2017 14th IAPR International Conference on Document Analysis and Recognition

New Word Pair Level Embeddings to Improve Word Pair Similarity

Asma Shaukat, Nazar Khan
Computer Vision and Machine Learning Group
Punjab University College of Information Technology (PUCIT)
Lahore, Pakistan
{asma.shaukat, nazarkhan}@pucit.edu.pk

Abstract—We present a novel approach for computing similarity of English word pairs. While many previous approaches compute cosine similarity of individually computed word embeddings, we compute a single embedding for the word pair that is suited for similarity computation. Such embeddings are then used to train a machine learning model. Testing results on MEN and WordSim-353 datasets demonstrate that for the task of word pair similarity, computing word pair embeddings is better than computing word embeddings only.

Keywords—word pair similarity, word pair embeddings, machine learning, natural language processing.

I. Introduction

As the name suggests, the word pair similarity (WPS) problem refers to computing a similarity level between two words. Some manually marked similarity estimates from the MEN dataset [29] can be seen in Table 1. WPS is one of most fundamental building blocks in many well-known applications of natural language processing, artificial intelligence, information retrieval and data mining. For example, textual similarity measurement including sentence pair similarity [1, 2, 34], document summarization [3], automatic thesauri generation [4], automatic retrieval of similar words [9] and word sense disambiguation [11] also involve WPS. Such applications have used different variety of similarity measures. In query expansion [5, 6], synonym words are used to modify user queries for improving search results.

Despite the widespread use of WPS, its measurement is still a challenging task. To compute semantic similarity of word pairs, manually compiled lexicons like WordNet and large text corpora have been used previously [7, 8]. Use of WordNet like databases has its own drawbacks. Such lexicons have limited knowledge. It is hard to keep them up to date. So focus of research has moved to utilization of large text corpora and ConceptNet-like semantic networks [15] that are continuously grown and updated.

In recent years, researchers have become increasingly interested in providing word embeddings to capture semantic similarities between words [12, 13, 16, 17, 10, 14]. A standard method used for measuring WPS uses cosine similarity of individual word embeddings. However, cosine similarity can be affected adversely by word frequency information in individual word embeddings [33].

TABLE 1: CROWD SOURCED SIMILARITY JUDGEMENTS FOR SOME WORD PAIRS RANGING FROM 0 TO 50.

Word$_1$	**Word$_2$**	**Similarity [29]**
car	automobile	50
cat	kitten	49
bakery	zebra	0
evening	walk	27
apartment	valley	14

We contend that embeddings used for WPS should be computed for the word pair instead of individual words. Word pair embeddings take advantage of word pair level information which is ignored in individual word embeddings. There is, however, less research and insight on how to use word embeddings to train machine learning models for measuring WPS effectively.

In this paper, we are proposing a new and useful way of looking for better vector space representation of word pairs for WPS measurement. We trained machine learning models on our resulting word pair embeddings for word pair similarity. We compared our results for WPS of our embeddings with results computed using some well-known individual word embeddings as Glove 42B [27], ConceptNet Numberbatch (CN) [16]. Our results are higher than previous state of the art results [Table 2, Table 3]. We found our embeddings a more useful way of looking at embeddings for word pair similarity than only word embeddings.

II. Methodology

We present a method for computing word pair embeddings instead of individual word embeddings. Many previous approaches present embeddings for individual words [14, 15, 16, 27] using their distributional semantics (Common Crawl corpus[1]) and structured knowledge from ConceptNet and PPDB [31]. These word embeddings, along with many other applications, have been used for measuring WPS using cosine similarity [15, 16, 27]. Although existing word embeddings for WPS measurement provide reasonable results, combining existing word embeddings with word pair specific information can improve performance of WPS measures. Figure 1 shows

[1] http://commoncrawl.org/

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.329

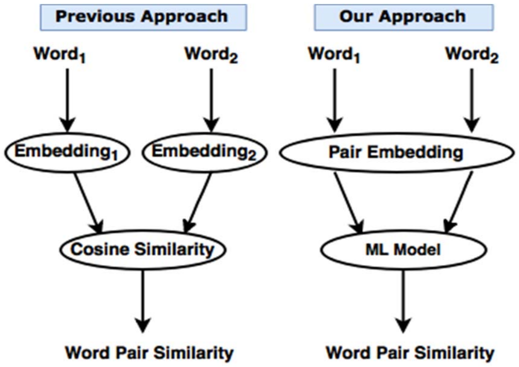

Figure 1: Overview of existing approaches and our approach for WPS.

an overview of previous approaches compared to our approach. The justification of our approach is very simple: if similarity is to be computed for a word pair, the embedding to be used should also be computed for a pair. Words have two types of similarities between them.

i) Attributional Similarity

Words sometimes share common attributes as cat and dog both are pet and animal. When two words share high degree of attributes, human beings consider them highly similar and vice versa. For example, a pair (car, automobile) is highly similar as both words share common attributes like ‘tire’, ‘brake’, ‘driving’ and others. These common attributes can have different relationships such as

tire—PartOf—car
tire—PartOf—automobile
driving—UseOf—car
driving—UseOf—automobile

Attributional similarity considers these phenomena and measures WPS using word attributes [20, 21]. Standard bag-of-words approach for information retrieval also uses attributional similarity [26].

ii)Relational Similarity

Words of a pair can be connected with each other through different relations. For example,

car— IsA—automobile
apple— IsA—fruit
come— Antonym—go
happy— Antonym—sad
happy— Synonym— cheerful
kitten— IsA—cat
kitten— RelatedTo—cat
walk— Antonym —stay
sit— Antonym—stand

Such type of information constitutes relational similarity. Relational similarity plays an important role in different applications including WPS [20], solving analogy questions [24] and classification of semantic relations [25].

Previous approaches based on cosine similarity of individual word embeddings exploit attributional similarity while ignoring the relational information between words. For the WPS task, relational information can be very helpful. However, to use such relational information, the embeddings should be computed at the level of word pairs instead of at the level of individual words only. Therefore, we incorporate information about relational similarity of a word pair in its embedding. A consequence of this is that we compute a single embedding for the word pair instead of two individual word embeddings. The next sections describe how our word pair embedding is computed.

A. ConceptNet

In order to extract relational information between words, we use ConceptNet, a semantic network of words. In recent years, ConceptNet has become a very valuable and rich source of common sense knowledge about words in machine readable form. It connects words with labeled relations. In addition to WPS, it has been explored for different applications including query expansion [18, 19, 22], word sense disambiguation [23], analogies [15, 16].

A well-known application of ConceptNet knowledge is contributing in providing effective vector space representations of individual words known as ConceptNet Numberbatch (CN) [15, 16, Figure 2]. Crowd sourced information about relations between words is also available. The 51 non-underlined features in Figure 3 represent relationship categories marked via crowd sourcing. This information can be utilized to learn machine learning models for measuring WPS effectively. Combining ConceptNet knowledge specific to pairs of words with existing word embeddings improves WPS results.

B. Word Pair Embeddings

For word pair (w_1, w_2), let ($\mathbf{e}_1$, $\mathbf{e}_2$) be the embeddings of the individual words obtained from an existing dataset such as Glove 42B [27] or ConceptNet Numberbatch (CN) [16]. We construct a 53 dimensional word pair embedding $\mathbf{v}$ from the relational information between words available in ConceptNet. The first element of $\mathbf{v}$ is the dot-product of individual embeddings. The remaining elements are described next.

ConceptNet contains a lot of information about relations of word pairs. For example,

car — SimilarTo — automobile
sleep — Antonym — awake

Figure 3 contains 51 relationship features (non-underlined) extracted from ConceptNet that we use in this work. They reflect human judgments of similarity and are documented in [15, 28] and an online interface for exploring them is also available[2]. Their availability and proper usage can significantly improve NLP systems. For example, for the WPS task, they allow us to answer questions such as “Which relationships connect words w_1 and w_2 with each other?” Specifically, for the pair (cat, dog), we can find out whether or not they satisfy relationships like

cat—NotDesires—dog
cat—LocatedNear—dog

[2] http://conceptnet.io/

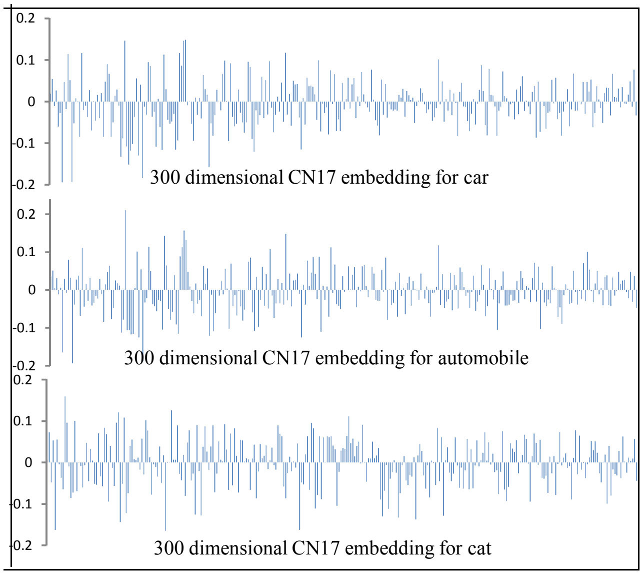

Figure 2: Individual word pair embeddings for car, automobile and cat. Dot product of embeddings of cat and automobile is 0.8933. Dot product of embeddings of car and cat is 0.1257.

The utility of all features for computing WPS is self-evident except that of ExternalURL. It captures similarity implicitly by recording the fact that w_1 and w_2 are considered related to each other by some knowledge source external to ConceptNet. For example, the URL "sw.opencyc.org" relates "car" to "automobile" and therefore our ExternalURL flag is set to 1.

car — ExternalURL — *sw.opencyc.org* automobile

For each of the non-underlined 51 relationships in Figure 3, we compute Boolean flags indicating whether w_1 and w_2 have that relationship or not. This gives us a 51 dimensional binary vector.

In addition to these 51 relational features, we construct a feature that we call TwoPhraseRelation. This feature is not available in ConceptNet as is but derived using two different ConceptNet relations. If w_1 and w_2 are connected by RelatedTo or IsA features using the information contained in two-word phrases then this flag will be 1, otherwise it will be 0. For example, for the pair (man, boy), each of the relationships

man — IsA — old boy
man — RelatedTo — adult boy
man — RelatedTo — grown boy

can cause this flag to be set to 1. After combining all features, our proposed WPE **v** consists of 53 values. Pseudocode of the WPE construction process is presented in Algorithm 1.

It can be observed that some entries in the 51 non-underlined features in Figure 3 capture more or less the same relationships. Therefore, the resulting embeddings will be redundant. However, we utilized all of the extracted features due to the following two reasons:

i. As mentioned above some examples of features show that ConceptNet features are interesting and can contribute in computation of WPS. Instead of manually choosing different features according to their contribution in WPS,

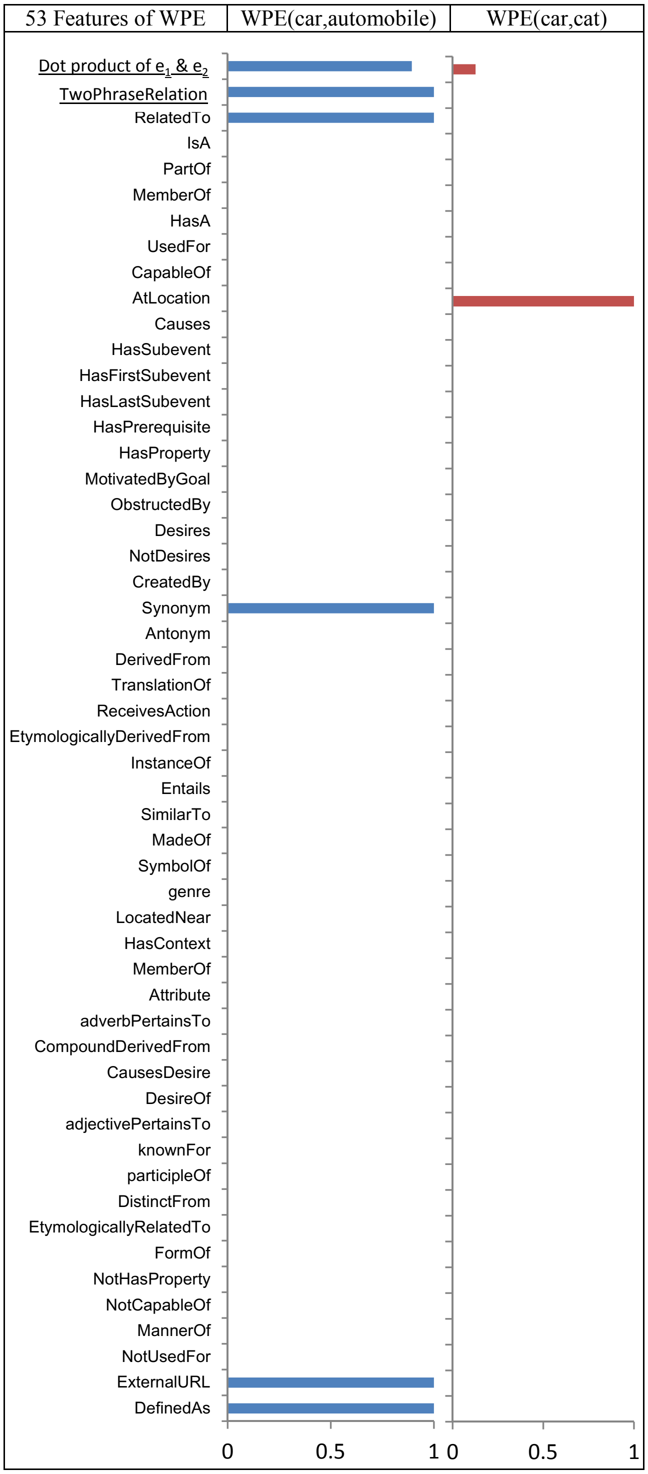

Figure 3: Examples of 53 dimensional word pair embeddings for the pairs (car,automobile) and (car,cat). Each dimension has a value ranging from 0 to 1.

Algorithm 1: Construction of WPE

```
    Input: two words $w_1$ and $w_2$, embedding dataset E, F
        //F is the set of 51 non-underlined features in Figure 3
    Output: word pair embedding $\mathbf{v}$
1   $\mathbf{e_1}$ = E($w_1$)
2   $\mathbf{e_2}$ = E($w_2$)
3   $\mathbf{v}$ = zero vector of length 2+|F|
4   S = set of all relations between $w_1$ and $w_2$ found in ConceptNet
5   for i = 1 to |S|
6       j = find(F, $s_i$)
7       $\mathbf{v}$(j+2) = 1
8   end
9   $\mathbf{v}$(1) = dot-product of $\mathbf{e_1}$and $\mathbf{e_2}$
10  R = {‘RelatedTo’, ‘IsA’}
11  for i = 1 to |R|
12      T = set of two-word phrases having relation $r_i$ with $w_1$ in
            ConceptNet (see page 3 for two-word phrases)
13      if T contains $w_2$
14          $\mathbf{v}$(2) = 1
15          break
16      end
17  end
18  if $\mathbf{v}$(2) is 0
19      Repeat lines 11 to 17 with roles of $w_1$ and $w_2$ swapped
20  end
21  return $\mathbf{v}$
```

it is better to leave this decision on subsequent machine learning models.

ii. Since ConceptNet is crowd sourced and is growing continuously, we cannot currently state that it correctly and completely represents all relational information between words. The probability of missing features is not zero. Utilization of all features provides robustness to missing features. For example, Synonym, SimilarTO and IsA capture similar ideas. If any one of them is missing, the others can substitute for it.

C. Computational complexity

Since our embeddings are computed by extracting information from ConceptNet, we analyze the time complexity of Algorithm 1 in terms of the number of relations between w_1 and w_2 found in ConceptNet. In line 4, ConceptNet is queried[3] to extract some relations that we store in set S. Each of the |S| iterations of lines 5-7 performs a search within a constant number of |F| entries. So the time complexity of lines 5-7 is O(|S|). At line 12, set T is also filled by querying ConceptNet. Line 13 corresponds to searching within |T| entries. So the time complexity of lines 11-17 is O(|T|). Therefore, the whole algorithm is linear in terms of the relations extracted from ConceptNet.

III. Experiments

To validate our hypothesis that word pair embeddings are more suitable for WPS computations, we perform experiments on two standard datasets. The first dataset is the MEN dataset [29] which contains 3000 word pairs with their crowd sourced similarity judgments. The second dataset is the WordSim-353 dataset [30] which provides 353 word pairs with their similarity judgments.

We obtained word pair embeddings for every pair using the method described in Section II. To train and test machine learning models for learning the mapping between embeddings and similarity values, we used development, validation and testing sets for each dataset. For the MEN dataset, we used the default development and testing splits that are already available. We applied a similar split to WordSim-353. Specifically, we selected every third pair for testing and remaining pairs were included in the development set. For cross validation, we further divided the development sets of both datasets randomly into training and validation sets.

For completeness, we mention that in the datasets that we used, there was only one “out of vocabulary” word “maradona” in WordSim-353. Specifically, no embedding for “maradona” was available in CN17 and CN16, although Glove42B contained an embedding for it. To obtain embeddings of “maradona” in CN17 and CN16, we carried out the following procedure. First, we found its most similar term with the longest prefix as “maradonian” in ConceptNet. We used this term to extract the 52 relational features. Unfortunately, CN17 and CN16 do not provide embeddings of “maradonia” as well. Therefore, we computed the base word for “maradona” which comes out to be “maradona” as well. Therefore, we eventually used web-based information to extract its nearest word as “footballer”. This is similar to the approach in [32]. Finally, embeddings for “footballer” were used to substitute for “maradona” in CN17 and CN16.

We compared results [Table 2, Table 3] of our embeddings with results of two well-known embeddings [15, 16, 27]. We downloaded two different embeddings of ConceptNet Numberbatch and one of Glove 42B available online. Before any similarity measurement using Glove embeddings, we first normalized each feature across the vocabulary as proposed in [27]. We reproduced their results using their embeddings only. Our reported reproduced results matched the published results with minor difference.

To learn the mapping from our word pair embeddings to human marked similarity judgments, we trained a regularized linear regression model and a non-linear neural network model. For the linear regression model, we cross-validated the regularization parameter in the range 0.001 to 1000 using 500 values at uniform intervals in log space. For the neural network model, we used the Matlab Neural Network Toolbox to train using Bayesian regularization. We used a lightweight neural network with one hidden layer containing just three neurons.

[3] http://api.conceptnet.io/query?node=/c/en/w_1&other=/c/en/w_2

TABLE 2: WPS FOR MEN DATASET USING DIFFERENT EMBEDDINGS AND DIFFERENT MODELS. SEE TEXT FOR DETAILS.

Embeddings	**Model**	**Spearman's correlation coefficient**		**Pearson's correlation coefficient**	
		Dev	**Test**	**Dev**	**Test**
CN17	CS	0.8574	0.8680	0.8414	0.8529
CN17+WPE	LR	0.8730	**0.8794**	0.8599	**0.8632**
CN17+WPE	NN	0.8797	**0.8830**	0.8848	**0.8851**
CN16	CS	0.8538	0.8612	0.8365	0.8451
CN16+WPE	LR	0.8723	**0.8736**	0.8568	**0.8562**
CN16+WPE	NN	0.8797	**0.8794**	0.8840	**0.8820**
Glove42	CS	0.8127	0.8143	0.7860	0.7885
Glove42+WPE	LR	0.8090	0.8109	0.6185	0.6079
Glove42+WPE	NN	0.8590	**0.8538**	0.8618	**0.8544**

TABLE 3: WPS FOR WORDSIM-353 DATASET USING DIFFERENT EMBEDDINGS AND DIFFERENT MODELS. SEE TEXT FOR DETAILS.

Embeddings	**Model**	**Spearman's correlation coefficient**		**Pearson's correlation coefficient**	
WordSim-353		Dev	Test	Dev	Test
CN17	CS	0.8137	0.8385	0.7485	0.7689
CN17,WPRS	LR	0.8318	**0.8524**	0.7812	**0.7996**
CN17,WPRS	NN	0.8446	**0.8435**	0.8188	**0.8202**
CN16	CS	0.8153	0.8610	0.7541	0.7909
CN16	LR	0.8327	**0.8722**	0.7862	**0.8176**
CN16,WPRS	NN	0.8401	**0.8720**	0.8144	**0.8434**
Glove42	CS	0.7469	0.8200	0.6791	0.7577
Glove42,WPRS	LR	0.7369	0.7986	0.5757	0.5679
Glove42,WPRS	NN	0.8178	**0.8649**	0.7972	**0.8455**

To evaluate any WPS technique, we compute the Spearman and Pearson correlation coefficients between predicted and actual similarity values. In Tables 2 and 3, CN17 refers to word embeddings corresponding to numberbatch-en-17.04b, CN16 refers to embeddings corresponding to conceptnet-numberbatch-201609_en_main[4], Glove42 refers to embeddings corresponding to [27] and WPE refers to our word pair embeddings. CS refers to direct computation of cosine similarity between individual word embeddings. This does not require any training. LR refers to similarity output by a linear regression model trained on our word pair embeddings. NN refers to similarity output by a neural network model trained on our word pair embeddings. Finally, Dev and Test refer to the development and test sets respectively. It can be seen in both tables that in most of the cases similarity values obtained from linear regression and neural network models trained on word pair embeddings have higher correlation with human estimates compared to direct cosine similarity of individual word embeddings. Neural networks performed better than linear regression in all instances except one. This can be attributed to the ability of linear regression to model only linear relationships between input and output. In contrast, neural networks can model nonlinear mappings. Linear regression models trained on our WPEs computed using Glove42 embeddings performed surprisingly worse than traditional cosine similarity. However, performance improved when CN17 and CN16 embeddings were used. This indicates the superiority of ConceptNet embeddings over Glove42 for both direct cosine similarity and learned WPS. Moreover, neural network models always learned better similarities than previous approaches based on single word embeddings.

IV. Conclusions

To solve the WPS problem, we have introduced embeddings at the level of word pairs. These embeddings use existing embeddings of individual words and append additional features specific to the pair. These additional features include relational similarity extracted from ConceptNet. We train supervised machine learning models on our embeddings to learn the mapping from word pair embeddings to human marked similarity values. Our results on the MEN and WordSim-353 datasets show that word pair embeddings are better than individual word embeddings for WPS computation.

In this work, we have only used relational features between words. As explained in Section II, there exists attributional similarity between words as well. An interesting future direction would be to incorporate attributional commonalities for the WPS task.

Acknowledgment

The authors would like to thank Abdul Mateen and Kamran Malik for providing helpful insights into the problem.

References

[1] Jiang Zhao, Man Lan, Zheng-Yu Niu, Yue Lu: Integrating word embeddings and traditional NLP features to measure textual entailment and semantic relatedness of sentence pairs. IJCNN 2015:1-7

[2] Hongzhe Liu, Pengfei Wang: Assessing Sentence Similarity Using WordNet based Word Similarity. JSW 8(6):1451-1458 (2013)

[3] Maria Soledad Pera, Yiu-Kai Ng: A naïve Bayes Classifier for Web Document Summaries Created by Using Word Similarity and Significant Factors. International Journal on Artificial Intelligence Tools (IJAIT) 19(4):465-486 (2010)

[4] J. Curran, "Ensemble menthods for automatic thesaurus extraction," EMNLP, pp.222–229, 2002.

[5] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using smart: Trec 3. In Proc. of 3rd Text REtreival Conference, pages 69–80, 1994.

[6] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In Proc. of 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pages 206–214, 1998.

[7] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In Proc. of 14th International Joint Conference on Aritificial Intelligence, 1995.

[8] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In Proc. of the International Conference on Research in Computational Linguistics ROCLING X, 1998.

[4] Embeddings for CN17 and CN16 were downloaded from https://github.com/commonsense/conceptnet-numberbatch

[9] D. Lin. Automatic retreival and clustering of similar words. In Proc. of the 17th COLING, pages 768–774, 1998.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space CoRR abs/1301.3781 (2013).

[11] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," IJCAI, pp.448–453, 1995.

[12] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," in Proceedings of NAACL HLT, Apr. 2013.

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in NIPS 2013: Advances in neural information processing systems, Oct. 2013.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in Proceedings of Workshop at ICLR, Jan. 2013

[15] Robert Speer, Joshua Chin, Catherine Havasi: ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. AAAI 2017:4444-4451

[16] Robert Speer, Joshua Chin: An Ensemble Method to Produce High-Quality Word Embeddings. CoRR abs/1604.01692 (2016)

[17] Robert Speer and Joanna Lowry-Duda. 2017. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, Vancouver, Canada, pages 85–89. http://www.aclweb.org/anthology/S17-2008.

[18] Ming-Hung Hsu, Ming-Feng Tsai, Hsin-Hsi Chen: Query Expansion with ConceptNet and WordNet: An Intrinsic Comparison. AIRS 2006:1-13

[19] Rajul Anand, Alexander Kotov: An Empirical Comparison of Statistical Term Association Graphs with DBpedia and ConceptNet for Query Expansion. FIRE 2015:27-30

[20] Lin Dai, Heyan Huang: An English-Chinese Cross-lingual Word Semantic Similarity Measure Exploring Attributes and Relations. PACLIC 2011:467-476

[21] D.L. Medin, R.L. Goldstone, and D. Gentner. Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. Psychological Science, 1(1): 64-69, 1990.

[22] Arbi Bouchoucha, Jing He, Jian-Yun Nie: Diversified query expansion using conceptnet. CIKM 2013:1861-1864

[23] Junpeng Chen, Juan Liu: Combining ConceptNet and WordNet for Word Sense Disambiguation. IJCNLP 2011:686-694

[24] Peter D. Turney: Measuring Semantic Similarity by Latent Relational Analysis. IJCAI 2005:1136-1141

[25] Preslav Nakov, Zornitsa Kozareva: Combining Relational and Attributional Similarity for Semantic Relation Classification. RANLP 2011:323-330

[26] G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. McGrawHill, New York, 1983.

[27] Jeffrey Pennington, Richard Socher, Christopher D. Manning: Glove: Global Vectors for Word Representation. EMNLP 2014:1532-1543

[28] https://github.com/commonsense/conceptnet5/wiki/Relations

[29] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. J. Artif. Intell. Res. (JAIR), 49:1–47.

[30] Minh-Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. CoNLL-2013, 104.

[31] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In HLT-NAACL, pages 758–764.

[32] Fanghuai Hu, Zhiqing Shao, Tong Ruan: Self-Supervised Synonym Extraction from the Web. J. Inf. Sci. Eng. (JISE) 31(3):1133-1148 (2015).

[33] Tobias Schnabel, Igor Labutov, David M. Mimno, Thorsten Joachims: Evaluation methods for unsupervised word embeddings. EMNLP 2015:298-307.

[34] Hua He, Jimmy J. Lin: Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement. HLT-NAACL 2016:937-948