}^M \alpha_{ij} K\left(\frac{\mathbf{x} - \mathbf{x}_{ij}}{h}\right) \\ &= \sum_{i=1}^k \sum_{j=1}^M \alpha_{ij} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2 h^2} (\mathbf{x} - \mathbf{x}_{ij})^T (\mathbf{x} - \mathbf{x}_{ij})\right) \end{aligned} \quad (8)$$

The density function estimation for a point $\mathbf{x} = (w, D_j)$ is shown in Equation 8. Substituting the values of $\mathbf{x}$ and the pivot points $\mathbf{x}_{ij}$ in Equation 8, we get Equation 9.

$$f(\mathbf{x}, \alpha) = \sum_{i=1}^k \sum_{j=1}^M \frac{\alpha_{ij}}{2\pi\sigma^2} \exp\left(\frac{(\vec{w} - \vec{q}_i)^2 + (P(w|D_j) - P(q_i|D_j))^2}{-2\sigma^2 h^2}\right) \quad (9)$$

In Equation 9, the values for the weights for the kernels, α_{ij} , are set to $P(w|D_j)P(q_i|D_j)$. From Equation 9, we can observe that a term w in document D_j , denoted as the two dimensional vector $\mathbf{x}$, will get a high value of the density function if: (i) the embedded vector of w is close to the query terms q_i , $i = 1 \dots, k$, or in other words, w is semantically related to the query terms; and (ii) w frequently co-occurs with the query terms in each top ranked document D_j , $j = 1 \dots, M$.

The empirical performance of the method is shown in Table 3. It can be seen that the performance of the proposed method is always significantly better than the RM3 baseline model.

Dataset	Method	Parameters		Metrics		
		M	N	MAP	Recall	P@5
TREC 6	LM	n/a	n/a	0.2355	0.5100	0.4040
	RLM	20	70	0.2634	0.5368	0.4360
	KDE	10	80	0.2850[†]	0.5724 [†]	0.4600 [†]
TREC 7	LM	n/a	n/a	0.1787	0.4893	0.4040
	RLM	20	70	0.2151	0.5432	0.4160
	KDE	10	80	0.2380	0.6108[†]	0.4400
TREC 8	LM	n/a	n/a	0.2466	0.5936	0.4560
	RLM	20	70	0.2701	0.6410	0.4760
	KDE	10	80	0.2957[†]	0.6887[†]	0.5120[†]
TREC Rb	LM	n/a	n/a	0.2699	0.7938	0.4464
	RLM	20	70	0.3304	0.8559	0.4949
	KDE	10	80	0.3456[†]	0.8772[†]	0.5152[†]

Table 3: Results of KDE feedback methods with QE. Parameters M (#fdbk docs) and N (#expansion terms) are tuned on the development sets - TREC 6. [†] denotes significance with respect to RLM.

4. CONCLUSION AND FUTURE WORK

There is a huge scope of improvements of retrieval results utilizing the features of word embeddings. A possible future work would be to look into the applications of word embeddings for other tasks in IR such as query categorization, novelty and diversity detection of documents.

REFERENCES

- Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth Jones. Word embedding based generalized language model for information retrieval. In *Proc. of 38th ACM SIGIR*, 2015.
- Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. Umass at TREC 2004: Novelty and HARD. In *Proc. of TREC '04*, 2004.
- Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proc. of 24th ACM SIGIR*, SIGIR '01, pages 120–127, 2001. ISBN 1-58113-331-6.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *EMNLP*, pages 1532–1543. ACL, 2014.
- Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth J. F. Jones. Word vector compositionality based relevance feedback using kernel density estimation. In *Proc. of 25th ACM CIKM 2016*, pages 1281–1290, 2016a.
- Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*, 2016b.