

Simple Alignment Sentence Classification for Aspect-Based Sentiment Analysis

Hiroataka Niitsuma, Ryo Asakura, Manabu Ohta

Graduate School of Natural Science and Technology, Okayama University
Okayama, 700-0530 Japan
niitsuma@de.cs.okayama-u.ac.jp, asakura@de.cs.okayama-u.ac.jp,
ohta@de.cs.okayama-u.ac.jp

Abstract. Aspect-based sentiment analysis (ABSA) is the task of identifying the polarities on various aspects of a given sentence. We propose a simple neural network model for supervised learning for ABSA. The proposed simple neural network model is actually single-layer neural network whose input is a list of word vectors. However, an attention network is attached to the simple neural network. The attention network provides properties of aspects of a sentence to the simple neural network. We evaluated the neural network model using the ABSA task in SemEval 2016 restaurant domain. The proposed model outperforms both convolutional neural network (CNN) -based and recurrent neural network (RNN) -based models. Since the simple neural network is a single-layer neural network, this model has only one parameter which is the vector dimension of the word vectors. When initial word vector is chosen appropriately, this model does not have any parameters.

1 Introduction

Sentiment extraction from text has many applications[1, 2]. The task of sentiment extraction is also known as sentiment analysis[3] or opinion mining[4]. Sentiment analysis[1, 2] is the task of assigning binary positive/negative labels to given sentences. However, a simple sentiment analysis only provides total evaluation of the given sentences. Usually, multiple and complex polarities are required for summarizing opinions of some given sentences. Aspect-based sentiment analysis (ABSA)[4] is the task of assigning positive/negative labels to a sentence for some aspects. For example, let us consider the following restaurant review:

“Food is good, but attitude of clerk is so rude.”

This review shows a positive feeling about the “food” and a negative feeling about the “attitude of clerk”. ABSA is the task of assigning a positive label to the “food” aspect and a negative label to the “attitude of clerk” aspect for this sentence.

In this research, we propose neural network models for ABSA. Our proposed models are evaluated based on benchmarks of the ABSA task in the SemEval 2016 Task 5 Subtask 1 (SE16T5S1) ¹.

This task has two sub-tasks. The first sub-task is extracting aspects from each sentence. The second sub-task is assigning sentiment labels to the extracted aspects. For the aspect extraction step, recurrent neural network (RNN) -based approach [5] outperformed existing models[6–11]. The existing models strongly depend on a lexical database for dictionary-based features. This is because the simple inference model is insufficient to capture numerous aspects. The RNN model[5] can capture several aspects without the need for a lexical database for dictionary-based features. The models achieved the highest scores in some subtasks in SE16T5S1. The RNN model was similar to the work of Collobert et al.[12]. In [12], each sub-sentence inside a sliding window is converted into an n-gram. The RNN model also used the n-grams-based feature as an input.

For the sentiment identification step, a neural network-based approach has been used [13–16]. In particular, sequential convolutional neural networks(CNNs) [12, 17] achieved good results for ABSA. Especially, [14, 13, 18] achieved the state-of-the-art performance for ABSA using CNN. Usually, the neural-attention mechanism performs well in NLP tasks. An RNN model with neural-attention was used in [16] for ABSA; however its performance was not as good as the top-ranked teams in SE16T5S1.

In [5], we used a recurrent convolutional network (RCNN) model for ABSA. RCNN is a combination of a convolution layer and an RNN [12]. RCNNs are used in several tasks: character-level text classification[19], image classification[20] and sentiment identification in videos[21]. However, RCNN model has many parameters.

In this research, we show that a simple neural-attention mechanism can achieve better results than RNN and CNN. The simple model that has only one parameter.

2 Related Work

RNNs and CNNs are the most successful methods in text classification. Poria et al.[18] and Tamchyna et al.[22] used sequential deep convolution and long short-term memory (LSTM)[23], respectively, for aspect information extraction in SemEval 2014 ABSA tasks. These methods showed very good performance. [22] used LSTM as a feature extractor and logistic regression as a classifier. Their model simply encodes the whole input sentence. Our model has the same mechanism. Our model also simply encodes the whole input sentence. However, our experiment shows the simple mechanism without CNNs and RNA has better performance.

Collobert et al.[12] showed that representing a sentence as a sequential convolution of word vectors gave good performance on several benchmarks. Kim[17]

¹ <http://alt.qcri.org/semeval2016/task5/>

introduced the variant convolution mechanism of [12]. However, the convolution mechanism of [12] in a sentiment identification model did not give good performance with RCNN model[5]. For the sentiment identification task in ABSA, Wang et al.[14] used sequential convolution and achieved state-of-the-art performance. They also reported that a fully-connected network was sufficed for the aspect information extraction task in SemEval ABSA 2015 because more complicated models may overfit the training data. In the SemEval ABSA 2015 competition, no team provided good performance using neural networks. In SE16T5S1, Toh et. al.[6], Khalil et al.[13] and Poria et al.[18] used sequential convolution for both aspect category identification and sentiment polarity identification; these teams were top-ranked in SE16T5S1.

3 Task Setting

3.1 Dataset

SE16T5S1 provides datasets of multilingual reviews. The reviews are annotated with aspect terms, which are linguistic expressions used in sentences with polarity, which can be considered a kind of simplified opinion of reviewers. The datasets consist of eight languages and seven domains. In this research, we use an English dataset in a restaurant domain. The dataset includes 2000 English restaurant reviews for training and 676 reviews for tests. Each review is tagged with its aspects with polarities. For example, let us consider the following restaurant review:

“The wine list is interesting and has many good values.”

This review is tagged in XML as follows:

```
<Opinions>
  <Opinion target="wine_list"
    category="DRINKS#STYLE_OPTIONS"
    polarity="positive" />
  <Opinion target="wine_list"
    category="DRINKS#PRICES"
    polarity="positive" />
</Opinions>
```

<Opinions> denotes a set of opinions. One review can have multiple opinions. Such multiple opinions are described in <Opinions> tag. <Opinion> denotes a single opinion. Each <Opinion> has target, category, and polarity attributes, which represent an aspect term, an aspect category, and a sentiment polarity, respectively. In the case that the target is not explicit, the target will be “NULL”. Sentiment polarity receives one of the following labels: positive, negative, or neutral. Same target values can appear more than once, such as the example above.

“NULL” refers to implicit aspects. In other cases, these are referred as explicit aspect[18, 11].

3.2 Sentiment Polarity Identification Task

We evaluate our model using a sentiment polarity identification task. This task is referred to as slot3 in SE16T5S1. Slot3 is the task of identifying polarities in a given sentence given its aspect categories.

4 Simple Alignment Sentence Classification

For the sentiment polarity identification task, we designed a simple neural network with an attention mechanism. Figure 1 describes the model. Let us call this proposed model simple alignment sentence classification(SASC).

Let $\mathbf{x}_i \in \mathbb{R}^d$ be the d -dimensional word vector corresponding to the i -th word in the sentence. The sentence is represented as the following list of the word vectors.

$$X = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_L] \quad (1)$$

where L is length of the sentence. Using these word vectors, SASC infers the sentiment polarity, as in the following.

$$\mathbf{y} = \text{softmax}(W \cdot X \cdot \alpha^T) \quad (2)$$

$\mathbf{y} \in \mathbb{R}^3$ is the probability distribution of the polarity. \mathbf{y} is the probability of { positive neutral negative } labels:

$$\mathbf{y} = (P(\text{positive}), P(\text{neutral}), P(\text{negative}))^T \quad (3)$$

$W \in \mathbb{R}^{3 \times d}$ is the projection matrix to { positive neutral negative } labels. $\alpha \in \mathbb{R}^L$ is the vector of attention weights. α is computed by

$$\alpha = \text{softmax}(X \mathbf{v}_a^T) \quad (4)$$

$\mathbf{v}_a \in \mathbb{R}^d$ represents the embedding of given aspect categories. The evaluation data in the restaurant domain has 12 categories. All categories have the vector \mathbf{v}_a :

$$V = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_{12}] \quad (5)$$

Note that this model has only one parameter d . d is the vector dimension of the word vector. When we use a pretrained appropriate word vector, d is fixed by the pretrained vector. In this case, this model does not have a parameter. In our experiment we utilize a pretrained 300-dimensional word vector using the Google News Corpus(GNC)². In our experiment, our model does not have any parameters.

² <https://code.google.com/p/word2vec/>

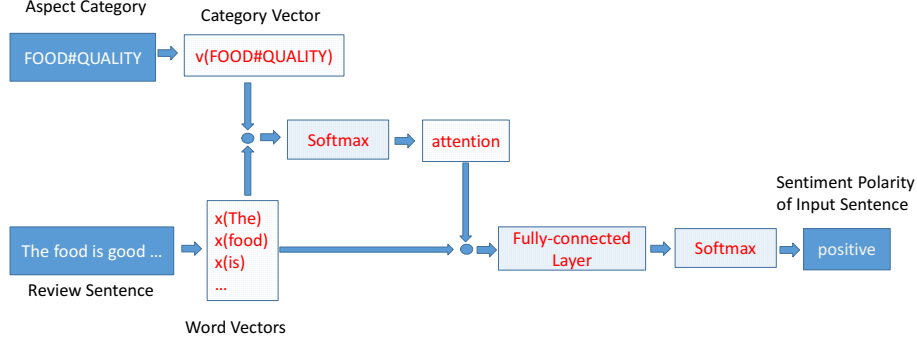


Fig. 1. Simple alignment sentence classification

5 Experiment

We trained the SASC model with a cross-entropy loss function using Adam optimizer[24]. The parameters of the optimizer are the same as the original paper[24]. V and W were initially set to random values. The word vectors X were initially set to a pretrained 300-dimensional word vector using the GNC. After setting the initial word vectors, they are adapted by the training of the SASC model. We trained the model using 2000 English restaurant reviews in the training data of SE16T5S1. The trained model was then evaluated using 676 reviews in the test data of SE16T5S1. The Number of epochs to train is fixed to eight. We evaluated the SASC model using other word vectors. In all cases, fixing the number of epochs to eight provided the best results.

5.1 CNN model

For comparison, we also evaluated the following CNN-based model based on [17].

$$\mathbf{y} = \text{softmax}(W \cdot \mathbf{z} + \mathbf{b}) \quad (6)$$

$\mathbf{y} \in \mathbb{R}^3$ is the probability of { positive neutral negative } labels:

$$\mathbf{y} = (P(\text{positive}), P(\text{neutral}), P(\text{negative}))^T \quad (7)$$

where

$$\mathbf{z} = \max Z \quad (8)$$

$$Z = \text{ReLU}(C) \quad (9)$$

$$C = W^{\text{conv}} \otimes X + \mathbf{b}^{\text{conv}} \quad (10)$$

$$X = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_L] \quad (11)$$

$W^{\text{conv}} \otimes$ represents convolution operation. In this experiment, 128 channel convolutions among three neighbor words were executed. $C \in \mathbb{R}^{(L-2) \times 128}$ is the convolved result. $\mathbf{b}^{\text{conv}} \in \mathbb{R}^{128}$ is bias of the convolution operation. The convolved result is applied to ReLU. Max pooling is then applied. Finally, the dense neural network of (6) is applied.

Note that, this model is not dependent on aspect categories. We also evaluated a model that is dependent on aspect categories. However, that model showed poor performance. For comparison, we evaluate a CNN model that was not dependent on aspect categories.

5.2 RNN model

For comparison, we also evaluated RNN models in [25]. [25] proposed an attention-based LSTM with aspect embedding (ATAE-LSTM) model. ATAE-LSTM is an attention network attached LSTM. The values of the attention network are switched by the selected aspect category. [25] also evaluated an LSTM model without the attached attention network. Let us refer to this simple LSTM model without an attached attention network as LSTM in the following.

5.3 Experimental Result

The table 1 shows the comparison results. The table also shows computing time per epoch. This experiment was evaluated by TensorFlow using an Intel i5 5300U. The results show that SASC has the best performance in terms of accuracy and computation speed. Especially, SASC is 20 times faster than ATAE-LSTM. This results show that ATAE-LSTM, LSTM, and CNN have redundant networks. The simple mechanism of SASC can avoid overfitting, which is the reason SASC shows the best performance in accuracy.

Table 1. model comparison

method	accuracy	sec/epoch
SASC	0.835	0.82
CNN	0.814	2.45
LSTM	0.790	13.8
ATAE-LSTM	0.819	15.2

5.4 Word Vector Dependence

In our experiment, we utilized pretrained 300-dimensional word vectors using the GNC. The tunable parameter of SASC is only initial word vectors. We discuss

dependency of the initial word vectors. Table 2 shows the word vectors we compared. We compared word vectors trained by continuous bag of words(CBOW), skip-gram with negative-sampling(SGNS) and fastText[26]. These word vectors were trained on the text8 corpus³ and English Wikipedia Snapshot in March 2015. We use CBOW and SGNS modules in the gensim library [27] with default settings except min_count. We use min_count=100 for Wikipedia, and min_count=5 for text8 corpus.

We evaluated each word vector by focusing on the similarity between words using the following six test datasets: WordSim Similarity(Sim)[28]; WordSim Relatedness(Rel)[29]; MEN dataset(MEN)[30]; Radinsky et al.’s Mechanical Turk dataset(Turk)[31]; Rare Words dataset(Rare)[32]; and SimLex-999 dataset(SimLex)[33]. We also evaluated each word vector based on word analogy by 3CosAdd using MSR’s analogy dataset [34] and Google’s analogy dataset [35] .

Table 2 shows that the GNC is best in all cases. However, we could not reproduce this result by parameter tuning of word2vec. We evaluated SASC with the word vectors whose characteristics and parameters were known.

Table 3 shows the dependence of the accuracy of SASC on the initial word vectors. This evaluation considers only words that appeared in the training restaurant reviews. Since a lot of words were missed in these test sets, the word similarity and analogy results are low values. Acc denotes accuracy of SASC for test reviews. random denotes word vectors that were random values. +SASC below random denotes that word vectors and SASC were trained using the training restaurant reviews. Other +SASC have same meaning.

The GNC was best in most cases. The accuracy of SASC with random word vectors was 0.711. This accuracy is comparable with that of initial training step with the GNC : 0.711. After training using the training restaurant reviews, the accuracy becomes 0.794. In addition, word vectors improved. This result shows SASC can improve word vectors. The results of fastText and SGNS also show the same characteristics. The word vectors of CBOW are degraded after training SASC. Except for GNC, CBOW shows the best performance in accuracy. These results show that CBOW provides appropriate initial word vectors for SASC, and that SASC removes some characteristics of word vectors after training.

Because word vectors of the missed words are filled as random vectors, some results of the random vector outperforms fastText, CBOW and SGNS. This tendency is stronger in the case of text8 corpus than Wikipedia, since text8 corpus has more missed words than Wikipedia. This tendency shows random vectors is not best for SASC.

6 Conclusion

In this paper, we proposed an SASC model for ABSA. SASC is actually single-layer neural network whose input is a list of word vectors. SASC model has only one parameter which is the vector dimension of the word vectors. When initial

³ <http://mattmahoney.net/dc/textdata.html>

Table 2. Word Vector

Training	method	Similarity						Analogy	
		Sim	Rel	MEN	Turk	Rare	SimLex	google	msr
text8	fastText	0.665	0.617	0.636	0.641	0.326	0.233	0.443	0.575
	CBOW	0.396	0.443	0.383	0.589	0.057	0.082	0.140	0.249
	SGNS	0.685	0.662	0.561	0.625	0.192	0.225	0.310	0.436
Wikipedia	fastText	0.705	0.637	0.672	0.632	0.352	0.283	0.528	0.366
	CBOW	0.668	0.550	0.665	0.663	0.314	0.256	0.532	0.375
	SGNS	0.721	0.631	0.674	0.661	0.372	0.292	0.507	0.426
GNC		0.735	0.704	0.714	0.789	0.800	0.294	0.768	0.746

Table 3. Word Vector Dependence

Training	Method	Similarity						Analogy		Acc
		Sim	Rel	MEN	Turk	Rare	SimLex	google	msr	
text8	random	0.183	-0.046	-0.008	-0.408	-0.527	-0.012	0.004	0.002	0.711
	+SASC	0.189	0.004	0.014	-0.177	-0.418	0.040	0.009	0.002	0.794
	fastText	-0.043	0.163	-0.010	-0.344	-0.373	-0.031	0.004	0.001	0.712
	+SASC	-0.119	-0.013	-0.005	-0.007	-0.045	0.068	0.007	0.005	0.775
	CBOW	-0.205	0.011	0.002	-0.389	-0.018	0.034	0.001	0.000	0.687
	+SASC	-0.034	-0.018	-0.023	0.389	0.245	0.085	0.006	0.001	0.773
	SGNS	-0.130	-0.075	-0.004	-0.411	0.255	0.078	0.006	0.005	0.715
	+SASC	0.002	-0.225	0.074	0.115	0.391	-0.025	0.001	0.002	0.775
Wikipedia	fastText	0.353	0.229	0.038	0.048	0.245	0.027	0.001	0.002	0.714
	+SASC	-0.007	-0.126	-0.029	0.314	0.409	-0.026	0.010	0.002	0.787
	CBOW	0.022	-0.013	-0.009	0.152	-0.091	-0.044	0.021	0.013	0.722
	+SASC	-0.195	-0.093	0.007	-0.091	-0.391	0.074	0.012	0.005	0.793
	SGNS	-0.128	0.108	-0.032	0.230	0.018	-0.038	0.016	0.011	0.710
	+SASC	0.070	-0.152	0.036	0.317	-0.282	0.034	0.003	0.002	0.775
GNC		0.735	0.704	0.714	0.789	0.800	0.294	0.768	0.746	0.711
	+SASC	0.647	0.535	0.522	0.665	0.818	0.342	0.291	0.249	0.835

word vector is chosen appropriately, SASC does not have any parameters. We evaluated SASC using the ABSA task in the SemEval 2016 restaurant domain. The proposed model outperformed CNN- and RNN-based models in terms of accuracy and computing speed. The tunable parameter of SASC is only initial word vectors. We showed that CBOW provides appropriate initial word vectors for SASC.

References

1. Cambria, E., White, B.: Jumping nlp curves: A review of natural language processing research [review article]. *IEEE Computational Intelligence Magazine* **9** (2014) 48–57
2. Cambria, E.: Affective computing and sentiment analysis. *IEEE Intelligent Systems* **31** (2016) 102–107
3. Nasukawa, T., Yi, J.: Sentiment analysis: Capturing favorability using natural language processing. In: *Proceedings of the 2Nd International Conference on Knowledge Capture. K-CAP '03*, New York, NY, USA, ACM (2003) 70–77
4. Liu, B.: Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* **5** (2012) 1–167
5. Asakura, R., Niitsuma, H., Ohta, M.: Recurrent neural networks on convoluted word vectors for aspect-based sentiment analysis. (In: *Computational Linguistics and Intelligent Text Processing: Proceedings of the 18th International Conference*)
6. Toh, Z., Su, J.: Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California, Association for Computational Linguistics (2016) 282–288
7. Brun, C., Perez, J., Roux, C.: Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California, Association for Computational Linguistics (2016) 277–281
8. Saïas, J.: Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, Association for Computational Linguistics (2015) 767–771
9. Toh, Z., Su, J.: Nlangp: Supervised machine learning system for aspect category classification and opinion target extraction. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, Association for Computational Linguistics (2015) 496–501
10. Hamdan, H., Bellot, P., Bechet, F.: Lsislif: Crf and logistic regression for opinion target extraction and sentiment polarity analysis. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, Association for Computational Linguistics (2015) 753–758
11. Poria, S., Cambria, E., Gui, C., Gelbukh, A.: (A rule-based approach to aspect extraction from product reviews)
12. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12** (2011) 2493–2537

13. Khalil, T., El-Beltagy, S.R.: Nlelrmrg at semeval-2016 task 5: Deep convolutional neural networks for aspect category and sentiment extraction. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, California, Association for Computational Linguistics (2016) 271–276
14. Wang, B., Liu, M.: Deep learning for aspect-based sentiment analysis (2015)
15. Chernyshevich, M.: Ihs-rd-belarus at semeval-2016 task 5: Detecting sentiment polarity using the heatmap of sentence. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, California, Association for Computational Linguistics (2016) 296–300
16. Yanase, T., Yanai, K., Sato, M., Miyoshi, T., Niwa, Y.: bunji at semeval-2016 task 5: Neural and syntactic models of entity-attribute relationship for aspect-based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, California, Association for Computational Linguistics (2016) 289–295
17. Kim, Y.: Convolutional neural networks for sentence classification. CoRR **abs/1408.5882** (2014)
18. Poria, S., Cambria, E., Gelbukh, A.: Aspect extraction for opinion mining with a deep convolutional neural network. Know.-Based Syst. **108** (2016) 42–49
19. Xiao, Y., Cho, K.: Efficient character-level document classification by combining convolution and recurrent layers. CoRR **abs/1602.00367** (2016)
20. Zuo, Z., Shuai, B., Wang, G., Liu, X., Wang, X., Wang, B., Chen, Y.: Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. (2015)
21. Poria, S., Chaturvedi, I., Cambria, E., Hussain, A.: Convolutional MKL based multimodal emotion recognition and sentiment analysis. In: IEEE 16th International Conference on Data Mining, ICDM 2016, December 12–15, 2016, Barcelona, Spain. (2016) 439–448
22. Tamchyna, A., Veselovská, K.: Ufal at semeval-2016 task 5: Recurrent neural networks for sentence classification. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, California, Association for Computational Linguistics (2016) 367–371
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory (1997)
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
25. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based lstm for aspect-level sentiment classification. In Su, J., Carreras, X., Duh, K., eds.: EMNLP, The Association for Computational Linguistics (2016) 606–615
26. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
27. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, ELRA (2010) 45–50 <http://is.muni.cz/publication/884893/en>.
28. Zesch, T., Muller, C., Gurevych, I.: Using wiktory for computing semantic relatedness. In: Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2. AAAI’08, AAAI Press (2008) 861–866
29. Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A.: A study on similarity and relatedness using distributional and wordnet-based approaches. In: Proceedings of Human Language Technologies: The 2009 Annual Conference

- of the North American Chapter of the Association for Computational Linguistics. NAACL '09, Stroudsburg, PA, USA, Association for Computational Linguistics (2009) 19–27
30. Bruni, E., Boleda, G., Baroni, M., Tran, N.K.: Distributional semantics in technicolor. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Jeju Island, Korea, Association for Computational Linguistics (2012) 136–145
 31. Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S.: A word at a time: Computing word relatedness using temporal semantic analysis. In: Proceedings of the 20th International Conference on World Wide Web. WWW '11, New York, NY, USA, ACM (2011) 337–346
 32. Luong, T., Socher, R., Manning, C.: Better word representations with recursive neural networks for morphology. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning, Sofia, Bulgaria, Association for Computational Linguistics (2013) 104–113
 33. Hill, F., Reichart, R., Korhonen, A.: Simlex-999: Evaluating semantic models with (genuine) similarity estimation. CoRR **abs/1408.3456** (2014)
 34. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: HLT-NAACL. (2013) 746–751
 35. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)