Specialising Paragraph Vectors for Text Polarity Detection

Fabio Tamburini
FICLIT, University of Bologna, Italy
fabio.tamburini@unibo.it

Abstract

This paper presents some experiments for specialising Paragraph Vectors, a new technique for creating text fragment (phrase, sentence, paragraph, text, ...) embedding vectors, for text polarity detection. The first extension regards the injection of polarity information extracted from a polarity lexicon into embeddings and the second extension aimed at inserting word order information into Paragraph Vectors. These two extensions, when training a logistic-regression classifier on the combined embeddings, were able to produce a relevant gain in performance when compared to the standard Paragraph Vector methods proposed by Le and Mikolov (2014).

Keywords: Text Polarity Detection, Word Embeddings, Paragraph Vectors, Polarity Lexica.

1. Introduction

Distributed word representations, built starting from the well-known Harris (1954) distributional hypothesis often stated as "*The degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear.*" and coded as sparse high dimensional vectors, have been playing a central role in Natural Language Processing (NLP) for years (Turney and Pantel, 2010; Baroni and Lenci, 2010). More recently, a large set of studies propose to represent words as dense vectors derived by training neural networks for language modelling (Bengio et al., 2003; Collobert and Weston, 2008; Mnih and Hinton, 2009; Mikolov et al., 2013). Such vectors are commonly called "word embeddings" and have been successfully used in a variety of NLP tasks (e.g. Turian et al. (2010), Collobert et al. (2011), Socher et al. (2013)).

These approaches and the cited applications use word-centered embeddings and are not suited, in general, for tasks requiring the classification of an entire text such as topic classification, sentiment analysis, language identification, etc. Le and Mikolov (2014) proposed an extension of a state-of-the-art word embedding method, namely the widely used `word2vec` package[1] (Mikolov et al., 2013), to create distributed dense representations for phrases, sentences, paragraph or even entire texts useful to be used as base vectors for any fragment/text classification task in the NLP domain.

Both standard word embeddings and the "Paragraph-Vector" (PV) model from Le and Mikolov produce vectors that do not contain any information about a specific NLP task, but, as showed by all works cited before, they indeed support the production of state-of-the-art classification results in various domains.

In this paper we propose to extend PVs in order to specialise them for text polarity classification by injecting specific knowledge connected with this task; we will show in some experiments as the combination of embeddings and external resources will produce increased performances. The idea of combining word embeddings and external knowledge to specialise them for a specific task has been actively explored in the last few years (Wang et al., 2014; Levy and Goldberg, 2014; Tang et al., 2014; Yu and Dredze, 2014; Wang et al., 2015; Pham et al., 2015; Liu et al., 2015), but all these studies extend the embeddings at word level. As far as we know, this is the first attempt to extend PVs with external knowledge.

[1] `https://code.google.com/p/word2vec/`

2. word2vec and PV models

Our work is grounded on the study presented in (Mikolov et al., 2013): they proposed two different word embedding models, namely CBOW and Skip-gram, and provide a very efficient implementation of them. The CBOW model uses the average context-words embeddings to predict the word in the middle of the context. The Skip-gram uses only one word to predict an entire context surrounding this word. See figure 1 for a schematic view of these models. Both methods use logistic regression to predict the target word and can apply hierarchical softmax or negative sampling techniques to improve classification efficiency (see (Rong, 2014) for an in depth description of the mathematical details of both approaches).

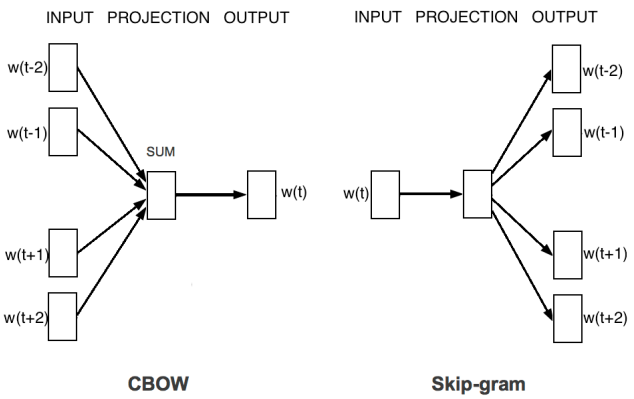


Figure 1: Word embedding models presented in Mikolov et al. (2013).

Le and Mikolov (2014) extended these approaches to build embeddings able to capture syntactic and semantic information from a complete text fragment (phrase, sentence, paragraph, text, ...), the PV models. The extension is quite simple: insert one artificial word at the beginning of each

text/fragment and apply the CBOW or Skip-gram techniques in the same way as before but considering the entire text/fragment as the context for the artificial word. At the end of the learning process the embedding vector corresponding to the artificial word is the PV representing the entire text/fragment, and can be successfully used in text classification tasks. Depending on which previous technique is extended, we obtain respectively the PV-DM and the PV-DBOW models showed in figure 2.

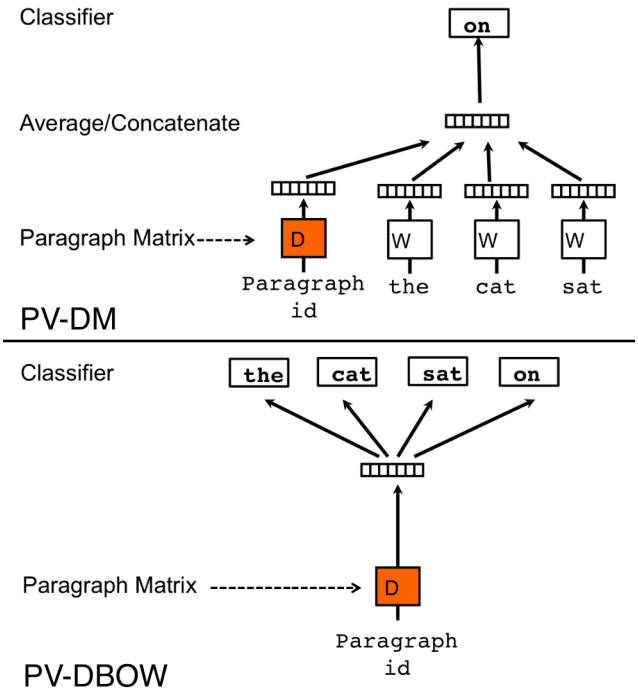


Figure 2: Paragraph Vector models presented in Le and Mikolov (2014).

All the models described before do not preserve word order information (Qiu et al., 2014; Lai et al., 2015; Trask et al., 2015) and do not contain any specific information useful for solving a given task. We propose to further extend PV models to insert such kind of information and specialise them for text polarity detection.

3. The proposed model

Our proposed model is based on two different ways of extending previous PV models. As in (Le and Mikolov, 2014), the PV embeddings obtained by these two extensions were concatenated (with the operator $\oplus$) to build the final embeddings used in our experiments.

3.1. First extension: Polarity Injection

There are various studies in literature aiming to extend the Skip-gram model in order to inject some kind of linguistic knowledge inside word embeddings (Yu and Dredze, 2014; Wang et al., 2014; Pham et al., 2015; Liu et al., 2015): all of them modify the standard Skip-gram objective function over the sequence of words $W = w_1, ..., w_M$

$$L(W) = \sum_{i=1}^{M} \sum_{c \in context} \log P(w_{i+c}|w_i),$$

where $P(\cdot)$ is computed using either a softmax function or the negative sampling technique, by adding a regulariser term $R(W, K)$ based on the external knowledge K in a weighted way

$$O(W, K) = (1 - \xi)\, L(W) + \xi\, R(W, K).$$

We propose to use a polarity lexicon, classifying words either as positive (+1) or negative (-1), or within the [-1,1] interval of real values, as specific knowledge to inject into the PV-DBOW model in order to specialise it for text polarity classification and define the regulariser function as

$$R(W, \pi) = \sum_{i=1}^{M} \sum_{c \in context} \left(\frac{|\pi(w_c) - \pi(w_{i+c})|}{2} + \frac{cs(w_c, w_{i+c}) + 1}{2} - 1 \right)^2,$$

where $\pi(w)$ is the word polarity value extracted from the polarity lexicon, and $cs(w_1, w_2)$ is the cosine similarity between word embeddings. The idea behind this regulariser function is to penalise words that have similar lexical polarity values but distant word embeddings, or, that is to say, words in the same polarity class, or exhibiting similar continuous polarity values, should receive similar word embeddings. These information should propagate to PVs giving the same results: text with similar polarities should receive similar PVs. The regulariser function is computed at each step over all words considered by the negative sampling technique used to compute $P(\cdot)$.

3.2. Second extension: Order preserving embeddings

As we said before all the word embedding methods we described do not preserve any information about word order into the produced embeddings, but Landauer (2002) estimated that about 20% of the meaning contained in a text derives from word order. The loss of word order caused by such "bag-of-word" models is particularly problematic for sentiment classification (Johnson and Zhang, 2015), thus the second extension to the standard PV models we propose is devoted to introduce some ordering information into the model.

In literature, there is a large bundle of studies for injecting some level of word order information into word embeddings (Qiu et al., 2014; Johnson and Zhang, 2015; Lai et al., 2015; Ling et al., 2015; Trask et al., 2015) or into general word-space models (Sahlgren et al., 2008; De Vine and Bruza, 2010; Basile et al., 2011). In particular we refer to the work of (Trask et al., 2015) that extends the CBOW model proposing to split the context into two partitions, the first containing the sum of all the context word embeddings before the target word and the other containing the sum of all the word embeddings after.

We are working on PVs and the corresponding model, namely PV-DM, uses an asymmetric context window containing only the words before the target word and summing them together with the text PV (see figure 2). Then, we split the context window into three partitions: the first contains the sum of the embeddings for the two words immediately

preceding the target word, the second the sum of the remaining word embeddings inside the window and the third the PV for the examined text. See figure 3 for a schematic view of the order preserving model.

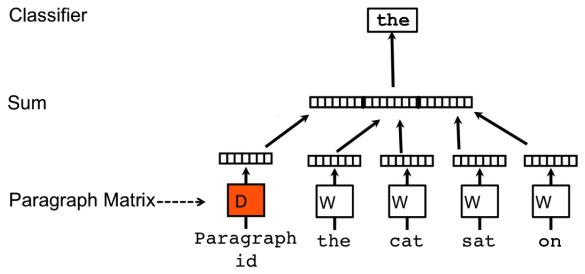


Figure 3: Order Preserving PV-DM model. We have three partitions: the first contains the sum of the embeddings for the two words immediately preceding the target word, the second the sum of the remaining word embeddings inside the window and the third the PV for the examined text/fragment.

4. Experiments

We performed some experiments for evaluating the proposed model using a standard benchmark for this field, the IMDB review dataset (Maas et al., 2011): this database contains 100.000 movie reviews, and 50.000 of them were classified with positive/negative polarity labels and divided into training and test set. Both sets contains an equal number of reviews and are also balanced with respect to polarity classes. Our experiments follow the same procedures described in (Le and Mikolov, 2014): we trained our models using the training set, consisting of 25.000 texts, and the 50.000 unclassified reviews and test them on the test set, containing the remaining 25.000 reviews, using the same logistic-regression classifier. As in the original experiments, we produced vector representations with 400 dimensions computed using a 10-word window with 5 negative sample words.

We chose to use the English polarity lexicon produced in (Hu and Liu, 2004) as extern knowledge source: it contains about 6.800 words with binary (+1,-1) polarity labels.

The original paper on PVs from Le and Mikolov (2014) reported an error rate of 7.42%, but subsequent experiments (Mesnil et al., 2015) declared that "*92.6% accuracy result only when the training and test data are not shuffled. Thus, we consider this result to be invalid.*". We verified this claim obtaining the same results, and then we completely agree with this position. Mesnil et al. (2015) provide also a correct evaluation on the same dataset for the original PV model properly shuffling the entire dataset. Actually, all these approaches are very sensitive to dataset shuffling, that is why we repeated all the experiments presented in this paper on 5 different shufflings averaging the results.

The first set of experiments were devoted to analyse the impact of the first enhancement we proposed, namely the injection of polarity values into the computed embeddings. Figure 4 outlines the Error Rate variation as a function of ξ: the profile exhibits a plateau between the values 0.05 and 0.20, thus we chose to perform all the subsequent experiments using $\xi = 0.12$.

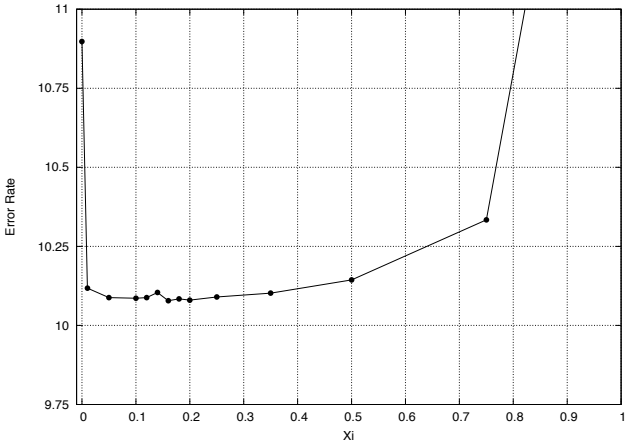


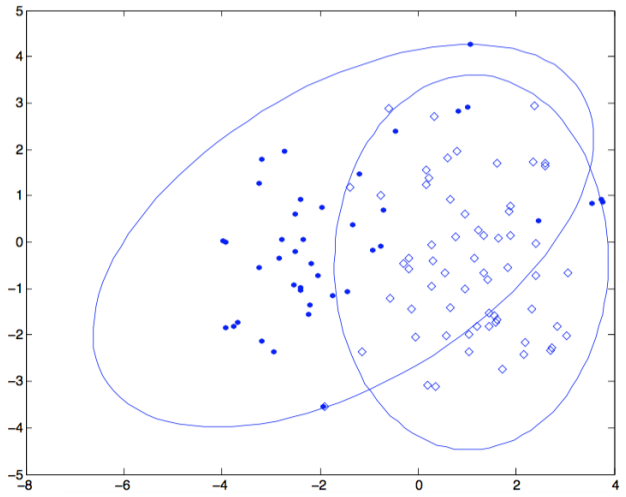
Figure 4: Mean Error Rate variation as a function of ξ for the first extension (LexPolarity PVs). The value for $\xi = 0$ is equivalent to the original algorithm.

In order to further test the effectiveness of this approach to inject information into embeddings we used the t-SNE package[2] (van der Maaten and Hinton, 2008) to project the high-dimensional embeddings into a 2D space in order to visually verify if the proposed method effectively helps in grouping the embedding of words exhibiting similar polarity values. We selected 106 target words from the lexicon having different polarities, projected their vectors into a bidimensional space using t-SNE and approximated the two clusters with multivariate gaussians in order to better understand the distributions. Figure 5 shows the two sets of points and the isoprobability ellipses corresponding to the 95% of the data for each cluster with, (b), or without, (a), the application of the proposed method. As we can see from the figure, the injection of polarity data tend to create better clusters of similar embeddings thus propagating such information into the embedding themselves. Given the small differences in Error Rate reduction for the proposed method, it is very difficult to make a similar visualisation directly on PVs even if it is evident, by looking at the performances, that some of the clustering properties showed by the proposed method on word embeddings affect also PVs.

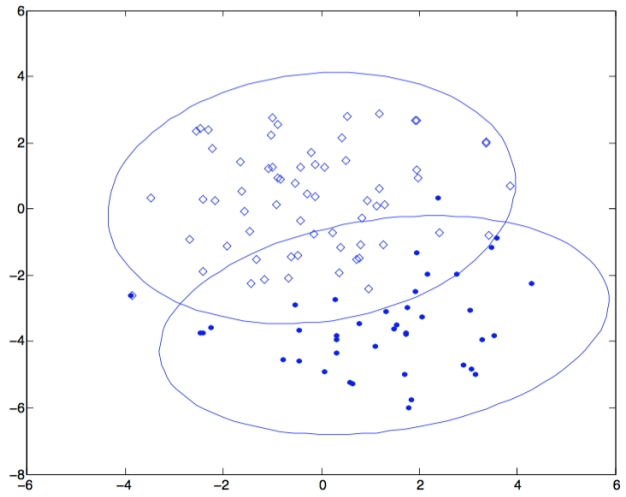
Table 1 outlines the final results of our experiments for the various enhancements we proposed compared with the original and correct results obtained by (Mesnil et al., 2015).

Following the suggestions of one of the anonymous reviewers, we devised some further experiments. We tested if the injection of polarity information extracted from an external lexical resource into PVs would achieve better results than simply combining the original PVs method with one simple lexicon-based sentiment-analysis method by interpolating or integrating the results of these two different techniques.

[2] https://lvdmaaten.github.io/tsne/



(a)



(b)

Figure 5: The bidimensional projections of word embeddings corresponding to some positive (diamonds) and negative (points) words extracted from the lexicon (a) for the original algorithm from (Le and Mikolov, 2014) and (b) for the proposed method for injecting polarities into embeddings.

We chose the VADER tool[3] (Hutto and Gilbert, 2014), a rule-based classifier for social media and web texts using a manually created polarity lexicon, as the reference tool for comparing the different techniques. We adopted such tool because, despite the simplicity of the approach, it contains a very good and manually checked polarity lexicon and obtained state-of-the-art results in text polarity classification. We devised a series of new experiments: sometimes, we considered only one of the five shufflings of the previous tests because VADER is not affected by text order:

- First of all we simply applied VADER for assigning a polarity value to each review in the test set obtaining an Error rate = 30.22, much higher than those obtained in all the PV experiments.
- In the second experiment we calculated the polarity of each text transforming the probability assigned by the logistic-regression classifier trained with the embeddings produced by the original PVs method into a continuous polarity value ($Pol = (P(1) - 0.5) * 2$, where $P(1)$ is the probability assigned to the positive polarity). Then we averaged it with the VADER polarity value, obtaining a mean Error rate = 18.56, still much higher than the values obtained in all the PVs experiments.
- The third experiment consisted in training the logistic-regression classifier using the original PVs adding the polarity values provided by VADER as an additional feature to the PVs, obtaining a mean Error Rate = 10.72.
- In the last experiments we replaced the lexicon from (Hu and Liu, 2004) we used in the previous set of experiments on PVs (see Table 1) with the VADER lexicon, in order to derive consistent results, and repeated all the tests on the five shufflings considered before (with $\xi = 0.12$). Combining the VADER-LexPolarity PVs and the Order Preserving PV, we obtained a mean Error Rate = 9.92.

[3] https://github.com/cjhutto/vaderSentiment

Method	mER	SD
Original PVs (Mesnil et al., 2015)	10.70	-
Original PVs (using our 5 shufflings)	10.90	0.13
Liu-LexPolarity PVs	10.09	0.16
Liu-LexPolarity⊕Order Preserving PVs	**9.89**	0.12

Table 1: The results of our experiments as mean Error Rate (mER) and Standard Deviation over the 5 shufflings. We present first the results obtained injecting only the lexical-polarity information contained in the (Hu and Liu, 2004) lexicon (with ξ=0.12) and then the results of both extension we proposed.

Table 2 summarises the results of all the experiments performed integrating the VADER classifier with the original PV method in various ways.

Method	mER	SD
Orig. VADER (Hutto and Gilbert, 2014)	30.22	-
Average of the Orig. PVs pol. & VADER pol.	18.56	0.08
LogReg Class. on Orig. PVs⊕VADER pol.	10.72	0.13
VADER-LexPolarity PVs	10.10	0.13
VADER-LexPolarity⊕Order Preserving PVs	9.92	0.11

Table 2: Results of all the experiments performed integrating the VADER classifier with the original PVs method in various ways. The last two experiments reproduce the same experiments presented in Table 1 but using the VADER lexicon.

5. Discussion and Conclusions

The results obtained by injecting information from a polarity lexicon and some word order information in the PV

model proposed by (Le and Mikolov, 2014) are quite encouraging, showing a relevant gain when compared to the original PV results on the same dataset.

In order to further validate such results we performed an extended set of experiments combining the original PVs method with a simple lexicon-based sentiment-analysis algorithm in various ways, showing that our approach produces better results that a simple system combination technique.

There are other studies in literature (e.g. (Mesnil et al., 2015) that combine various techniques for deriving word embeddings, namely n-grams, Recurrent Neural Networks-Language Models, PVs and Naive Bayes-SVM, in a unique ensemble of generative and discriminative techniques) obtaining better results than the ones presented in this study. Nevertheless, the aim of this paper is not to present the best solution for solving the text polarity classification problem on the IMDB review dataset, but to investigate possible forms of extensions for the PV models in order to specialise them for a specific task.

On the one side we obtained an increase in performances for the examined task, but, on the other side, the PVs will be bound to this task and they will loose their ability to be applied to different tasks without any modification. We believe that this is a price to be paid for increasing system performances adapting them to the required task.

Our future plans envisage the testing of the proposed models on different datasets, different kind of text types (e.g. on tweets) and on different languages.

6. Acknowledgements

We wish to thank one of the anonymous reviewers for the precious suggestions for enlarging the set of experiments used to validate the proposed method.

7. Bibliographical References

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36:673–721.

Basile, P., Caputo, A., and Semeraro, G. (2011). Encoding syntactic dependencies by vector permutation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 43–51, Edinburgh, UK.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning, ICML*, pages 160–167.

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

De Vine, L. and Bruza, P. D. (2010). Semantic oscillations : encoding context and structure in complex valued holographic vectors. In *2010 AAAI Fall Symposium Series : Quantum Informatics for Cognitive, Social, and Semantic Processes*, Arlington, Virginia, December. AAAI Press.

Harris, Z. (1954). Distributional structure. *Word*, 10:146–162.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pages 168–177, Seattle, WA, USA.

Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, Ann Arbor, MI, USA.

Johnson, R. and Zhang, T. (2015). Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, Denver, Colorado.

Lai, S., Liu, K., Xu, L., and Zhao, J. (2015). How to generate a good word embedding? *CoRR*, abs/1507.05523.

Landauer, T. K. (2002). On the computational basis of learning and cognition: Arguments from lsa. In N. Ross, editor, *The psychology of learning and motivation*, volume 41, pages 43–84. Elsevier.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, pages 1188–1196, Beijing, China.

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland.

Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado.

Liu, Y., Liu, Z., Chua, T.-S., and Sun, M. (2015). Topical word embeddings. In *29th AAAI Conference on Artificial Intelligence*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA.

Mesnil, G., Mikolov, T., Ranzato, M., and Bengio, Y. (2015). Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. In *Proceedings of the International Conference on Learning Representations, ICLR2015*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Mnih, A. and Hinton, G. (2009). A scalable hierarchical

distributed language model. In *Advances in Neural Information Processing Systems*, volume 21, pages 1081–1088.

Pham, N. T., Lazaridou, A., and Baroni, M. (2015). A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 21–26, Beijing, China.

Qiu, L., Cao, Y., Nie, Z., Yu, Y., and Rui, Y. (2014). Learning word representation considering proximity and ambiguity. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1572–1578.

Rong, X. (2014). word2vec parameter learning explained. *CoRR*, abs/1411.2738.

Sahlgren, M., Holst, A., and Kanerva, P. (2008). Permutations as a means to encode order in word space. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society - CogSci 2008*, pages 1300–1305.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, Baltimore, Maryland.

Trask, A., Gilmore, D., and Russell, M. (2015). Modeling order in neural word embeddings at scale. In David Blei et al., editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2266–2275.

Turian, J., Ratinov, L.-A., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

van der Maaten, L. and Hinton, G. (2008). Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, pages 2579–2605.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar.

Wang, Y., Liu, Z., and Sun, M. (2015). Incorporating linguistic knowledge for learning distributed word representations. *PLoS ONE*, 10(4):e0118437.

Yu, M. and Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550, Baltimore, Maryland.