A Document Descriptor using Covariance of Word Vectors

Marwan Torki
Alexandria University, Egypt
mtorki@alexu.edu.eg

Abstract

In this paper, we address the problem of finding a novel document descriptor based on the covariance matrix of the word vectors of a document. Our descriptor has a fixed length, which makes it easy to use in many supervised and unsupervised applications. We tested our novel descriptor in different tasks including supervised and unsupervised settings. Our evaluation shows that our document covariance descriptor fits different tasks with competitive performance against state-of-the-art methods.

1 Introduction

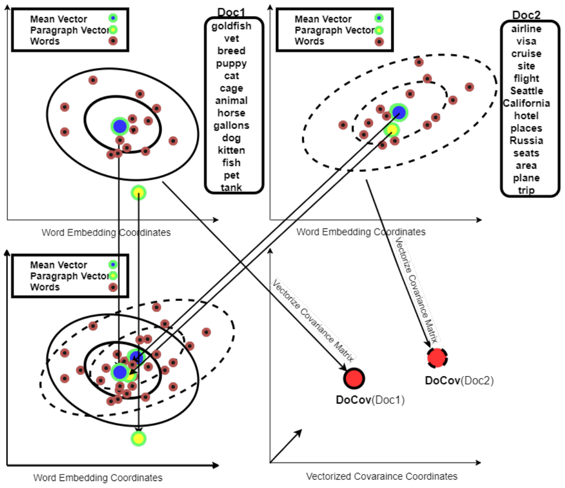

Figure 1: Doc1 is about "pets" and Doc2 is about "travel". **Top:** The first two dimensions of a word embedding for each document. **Bottom Left:** The embedding of the words of the two documents. The Mean vectors and the paragraph vectors are shown. Covariance matrices are shown via the confidence ellipses. **Bottom Right:** Corresponding covariance matrices are represented as points in a new space.

Retrieving documents that are similar to a query using vectors has a long history. Earlier methods modeled documents and queries using vector space models via bag-of-words (BOW) representation (Salton and Buckley, 1988). Other representations include latent semantic indexing (LSI) (Deerwester et al., 1990), which can be used to define dense vector representation for documents and/or queries. The past few years have witnessed a big interest in distributed representation for words, sentences, paragraphs and documents. This was achieved by leveraging deep learning methods that learn word vector representation. Introduction of neural language models (Bengio et al., 2003) using deep learning allowed to learn word vector representation (word embedding for simplicity). The seminal work of Mikolov et al. introduced an efficient way to compute dense vectorized representation of words (Mikolov et al., 2013a,b). A more recent step was taken to move beyond distributed representation of words. This is to find a distributed representation for sentences, paragraphs and documents. Most of the presented works study the interrelationship between words in a text snippet (Hill et al., 2016; Kiros et al., 2015; Le and Mikolov, 2014) in an unsupervised fashion. Other methods build a task specific representation (Kim, 2014; Collobert et al., 2011).

In this paper we propose to use the covariance matrix of the word vectors in some document to define a novel descriptor for a document. We call our representation DoCoV descriptor. Our descriptor obtains a fixed-length representation of the paragraph which captures the interrelationship between the dimensions of the word embedding via the covariance matrix elements. This makes our work distinguished from to the work of (Le and Mikolov, 2014; Hill et al., 2016; Kiros et al., 2015) where they study the interrelationship of words in the text

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 527–532
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

snippet.

1.1 Toy Example

We show a toy example to highlight the differences between DoCoV vector, the Mean vector and paragraph vector (Le and Mikolov, 2014). First, we used Gensim library[1] to generate word vectors and paragraph vectors using a dummy training corpus. Next, we formed two hypothetical documents; first document contains words about "pets" and second document contains words about "travel". In figure 1 we show on the top part the first two dimensions of a word embedding for each document separately. On the bottom Left, we show embedding of the two documents' words in the same space. We also show the Mean vectors and the paragraph vectors. In the word embedding space the covariance matrices are represented via the confidence ellipses. On the bottom right we show the corresponding covariance matrices as points in a new space after vectorization step.

1.2 Motivation and Contributions

Below we describe our motivation towards the proposal of our novel representation:

(1) Some neural-based paragraph representations such as paragraph vectors (Le and Mikolov, 2014) , FastSent (Hill et al., 2016) use a shared space between the words and paragraphs. This is counter intuitive, as the paragraph is a different entity other than the words. Figure 1 illustrates that point, we do not see a clear interpretation of why the paragraph vectors (Le and Mikolov, 2014) are positioned in the space as in figure 1.

(2) The covariance matrix represents the second order summary statistic of multivariate data. This distinguishes the covariance matrix from the mean vector. In figure 1 we visualize the covariance matrix using confidence ellipse representation.We see that the covariance encodes the shape of the density composed of the words of interest. In the earlier example the Mean vectors of two dissimilar documents are put close by the word embedding. On the other hand, the covariance matrices capture the distinctness of the two documents.

(3) The use of the covariance as a spatial descriptor for multivariate data has a great success in different domains like computer vision (Tuzel et al., 2006; Hussein et al., 2013; Sharaf et al., 2015) and brain signal analysis (Barachant et al., 2013). With this global success of this representation, we believe this method can be useful for text-related tasks.

(4) The computation of the covariance descriptor is known to be fast and highly parallelizable. Moreover, there is no inference steps involved while computing the covariance matrix given its observations. This is an advantage compared to existing methods for generating paragraph vectors, such as (Le and Mikolov, 2014; Hill et al., 2016).

Our contribution in this work is two-fold:

(1) We propose the Document-Covariance descriptor (**DoCoV**) to represent every document as the covariance of the word embedding of its words. To the best of our knowledge, we are the first to explicitly compute covariance descriptors on word embedding such as word2vec (Mikolov et al., 2013b) or similar word vectors.

(2) We empirically show the effectiveness of our novel descriptor in comparison to the state-of-the-art methods in various unsupervised and supervised classification tasks. Our results show that our descriptor can attain comparable accuracy to state-of-the-art methods in a diverse set of tasks.

1.3 Related Work

We can see the word embedding at the core of recent state-of-art methods for solving many tasks like semantic textual similarity, sentiment analysis and more. Among the approaches of finding word embedding are (Pennington et al., 2014; Levy and Goldberg, 2014; Mikolov et al., 2013b). These alternatives share the same objective of finding a fixed-length vectorized representation for words to capture the semantic and syntactic regularities between words.

These efforts paved the way for many researchers to judge document similarity based on word embedding. Some efforts aimed at finding a global representation of a text snippet using a paragraph-level representation such as paragraph vectors (Le and Mikolov, 2014). Recently other neural-based sentence and paragraph level representations appeared to provide a fixed length representation like Skip-Thought Vectors (Kiros et al., 2015) and FastSent (Hill et al., 2016). Some efforts focused on defining a Word Mover Distance(WMD) based on word level representation (Kusner et al., 2015).

Prior to this work, we proposed earlier trials for using covariance features in community question answering (Malhas et al., 2016b,a; Torki et al.,

[1] https://radimrehurek.com/gensim/

2017). In these trials we used the covariance features in combination with lexical and semantic features. Close to our work is (Nikolentzos et al., 2017), they build an implicit representation of documents using multidimensional Gaussian distribution. Then they compute a similarity kernel to be used in document classification task. Our work is distinguished from (Nikolentzos et al., 2017) as we compute an explicit descriptor for any document. Moreover, we use linear models which scale much better than non-linear kernels as introduced in (Nikolentzos et al., 2017).

2 Document Covariance Descriptor

We present our DoCoV descriptor. First, we define a document observation matrix. Second, we show how to extract our DoCoV descriptor.

Document Observation Matrix

Given a d-dimensional word embedding model and an n-terms document. We can define a document observation matrix $O \in \mathbb{R}^{n \times d}$. In the matrix O, a row represents a term in the document and columns represent the d-dimensional word embedding representation for that term.

Assume that we have observed n terms of a d-dimensional random variable; we have a data matrix $O(n \times d)$:

$$O = \begin{pmatrix} x_{11} & \cdots & x_{1d} \\ x_{21} & \cdots & x_{2d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{pmatrix} \qquad (1)$$

The rows $x_i = \begin{bmatrix} x_1 & x_2 & \cdots & x_d \end{bmatrix}^T \in R^d$, denote the i-th observation of a d-dimensional random variable $X \in R^d$. The "sample mean vector" of the n observations $\in R^d$ is given by the vector $\bar{x}$ of the means $\bar{x_j}$ of the d variables:

$$\bar{x} = \begin{bmatrix} \bar{x_1} & \bar{x_2} & \cdots & \bar{x_d} \end{bmatrix}^T \in R^d \qquad (2)$$

From hereafter, when we mention the Mean vector we mean the sample Mean Vector $\bar{x}$.

Document-Covariance Descriptor (DoCoV)

Given an observation matrix O for a document, we compute the covariance matrix entriesfor every pair of dimensions (X, Y).

$$\sigma_{X,Y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N} \qquad (3)$$

The matrix $C \in \mathbb{R}^{d \times d}$ is a symmetric matrix and is defined as

$$C = \begin{pmatrix} \sigma_{X_1}^2 & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_d} \\ \sigma_{X_1X_2} & \sigma_{X_2}^2 & \cdots & \sigma_{X_2X_d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_1X_d} & \sigma_{X_2X_d} & \cdots & \sigma_{X_d}^2 \end{pmatrix} (4)$$

We Compute a vectorized representation of the matrix C as the stacking of the upper triangular part of matrix C as in eq. 5. This process produces a vector $v \in \mathbb{R}^{d(d+1)/2}$. The Euclidean distance between vectorized matrices is equivalent to the Frobenius norm of the original covariance matrices.

$$v = \textbf{vect}(C) = \begin{cases} \sqrt{2}C_{p,q} & \text{if } p < q \\ C_{p,q} & \text{if } p = q \end{cases} \qquad (5)$$

3 Experimental Evaluation

We show an extensive comparative evaluation for unsupervised paragraph representation approaches. We test the *unsupervised* semantic textual similarity task. Next, we show a comparative evaluation for text classification benchmarks.

3.1 Effect of Word Embedding Source and Dimensionality on Classification Results

We evaluate classification performance over the IMDB movie reviews dataset using error rate as the evaluation measure. We report our results using a linear SVM classifier.We chose the default parameters for Linear SVM classifier in scikit-learn library[2].

The IMDB movie review dataset was first proposed by Maas et al. (Maas et al., 2011) as a benchmark for sentiment analysis. The dataset consists of 100K IMDB movie reviews and each review has several sentences. The 100K reviews are divided into three datasets: 25K labelled training instances, 25K labelled test instances and 50K unlabelled training instances. Each review has one label representing the sentiment of it: Positive or Negative. These labels are balanced in both the training and the test set.

The objective is to show that theDoCoV descriptor can be used with different alternatives for word representations. Also, the experiment shows that pre-trained models are giving the best results, namely the word2vec model built on Google news. This alleviates the need of computing a problem

[2] http://scikit-learn.org/

specific word embedding. In some cases there is no available data to construct the word embedding. To illustrate that we tried different alternatives for word representation.

(1) We computed our own skipgram models using Gensim Library. We used the Training and unlabelled subsets of IMDB dataset to obtain different embedding by setting number of dimensions to 100, 200 and 300.

(2) We used pre-trained GloVe models trained on wikipedia2014 and Gigaword5. We tested the available different dimensionality 100, 200 and 300. We also used the 300 dimensions GloVe model that used commoncrawl with 42 Billion tokens We call the last one *Lrg*. This model provides word vectors of 300 dimensions for each word.

(3) We used pre-trained word2vec model trained on Google news. We call it *Gnews*. This model provides word vectors of 300 dimensions for each word. Table 1 shows the results when using DoCoV computed at different dimensions of word embedding in classification. The table also compares classification performance when using DoCoV to the performance when using the Mean of word embedding as a baseline. Also, we show the effect of fusing DoCoV with other feature sets. We mainly experiment with the following sets: DoCoV, Mean, and bag-of-words (BOW). We use the mean and DoCoV features.

Observations

From the results we can observe the following

(1) We observe that the DoCoV is consistently outperforming the **Mean** vector for different dimensionality of the word embedding regardless of the embedding source.

(3) The best performing feature concatenation is DoCoV+BOW. This ensures that the concatenation in fact is benefiting from both representations.

(3) In general the best results are achieved using the available 300-dimensions *Gnews* word embedding. In the subsequent experiments we will use that embedding such that we do not need to build a different word embedding for every task on hand.

Unsupervised Semantic Textual Similarity

We conduct a comparative evaluation against the state-of-the-art approaches in unsupervised paragraph representation. We follow the setup used in (Hill et al., 2016).

Datasets and Baselines

We contrast our results against the methods reported in (Hill et al., 2016). The competing methods are the paragraph vectors (Le and Mikolov, 2014), skip-thought vectors (Kiros et al., 2015), Fastsent (Hill et al., 2016), Sequential (Denoising) Autoencoders (SDAE) (Hill et al., 2016). The Mean vector baseline is also implemented. Also, we use the sum of the similarities generated by the DoCoV and the mean vectors. All of our results are reported using the freely available *Gnews* word2vec of $dim = 300$. We use same evaluation measures (Hill et al., 2016). We use the Pearson correlation and Spearman correlation with the manual relatedness judgements.

The semantic sentence relatedness datasets used in the comparative evaluation the SICK dataset (Marelli et al., 2014) consists of 10,000 pairs of sentences and relatedness judgements and the STS 2014 dataset (Agirre et al., 2014) consists of 3,750 pairs and ratings from six linguistic domains.

Results and Discussion

We show the correlation values between the similarities computed via DoCoV and the human judgements. We contrast the performance of other representations in table 2.

We observe that DoCoV representation outperforms other representations in this task. Other models such as skipthought vectors (Kiros et al., 2015) and SDAE (Hill et al., 2016) requires building an encoder-decoder model which takes time[3] to learn. For other models like paragraph vectors (Le and Mikolov, 2014) and Fastsent vectors (Hill et al., 2016), they require a gradient descent inference step to compute the paragraph/sentence vectors. Using the DoCoV, we just require a pre-trained word embedding model and we do not need any additional training like encoder-decoder models or inference steps via gradient descent.

Text Classification Benchmarks

The datasets used in this experiment form a text-classification benchmark for sentence and paragraph classification. Towards the end of this section we can clearly identify the value of the DoCoV descriptor as a generic descriptor for text classification tasks.

Datasets and Baselines

We contrast our results against the same methods of unsupervised paragraph representations. In addition to the results of DoCoV we examined concatenation of BoW with tf-idf weighting and Mean vectors with our DoCoV descriptors. We use linear

[3] Up to weeks.

Table 1: Error-Rate performance when changing word vectors dimensionality.

Model/Dim					
BOW	9.66%				
Gensim	Mean	DoCoV	DoCoV +Mean	DoCoV Bow	DoCoV +Mean+Bow
d=100	14.13%	11.64%	11.16%	9.39%	9.44%
d=200	12.86%	11.08%	10.80%	9.39%	9.58%
d=300	12.83%	11.08%	10.85%	9.41%	9.47%
Glove	Mean	DoCoV	DoCoV +Mean	DoCoV Bow	DoCoV +Mean+Bow
d=100	20%	13.07%	12.88%	9.63%	9.62%
d=200	16.95%	12.36%	12.22%	9.64%	9.65%
d=300	16.29%	12.00%	11.91%	9.63%	9.66%
d=300,*Lrg*	14.94%	11.70%	11.56%	9.5%	9.6%
Gnews	Mean	DoCoV	DoCoV +Mean	DoCoV Bow	DoCoV +Mean+Bow
d=300	14.03%	11.11%	10.75%	**9.32%**	9.6%

Table 2: Spearman/Pearson correlations on *unsupervised* (relatedness) evaluations.

	STS-2014							SICK
Model	News	Forums	Wordnet	Twitter	Images	Headlines	All	
P2vec (Le and Mikolov, 2014)	0.42/0.46	0.33/0.34	0.51/0.48	0.54/0.57	0.32/0.30	0.46/0.47	0.44/0.44	0.44/0.46
FastSent (Hill et al., 2016)	0.44/0.45	0.14/0.15	0.39/0.34	0.42/0.43	0.55/0.60	0.43/0.44	0.27/0.29	0.57/0.60
FastSent+AE (Hill et al., 2016)	**0.58/0.59**	**0.41**/0.36	**0.74/0.70**	0.63/0.66	**0.74/0.78**	0.57/0.59	**0.63/0.64**	**0.61/0.72**
Skip-Thought (Kiros et al., 2015)	0.56/**0.59**	**0.41/0.40**	0.69/0.64	**0.70/0.74**	0.63/0.65	**0.58/0.60**	0.62/0.62	0.60/0.65
SAE (Hill et al., 2016)	0.17/0.16	0.12/0.12	0.30/0.23	0.28/0.22	0.49/0.46	0.13/0.11	0.12/0.13	0.32/0.31
SAE+embs (Hill et al., 2016)	0.52/0.54	0.22/0.23	0.60/0.55	0.60/0.60	0.64/0.64	0.41/0.41	0.42/0.43	0.47/0.49
SDAE (Hill et al., 2016)	.07/0.04	0.11/0.13	0.33/0.24	0.44/0.42	0.44/0.38	0.36/0.36	0.17/0.15	0.46/0.46
SDAE+embs (Hill et al., 2016)	0.51/0.54	0.29/0.29	0.56/0.50	0.57/0.58	0.59/0.59	0.43/0.44	0.37/0.38	0.46/0.46
Mean	**0.65**/0.68	0.46/0.45	0.75/0.78	**0.71**/0.75	0.76/0.78	0.59/0.64	0.64/0.66	**0.63/0.73**
DoCoV	0.62/0.68	0.50/**0.51**	0.77/**0.79**	0.69/0.75	**0.78**/0.80	0.60/0.63	**0.67/0.70**	0.61/0.69
DoCoV+Mean	0.64/**0.70**	**0.51/0.51**	**0.79**/0.78	**0.71/0.76**	**0.78/0.81**	**0.61/0.65**	**0.67/0.70**	0.62/0.71

Table 3: Accuracy of sentence representation models on text classification benchmarks.

Representation \Dataset	MR	CR	Trec	Subj	Overall
Mean	77.4	79.2	80	91.3	81.98
BOW +tf-idf weights	77.1	78.5	89.3	89.3	83.55
P2vec (Le and Mikolov, 2014)	74.8	78.1	91.8	90.5	83.8
Skip-uni (Kiros et al., 2015)	75.5	79.3	91.4	92.1	84.58
bi-skip (Kiros et al., 2015)	73.9	77.9	89.4	92.5	84.43
comb-skip (Kiros et al., 2015)	**76.5**	**80.1**	**92.2**	**93.6**	**85.6**
FastSent (Hill et al., 2016)	70.8	78.4	76.8	88.7	78.68
FastSentAE (Hill et al., 2016)	71.8	76.7	80.4	88.8	79.43
SAE (Hill et al., 2016)	62.6	68	80.2	86.1	74.23
SAE+embs (Hill et al., 2016)	73.2	75.3	80.4	89.8	79.68
SDAE (Hill et al., 2016)	67.6	74	77.6	89.3	77.13
SDAE+embs (Hill et al., 2016)	74.6	78	78.4	90.8	80.45
COV	79.7	79.4	89.5	92.8	85.35
COV+Mean	80.2	80.1	90.3	93.1	85.93
COV+Bow	80.7	80.5	**91.8**	**93.3**	86.58
COV+Mean+BOW	**81.1**	**81.5**	91.6	93.2	***86.85***

SVM for all the tasks. All of our results are reported using the freely available *Gnews* word2vec of $dim = 300$. We use classification accuracy as the evaluation measure for this experiment as (Hill et al., 2016).

The subsets used in comparative benchmark evaluation are: Movie Reviews **MR** (Pang and Lee, 2005), Subjectivity **Subj** (Pang and Lee, 2004),Customer Reviews **CR** (Hu and Liu, 2004) and TREC Question **TREC** (Li and Roth, 2002).

Results and Discussion

Table 3 shows the results of our variants against state-of-art algorithms that use unsupervised paragraph representation.

We observe that DoCoV is consistently better than the Mean vector and BOW with tf-idf weights. Also, DoCoV is improving consistently when concatenated with baselines such as Mean vector and BOW vectors. This means each feature is capturing different discriminating information. This justifies the choice of concatenating DoCoV with other features. We further observe that DoCoV is consistently better than the paragraph vectors (Le and Mikolov, 2014), Fastsent and SDAE (Hill et al., 2016). The overall accuracy of DoCoV is highlighted and it outperforms other methods on the text classification benchmark.

4 Conclusion

We presented a novel descriptor to represent text on any level such as sentences, paragraphs or documents. Our representation is generic which makes it useful for different supervised and unsupervised tasks. It has fixed-length property which makes it useful for different learning algorithms. Also, our descriptor requires minimal training. We do not require a encoder-decoder model or a gradient descent iterations to be computed.

Empirically we showed the effectiveness of the descriptor in different tasks. We showed better performance against other state-of-the-art methods in supervised and unsupervised settings.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *SemEval*.

Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. 2013. Classification of covariance matrices using a riemannian-based kernel for bci applications. *Neurocomputing*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *NAACL*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *SIGKDD*.

Mohamed E Hussein, Marwan Torki, Mohammad Abdelaziz Gowayyed, and Motaz El-Saban. 2013. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*.

Matt J Kusner, Yu Sun, Nicholas I Kolkin, Kilian Q Weinberger, et al. 2015. From word embeddings to document distances. In *ICML*.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *NIPS*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *ACL*.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL-HLT*.

Rana Malhas, Marwan Torki, Rahma Ali, Tamer Elsayed, and Evi Yulianti. 2016a. Real, live, and concise: Answering open-domain questions with word embedding and summarization. In *TREC*.

Rana Malhas, Marwan Torki, and Tamer Elsayed. 2016b. Qu-ir at semeval 2016 task 3: Learning to rank on arabic community question answering forums with word embedding. In *SemEval*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Giannis Nikolentzos, Polykarpos Meladianos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. 2017. Multivariate gaussian document representation from word embeddings for text categorization. In *EACL*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Gerard Salton and Christopher Buckley. 1988. Termweighting approaches in automatic text retrieval. *Information processing & management*.

Amr Sharaf, Marwan Torki, Mohamed E Hussein, and Motaz El-Saban. 2015. Real-time multi-scale action detection from 3d skeleton data. In *WACV*.

Marwan Torki, Maram Hasanain, and Tamer Elsayed. 2017. Qu-bigir at semeval 2017 task 3: Using similarity features for arabic community question answering forums. In *SemEval*.

Oncel Tuzel, Fatih Porikli, and Peter Meer. 2006. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV 2006*.