SCDV : Sparse Composite Document Vectors using soft clustering over distributional representations

Dheeraj Mekala*
IIT Kanpur
dheerajm@iitk.ac.in

Vivek Gupta*
Microsoft Research
t-vigu@microsoft.com

Bhargavi Paranjape
Microsoft Research
t-bhpara@microsoft.com

Harish Karnick
IIT Kanpur
hk@iitk.ac.in

Abstract

We present a feature vector formation technique for documents - Sparse Composite Document Vector (SCDV) - which overcomes several shortcomings of the current distributional paragraph vector representations that are widely used for text representation. In SCDV, word embeddings are clustered to capture multiple semantic contexts in which words occur. They are then chained together to form document topic-vectors that can express complex, multi-topic documents. Through extensive experiments on multi-class and multi-label classification tasks, we outperform the previous state-of-the-art method, NTSG (Liu et al., 2015a). We also show that SCDV embeddings perform well on heterogeneous tasks like Topic Coherence, context-sensitive Learning and Information Retrieval. Moreover, we achieve significant reduction in training and prediction times compared to other representation methods. SCDV achieves best of both worlds - better performance with lower time and space complexity.

1 Introduction

Distributed word embeddings represent words as dense, low-dimensional and real-valued vectors that can capture their semantic and syntactic properties. These embeddings are used abundantly by machine learning algorithms in tasks such as text classification and clustering. Traditional bag-of-word models that represent words as indices into a vocabulary don't account for word ordering and long-distance semantic relations. Representations based on neural network language models (Mikolov et al., 2013b) can overcome these flaws and further reduce the dimensionality of the vectors. The success of the method is recently mathematically explained using the random walk on discourses model (Arora et al., 2016a). However, there is a need to extend word embeddings to entire paragraphs and documents for tasks such as document and short-text classification.

Representing entire documents in a dense, low-dimensional space is a challenge. A simple weighted average of the word embeddings in a large chunk of text ignores word ordering, while a parse tree based combination of embeddings (Socher et al., 2013) can only extend to sentences. (Le and Mikolov, 2014) trains word and paragraph vectors to predict context but shares word-embeddings across paragraphs. However, words can have different semantic meanings in different contexts. Hence, vectors of two documents that contain the same word in two distinct senses need to account for this distinction for an accurate semantic representation of the documents. (Ling et al., 2015), (Liu et al., 2015a) map word embeddings to a latent topic space to capture different senses in which words occur. However, they represent complex documents in the same space as words, reducing their expressive power. These methods are also computationally intensive.

In this work, we propose the Sparse Composite Document Vector(SCDV) representation learning technique to address these challenges and create efficient, accurate and robust semantic representations of large texts for document classification tasks. SCDV combines syntax and semantics learnt by word embedding models together with a latent topic model that can handle different senses of words, thus enhancing the expressive power of document vectors. The topic space is learnt efficiently using a soft clustering technique over embeddings and the final document vectors are made

*Represents equal contribution

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 659–669
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

sparse for reduced time and space complexity in tasks that consume these vectors.

The remaining part of the paper is organized as follows. Section 2 discusses related work in document representations. Section 3 introduces and explains SCDV in detail. This is followed by extensive and rigorous experiments together with analysis in section 4 and 5 respectively.

2 Related Work

(Le and Mikolov, 2014) proposed two models for distributional representation of a document, namely, *Distributed Memory Model Paragraph Vectors (PV-DM)* and *Distributed BoWs paragraph vectors (PV-DBoW)*. In *PV-DM*, the model is learned to predict the next context word using word and paragraph vectors. In *PV-DBoW*, the paragraph vector is directly learned to predict randomly sampled context words. In both models, word vectors are shared across paragraphs. While word vectors capture semantics across different paragraphs of the text, documents vectors are learned over context words generated from the same paragraph and potentially capture only local semantics (Singh and Mukerjee, 2015). Moreover, a paragraph vector is embedded in the same space as word vectors though it can contain multiple topics and words with multiple senses. As a result, doc2vec (Le and Mikolov, 2014) doesn't perform well on Information Retrieval as described in (Ai et al., 2016a) and (Roy et al., 2016). Consequently, we expect a paragraph vector to be embedded in a higher dimensional space.

A paragraph vector also assumes all words contribute equally, both quantitatively (weight) and qualitatively (meaning). They ignore the importance and distinctiveness of a word across all documents (Singh and Mukerjee, 2015). Mukerjee et al. (Singh and Mukerjee, 2015) proposed idf-weighted averaging of word vectors to form document vectors. This method tries to address the above problem. However, it assumes that all words within a document belong to the same semantic topic. Intuitively, a paragraph often has words originating from several semantically different topics. In fact, Latent Dirichlet Allocation (Blei et al., 2003) models a document as a distribution of multiple topics.

These shortcomings are addressed in three novel composite document representations called *Topical word embedding (TWE-1,TWE-2 and TWE-3)* by (Liu et al., 2015a). *TWE-1* learns word and topic embeddings by considering each topic as a pseudo word and builds the topical word embedding for each word-topic assignment. Here, the interaction between a word and the topic to which it is assigned is not considered. *TWE-2* learns a topical word embedding for each word-topic assignment directly, by considering each word- topic pair as a pseudo word. Here, the interaction between a word and its assigned topic is considered but the vocabulary of pseudo-words blows up. For each word and each topic, *TWE-3* builds distinct embeddings for the topic and word and concatenates them for each word-topic assignment. Here, the word embeddings are influenced by the corresponding topic embeddings, making words in the same topic less discriminative.

(Liu et al., 2015a) proposed an architecture called *Neural tensor skip-gram model (NTSG-1, NTSG-2, NTSG-3, NTSG-4)*, that learns multi-prototype word embeddings and uses a tensor layer to model the interaction of words and topics to capture different senses. $NTSG$ outperforms other embedding methods like $TWE-1$ on the 20 newsgroup data-set by modeling context-sensitive embeddings in addition to topical-word embeddings. $LTSG$ (Law et al., 2017) builds on $NTSG$ by jointly learning the latent topic space and context-sensitive word embeddings. All three, TWE, $NTSG$ and $LTSG$ use LDA and suffer from computational issues like large training time, prediction time and storage space. They also embed document vectors in the same space as terms. Other works that harness topic modeling like WTM (Fu et al., 2016), $w2v-LDA$ (Nguyen et al., 2015), $TV + MeanWV$ (Li et al., 2016a), $LTSG$ (Law et al., 2017), $Gaussian - LDA$ (Das et al., 2015), $Topic2Vec$ (Niu et al., 2015), (Moody, 2016) and $MvTM$ (Li et al., 2016b) also suffer from similar issues.

(Gupta et al., 2016) proposed a method to form a composite document vector using word embeddings and tf-idf values, called the *Bag of Words Vector (BoWV)*. In $BoWV$, each document is represented by a vector of dimension $D = K * d + K$, where K is the number of clusters and d is the dimension of the word embeddings. The core idea behind $BoWV$ is that semantically different words belong to different topics and their word vectors should not be averaged. Further, $BoWV$ computes inverse cluster frequency of each clus-

ter (icf) by averaging the idf values of its member terms to capture the importance of words in the corpus. However, $BoWV$ does hard clustering using K-means algorithm, assigning each word to only one cluster or semantic topic but a word can belong to multiple topics. For example, the word *apple* belongs to topic *food* as a fruit, and belongs to topic *Information Technology* as an IT company. Moreover, $BoWV$ is a non-sparse, high dimensional continuous vector and suffers from computational problems like large training time, prediction time and storage requirements.

3 Sparse Composite Document Vectors

In this section, we present the proposed Sparse Composite Document Vector (SCDV) representation as a novel document vector learning algorithm. The feature formation algorithm can be divided into three steps.

3.1 Word Vector Clustering

We begin by learning d dimensional word vector representations for every word in the vocabulary V using the skip-gram algorithm with negative sampling (SGNS) (Mikolov et al., 2013a). We then cluster these word embeddings using the Gaussian Mixture Models(GMM) (Reynolds, 2015) soft clustering technique. The number of clusters, *K*, to be formed is a parameter of the SCDV model. By inducing soft clusters, we ensure that each word belongs to every cluster with some probability $P(c_k|w_i)$.

$$p(c_k = 1) = \pi_k$$

$$p(c_k = 1|w) = \frac{\pi_k \mathcal{N}(w|\mu_k, \Sigma_k)}{\Sigma_{j=1}^{K} \pi_j \mathcal{N}(w|\mu_j, \Sigma_j)}$$

3.2 Document Topic-vector Formation

For each word w_i, we create K different word-cluster vectors of d dimensions ($w\vec{c}v_{ik}$) by weighting the word's embedding with its probability distribution in the kth cluster, $P(c_k|w_i)$. We then concatenate all K word-cluster vectors ($w\vec{c}v_{ik}$) into a K×d dimensional embedding and weight it with inverse document frequency of w_i to form a word-topics vector ($w\vec{t}v_i$). Finally, for all words appearing in document D_n, we sum their word-topic vectors $w\vec{t}v_i$ to obtain the document vector $\vec{dv}_{D_n}$.

$$w\vec{c}v_{ik} = w\vec{v}_i \times P(c_k|w_i)$$

Algorithm 1: Sparse Composite Document Vector

Data: Documents D_n, n = 1 . . . N
Result: Document vectors $S\vec{CDV}_{D_n}$, n = 1 . . . N

1 Obtain word vector ($w\vec{v}_i$), for each word w_i;
2 Calculate idf values, $idf(w_i)$, $i = 1..|V|$;
  /* $|V|$ is vocabulary size */
3 Cluster word vectors $w\vec{v}$ using GMM clustering into K clusters;
4 Obtain soft assignment $P(c_k|w_i)$ for word w_i and cluster c_k;
  /* Loop 5-10 can be pre-computed */
5 **for** *each word* w_i *in vocabulary* V **do**
6 **for** *each cluster* c_k **do**
7 $w\vec{c}v_{ik} = w\vec{v}_i \times P(c_k|w_i)$;
8 **end**
9 $w\vec{t}v_i = idf(w_i) \times \bigoplus_{k=1}^{K} w\vec{c}v_{ik}$;
  /* $\bigoplus$ is concatenation */
10 **end**
11 **for** $n \in (1..N)$ **do**
12 Initialize document vector $\vec{dv}_{D_n} = \vec{0}$;
13 **for** *word* w_i *in* D_n **do**
14 $\vec{dv}_{D_n}$ += $w\vec{t}v_i$;
15 **end**
16 $S\vec{CDV}_{D_n}$ = make-sparse($\vec{dv}_{D_n}$);
  /* as mentioned in sec 3 */
17 **end**

$$w\vec{t}v_i = idf(w_i) \times \bigoplus_{k=1}^{K} w\vec{c}v_{ik}$$

where, $\bigoplus$ is concatenation

3.3 Sparse Document Vectors

After normalizing the vector, we observed that most values in $\vec{dv}_{D_n}$ are very close to zero. Figure 3 verifies this observation. We utilize this fact to make the document vector $\vec{dv}_{D_n}$ sparse by zeroing attribute values whose absolute value is close to a threshold (specified as a parameter), which results in the Sparse Composite Document Vector $S\vec{CDV}_{D_n}$.

In particular, let p be percentage sparsity threshold parameter, a_i the value of the ith attribute of the non-Sparse Composite Document Vector and n represent the nth document in the training set:

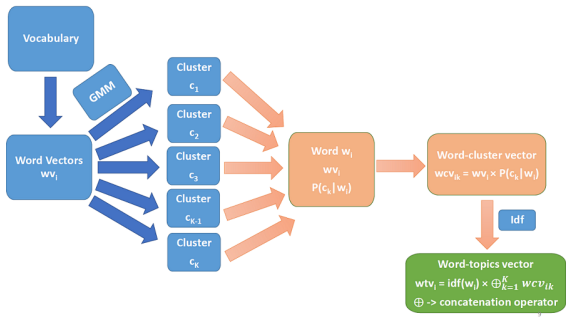

Figure 1: Word-topics vector formation.

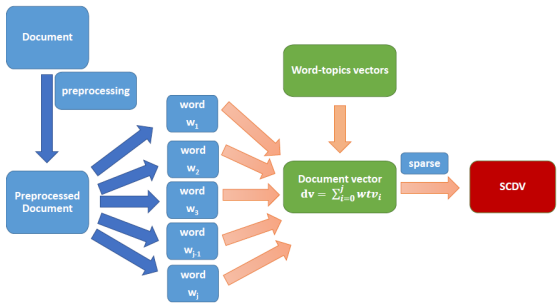

Figure 2: Sparse Composite Document Vector formation.

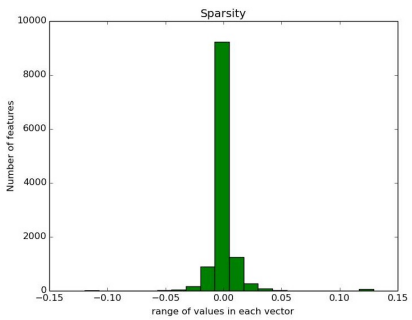

Figure 3: Distribution of attribute feature vector values.

$$a_i = \begin{cases} a_i & \text{if } |a_i| \geq \frac{p}{100} * t \\ 0 & \text{otherwise} \end{cases}$$

$$t = \frac{|a_{min}| + |a_{max}|}{2}$$

$$a_{min} = avg_n(min_i(a_i))$$

$$a_{max} = avg_n(max_i(a_i))$$

Flowcharts depicting the formation of word-topics vector and Sparse Composite Document Vectors are shown in figure 1 and figure 2 respectively. Algorithm 1 describes SCDV in detail.

4 Experiments

We perform multiple experiments to show the effectiveness of SCDV representations for multi-class and multi-label text classification. For all experiments and baselines, we use Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz, 40 working cores, 128GB RAM machine with Linux Ubuntu 14.4. However, we utilize multiple cores only during Word2Vec training and when we run the one-vs-rest classifier for Reuters.

4.1 Baselines

We consider the following baselines: Bag-of-Words (BoW) model (Harris, 1954), Bag of Word Vector (BoWV) (Gupta et al., 2016) model, paragraph vector models (Le and Mikolov, 2014), Topical word embeddings (TWE-1) (Liu et al., 2015b), Neural Tensor Skip-Gram Model (NTSG-1 to NTSG-3) (Liu et al., 2015a), tf-idf weighted average word-vector model (Singh and Mukerjee, 2015) and weighted Bag of Concepts (weight-BoC) (Kim et al., 2017), where we build topic-document vectors by counting the member words in each topic.

We use the best parameter settings as reported in all our baselines to generate their results. We use 200 dimensions for tf-idf weighted word-vector model, 400 for paragraph vector model, 80 topics and 400 dimensional vectors for TWE, NTSG, LTSG and 60 topics and 200 dimensional word vectors for BOWV. We also compare our results with reported results of other topic modeling based document embedding methods like WTM (Fu et al., 2016), $w2v - LDA$ (Nguyen et al., 2015), LDA (Chen and Liu, 2014), $TV + MeanWV$ (Li et al., 2016a), $LTSG$ (Law et al., 2017), $Gaussian - LDA$ (Das et al., 2015), $Topic2Vec$ (Niu et al., 2015), (Moody, 2016) and $MvTM$ (Li et al., 2016b). Implementation of SCDV and related experiments is available here [1].

4.2 Text Classification

We run multi-class experiments on 20NewsGroup dataset [2] and multi-label classification experiments on Reuters-21578 dataset [3]. We use the script[4] for preprocessing the Reuters-21578 dataset. We use LinearSVM for multi-class classi-

[1] https://github.com/dheeraj7596/SCDV
[2] http://qwone.com/~jason/20Newsgroups/
[3] https://goo.gl/NrOfu
[4] https://gist.github.com/herrfz/7967781

fication and Logistic regression with OneVsRest setting for multi-label classification in baselines and SCDV.

For SCDV, we set the dimension of word-embeddings to 200 and the number of mixture components in GMM to 60. All mixture components share the same spherical co-variance matrix. We learn word vector embedding using Skip-Gram with window size of 10, Negative Sampling (SGNS) of 10 and minimum word frequency as 20. We use 5-fold cross-validation on F1 score to tune parameter C of SVM and the sparsity threshold for SCDV.

4.2.1 Multi-class classification

We evaluate classifier performance using standard metrics like accuracy, macro-averaging precision, recall and F-measure. Table 1 shows a comparison with the current state-of-art (NTSG) document representations on the 20Newsgroup dataset. We observe that SCDV outperforms all other current models by fair margins. We also present the class-wise precision and recall for 20Newsgroup on an almost balanced dataset with SVM over Bag of Words model and the SCDV embeddings in Table 2 and observe that SCDV improves consistently over all classes.

Table 1: Performance on multi-class classification (Values in red show best performance, the SCDV algorithm of this paper)

Model	Acc	Prec	Rec	F-mes
SCDV	**84.6**	**84.6**	**84.5**	**84.6**
NTSG-1	82.6	82.5	81.9	81.2
NTSG-2	82.5	83.7	82.8	82.4
BoWV	81.6	81.1	81.1	80.9
NTSG-3	81.9	83.0	81.7	81.1
LTSG	82.8	82.4	81.8	81.8
WTM	80.9	80.3	80.3	80.0
w2v-LDA	77.7	77.4	77.2	76.9
TV+MeanWV	72.2	71.8	71.5	71.6
MvTM	72.2	71.8	71.5	71.6
TWE-1	81.5	81.2	80.6	80.6
lda2Vec	81.3	81.4	80.4	80.5
lda	72.2	70.8	70.7	70.0
weight-AvgVec	81.9	81.7	81.9	81.7
BoW	79.7	79.5	79.0	79.0
weight-BOC	71.8	71.3	71.8	71.4
PV-DBoW	75.4	74.9	74.3	74.3
PV-DM	72.4	72.1	71.5	71.5

Table 2: Class-level results on the balanced 20newsgroup dataset.

	BoW		SCDV	
Class Name	**Pre.**	**Rec.**	**Pre.**	**Rec.**
alt.atheism	67.8	72.1	80.2	79.5
comp.graphics	67.1	73.5	75.3	77.4
comp.os.ms-windows.misc	77.1	66.5	78.6	77.2
comp.sys.ibm.pc.hardware	62.8	72.4	75.6	73.5
comp.sys.mac.hardware	77.4	78.2	83.4	85.5
comp.windows.x	83.2	73.2	87.6	78.6
misc.forsale	81.3	88.2	81.4	85.9
rec.autos	80.7	82.8	91.2	90.6
rec.motorcycles	92.3	87.9	95.4	95.7
rec.sport.baseball	89.8	89.2	93.2	94.7
rec.sport.hockey	93.3	93.7	96.3	99.2
sci.crypt	92.2	86.1	92.5	94.7
sci.electronics	70.9	73.3	74.6	74.9
sci.med	79.3	81.3	91.3	88.4
sci.space	90.2	88.3	88.5	93.8
soc.religion.christian	77.3	87.9	83.3	92.3
talk.politics.guns	71.7	85.7	72.7	90.6
talk.politics.mideast	91.7	76.9	96.2	95.4
talk.politics.misc	71.7	56.5	80.9	59.7
talk.religion.misc	63.2	55.4	73.5	57.2

4.2.2 Multi-label classification

We evaluate multi-label classification performance using Precision@K, nDCG@k (Bhatia et al., 2015), Coverage error, Label ranking average precision score (LRAPS)[5] and F1-score. All measures are extensively used for the multi-label classification task. However, F1-score is an appropriate metric for multi-label classification as it considers label biases when train-test splits are random. Table 3 show evaluation results for multi-label text classification on the Reuters-21578 dataset.

4.2.3 Effect of Hyper-Parameters

SCDV has three parameters: the number of clusters, word vector dimension and sparsity threshold parameter. We vary one parameter by keeping the other two constant. Performance on varying all three parameters in shown in Figure 4. We observe that performance improves as we increase the number of clusters and saturates at 60. The performance improves until a word vector dimension of 300 after which it saturates. Similarly, we observe that the performance improves as we increase p till 4 after which it declines. At 4% thresholding, we reduce the storage space by 80% compared to the dense vectors. We observe that SCDV is robust to variations in training Word2Vec

[5]Section 3.3.3.2 of https://goo.gl/4GrR3M

Table 3: Performance on various metrics for multi-label classification for Reuters(Values in red show best performance, the SCDV algorithm of this paper)

Model	Prec@1 nDCG@1	Prec @5	nDCG @5	Coverage Error	LRAPS	F1-Score
SCDV	**94.20**	**36.98**	**49.55**	**6.48**	**93.30**	**81.75**
BoWV	92.90	36.14	48.55	8.16	91.46	79.16
TWE-1	90.91	35.49	47.54	8.16	91.46	79.16
PV-DM	87.54	33.24	44.21	13.15	86.21	70.24
PV-DBoW	88.78	34.51	46.42	11.28	87.43	73.68
AvgVec	89.09	34.73	46.48	9.67	87.28	71.91
tfidf AvgVec	89.33	35.04	46.83	9.42	87.90	71.97

and GMM. The performance metrics reported in Tables 1, 3 are the average values obtained across 5 separate runs of $SCDV$, each run training a different Word2Vec and GMM model with identical hyper-parameters.

4.3 Topic Coherence

We evaluate the topics generated by GMM clustering on 20NewsGroup for quantitative and qualitative analysis. Instead of using perplexity (Chang et al., 2011), which doesn't correlate with semantic coherence and human judgment of individual topics, we used the popular topic coherence (Mimno et al., 2011), (Arora et al., 2013), (Chen and Liu, 2014) measure. A higher topic coherence score indicates a more coherent topic.

We used Bayes rule to compute the $P(w_k|c_i)$ for a given topic c_i and given word w_j and compute the score of the top 10 words for each topic.

$$P(w_k|c_i) = \frac{P(c_i|w_k)P(w_k)}{P(c_i)}$$

where,

$$P(c_i) = \sum_{i=1}^{K} P(c_i|w_k)P(w_k)$$

$$P(w_k) = \frac{\#(w_k)}{\sum_{i=1}^{V} \#(w_i)}$$

Here, $\#(w_k)$ denotes the number of times word w_k appears in the corpus and V represents vocabulary size.

We calculated the topic coherence score for all topics for $SCDV$, LDA and $LTSG$ (Law et al., 2017). Averaging the score of all 80 topics, GMM clustering scores -85.23 compared to -108.72 of LDA and -92.23 of LTSG. Thus, SCDV creates more coherent topics than both LDA and LTSG.

Table 4 shows top 10 words of 3 topics from GMM clustering, LDA model and $LTSG$ model on 20NewsGroup and $SCDV$ shows higher topic coherence. Words are ranked based on their probability distribution in each topic. Our results also support the qualitative results of (Randhawa et al., 2016), (Sridhar, 2015) paper, where k-means, GMM was used respectively over word vectors to find topics.

4.4 Context-Sensitive Learning

In order to demonstrate the effects of soft clustering (GMM) during SCDV formation, we select some words (w_j) with multiple senses from 20Newsgroup and their soft cluster assignments to find the dominant clusters. We also select top scoring words (w_k) from each cluster (c_i) to represent the meaning of that cluster. Table 5 shows polysemic words and their dominant clusters with assignment probabilities. This indicates that using soft clustering to learn word vectors helps combine multiple senses into a single embedding vector. (Arora et al., 2016b) also reported similar results for polysemous words.

4.5 Information Retrieval

(Ai et al., 2016b) used (Mikolov et al., 2013b)'s paragraph vectors to enhance the basic language model based retrieval model. The language model(LM) probabilities are estimated from the corpus and smoothed using a Dirichlet prior (Zhai and Lafferty, 2004). In (Ai et al., 2016b), this language model is then interpolated with the paragraph vector (PV) language model as follows.

$$P(w|d) = (1-\lambda)P_{LM}(w|d) + \lambda P_{PV}(w|d)$$

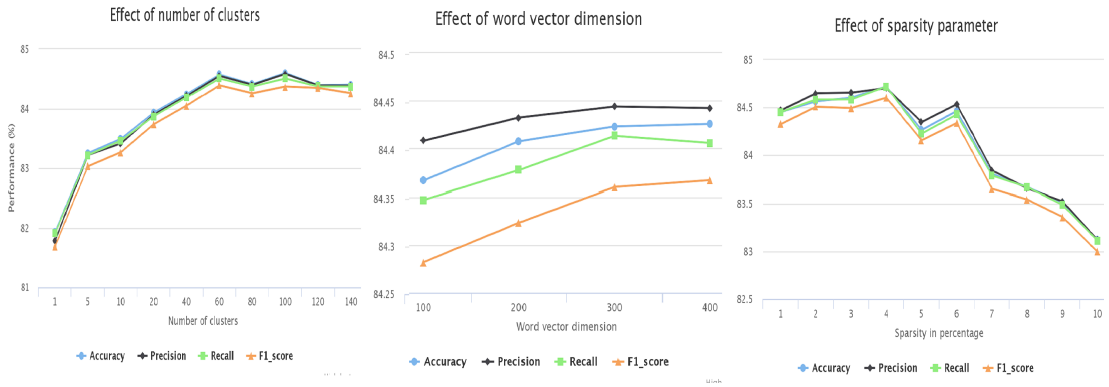

Figure 4: Effect of varying number of clusters (left), varying word vector dimension (center) and varying sparsity parameter (right) on performance for 20NewsGroup with SCDV

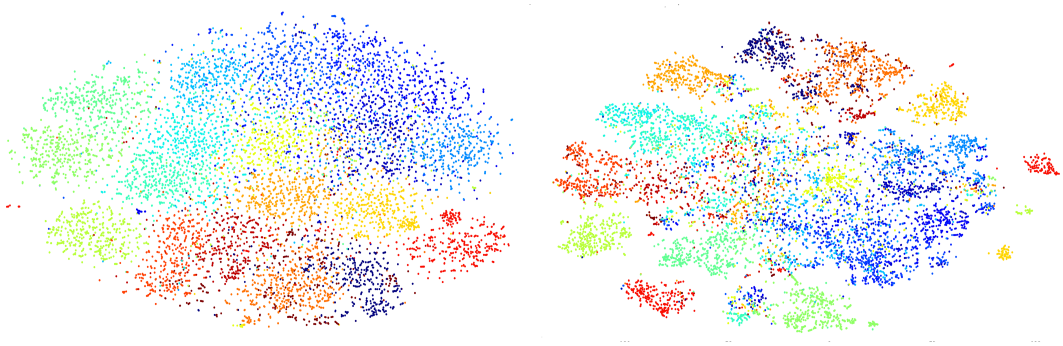

Figure 5: Visualization of paragraph vectors(left) and SCDV(right) using t-SNE

where,

$$P_{PV}(w|d) = \frac{exp(\vec{w}.\vec{d})}{\sum_{i=1}^{V} exp(\vec{w_i}.\vec{d})}$$

and the score for document d and query string Q is given by

$$score(q,d) = \sum_{w \in Q} P(w)P(w|d)$$

where $P(w)$ is obtained from the unigram query model and $score(q,d)$ is used to rank documents. (Ai et al., 2016b) do not directly make use of paragraph vectors for the retrieval task, but improve the document language model. To directly make use of paragraph vectors and make computations more tractable, we directly interpolate the language model query-document score $score(q,d)$ with the similarity score between the normalized query and document vectors to generate $score_{PV}(q,d)$, which is then used to rank documents.

$$score_{PV}(q,d) = (1-\lambda)score(q,d) + \lambda\vec{q}.\vec{d}$$

Directly evaluating the document similarity score with the query paragraph vector rather than collecting similarity scores for individual words in the query helps avoid confusion amongst distinct query topics and makes the interpolation operation faster. In Table 6, we report Mean Average Precision(MAP) values for four datasets, Associated Press 88-89 (topics 51-200), Wall Street Journal (topics 51-200), San Jose Mercury (topics 51-150) and Disks 4 & 5 (topics 301-450) in the TREC collection. We learn λ on a held out set of topics. We observe consistent improvement in MAP for all datasets. We marginally improve the MAP reported by (Ai et al., 2016b) on the Robust04 task. In addition, we also report the improvements in MAP score when Model based relevance feedback (Zhai and Lafferty, 2001) is applied over the initially retrieved results from both models. Again, we notice a consistent improvement in MAP.

5 Analysis and Discussion

SCDV overcomes several challenges encountered while training document vectors, which we had mentioned above.

Table 4: Top words of some topics from GMM and LDA on 20NewsGroup for K = 80. Higher score represent better coherent topics.

Topic Image			Topic Health			Topic Mail		
GMM	**LTSG**	**LDA**	**GMM**	**LTSG**	**LDA**	**GMM**	**LTSG**	**LDA**
file	image	image	heath	stimulation	doctor	ftp	anonymous	list
bit	jpeg	file	study	diseases	disease	mail	faq	mail
image	gif	color	medical	disease	coupons	internet	send	information
files	format	gif	drug	toxin	treatment	phone	ftp	internet
color	file	jpeg	test	toxic	pain	email	mailing	send
format	files	file	drugs	newsletter	medical	send	server	posting
images	convert	format	studies	staff	day	opinions	mail	email
jpeg	color	bit	disease	volume	microorganism	fax	alt	group
gif	formats	images	education	heaths	medicine	address	archive	news
program	images	quality	age	aids	body	box	email	anonymous
-67.16	-75.66	-88.79	-66.91	-96.98	-100.39	-77.47	-78.23	-95.47

Table 5: Words with multiple senses assigned to multiple clusters with significant probabilities

Word	**Cluster Words**	$\mathbf{P}(c_i \mid w_j)$
subject:1	physics, chemistry, math, science	0.27
subject:2	mail, letter, email, gmail	0.72
interest:1	information, enthusiasm, question	0.65
interest:2	bank, market, finance, investment	0.32
break:1	vacation, holiday, trip, spring	0.52
break:2	encryption, cipher, security, privacy	0.22
break:3	if, elseif, endif, loop, continue	0.23
unit:1	calculation, distance, mass, length	0.25
unit:2	electronics, KWH, digital, signal	0.69

1. Clustering word-embeddings to discover topics improves performance of classification as Figure 4 (left) indicates, while also generating coherent clusters of words (Table 4). Figure 5 shows that clustering gives more discriminative representations of documents than paragraph vectors do since it uses K $\times$ d dimensions while paragraph vectors embed documents and words in the same space. This enables SCDV to represent complex documents. Fuzzy clustering allows words to belong to multiple topics, thereby recognizing polysemic words, as Table 5 indicates. Thus it mimics the word-context interaction in NTSG and LTSG.

2. Semantically different words are assigned to different topics. Moreover, a single document can contain words from multiple different topics. Instead of a weighted averaging of word embeddings to form document vectors, as in most previous work, concatenating word embeddings for each topic (cluster) avoids merging of semantically different topics.

3. It is well-known that in higher dimensions, structural regularizers such as sparsity help overcome the curse of dimensionality (Wainwright, 2014).Figure 3 demonstrates this, since majority of the features are close to zero. Sparsity also enables linear SVM to scale to large dimensions. On 20NewsGroups, BoWV model takes up 1.1 GB while SCDV takes up only 236MB(80% decrease). Since GMM assigns a non-zero probability to every topic in the word embedding, noise can accumulate when document vectors are created and tip the scales in favor of an unrelated topic. Sparsity helps to reduce this by zeroing out very small values of probability.

4. SCDV uses Gaussian Mixture Model (GMM) while TWE, $NTSG$ and $LTSG$ use LDA for finding semantic topics respectively. GMM time complexity is $\mathcal{O}(VNT^2)$ while that of LDA is $\mathcal{O}(V^2NT)$. Here, V = Vocabulary size, N = number of documents and T = number of topics. Since number of topics T $<$ vocabulary size V, GMM is faster. Empirically, compared to TWE, $SCDV$ reduces document vector formation, training and prediction time significantly. Table 7 shows training and prediction times for BoWV, SCDV and TWE models.

6 Conclusion

In this paper, we propose a document feature formation technique for topic-based document representation. SCDV outperforms state-of-the-art models in multi-class and multi-label classification tasks. SCDV introduces sparsity in document vectors to handle high dimensionality. Table 7 in-

Table 6: Mean average precision (MAP) for IR on four IR datasets

Dataset	LM	LM+SCDV	MB	MB + SCDV
AP	0.2742	**0.2856**	0.3283	**0.3395**
SJM	0.2052	**0.2105**	0.2341	**0.2409**
WSJ	0.2618	**0.2705**	0.3027	**0.3126**
Robust04	0.2516	**0.2684**	0.2819	**0.2933**

Table 7: Time Comparison (20NewsGroup) (Values in red show least time, the SCDV algorithm of this paper)

Time (sec)	BoWV	TWE-1	SCDV
DocVec Formation	1250	700	**160**
Total Training	1320	740	**200**
Total Prediction	780	120	**25**

dicates that SCDV shows considerable improvements in feature formation, training and prediction times for the 20NewsGroups dataset. We show that fuzzy GMM clustering on word-vectors lead to more coherent topic than LDA and can also be used to detect Polysemic words. SCDV embeddings also provide a robust estimation of the query and document language models, thus improving the MAP of language model based retrieval systems. In conclusion, SCDV is simple, efficient and creates a more accurate semantic representation of documents.

Acknowledgments

The authors want to thank Nagarajan Natarajan (Post-Doc, Microsoft Research, India), Praneeth Netrapalli (Researcher, Microsoft Research, India), Raghavendra Udupa (Researcher, Microsoft Research, India), Prateek Jain (Researcher, Microsoft Research, India) for encouraging and valuable feedback .

References

Qingyao Ai, Liu Yang, Jiafeng Guo, and W Bruce Croft. 2016a. Analysis of the paragraph vector model for information retrieval. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval*, pages 133–142. ACM.

Qingyao Ai, Liu Yang, Jiafeng Guo, and W Bruce Croft. 2016b. Improving language estimation with the paragraph vector model for ad-hoc retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 869–872. ACM.

Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*, pages 280–288.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016a. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016b. Linear algebraic structure of word senses, with applications to polysemy. *arXiv preprint arXiv:1601.03764*.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 730–738.

David M. Blei, David M. Ng, Michael I. Jordan, and John Lafferty. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003.

Jonathan Chang, Jordan L Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. 2011. Reading tea leaves: How humans interpret topic models. pages 262–272.

Zhiyuan Chen and Bing Liu. 2014. Topic modeling using topics from many domains, lifelong learning and big data. In *International Conference on Machine Learning*, pages 703–711.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian lda for topic models with word embeddings. In *Proceedings of The 53th Annual Meeting of the Association of Computational Linguistic (ACL)*, pages 795–804. Association of Computational Linguistic.

Xianghua Fu, Ting Wang, Jing Li, Chong Yu, and Wangwang Liu. 2016. Improving distributed word representation and topic model by word-topic mixture model. In *Proceedings of The 8th Asian Conference on Machine Learning*, pages 190–205.

Vivek Gupta, Harish Karnick, Ashendra Bansal, and Pradhuman Jhala. 2016. Product classification in e-commerce using distributional semantics. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 536–546.

Zellig Harris. 1954. Distributional structure. *Word*, 10:146–162.

Han Kyul Kim, Hyunjoong Kim, and Sungzoon Cho. 2017. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*.

Jarvan Law, Hankz Hankui Zhuo, Junhua He, and Erhu Rong. 2017. Ltsg: Latent topical skip-gram for mutually learning topic model and vector representations. *arXiv preprint arXiv:1702.07117*.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

Shaohua Li, Tat-Seng Chua, Jun Zhu, and Chunyan Miao. 2016a. Generative topic embedding: a continuous representation of documents. In *Proceedings of The 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association of Computational Linguistic.

Ximing Li, Jinjin Chi, Changchun Li, Jihong Ouyang, and Bo Fu. 2016b. Integrating topic modeling with word embeddings by mixtures of vmfs. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 151–160.

Wand Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word-vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. North American Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2015a. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of IJCAI 2014, the 24th International Joint Conference on Artificial Intelligence*, pages 1284–1290.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015b. Topical word embeddings. In *Proceedings of AAAI 2015, the 29th Twenty-Ninth Association for the Advancement of Artificial Intelligence*, pages 2418–2424.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics.

Christopher E Moody. 2016. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

Liqiang Niu, Xinyu Dai, Jianbing Zhang, and Jiajun Chen. 2015. Topic2vec: learning distributed representations of topics. In *2015 International Conference on Asian Language Processing (IALP)*, pages 193–196. IEEE.

Ramandeep S Randhawa, Parag Jain, and Gagan Madan. 2016. Topic modeling using distributed word embeddings. *arXiv preprint arXiv:1603.04747*.

Douglas Reynolds. 2015. Gaussian mixture models. *Encyclopedia of biometrics*, pages 827–832.

Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth JF Jones. 2016. Representing documents and queries as sets of word embedded vectors for information retrieval. *arXiv preprint arXiv:1606.07869*.

Pranjal Singh and Amitabha Mukerjee. 2015. Words are not equal: Graded weighting model for building composite document vectors. In *Proceedings of the twelfth International Conference on Natural Language Processing (ICON-2015)*.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, Christopher Potts, et al. 2013. 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.

Vivek Kumar Rangarajan Sridhar. 2015. Unsupervised topic modeling for short texts using distributed representations of words. In *VS@ HLT-NAACL*, pages 192–200.

Martin J Wainwright. 2014. Structured regularizers for high-dimensional problems: Statistical and computational issues. *Annual Review of Statistics and Its Application*, 1:233–253.

Chengxiang Zhai and John Lafferty. 2001. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410. ACM.

Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214.