

# Toward Incorporation of Relevant Documents in word2vec

Navid Rekabsaz, Mihai Lupu, Allan Hanbury\*  
Information & Software Engineering Group  
TU WIEN  
rekabsaz/lupu/hanbury@ifs.tuwien.ac.at

Bhaskar Mitra†  
Microsoft, UCL  
Cambridge, UK  
bmitra@microsoft.com

## ABSTRACT

Recent advances in neural word embedding provide significant benefit to various information retrieval tasks. However as shown by recent studies, adapting the embedding models for the needs of IR tasks can bring considerable further improvements. The embedding models in general define the term relatedness by exploiting the terms' co-occurrences in short-window contexts. An alternative (and well-studied) approach in IR for related terms to a query is using local information i.e. a set of top-retrieved documents. In view of these two methods of term relatedness, in this work, we report our study on incorporating the local information of the query in the word embeddings. One main challenge in this direction is that the dense vectors of word embeddings and their estimation of term-to-term relatedness remain difficult to interpret and hard to analyze. As an alternative, explicit word representations propose vectors whose dimensions are easily interpretable, and recent methods show competitive performance to the dense vectors. We introduce a neural-based explicit representation, rooted in the conceptual ideas of the word2vec Skip-Gram model. The method provides interpretable explicit vectors while keeping the effectiveness of the Skip-Gram model. The evaluation of various explicit representations on word association collections shows that the newly proposed method outperforms the state-of-the-art explicit representations when tasked with ranking highly similar terms. Based on the introduced explicit representation, we discuss our approaches on integrating local documents in globally-trained embedding models and discuss the preliminary results.

## KEYWORDS

explicit representations, word2vec, PMI, similarity, relevant model

### ACM Reference format:

Navid Rekabsaz, Mihai Lupu, Allan Hanbury and Bhaskar Mitra. 2017. Toward Incorporation of Relevant Documents in word2vec. In *Proceedings of SIGIR '17 NeuIR workshop, August 07-11, 2017, Shinjuku, Tokyo, Japan*, . 5 pages.  
DOI:

## 1 INTRODUCTION

Word embedding i.e. representation of words in a low-dimensional space was shown to benefit IR tasks like document retrieval [16, 21, 22, 28], query expansion [5, 25], or sentiment analysis [20]. Despite

\*Funded by: Self-Optimizer (FFG 852624) in the EUROSTARS programme, funded by EUREKA, BMWFW and European Union, and ADMIRE (P 25905-N23) by FWF. Thanks to Joni Sayeler and Linus Wretblad for their contributions in SelfOptimizer.

†The author is a part-time PhD student at University College London.

the relative improvement of various IR tasks, recently Rekabsaz et al. [23] points out the issue of topic shifting when directly using window-based word embedding models (i.e. word2vec [15] and GloVe [18]). Their study shows a high potential in further performance improvement in document retrieval when filtering out the terms that cause the topic shifting.

One fundamental issue in using word embedding (or most other methods of term similarity) is rooted in the assumption of independence of query terms i.e. the similar terms to a query term is independent of the other query terms. Diaz et al. [5] approaches this issue by training separate embedding models on local information of the query i.e. a set of top retrieved documents (top 1000 in their study). They show that the locally-trained word embeddings outperform the global embedding model as the similar terms are relevant to the whole the query. However, as mentioned in the study, it is not a feasible approach because a new model has to be trained at query time (albeit on a small set of documents). The use of local information for retrieval has extensively studied [10, 14, 27]. Recent studies use the local information to train a relevant-based embedding [26] or multi-layer neural network ranking model [4]. In contrast, we focus on exploiting window-based term similarities adjusted with local information.

In this report, we present our ideas and the intermediary steps in this direction, asking *How can we efficiently incorporate the local information in the word embedding models?* The word embeddings, while easy to construct based on raw unannotated corpora, are hardly interpretable, especially in comparison to lexical semantics. It remains opaque what the dimensions of the vectors refer to, or which factors in language (corpus) cause the relatedness of two words. Having interpretable vectors would enable the integration of important topics of the local documents and make a causal analysis possible.

A natural solution to this problem is using explicit representations of words i.e. vectors with clearly-defined dimensions, which can be words, windows of words, or documents. These vectors appeared decades ago as the initial form of word/document representations. In IR, the word-document matrix, populated with an IR weighting schema (e.g. TFIDF, BM25) has been extensively used, at least as a conceptual model. This matrix when being subjected to SVD matrix factorization produces the LSI model [3] with dense word/document representation. The well-known Pointwise Mutual Information (PMI) method is an alternative representation method, rooted in information theory, which provides a word-context matrix based on the co-occurrences between words in contexts.

Despite high interpretability, the explicit representations are too large to be stored in memory (i.e. are not efficient) and have been often shown to be less effective in comparison to the low-dimensional dense vectors. In practice, the memory issue can be mitigated by suitable data structures if the vectors are highly sparse. Regarding

the effectiveness, Levy et al. [12] show competitive performance of a proposed explicit representation (Section 2) in comparison to the state-of-the-art word embeddings on a set of word association test collections.

As an alternative approach to improve interpretability, some recent reports [7, 24] propose methods to increase the sparsity of the dense vectors. The rationale of these approaches is that by having more sparse vectors, it becomes more clear which dimension of the vectors might be referring to which concepts in language.

In contrast to this approach, our main contribution is in line with previous studies [11, 12] on providing fully interpretable vectors by proposing a novel explicit representation, based on the advancements in the neural word embedding methods. Our approach originates from the word2vec Skip-Gram model i.e. the proposed representation benefits from interpretability and also the substantial subtleties of the Skip-Gram model (Section 3).

We evaluate our method on the task of retrieving highly similar words to a given word, as an essential scenario in many IR tasks. We show that our proposed explicit representation, as similar to the Skip-Gram model, outperforms the state-of-the-art explicit vectors in selecting highly similar words (Section 4).

Given the explicit representation, we discuss our ideas and the preliminary results on integrating the local information of the queries in the representations (Section 5).

## 2 BACKGROUND

In this section, we review the word2vec Skip-Gram (SG) model as well as the state-of-the-art explicit representations.

### 2.1 Embedding with Negative Sampling

The SG model starts with randomly initializing two sets of vectors: word ( $V$ ) and context ( $\tilde{V}$ ) vectors, both of size  $|W| \times d$ , where  $W$  is the set of words in the collection and  $d$  is the embedding dimensionality.

The objective of SG is to find a set of  $V$  and  $\tilde{V}$ —as the parameters of an optimization algorithm—by increasing the conditional probability of observing a word  $c$  given another word  $w$  when they co-occur in a window, and decreasing it when they do not. In theory, this probability is defined as follows:

$$p(c|w) = \frac{\exp(V_w \tilde{V}_c)}{\sum_{c' \in W} \exp(V_w \tilde{V}_{c'})} \quad (1)$$

where  $V_w$  and  $\tilde{V}_c$  are the word vector of the word  $w$  and the context vector of the word  $c$ , respectively.

Obviously, calculating the denominator of Eq. 1 is highly expensive and a bottleneck for scalability. One proposed approach for this problem is *Noisy Contrastive Estimation (NCE)* [17] method. The NCE method, instead of computing the probability in Eq. 1, measures the probability which contrasts the *genuine* distribution of the words-context pairs (given from the corpus) from a *noisy* distribution. The noisy distribution  $\mathcal{N}$  is defined based on the unigram distribution of the words in the corpus. Formally, it defines a binary variable  $y$ , showing whether a given pair belongs to the genuine distribution:  $p(y = 1|w, c)$ . Further on, Mikolov et al. [15] proposed the *Negative Sampling* method by some simplifications in calculating  $p(y = 1|w, c)$  (further details are explained in the original papers), resulting in the following formula:

$$p(y = 1|w, c) = \frac{\exp(V_w \tilde{V}_c)}{\exp(V_w \tilde{V}_c) + 1} = \sigma(V_w \tilde{V}_c) \quad (2)$$

where  $\sigma$  is the sigmoid function ( $\sigma(x) = 1/(1 + \exp(-x))$ ). Based on this probability, the cost function of the SG method is defined as follows:

$$J = - \sum_{\langle w, c \rangle \in X} \left[ \log p(y = 1|w, c) + k \mathbb{E}_{\check{c}_i \sim \mathcal{N}} \log p(y = 0|w, \check{c}_i) \right] \quad (3)$$

where  $X$  is the collection of co-occurrence pairs in the corpus,  $\check{c}_i$  is each of the  $k$  sampled words from the noisy distribution  $\mathcal{N}$ , and  $\mathbb{E}$  denotes expectation value, calculated as the average for the  $k$  sampled words in every iteration.

In addition, two preprocessing steps dampen the dominating effect of very frequent words: First is *subsampling* which randomly removes an occurrence of word  $w$  in the corpus  $C$  when the word's frequency in the corpus  $f(w, C)$  is more than some threshold  $t$  with a probability value of  $1 - \sqrt{t/f(w, C)}$ . The second is *context distribution smoothing (cDS)* which dampens the values of the probability distribution  $\mathcal{N}$  by raising them to power  $\alpha$  where  $\alpha < 1$ .

### 2.2 Explicit Representation

A well-known explicit representations is defined based on PMI. In this representation, for the word  $w$ , the value of the corresponding dimension to the context word  $c$  is defined as follows:

$$PMI(w, c) = \log \frac{p(w, c)}{p(w)p(c)} \quad (4)$$

where  $p(w, c)$  is the probability of appearance of  $\langle w, c \rangle$  in the co-occurrence collection:  $f(\langle w, c \rangle, X)/|X|$  and  $p(w)$  is calculated by counting the appearance of  $w$  with any other word:  $f(\langle w, \cdot \rangle, X)/|X|$  (same for  $p(c)$ ).

A widely-used alternative is *Positive PMI (PPMI)* which replaces the negative values with zero:  $PPMI(w, c) = \max(PMI(w, c), 0)$ .

Levy and Goldberg [11] show an interesting relation between *PMI* and *SG* representations, i.e. when the dimension of the vectors is very high (as in explicit representations), the optimal solution of *SG* objective function (Eq. 3) is equal to *PMI* shifted by  $\log k$ . They call this representation *Shifted Positive PMI (SPPMI)*:

$$SPPMI(w, c) = \max(PMI(w, c) - \log(k), 0) \quad (5)$$

They further integrate the ideas of subsampling and *cDS* into *SPPMI*. Subsampling is applied when creating the  $X$  set by randomly removing very frequent words. The *cDS* method adds a smoothing on the probability of the context word, as follows:

$$PMI_\alpha(w, c) = \log \frac{p(w, c)}{p(w)p_\alpha(c)} \quad p_\alpha(c) = \frac{f(\langle w, \cdot \rangle, X)^\alpha}{\sum_{w' \in W} f(\langle w', \cdot \rangle, X)^\alpha} \quad (6)$$

They finally show the competitive performance of the *SPPMI* model on word association tasks to the *SG* model. Their definitions of *PPMI* and *SPPMI* are the current state-of-the-art in explicit representations, against which we will compare our method.

**Table 1: Word association evaluation. Best performing among explicit/all embeddings are shown with bold/underline.**

Method	Sparsity	WS Sim.	WS Rel.	MEN	Rare	SCWS	SimLex
PPMI	98.6%	.681	.603	.702	.309	.601	.284
SPPMI	99.6%	<b>.722</b>	<b>.661</b>	.704	.394	.571	<b>.296</b>
ExpSG	0%	.596	.404	.645	.378	.549	.231
RExpSG	0%	.527	.388	.606	.311	.507	.215
PRExpSG	94.1%	.697	.626	<b>.711</b>	<b>.406</b>	<b>.614</b>	.272
SG	0%	<u>.770</u>	.620	<u>.750</u>	<u>.488</u>	<u>.648</u>	<u>.367</u>

### 3 EXPLICIT SKIP-GRAM

Let us revisit the  $p(y = 1|w, c)$  probability in the Negative Sampling method (Eq. 2) i.e. the probability that the co-occurrence of two terms comes from the training corpus and not from the random distribution. The purpose of this probability in fact shares a meaningful relation with the conceptual goal of the *PMI*-based representations i.e. to distinguish a genuine from a random co-occurrence. Based on this idea, an immediate way of defining an explicit representation would be to use Eq. 2 as follows:

$$\text{ExpSG}(w, c) = p(y = 1|w, c) = \sigma(V_w \tilde{V}_c) \quad (7)$$

This *Explicit Skip-Gram (ExpSG)* representation assigns a value between 0 to 1 to the relation between each pair of words. It is however intuitive to consider that the very low values do not represent a meaningful relation and can potentially introduce noise in computation. Such very low values can be seen in the relation of a word to very frequent or completely unrelated words. We can extend this idea to all the values of *ExpSG*, i.e. some portion (or all) of every relation contains noise.

To measure the noise in *ExpSG(w, c)*, we use the definition of noise probabilities in the Negative Sampling approach: the expectation value of  $p(y = 1|w, c)$  where  $c$  (or  $w$ ) is randomly sampled from the dictionary for several times. Based on this idea, we define the *Reduced Explicit Skip-Gram (RExpSG)* model by subtracting the two expectation values from *ExpSG*:

$$\text{RExpSG}(w, c) = \text{ExpSG}(w, c) - \mathbb{E}_{\tilde{c} \sim \mathcal{N}} p(y = 1|w, \tilde{c}) - \mathbb{E}_{\tilde{w} \sim \mathcal{N}} p(y = 1|\tilde{w}, c) \quad (8)$$

Since the expectation values can be calculated off-line, in contrast to Negative Sampling, we compute it over all the vocabulary:

$$\mathbb{E}_{\tilde{w} \sim \mathcal{N}} p(y = 1|\tilde{w}, c) = \frac{\sum_{i=1}^{|W|} f(\tilde{w}_i, C) \cdot \sigma(V_{\tilde{w}_i} \tilde{V}_c)}{\sum_{i=1}^{|W|} f(\tilde{w}_i, C)} \quad (9)$$

For the sampling of the context word  $\tilde{c}$ , similar to *SG* and  $PMI_\alpha$ , we apply *cds* by raising frequency to the power of  $\alpha$ , as follows:

$$\mathbb{E}_{\tilde{c} \sim \mathcal{N}} p(y = 1|w, \tilde{c}) = \frac{\sum_{i=1}^{|W|} f(\tilde{c}_i, C)^\alpha \cdot \sigma(V_w \tilde{V}_{\tilde{c}_i})}{\sum_{i=1}^{|W|} f(\tilde{c}_i, C)^\alpha} \quad (10)$$

Similar to *PPMI*, our last proposed representation removes the negative values. The *Positive Reduced Explicit SkipGram (PRExpSG)* is defined as follows:

$$\text{PRExpSG}(w, c) = \max(\text{RExpSG}(w, c), 0) \quad (11)$$

Setting the values to zero in *PRExpSG* facilitates the use of efficient data structures i.e. sparse vectors. We analyze the efficiency and effectiveness of the explicit representations in the next section.

### 4 EVALUATION OF EXPLICIT SKIP-GRAM

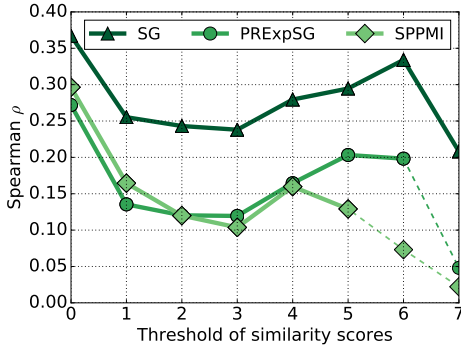
To analyze the representations, we create a Skip-Gram model with 300 dimensions on the Wikipedia dump file for August 2015 using the gensim toolkit [19]. As suggested by Levy et al. [12], we use a window of 5 words, negative sampling of  $k = 10$ , down sampling of  $t = 10^{-5}$ , a *cds* value of  $\alpha = 0.75$ , trained on 20 epochs, and filtering out words with frequency less than 100. The final model contains 199851 words. The same values are used for the common parameters in the *PPMI* and *SPPMI* representations.

We conduct our experiments on 6 word association benchmark collections. Each collection contains a set of word pairs where the association between each pair is assessed by human annotators (annotation score). The evaluation is done by calculating the Spearman correlation between the list of pairs scored by similarity values versus by human annotations. The collections used are: WordSim353 partitioned into Similarity and Relatedness [1]; MEN dataset [2]; Rare Words dataset [13]; SCWS [9]; and SimLex dataset [8].

The evaluation results for the explicit representations as well as *SG* are reported in Table 1. The bold values show the best performing explicit representation and the values with underline refer to the best results among all representations. Based on the results, *PRExpSG* and *SPPMI* show very similar performance (in 3 benchmarks *PRExpSG* and in the other 3 *SPPMI* shows the best performance), both considerably outperforming the other explicit representations. As also shown in previous studies [12], *SG* in general performs better than the best performing explicit representations.

However, when considering downstream tasks, despite the pervasive use of word association benchmarks, they do not provide a comprehensive assessment on many subtleties of representations which might be crucial (see Faruqui et al. [6]). For example, in tasks such as query expansion [5, 25] or the integration of embeddings in IR models [21, 22], what is expected from an effective word representation is a set of *highly* similar words for each query word.

To have a more relevant evaluation for these sort of tasks, we need to consider (1) the word *similarity* benchmarks (in contrast to *relatedness*) and (2) the effectiveness of the representation on highly similar words. Among the benchmarks, SimLex is a recent collection, specifically designed to evaluate the concept of word similarity. In particular, SimLex’s creators criticize the WordSim353 Similarity collection as it does not distinguish between word similarity and relatedness [8]. We therefore focus on the SimLex collection for further evaluations. To assess the effectiveness of representations



**Figure 1: Correlation of subsets of SimLex for various similarity thresholds.**

on highly similar pairs, we extract subsets of the SimLex collection that have higher annotation score than a threshold, and calculate the Spearman correlation of the results, separately for each subset. We conduct evaluation on 10 subsets with the thresholds from 9 (highly similar) to 0 (all pairs).

Figure 1 shows the evaluation results on the subsets for the *SG*, *PRExpSG*, and *SPPMI* representations. When the threshold is higher than 7, none of the models has significant correlation values ( $p < 0.05$ ) and are therefore not depicted. The non-significant results for *PRExpSG* and *SPPMI* are indicated with dashed lines. The *SG* model constantly shows high correlation values over all the thresholds. Interestingly, while the *PRExpSG* has slightly worse performance than *SPPMI* at lower thresholds, it reaches better correlation results at high thresholds. We argue that the better performance for highly similar pairs is due to the conceptual similarity of *PRExpSG* to the *SG* model i.e. *PRExpSG* benefits from the subtleties of the *SG* model.

Finally, let us have a look at the sparsity of the explicit representations, reported in Table 1. The *PRExpSG* and *SPPMI* representations are highly sparse, making them amenable to storage in volatile memory in practical scenarios. It is also worth noticing that in contrast to *SPPMI*, in the *PRExpSG* vectors, there might be non-zero values also for pairs of terms that do not necessarily ever co-occur in text. This characteristic—inherited from the *SG* model—is arguable the reason for better performance in high-similarity values, by mitigating the sparseness problem of a corpus.

## 5 INTEGRATION OF LOCAL INFORMATION

In this section, we discuss our ideas on incorporating the information of the local documents of a query in an explicit representation. Each cell value of any of the discussed explicit representations defines the degree of a genuine co-occurrence relationship between the term  $w$  to a context term  $c$ . We refer to the relation as  $v(w, c)$ . Our objective is to alter  $v$  to  $\hat{v}$  based on the information in the local document i.e. the top  $k$  retrieved documents ( $k$  has a small number like 10). To retrieve the local documents, we use language modeling with Dirichlet smoothing ( $\mu = 1500$ ) while one may apply any other retrieval model.

We define the relation between  $v$  and  $\hat{v}$  by introducing the coefficient  $\delta$ , defined from local information, as follows:  $\hat{v}(w, c) = \delta(w, c)v(w, c)$ . In the following, we suggest various definition for  $\delta$ .

The first suggestion only keeps the dimensions (context terms) of the representation whose corresponding terms appear in the relevant documents and sets the values related to other dimensions to zero.

$$\delta_1(w, c) = \delta_1(c) = \mathbb{1}[f(c, F) > 0] \quad (12)$$

where  $\mathbb{1}$  is the indicator function, and  $F$  is the collection of local documents. As mentioned in the Eq. 12, it is only based on the context terms  $c$  and independent of  $w$  (it also holds in some other formulas in the following as it is mentioned in the equation). Our initial results using the  $\delta_1$  method show little differences in the orders of top similar terms to the query terms when comparing the global embedding with the locally-adapted representation. We argue that it is due to the neglect of the weights of the terms in the local documents. In fact, since (very probably) there are still irrelevant documents in the top retrieved ones, by considering only the occurrence of the terms some irrelevant dimensions/terms still affect the similarity calculation.

To exploit the importance of the terms of the local documents, in the second method, we use the probability of the occurrence of the context term  $c$  in the global and local documents, defined as follows:

$$\delta_2(w, c) = \delta_2(c) = \frac{p(c|F)}{p(c|C)} = \frac{f(c, F)/\sum_{d \in F} |d|}{f(c, C)/\sum_{d \in C} |d|} \quad (13)$$

The third method reflects the idea of the *PMI*-based representations by measuring the probability of co-occurrences of  $w$  and  $c$  in some defined contexts in local and global documents.

$$\delta_3(w, c) = \frac{p(w, c|X_F)}{p(w, c|X_C)} = \frac{f(\langle w, c \rangle, X_F)/|X_F|}{f(\langle w, c \rangle, X_C)/|X_C|} \quad (14)$$

where  $X_F$  and  $X_C$  refer to the local and global co-occurrence sets respectively. The context in  $\delta_3$  can be defined as short-window around  $w$ , in paragraph-level, or whole the document.

The next method exploits the idea of the relevance model [10] and is defined as follows:

$$\delta_4(w, c) = \delta_4(c) = p(c|\Theta_F) = \sum_{\theta_d \in \Theta_F} p(c|\theta_d) \prod_{q \in Q} p(q|\theta_d) \quad (15)$$

where  $Q$  is the set of query terms  $q$ ,  $\Theta_F$  is the collection of documents' language models  $\theta_d$  for each document  $d \in F$ .

The last approach expands the  $\delta_4$  method by also using the probability of  $w$  in the joint probability  $p(w, c|\theta_d)$ . We assume the independence of  $w$  and  $c$  when conditioned on  $\theta_d$  i.e.  $p(w, c|\theta_d) = p(w|\theta_d)p(c|\theta_d)$ , resulting to the following definition:

$$\delta_5(w, c) = p(w, c|\Theta_F) = \sum_{\theta_d \in \Theta_F} p(w|\theta_d)p(c|\theta_d) \prod_{i \in Q} p(q|\theta_d) \quad (16)$$

We expect that the report provides the ground for circulating discussions on the theme of the study and suggested ideas.

## 6 CONCLUSION

Incorporating the important topics of the local documents in word embeddings is a crucial step for specializing the embedding models for IR tasks. To move toward this direction, we propose a novel explicit representation of words by capturing the probability of genuine co-occurrence of the words, achieved from the word2vec Skip-Gram model. The proposed representation inherits the characteristics of the Skip-Gram model while making it possible to interpret the vector representations. The evaluation on term association benchmarks

shows similar results to the state-of-the-art explicit representations, but our method outperforms the state-of-the-art in the scenario of retrieving top-similar words to a given word. Further on, based on the introduced explicit representation, we discuss our proposed methods to redefine the terms' vectors using the importance of the terms in the top-retrieved documents.

## REFERENCES

- [1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proc. of HLT-NAACL*.
- [2] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal Distributional Semantics. *JAIR* (2014).
- [3] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *American society for inf. science* (1990).
- [4] Mostafa Dehghani, Hamed Zamani, A. Severyn, J. Kamps, and W Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proc. of SIGIR*.
- [5] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. *Proc. of ACL* (2016).
- [6] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proc. of ACL*.
- [7] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. 2015. Sparse Overcomplete Word Vector Representations. In *Proc. of ACL*.
- [8] Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* (2016).
- [9] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proc. of ACL*.
- [10] Victor Lavrenko and W Bruce Croft. 2001. Relevance based language models. In *Proc. of SIGIR*.
- [11] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proc. of NIPS*.
- [12] Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL* (2015).
- [13] Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proc. of CoNLL*.
- [14] Yuanhua Lv and ChengXiang Zhai. 2009. A comparative study of methods for estimating query language models with pseudo feedback. In *Proc. of CIKM*.
- [15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
- [16] Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. 2016. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137* (2016).
- [17] Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proc. of ICML*.
- [18] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *Proc. of EMNLP*.
- [19] Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proc. of LREC Workshop on New Challenges for NLP Frameworks*.
- [20] Navid Rekasaz, Mihai Lupu, Artem Baklanov, Allan Hanbury, Alexander Dür, and Linda Anderson. 2017. Volatility Prediction using Financial Disclosures Sentiments with Word Embedding-based IR Models. In *Proc. of ACL*.
- [21] Navid Rekasaz, Mihai Lupu, and Allan Hanbury. 2016. Generalizing Translation Models in the Probabilistic Relevance Framework. In *Proc. of CIKM*.
- [22] Navid Rekasaz, Mihai Lupu, and Allan Hanbury. 2017. Exploration of a threshold for similarity based on uncertainty in word embedding. In *Proc. of ECIR*.
- [23] Navid Rekasaz, Mihai Lupu, Allan Hanbury, and Hamed Zamani. 2017. Word Embedding Causes Topic Shifting; Exploit Global Context!. In *Proc. of SIGIR*.
- [24] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2016. Sparse word embeddings using l1 regularized online learning. In *Proc. of IJCAI*.
- [25] Hamed Zamani and W Bruce Croft. 2016. Embedding-based query language models. In *Proc. of ICTIR*.
- [26] Hamed Zamani and W Bruce Croft. 2017. Relevance-based Word Embedding. In *Proc. of SIGIR*.
- [27] Chengxiang Zhai and John Lafferty. 2001. Model-based feedback in the language modeling approach to information retrieval. In *Proc. of CIKM*.
- [28] Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. 2015. Integrating and evaluating neural word embeddings in information retrieval. In *Proc. of Australasian Document Computing Symposium*.