Unsupervised Sparse Vector Densification for Short Text Similarity

Yangqiu Song and Dan Roth
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{yqsong,danr}@illinois.edu

Abstract

Sparse representations of text such as bag-of-words models or extended explicit semantic analysis (ESA) representations are commonly used in many NLP applications. However, for short texts, the similarity between two such sparse vectors is not accurate due to the small term overlap. While there have been multiple proposals for dense representations of words, measuring similarity between short texts (sentences, snippets, paragraphs) requires combining these token level similarities. In this paper, we propose to combine ESA representations and word2vec representations as a way to generate denser representations and, consequently, a better similarity measure between short texts. We study three densification mechanisms that involve aligning sparse representation via many-to-many, many-to-one, and one-to-one mappings. We then show the effectiveness of these mechanisms on measuring similarity between short texts.

1 Introduction

Bag-of-words model has been used for many applications as the state-of-the-art method for tasks such as document classifications and information retrieval. It represents each text as a bag-of-words, and computes the similarity, e.g., cosine value, between two sparse vectors in the high-dimensional space. When the contextual information is insufficient, e.g., due to the short length of the document, explicit semantic analysis (ESA) has been used as a way to enrich the text representation (Gabrilovich and Markovitch, 2006; Gabrilovich and Markovitch, 2007). Instead of using only the words in a document, ESA uses a bag-of-concepts retrieved from Wikipedia to represent the text. Then the similarity between two texts can be computed in this enriched concept space.

Both bag-of-words and bag-of-concepts models suffer from the sparsity problem. Because both models use sparse vectors to represent text, when comparing two pieces of texts, the similarity can be zero even when the text snippets are highly related, but make use of different vocabulary. We can expect that these two texts are related but the similarity value does not reflect that. ESA, despite augmenting the lexical space with relevant Wikipedia concepts, still suffers from the sparsity problem. We illustrate this problem with the following simple experiment, done by choosing a documents from the "rec.autos" group in the 20-newsgroups data set[1]. For both documents and the label description "cars" (here we follow the description shown in (Chang et al., 2008; Song and Roth, 2014)), we computed 500 concepts using ESA. Then we identified the concepts that appear both in the document ESA representation and in the label ESA representation. The average sizes of this intersection (number of overlapping concepts in the document and label representation) are shown in Table 1. In addition to the original documents, we also split each document into 2, 4, 8, 16 equal length parts, computed the ESA representation of each, and then the intersection with the ESA representation of the label. Table 1 shows that the number of concepts shared by the label and the document representation decreases significantly, even if not as significantly

[1] http://qwone.com/~jason/20Newsgroups/

Table 1: Average sizes of the intersection between the ESA concept representations of documents and labels. Both documents and label are represented with 500 Wikipedia concepts. Documents are split into different lengths.

# of split	Avg. # of words per doc.	Avg. # of concepts
1	209.6	23.1
2	104.8	18.1
4	52.4	13.8
8	26.2	10.6
16	13.1	8.4

as the drop in the document size. For example, there are on average 8 concepts in the intersection of two vectors with 500 non-zero concepts when we split each document into 16 parts.

When there are fewer overlapping terms between two pieces of texts, it can cause mismatch or biased match and result in less accurate comparison. In this paper, we propose to use unsupervised approaches to improve the representation, along with a corresponding similarity approach between these representations. Our contribution is twofold. First, we incorporate the popular word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b) representations into ESA representation, and show that incorporating semantic relatedness between Wikipedia titles can indeed help the similarity measure between short texts. Second, we propose and evaluate three mechanisms for comparing the resulting representations. We verify the superiority of the proposed methods using three different NLP tasks.

2 Sparse Vector Densification

In this section, we introduce a way to compute the similarity between two sparse vectors by augmenting the original similarity measure, i.e., cosine similarity. Suppose we have two vectors $\mathbf{x} = (x_1, \ldots, x_V)^T$ and $\mathbf{y} = (y_1, \ldots, y_V)^T$ where V is the vocabulary size. Traditional cosine similarity computes the dot product between these two vectors and normalizes it by their norms: $\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{||\mathbf{x}||\cdot||\mathbf{y}||}$. This requires each dimension of $\mathbf{x}$ to be aligned with the same dimension of $\mathbf{y}$. Note that for sparse vectors $\mathbf{x}$ and $\mathbf{y}$, most of the the elements can be zero. Aligning the indices can result in zero similarity even though the two pieces of texts are related. Thus, we propose to align different indices of $\mathbf{x}$ and $\mathbf{y}$ together to increase the similarity value.

We can rewrite the vectors $\mathbf{x}$ and $\mathbf{y}$ as $\mathbf{x} = \{x_{a_1}, \ldots, x_{a_{n_x}}\}$ and $\mathbf{y} = \{y_{b_1}, \ldots, y_{b_{n_y}}\}$, where a_i and b_j are indices of the non-zero terms in $\mathbf{x}$ and $\mathbf{y}$ ($1 \leq a_i, b_j \leq V$). x_{a_i} and y_{b_i} are the weights associated to the terms in the vocabulary. Suppose there are non-zero terms n_x and n_y in $\mathbf{x}$ and $\mathbf{y}$ respectively. Then cosine similarity can be rewritten as:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\delta(a_i - b_j)x_{a_i}y_{b_j}}{||\mathbf{x}|| \cdot ||\mathbf{y}||}, \quad (1)$$

where $\delta(\cdot)$ is the Dirac function $\delta(0) = 1$ and $\delta(\text{other}) = 0$. Suppose we can compute the similarity between terms a_i and b_j, which is denoted as $\phi(a_i, b_j)$, then the problem is how to aggregate the similarities between all a_i's and b_j's to augment the original cosine similarity.

2.1 Similarity Augmentation

The most intuitive way to integrate the similarities between terms is averaging them:

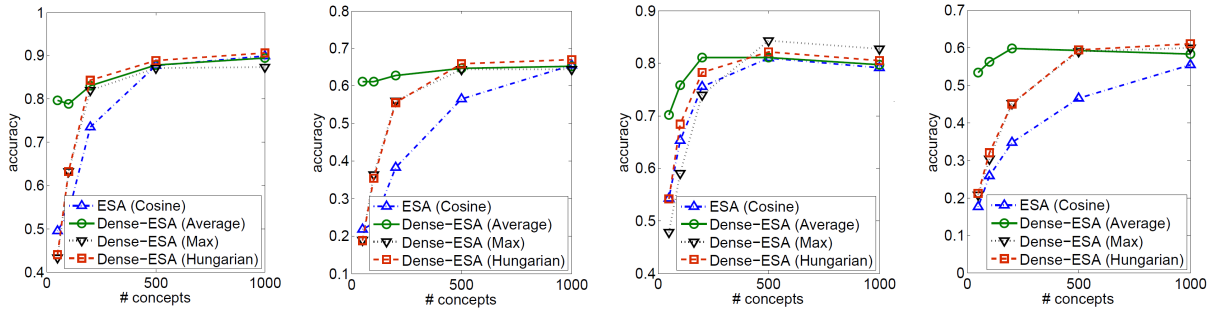
$$S_A(\mathbf{x}, \mathbf{y}) = \frac{1}{n_x||\mathbf{x}|| \cdot n_y||\mathbf{y}||}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}x_{a_i}y_{b_j}\phi(a_i, b_j). \quad (2)$$

This similarity averages all the pairwise similarities between terms a_i's and b_j's. However, we can expect a lot of the similarities $\phi(a_i, b_j)$ to be close to zero. In this case, instead of introducing the relatedness between nonidentical terms, it will also introduce noise. Therefore, we also consider an alignment mechanism that we implement greedily via a maximum matching mechanism:

$$S_M(\mathbf{x}, \mathbf{y}) = \frac{1}{||\mathbf{x}|| \cdot ||\mathbf{y}||}\sum_{i=1}^{n_x}x_{a_i}y_{b_j}\max_j\phi(a_i, b_j). \quad (3)$$

We choose j as $\text{argmax}_{j'}\,\phi(a_i, b_{j'})$ and substitute the similarity $\phi(a_i, b_j)$ between terms a_i and b_j into the final similarity between $\mathbf{x}$ and $\mathbf{y}$. Note that this similarity is not symmetric. Thus, if one needs a symmetric similarity, the similarity can be computed by averaging two similarities $S_M(\mathbf{x}, \mathbf{y})$ and $S_M(\mathbf{y}, \mathbf{x})$.

The above two similarity measurements are simple and intuitive. We can think about $S_A(\mathbf{x}, \mathbf{y})$ as leveraging term many-to-many mapping, while



(a) rec.autos vs. sci.electronics (full doc.) (b) rec.autos vs. sci.electronics (1/16 doc.) (c) rec.autos vs. rec.motorcycles (full doc.) (d) rec.autos vs. rec.motorcycles (1/16 doc.)

Figure 1: Accuracy of dataless classification using ESA and Dense-ESA with different numbers of concepts.

$S_M(\mathbf{x}, \mathbf{y})$ uses only one-to-many term mapping. $S_A(\mathbf{x}, \mathbf{y})$ can introduce small and noisy similarity values between terms. While $S_M(\mathbf{x}, \mathbf{y})$ essentially aligns each term in $\mathbf{x}$ with it's best match in $\mathbf{y}$, we run the risk that multiple components of $\mathbf{x}$ will select the same element in $\mathbf{y}$. To ensure that all the non-zero terms in $\mathbf{x}$ and $\mathbf{y}$ are matched, we propose to constrain this metric by disallowing many-to-one mapping. We do that by using a similarity metric based on the Hungarian method (Papadimitriou and Steiglitz, 1982). The Hungarian method is a combinatorial optimization algorithm that solves the bipartite graph matching problem by finding an optimal assignment matching the two sides of the graph on a one-to-one basis. Assume that we run the Hungarian method on the the pair $\{\mathbf{x}, \mathbf{y}\}$, and let $h(a_i) = b_j$ denote the outcome of the algorithm, that is a_i is aligned with b_j. (We assume here, for simplicity, that $n_x = n_y$; we can always achieve that by adding some zero weighted terms that are not aligned). The we define the similarity as:

$$S_H(\mathbf{x}, \mathbf{y}) = \frac{1}{||\mathbf{x}|| \cdot ||\mathbf{y}||} \sum_{i=1}^{n_x} x_{a_i} y_{h(a_i)} \phi(a_i, h(a_i)). \quad (4)$$

2.2 Term Similarity Measure

To evaluate the term similarity $\phi(\cdot, \cdot)$, we use local contextual similarity based on distributed representations. We adopt the word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b) approach to obtain a dense representation of words. The representation of each word is predicted based on the context word distribution in a window around it. We trained word2vec on the Wikipedia dump data using the default parameters (CBOW model with window size as five). For each word, we finally obtained a 200 dimensional vector. If the term is a phrase, we simply average words' vectors of each phrase to obtain the representation following the original word2vec approach (Mikolov et al., 2013a; Mikolov et al., 2013b). We use two vectors $\mathbf{a}$ and $\mathbf{b}$ to represent the vectors for the two terms. To evaluate the similarity between two terms, for the average approach as Eq. (2), we use the RBF kernel over the two vectors $\exp\{-||\mathbf{a} - \mathbf{b}||^2/(0.03 \cdot ||\mathbf{a}|| \cdot ||\mathbf{b}||)\}$ as the similarity for all the experiments, since this will have a good property to cut the terms with small similarities. For the max and Hungarian approach as Eqs. (3) and (4), we simply use the cosine similarity between the two word2vec vectors. In addition, we cut off all similarities below threshold γ and map them to zero.

3 Experiments

We experiment on three data sets. We use dataless classification (Chang et al., 2008; Song and Roth, 2014) over 20-newsgroups data set to verify the correctness of our argument of short text problems, and use two short text data sets to evaluate document similarity measurement and event classification for sentences.

3.1 Dataless Classification

Dataless classification uses the similarity between documents and labels in an enriched "semantic" space to determine in which category the given document is. In this experiment, we used the label descriptions provided by (Chang et al., 2008). It has been shown that ESA outperforms other representations for dataless classification (Chang et al., 2008; Song and Roth, 2014). Thus, we chose ESA as our

Table 2: Accuracy of dataless classification using ESA and Dense-ESA with 500 dimensions.

	rec.autos vs. sci.electronics (easy)		rec.autos vs. rec.motorcycles (difficult)	
Method	Full document	Short (1/16 doc.)	Full document	Short (1/16 doc.)
ESA (Cosine)	87.75%	56.55%	80.95%	46.64%
Dense-ESA (Average)	87.80%	64.67%	81.11%	59.38%
Dense-ESA (Max)	87.10%	64.34%	**84.30%**	59.11%
Dense-ESA (Hungarian)	**88.85%**	**65.95%**	82.15%	**59.65%**

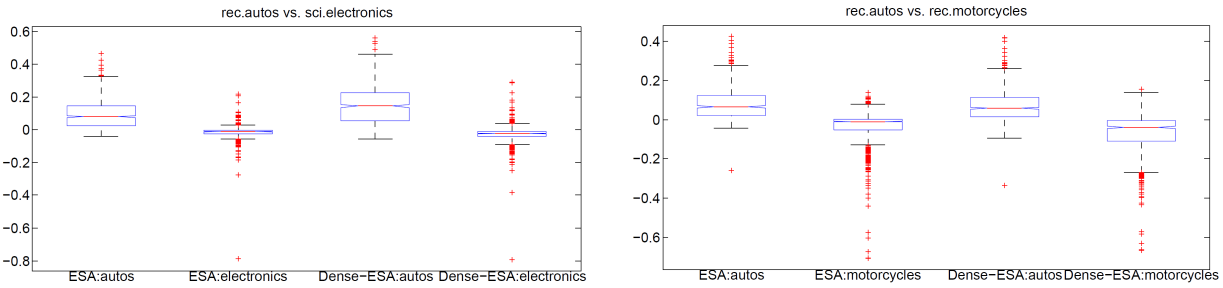


Figure 2: Boxplot of similarity scores for "rec.autos vs. sci.electronics" (easy, left) and "rec.autos vs. rec.motorcycles" (difficult, right). For each method of ESA and Dense-ESA with max matching in Eq. (3), we compute $S(d, l_1)$ and $S(d, l_2)$ between a document d and the labels l_1 and l_2. Then we compute $S(d) = S(d, l_1) - S(d, l_2)$. For each ground truth label, we draw the distribution of $S(\cdot)$ with outliers in the figures. For example, "ESA:autos" shows the $S(\cdot)$'s distribution of the data with label "rec.autos." The t-test results show that the distributions of different labels are significantly different (99%). We can see that Dense-ESA pulls apart the distributions of different labels and that the separation is more significant for the more difficult problem (right).

baseline method. To demonstrate how the length of documents affects the classification result, we used both full documents and the 16 split parts (the parts are associated with the same label as the original document). To demonstrate the impact of densification, we selected two problems as an illustration: "rec.autos vs. sci.electronics" and "rec.autos vs. rec.motorcycles." While the former problem is relatively easy since they belong to different super-classes, the latter problem is more difficult since they are under the same super-class. The value of threshold γ for max matching and Hungarian based densification is set to 0.85 empirically.

Figure 1 shows the results of the dataless classification using ESA and ESA with densification (Dense-ESA) with different numbers of Wikipedia concepts as the representation dimensionality. We can see that Dense-ESA significantly improves the dataless classification results. As shown in Table 2, while the max matching and Hungarian matching based methods are typically the best metrics the most significant results, the improvements are more significant for shorter documents, and for more difficult problems. Figure 2 highlights this observation.

Table 3: Spearman's correlation of document similarity using ESA and Dense-ESA with 500 concepts.

Method	Spearman's correlation
ESA (Cosine)	0.5665
Dense-ESA (Average)	0.5814
Dense-ESA (Max)	0.5888
Dense-ESA (Hungarian)	**0.6003**

3.2 Document Similarity

We used the data set provided by Lee et al.[2] (Lee et al., 2005) to evaluate pairwise short document similarity. There are 50 documents and the average number of words is 80.2. We averaged all the human annotations for the same document pair as the similarity score. After computing the scores for pairs of documents, we used Spearman's correlation to evaluate the results. Larger correlation score means that the similarity is more consistent with human annotation. The best word level based similarity result is close to 0.5 (Lee et al., 2005). We tried the cosine similarity between ESA representations and

[2]http://faculty.sites.uci.edu/mdlee/similarity-data/

Table 4: F_1 of sentence event type classification using ESA and Dense-ESA with 500 concepts.

Method	F_1 (mean±std)
ESA (Cosine)	0.469±0.011
Dense-ESA (Average)	0.451±0.010
Dense-ESA (Max)	**0.481±0.008**
Dense-ESA (Hungarian)	0.475±0.016

also Dense-ESA. The value of γ for max matching based densification is set to 0.95, and for Hungarian based densification it is set to 0.89. We can see that from Table 3, ESA is better than the word based method, and that all versions of Dense-ESA outperform the original ESA.

3.3 Event Classification

In this experiment, we chose the ACE2005[3] data set to test how well we can classify sentences into event types without any training. There are eight types of events: life, movement, conflict, contact, etc. We chose all the sentences that contain event information as the data set. Following the dataless classification protocol, we compare the similarity between sentences and label descriptions to determine the event types. There are 3,644 unique sentences with events, including 2,712 sentences having only one event type, 421 having two event types, and 30 having three event types. The average length of the sentences is 23.71. Thus, this is a multi-label classification problem. To test the approaches, we used five-fold cross validation to select the thresholds for each class to classify whether the sentence belongs to an event type. The value of threshold γ for both max matching and Hungarian based densification is also set to 0.85 empirically. Then we report the mean and standard derivation over five runs. The results are shown in Table 4. We can see that Dense-ESA also outperforms ESA.

4 Related Work

ESA (Gabrilovich and Markovitch, 2006; Gabrilovich and Markovitch, 2007) and distributed word representations (Ratinov and Roth, 2009; Turian et al., 2010; Collobert et al., 2011; Mikolov et al., 2013a; Mikolov et al., 2013b; Pennington et al., 2014) are popular text representations that encode world knowledge. Recently, several representations were proposed to extend word representations for phrases or sentences (Lu and Li, 2013; Hermann and Blunsom, 2014; Passos et al., 2014; Kalchbrenner et al., 2014; Le and Mikolov, 2014; Hu et al., 2014; Sutskever et al., 2014; Zhao et al., 2015). In this paper, we evaluate how to combine two off-the-shelf representations to densify the similarity between text data.

Yih et al. also used average matching and a different maximum matching for QA problem (Yih et al., 2013). However, their sparse representation is still at the word level while ours is based on ESA. Interestingly, related ideas to our average matching mechanism have been proposed also in the computer vision community, which is the set kernel (or set similarity) (Smola et al., 2007; Gretton et al., 2012; Xiong et al., 2013).

5 Conclusion

In this paper, we study the mechanisms of combining two popular representations of text, i.e., ESA and word2vec, to enhance computing short text similarity. Furthermore, we proposed three different mechanisms to compute the similarity between these representations, and demonstrated, using three different data sets that the proposed method outperforms the traditional ESA.

Acknowledgments

This work is supported by the Multimodal Information Access & Synthesis Center at UIUC, part of C-CICADA, a DHS Science and Technology Center of Excellence, by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, and by DARPA under agreement number FA8750-13-2-0008. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied by these agencies or the U.S. Government.

[3] http://www.itl.nist.gov/iad/mig/tests/ace/2005/

References

M. Chang, L. Ratinov, D. Roth, and V. Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI*, pages 830–835.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

E. Gabrilovich and S. Markovitch. 2006. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, pages 1301–1306.

E. Gabrilovich and S. Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI*, pages 1606–1611.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. 2012. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773.

K. M. Hermann and P. Blunsom. 2014. Multilingual models for compositional distributed semantics. In *ACL*, pages 58–68.

B. Hu, Z. Lu, H. Li, and Q. Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, pages 2042–2050.

N. Kalchbrenner, E. Grefenstette, and P. Blunsom. 2014. A convolutional neural network for modelling sentences. In *ACL*, pages 655–665.

Q. V. Le and T. Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196.

M. D. Lee, B. Pincombe, and M. Welsh. 2005. An empirical evaluation of models of text document similarity. In *CogSci*, pages 1254–1259.

Z. Lu and H. Li. 2013. A deep architecture for matching short texts. In *NIPS*, pages 1367–1375.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

T. Mikolov, W.-t. Yih, and G. Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

C. H. Papadimitriou and K. Steiglitz. 1982. *Combinatorial Optimization: Algorithm und Complexity*. Englewood Cliffs, NJ: Prentice-Hall.

A. Passos, V. Kumar, and A. McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *CoNLL*, pages 78–86.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*, pages 147–155.

A. J. Smola, A. Gretton, L. Song, and B. Schölkopf. 2007. A hilbert space embedding for distributions. In *ALT*, pages 13–31.

Y. Song and D. Roth. 2014. On dataless hierarchical text classification. In *AAAI*, pages 1579–1585.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*, pages 384–394.

L. Xiong, B. Póczos, and J. G. Schneider. 2013. Efficient learning on point sets. In *ICDM*, pages 847–856.

W. Yih, M. Chang, C. Meek, and A. Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *ACL*, pages 1744–1753.

Y. Zhao, Z. Liu, and M. Sun. 2015. Phrase type sensitive tensor indexing model for semantic composition. In *AAAI*.