Sparsifying Word Representations for Deep Unordered Sentence Modeling

Prasanna Sattigeri
IBM T. J. Watson Research Center
Yorktown Heights, NY
psattig@us.ibm.com

Jayaraman J. Thiagarajan
Lawrence Livermore National Laboratory
Livermore, CA
jayaramanthi1@llnl.gov

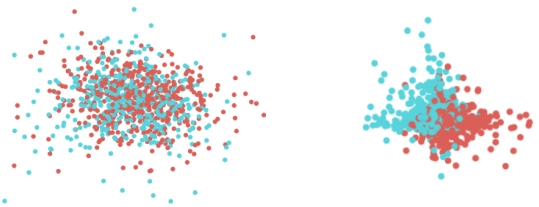
Abstract

Sparsity often leads to efficient and interpretable representations for data. In this paper, we introduce an architecture to infer the appropriate sparsity pattern for the word embeddings while learning the sentence composition in a deep network. The proposed approach produces competitive results in sentiment and topic classification tasks with high degree of sparsity. It is computationally cheaper to compute sparse word representations than existing approaches. The imposed sparsity is directly controlled by the task considered and leads to more interpretability.

1 Introduction

The recent surge in representation learning has resulted in remarkable advances in a variety of applications including computer vision and speech processing. In the context of natural language processing, much effort has been focused on constructing vector space representations for words through neural language models (Mikolov et al., 2013; Pennington et al., 2014) and designing appropriate composition functions to apply word embeddings for modeling sentences or documents. By design, the goal of neural word embedding approaches is to build *dense* vector representations that capture syntactic and semantic similarities in data (e.g., beautiful, and attractive have similar meanings, as opposed to ugly, and repulsive), that the classic categorical representation of words as indices of a vocabulary fails to capture.

The composition function based on these embeddings can be either unordered (e.g. average of the word representations) or syntactic, wherein the word order is explicity modeled (Socher et al., 2013a; Sutskever et al., 2011; Bowman, 2013).



(a) Original word vectors. (b) Sparsified word vectors.

Figure 1: t-SNE embedding of the sentence representations obtained as the average of word vectors for a random set of 1000 sentences from the SUBJ dataset.

While the former class of approaches results in simple architectures that are easily scalable, the latter can provide richer models with much severe computational complexity during training. Furthermore, the input word vectors are often fine-tuned during the training phase to improve the sentence (or document) classification performance. However, this can lead to severe overfitting and hence regularization strategies such as *word-dropout* are used (Iyyer et al., 2015) and in other cases the original word vectors are augmented to the input as a *static* channel (Kim, 2014; Zhang and Wallace, 2015).

Alternately, approaches that build word representations using different forms of regularization inspired by the linguistic study of word meanings have been effective in modeling sentences. For example, *sparsity* regularization can be been used to construct distributed representations (Eisenstein et al., 2011) that capture some of the crucial lexical semantics largely based on familiar, discrete classes (e.g., supersenses) and relations (e.g., synonymy and hypernymy).

Instead of employing sparsity to regularize word embeddings, we propose to infer appropriate sparsity patterns for pre-learned word vectors

Proceedings of the 1st Workshop on Representation Learning for NLP, pages 206–214,
Berlin, Germany, August 11th, 2016.

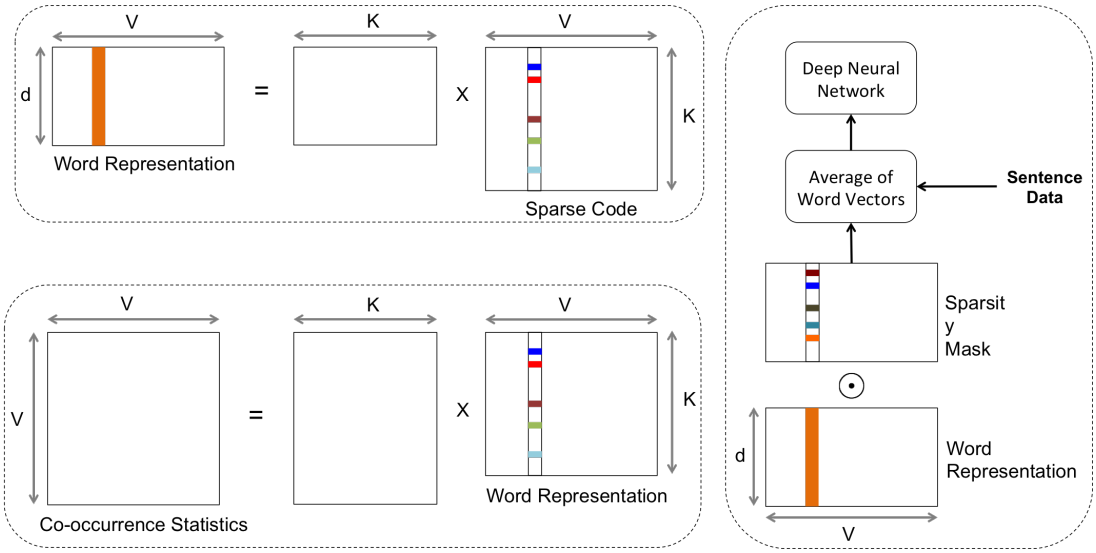


Figure 2: Sparsity has been commonly used to regularize word embeddings in order to effectively govern the relationship between word dimensions and provide interpretable representations. Examples include the approaches in (Yogatama et al., 2015) (left-top) and (Faruqui et al., 2015) (left-bottom). In contrary, we propose to infer sparsity patterns for pre-learned word embeddings in order to preserve only the key semantics required for the sentence classification task (right).

to improve the discrimination of sentence representations. In particular, we consider a unordered composition setting, similar to (Iyyer et al., 2015), wherein the sentence representation is obtained as the average of the words. Intuitively, sparsity is imposed to govern the relationship between word dimensions to capture only the semantics crucial to the particular task considered. For example, in a sentiment analysis task, opposite relationships between adjectives such as beautiful and ugly are more important than gender relationships such as king and queen. Surprisingly, without any additional regularization such as word-dropout or static channel of word vectors, the proposed approach produces competitive results in sentiment and topic classification tasks with high degree of sparsity. While it is cheaper to compute sparse word representations than existing approaches (Faruqui et al., 2015; Yogatama et al., 2015) the imposed sparsity is not merely based on the semantics of the space of words, but directly controlled by the task considered. Furthermore, by automatically learning sparsity masks that preserve only the semantic relationships appropriate for the task at hand, the resulting sentence models are highly discriminative. For example, in Figures 6(a) and 6(b), we show the sentence representations (word averaging) obtained using the original Glove word embeddings and the proposed approach.

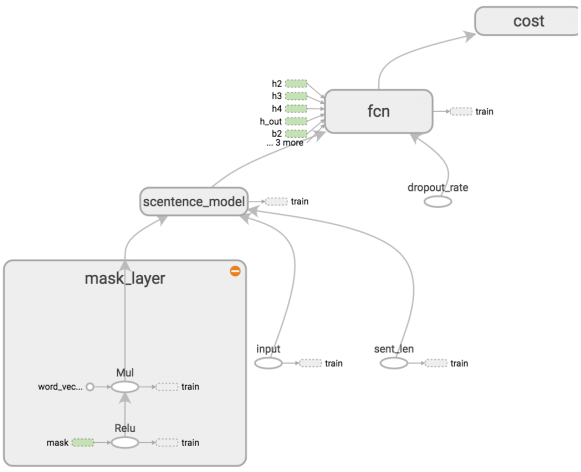


Figure 3: Tensorflow architecture for the proposed approach of sparsifying word embeddings based on labeled sentence data.

2 Sparsity in Word Embeddings

Though neural word representations are highly effective in enabling inference of complex semantic relationships between words, the interpretability of the word dimensions themselves is highly opaque. Hence, there is a disconnect between such dense representations and the word representa-

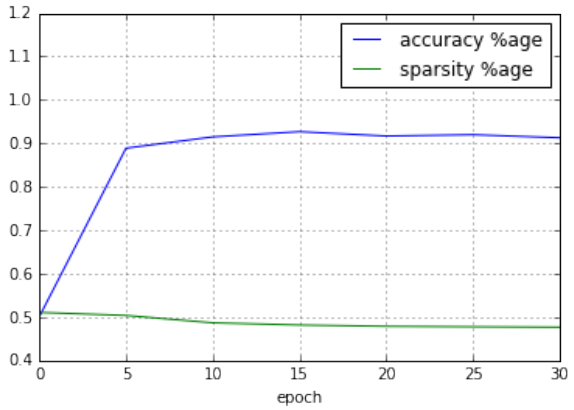
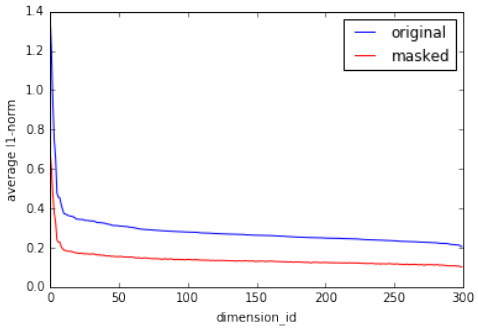
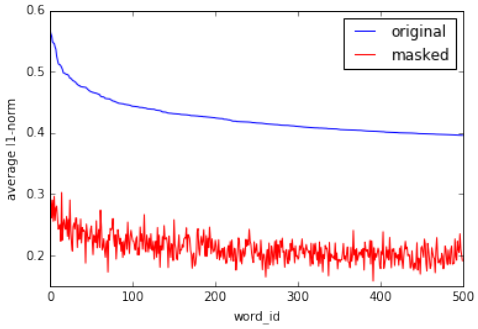


Figure 4: Convergence behavior of the proposed approach on the SUBJ dataset - Both the %Sparsity of the inferred mask and the testing accuracy are shown. Interestingly, only less than 50% of the entries are retained.



(a) Average ℓ_1 norm per dimension



(b) Average ℓ_1 norm for the top 500 words based with the largest ℓ_1 norm in the original word vector space.

Figure 5: Measuring changes in the original dense word vectors and word vectors sparsified using mask inferred from the newswire dataset.

tions typically found in lexical semantics, wherein each word can be represented sparsely in terms of an extensive set of discrete classes. For example, the word *apple* can be sparsely represented in terms of discrete concepts such as *fruit*, *edible food*, *red* etc. In contrast, learned word representations such as the Word2vec produces dense vectors, where the word dimensions that actually reveal a particular semantic relationship are not transparent. This motivated NLP researchers to explicitly impose sparsity into the word embedding inference. Sparse modeling with an overcomplete set of features is well known to produce simple, interpretable representations, while retaining the approximation power of dense models. Authors in (Andreas and Klein, 2014), use the creation of a word such as *northeast* from words north and east to illustrate that linguistic descriptors orient along a sparse set of perceptual basis. In the context of nlp tasks, it has been showed that sparse codes inferred from the pre-learned word embeddings (Figure 2) are more interpretable and hence sparsity can be used to govern relationships between word dimensions (Fyshe et al., 2014; Faruqui et al., 2015). Since sparsity can reveal the word dimensions pertinent to specific semantics, the resulting sparse representations were more effective in sentence classification. Similarly Chang et.al. found that sparse word vectors performed better in the behavioral task used to quantify interpretability (Chang et al., 2009). Furthermore, in (Yogatama and Smith, 2014), the authors advocate several sparsity based structural regularization schemes as a more suitable inductive bias and show improvements over dense representations several NLP tasks. In addition to the inherent computational complexity, an important downside of these approaches is that sparsity is merely used to regularize the word embeddings and hence cannot directly improve the discrimination of sentence representations constructed using these word vectors.

A striking similarity between all existing approaches for learning sparse word embeddings is that they aim to make the word dimensions corresponding to different semantic groups disjoint. However, given the large range of potential semantic relationships, it becomes computationally challenging to infer sparse representations that can discriminate all of them. This challenge is even more severe when word embeddings are applied to NLP tasks such as sentence classification. This motivates the need to infer the appropriate sparsity patterns for word embeddings such that they can easily discriminate the semantic concepts crucial for

Model	SST-fine	IMDB	SUBJ	Reuters
CNN-MC (Kim, 2014)	47.4	–	93.2	–
F-Dropout (Wang and Manning, 2013)	–	91.1	**93.6**	–
TreeLSTM (Tai et al., 2015)	**50.6**	–	–	–
PVEC (Le and Mikolov, 2014)	48.7	**92.6**	–	–
DAN + Word-Drop (Iyyer et al., 2015)	46.9	89.4	92.4	72.6
DAN + Sparsity-Mask	47.4	91.1	92.9	**73.7**
DAN + Binary-Mask	47.2	88.7	92.4	72.1

Table 1: Sentence classification performance of the proposed approach in comparison to other methods. In addition to outperforming the deep averaging architecture, our approach achieves competitive performances in comparison to state-of-the-art syntactic sentence classification methods.

Word / NNs	Sentiment Mask	Newswire Mask	Original
uncomfortable	uneasy, enough, terribly, hence	renovations, racket, contingent, competing	awkward, uneasy, unpleasant, bothered
president	being, concerned, nothing, then	between, growth, bank, earnings	vice, chief, executive, former

Table 2: Neighborhood of words obtained with two different sparsity masks: (a) sentiment mask from the SST dataset, (b) Newswire mask from the Reuters newswire dataset. In addition, we show the neighbors identified using the original word embeddings.

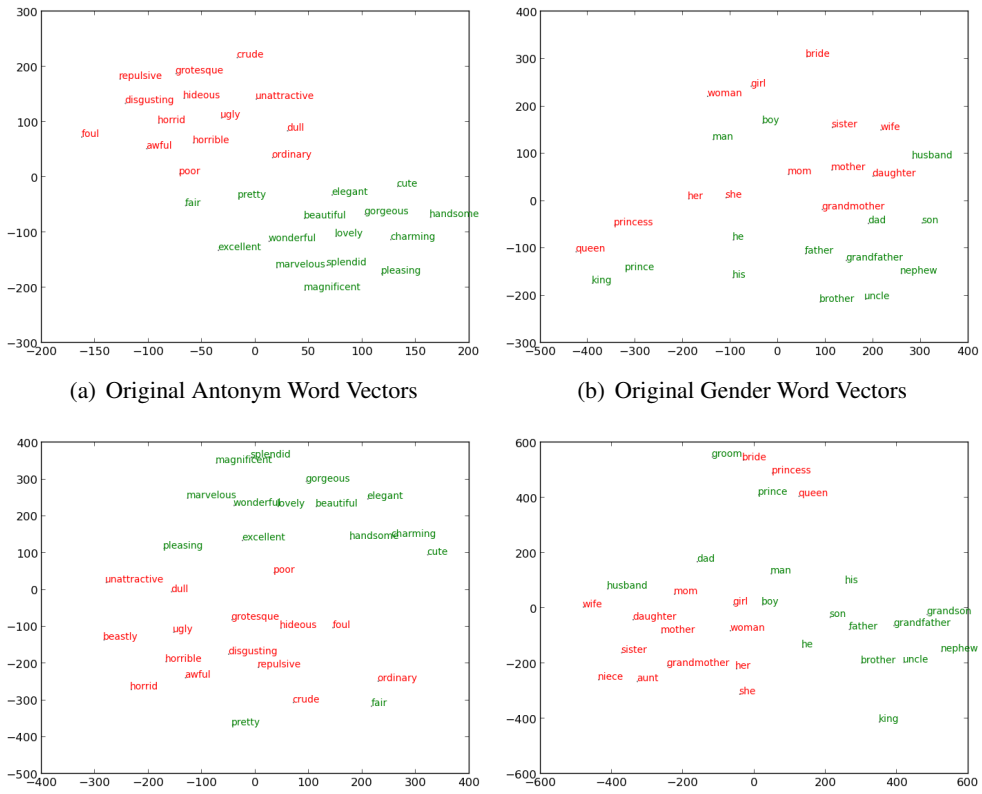
the NLP task at hand. Such a task-driven approach has two important advantages: (a) By inferring sparsity patterns specific to the task/dataset there is improved discrimination, (b) We can circumvent the computationally intensive sparse learning by adding this as a layer into the traditional deep learning architectures used for sentence classification. In the rest of paper, we describe our approach to couple the process of sparsifying word embeddings in deep undordered sentence classification framework similar to (Iyyer et al., 2015).

3 Proposed Approach

The proposed architecture shown in Figure 2(right) aims to infer a sparsity mask for the word embeddings using a deep unordered composition network (Iyyer et al., 2015). Note that, the sentence modeling corresponds to simply averaging the word vectors in that sentence. Let the word vectors be denoted by a matrix $\mathbf{W} \in R^{V,d}$. In our architecture, we introduce the sparsity mask M which is applied to the word vector matrix W as an element-wise product. The mask is a real valued matrix which is passed through $Relu$ non-linear activation to transform into a sparse mask with non-negative entries. This mask is applied in a multiplicative manner on W to obtain the masked word vector matrix $\hat{W} = W \odot Relu(M)$ and is optimized such that the sentence-level classification performance is maximized. We also consider a variant of this architecture, wherein the entries are thresholded to discrete values 0 or 1 based on the sign of entries in the real valued mask.

For all analysis and results reported in this paper, we used the pre-trained $300-$dimensional *Glove* (Pennington et al., 2014) word vectors. As described earlier, a sentence level representation is created by averaging the word vectors corresponding to the constituent words. This 300-dimensional representation is then passed through a series of fully connected layers and finally a softmax layer for prediction of labels. In contrast to the architecture in (Iyyer et al., 2015) no word-dropout regularization is used. Apart form the standard cross-entropy loss with a weight-decay regularization, we also include a term in the loss function to minimize the ℓ_1-norm of the mask to explicitly enforce sparsity on the latter. We implemented the architectures for both the real mask and binary mask versions using Tensorflow and Figure 3 shows the masking operation in detail using the tensorboard network architecture. visualizer.

The fully-connected deep network (FCN) on top of the sentence model is maintained the same for all datasets. The FCN is made up of three non-



(a) Original Antonym Word Vectors

(b) Original Gender Word Vectors

(c) Antonym Word Vectors with Sentiment Mask

(d) Gender Word Vectors with Sentiment Mask

Figure 6: t-SNE plots of original and sparsified word vectors illustrating the ability of the learned mask to retain only semantic relationships relevant to the task at hand.

linear layers followed by the soft-max layer. The non-linear layer consisted of a linear transformation followed by the ReLu unit. The hidden layers have a constant dimension of 300 and dropout is applied at each of these layers. The same hyperparameters were used across all datasets. Adam optimizer was used with learning rate set to $1e-4$ and the dropout out rate was set 0.5. The ℓ_2 regularization parameter for weight-decay was set to $1e-4$.. The weights of the FCN layers were initialized randomly from uniform distribution $[-1, 1]$ and scaled with a factor of 0.08. 10-fold CV was applied to datasets with no explicit train/test splits.

4 Experiments and Results

We evaluated the proposed approaches using a set of commonly used text classification datasets both at the sentence level and the document level. We report the performance of the proposed architecture with respect to the classification task pertaining to each dataset. This is followed up by investigation of the properties of the sparsified word vectors. For all classification performance comparisons we used the vanilla DAN with word-dropout regularization (Iyyer et al., 2015), and the proposed DAN + sparsity mask and DAN + binary mask variants.

Datasets:

- IMDB (document level): This dataset (Maas et al., 2011) consists of 50,000 labeled instances of movie reviews taken form the movie review site, IMDB. Each review can be made up of several sentences and is labeled as either positive or negative. The dataset also provides a balanced split of 25,000 instances for training and 25,000 instances for testing.

- SST-fine (sentence level): This sentence level dataset was created by (Pang and Lee, 2005) and extended by (Socher et al., 2013b). The sentences are taken from movie review site, Rotten Tomatoes (RT). In our experiments, we use the fine-grain labels for the classification task. The dataset provides three set for training, validation and testing with each containing and, respectively. Note that, sev-

original: fanatically, stylings, melding, inimitable, ardently masked: whole-heartedly, uncompromising, rosily, principled, hard-driving
original: post-camp, larceny, family-friendly, light-years, matchmaking masked: post-camp, family-friendly, voyages, four-star, cabins
original: ballot, ontiveros, candidate, nomination, badge masked: candidate, nomination, laziest, vote, voting
original: 95, shave, grad, veggietales, colgate masked: shave, grad, veggietales, colgate, golf

Table 3: Words in the newswire dataset with largest coefficient along a random dimension of word vectors. Each row belongs to different dimension.

Original	blinddate, micro-device, bible-study, greenfingers, fever-pitched, bogosian, darabont, navona, 66-day, murri
Masked	screenplay, cinematic, entertaining, fascinating, movie, daughter, he, micro-device, secret, discovers

Table 4: Demonstration of the discriminative power of sparsified word embeddings - Words with largest ℓ_1-norm in the SUBJ dataset. The words colored in blue occur most commonly found in sentences from the subjective class while words marked in red occur commonly in objective sentences.

eral existing syntactic approaches also utilize the phrase level labels by augmenting them to the training set. However, we evaluate the three DAN architectures without the phrase-level labels.

- SUBJ (sentence level): This dataset called as the Subjective dataset (Pang and Lee, 2004) involves classifying a sentence as either being subjective or objective. This dataset provides 10,000 instances in total and contain separate validation/test set.

- Reuters (document level): This dataset comprises of 11228 newswires from Reuters. The task is to classify the newswires into one of the given 46 topics. There is no standard train/test split for this dataset.

The classification performance on these datasets is reported in table 1. As it can be observed, the sparsified word vectors ourperform the conventional word embeddings with the DAN architecture and perform competitively with respect to state-of-the-art syntactic methods. Investigating the properties of the masked work vectors and comparing them to original work vectors can shed some light on the behavior of the sparsification procedure. Figure 5(a) shows the mean ℓ_1-norm of each dimension of the word vector across all the words in the vocabulary for the SST sentiment classification dataset. The dimensions are ordered by their ℓ_1-norm in the original word vector space. The general behavior remains the same, however with an overall reduction in norm that can be attributed to the sparsity in the masked word vectors. Similar analysis can be performed with respect to words instead of each word vector dimensions. In figure 5(b), the blue line corresponds to ℓ_1-norm of the top 500 words ordered by the norm. The norm for the same words in the masked space is shown in red, which indicates that the mask is word-specific and can tune the entries as suited for the task in hand.

Analysis of task-specific mask: To understand the effect of the task-specific mask, we study the similarity of words and compare them in the original word vector space and the sparsified word vector spaces. Table 4 shows a couple of example neighborhoods of words in these spaces. Subjectively, we can see that the word vector semantic space is modified such that word neighborhoods that are more important for the task are preserved and enhanced. We can draw similar inferences by looking at the t-SNE (Van der Maaten and Hinton, 2008; Faruqui and Dyer, 2014) plots as seen in figure 6.

Another approach is to investigate the individual dimensions of word vector and how the mask affects their behavior is isolation. In table 3, we show the top words for 4 random dimensions

original: officials, he, government, who, political masked: she, he, local, until, decision
original: societal, well-defined, physiological, mechanisms, perceptible masked: predictable, least, familial, elements, unconditional
original: wanna, song, lil, bitch, gonna masked: movies, laugh, fans, moments, wit
original: voltage, layer, cells, surface, battery masked: easy, i, velocity, provides, functions
original: goldie, knowles, hailey, dick, dildo masked: rachel, peter, patricia, alex, johnny

Table 5: Words from SUBJ dataset with largest coefficient along the top-5 dimensions (in terms of ℓ_1-norm) of word vectors.

of the original word vectors and the corresponding dimensions from the masked counterparts obtained from the SST-fine sentiment classification task. The top words are obtained by sorting the absolute value of the words along each of those dimensions. Since, there is a direct correspondence between original and masked word vector dimensions, we can directly compare them. The examples in table 3 show that mask improves the semantic consistency and hence improves interpretation of individual dimensions. Similar analysis is carried our for the SUBJ dataset and the results are reported in Table 5

Finally, we use the SUBJ dataset to demonstrate the discriminative power of sparsified word embeddings in sentence classification. The words with the largest ℓ_1-norm in the masked vector space in Table 4 reveal that the sparsity mask identifies a set of words crucial for discriminating the two classes. Finally, we consider an example sentence in each of the two classes and show the average ℓ_1 norms for words in the sentences in Figure 7. As it can be observed, words such as *emotional* and *material* are crucial to identifying the subjective nature of the sentence while words such as $125 - year$ which has prominence in the original word vector space has no relevance.

5 Conclusions

We have described an architecture that performs fine tuning of the word vectors in a classification setup while promoting sparsity in them. The resulting network achieves competitive results on several text classification datasets. This approach of inducing sparsity is computationally much cheaper than the traditional sparse models. The fine-tuned word vectors are also shown to be more interpretable, task specific and in process enhance the effectiveness of architectures based on simple unordered composition model. Also, the resulting word vectors posses improved discriminatory power suggesting that the use of this method as a pre-processing step can potentially lead to improved performance in other tasks which utilize word vectors.

References

Jacob Andreas and Dan Klein. 2014. Grounding language with points and paths in continuous spaces. In *CoNLL*, pages 58–67.

Samuel R Bowman. 2013. Can recursive neural tensor networks learn logical reasoning? *arXiv preprint arXiv:1312.6192*.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

Jacob Eisenstein, Noah A Smith, and Eric P Xing. 2011. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1365–1374. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, USA, June. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. 2015. Sparse overcomplete word vector representations. *arXiv preprint arXiv:1506.02004*.

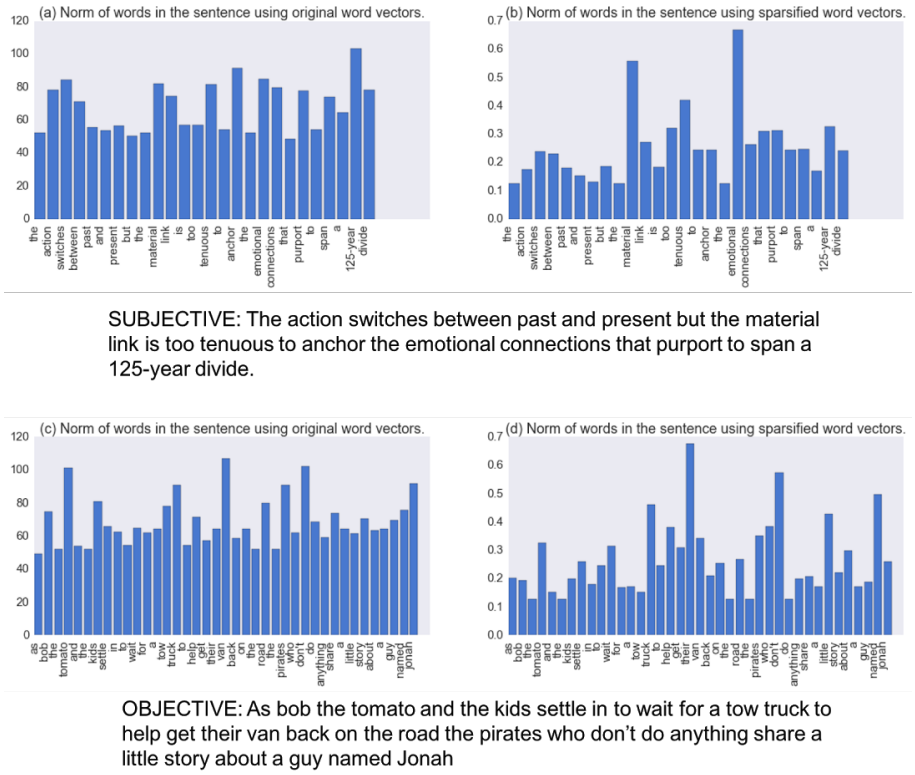


Figure 7: An example from the SUBJ dataset demonstrating the ability of the proposed sparisification to choose words that easily discriminate the two classes.

Alona Fyshe, Partha P Talukdar, Brian Murphy, and Tom M Mitchell. 2014. Interpretable semantic vectors from a joint model of brain-and text-based meaning. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2014, page 489. NIH Public Access.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1681–1691.

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

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

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 *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013a. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013b. Recursive deep

models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

Sida Wang and Christopher Manning. 2013. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning*, pages 118–126.

Dani Yogatama and Noah A Smith. 2014. Linguistic structured sparsity in text categorization.

Dani Yogatama, Manaal Faruqui, Chris Dyer, and Noah Smith. 2015. Learning word representations with hierarchical sparse coding. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 87–96.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.