Concurrent Inference of Topic Models and Distributed Vector Representations

Debakar Shamanta[1], Sheikh Motahar Naim[1], Parang Saraf[2], Naren Ramakrishnan[2], and M. Shahriar Hossain[1]

[1]Dept of CS, University of Texas at El Paso, El Paso, TX 79968
[2]Dept of CS, Virginia Tech, Arlington, VA 22203
dshamanta@miners.utep.edu,snaim@miners.utep.edu,
parang@cs.vt.edu,naren@cs.vt.edu,mhossain@utep.edu

Abstract. Topic modeling techniques have been widely used to uncover dominant themes hidden inside an unstructured document collection. Though these techniques first originated in the probabilistic analysis of word distributions, many deep learning approaches have been adopted recently. In this paper, we propose a novel neural network based architecture that produces distributed representation of topics to capture topical themes in a dataset. Unlike many state-of-the-art techniques for generating distributed representation of words and documents that directly use neighboring words for training, we leverage the outcome of a sophisticated deep neural network to estimate the topic labels of each document. The networks, for topic modeling and generation of distributed representations, are trained concurrently in a cascaded style with better runtime without sacrificing the quality of the topics. Empirical studies reported in the paper show that the distributed representations of topics represent intuitive themes using smaller dimensions than conventional topic modeling approaches.

Keywords: Topic Modeling, Distributed Representation

1 Introduction

The representation of textual datasets in vector space has been a long-standing central issue in data mining with a veritable cottage industry devoted to representing domain-specific information. Most representations consider features as localized chunks as a result of which the interpretation of the features might lack generalizability. Researchers have recently become interested in distributed representations [12, 8, 14, 19] because distributed representations generalize features based on the facts captured from the entire dataset rather than one single object or a small group of objects. Moreover, modern large and unstructured datasets involve too many heterogeneous entries for which local subspaces cannot capture relationships between the features. For example, publication datasets nowadays come with a substantial number of features like author information, scientific area, and keywords along with the actual text for each document. News article datasets have author information, time stamp data, category, and sometimes

tweets and comments posted against the articles. Movie clips are accompanied by synopsis, production information, rating, and text reviews. The focus of this paper is on the design of a flexible mechanism that can generate multiple types of features in the same space. We show that the proposed method is not only able to generate feature vectors for labeled information available with the datasets but also for discovered information that are not readily available with the dataset as labels, for example, topics. Current state-of-the-art of distributed representations for unstructured text datasets can model two different types of elements in the same hyperspace, as described by Le and Mikolov [16]. Le and Mikolov's framework generates distributed vectors of documents (or paragraphs) and words in the same space using a deep neural network. Further generalization, that we have described in this paper, can provide distributed representations for heterogeneous elements of a dataset in the same hyperspace. However, the problem of creating distributed representations becomes more challenging when the label information is not contained within the dataset. The focus of this paper is on the generation of topical structures and their representations in the same space as documents and words. The capability of representing topics, documents, words, and other labeled information in the same space opens up the opportunity to compute syntactic and semantic relationships between not only words but also between topics and documents by directly by using simple vector algebra.

Estimating the topic labels for documents is another challenge while using distributed representations. Earlier topic modeling techniques [9, 13] used to define a document as a mixture of topics and estimate the probability $p(t|d)$ of a topic (t) of a document (d) through probabilistic reasoning. More recently, topic models are seen from a neural network point of view [26, 15, 6] where these probabilities are generated from the hidden nodes of a network. Such neural networks require compact numeric representations of words and documents for effective training, which are not easy to estimate with traditional vector space based document modeling techniques that represent the documents using a very high dimensional space. There have been attempts to use the compact distributed representations of words and documents learned from a general purpose large dataset [6] but the precomputed vectors may not be always appropriate for many new domain specific datasets. Furthermore, the vocabulary shifts in a new direction over time resulting in changes in the distributed representations.

Specific contributions of this paper are as follows.

1. We formulate the problem of computing distributed representation of topics in the same space as documents and words using a novel fusion of a neural network based topic modeling and a distributed representation generation technique.
2. The tasks of computing topics for documents and generating distributed representations are simultaneous in the proposed method unlike closely related state-of-the-art techniques where precomputed distributed vectors of words are leveraged to compute topics. Additionally, none of the state-of-the-art methods generates distributed representation of topics to the best of our knowledge.

3. Our proposed method generates the distributed vectors using a smaller number of dimensions than the actual text feature space. Even if the space is of lower number of dimensions, the vectors capture syntactic and semantic relationships between language components.
4. We demonstrate that the generated topic vectors explain domain specific properties of datasets, help identify topical similarities, and exhibit topic-specific relationships with document vectors.

2 Related Work

Distributed representations have been used in diverse fields of scientific research with notable success due to their superiority in capturing generalized view of information over local representations. Rumelhart et al. [22] designed a neural network based approach for distributed representation of words which has been followed by many efforts in language modeling. One such model is the neural probabilistic model [2] proposed by Bengio et al. This framework uses a sliding window based context of a word to generate compact representations. Mikolov et al. [17] brings in continuous bag-of-words (CBOW) and skip-gram models to compute continuous vector representations of words efficiently from very large data sets. The skip-gram model was significantly improved in [18], which includes phrase vectors along with words. Le and Mikolov [16] extended the CBOW model to learn distributed representation of higher level texts like paragraphs and documents. Our proposed model further enriches the literature by including the capability to generate (1) vectors for arbitrary labels in the dataset and (2) vectors for topics for which a text dataset does not contain any labeled information.

Finding hidden themes in a document collection has been of great interest to data mining and information retrieval researchers for more than two decades. An earlier work in the literature is latent semantic indexing (LSI) [9] that maps document and terms in a special "latent semantic" space by applying dimensionality reduction on traditional bag-of-words vector space representations of documents. A probabilistic version of LSI, pLSI [13], introduces a mixture model where each document is represented by a mixing proportion of hidden "topics". Latent Dirichlet Allocation (LDA) [5], a somewhat generalized but more sophisticated version of pLSI, is one of the most notable ones in the literature. It provides a generative probabilistic approach for document modeling assuming a random process by which the documents are created. LDA spawned a deluge of work exploring different aspects of topic modeling. For example, the Dynamic Topic Model (DTM) [4] captures the evolution of topics in a time-labeled corpus. Online LDA (OLDA) [1] handles streams of documents with dynamic vocabulary, Wallach [25] and Griffiths et al. [11] exploit the sentence structures of documents and Correlated Topic Model (CTM) [3] captures the correlation between topics.

More recently, neural network based models have received great attention from the data mining community. Wan [26] et al. introduce a hybrid model in computer vision settings; DocNADE [15] provides an autoregressive neural network for topic modeling; Cao et al. [6] propose a neural topic model (NTM)

with supervised extension. The latter work has close resemblance to a part of our proposed model that focuses on generating topics for each document.

3 Problem Formulation

Let $D = \{d_1, d_2, \ldots, d_N\}$ be a text dataset containing N documents taking terms from the set of M words $W = \{w_1, w_2, \ldots, w_M\}$. Each document can contain an arbitrary number of words in any sequence. The objective is to generate a universal distributed representation for the labeled items (e.g., words and documents) and latent topics of each document of dataset D. Let $T = \{t_1, t_2, \ldots, t_K\}$ be the set of topics. Consider that the expected number of dimensions in the distributed representation of words, documents, and topics is L. L should be much smaller than the number of words M. Word vectors $\mathcal{W} \in \mathbb{R}^{M \times L}$, document vectors, $\mathcal{D} \in \mathbb{R}^{N \times L}$ and topic vectors, $\mathcal{T} \in \mathbb{R}^{K \times L}$ generated in the same L-dimensional space should maintain two specific properties: (1) distributed representation of each type should be capable of capturing the semantic, syntactic, and topical aspect of conventional language models, and (2) all types of vectors (topics, documents, and words) organized in the L-dimensional hyperspace must be comparable to each other.

The first property aligns the framework with the objectives of any language model where features are generated for most common data mining tasks likes clustering and classification. The second property, however, is unique and specific to relating vectors of different types of entities like topics, documents, and vectors. In word2vec [17], the authors show that distributed representations of word can retrieve linguistic similarities between pairs of words. For example, $\mathcal{W}_{\text{King}} - \mathcal{W}_{\text{Man}}$ is close to $\mathcal{W}_{\text{Queen}} - \mathcal{W}_{\text{Woman}}$. The ability to model topics in the same hyperspace extends this property by capturing similarity between relationships among topics and documents. For example, if two documents d_i and d_j are drawn from the same topic t_p then $\mathcal{T}_p - \mathcal{D}_i$ should be closer to $\mathcal{T}_p - \mathcal{D}_j$. Similarly, if two documents d_i and d_j are drawn from two different topics t_p and t_q, then $\mathcal{T}_p - \mathcal{D}_i$ should tend to be different than $\mathcal{T}_p - \mathcal{D}_j$.

4 Methodology

The main objective of the proposed framework is to generate a compact distributed representation for topics, documents, and words of a document collection in the same hyperspace in such a way that all these heterogeneous objects are comparable to each other and capture the semantic, syntactic and thematic properties. The proposed framework has three major components. First, we adopt a generic neural network that can generate distributed vectors for documents, words, and any given labels. Second, we propose a deep neural network based topic modeling that can take distributed representations of words and documents, and estimate topic distribution for each document. Finally, we convolute both these networks so that they can share information and train simultaneously. Fig. 1 shows the proposed framework. The following subsections describe the model in a sequence.

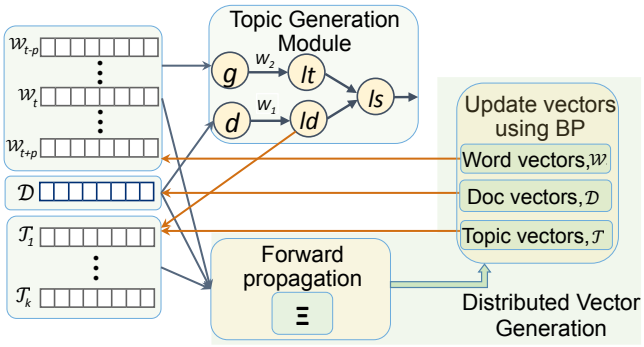


Fig. 1. The proposed framework.

4.1 Distributed Representation of Heterogeneous Entities

Inferring a distributed representation $\mathcal{W}$ for the words of a document collection D having vocabulary W is based on predicting a word given other words in the same context. The objective of such a word representation model is to maximize the average log probability

$$\frac{1}{M}\sum_{m=p}^{M-p}\log p(w_m|w_{m-p},\ldots,w_{m+p}) \tag{1}$$

The individual probabilities in Equation 1 are estimated by training a multi-class deep neural network, such as softmax. They can be computed as:

$$p(w_m|w_{m-p},\ldots,w_{m+p}) = \frac{e^{y_m}}{\sum_i e^{y_i}} \tag{2}$$

where y_i is the unnormalized log-probability for every output word w_i.

$$y_i = b + Uh(w_m|w_{m-p},\ldots,w_{m+p};\mathcal{W}) \tag{3}$$

Here, U and b are the softmax parameters. h is constructed by a concatenation or average of relevant word vectors. We use hierarchical softmax [17] instead of softmax for faster training, and calculate the gradient using stochastic gradient descent. After the training converges, words with similar meaning are mapped to

Algorithm 1: *LearnDistRep* – algorithm for learning topic vectors

input : Document id, d
Set of topics in d, T_d
Word to predict, w
Context of w, C_w
parameter: Distributed representations $\mathcal{D}$, $\mathcal{W}$ and $\mathcal{T}$

1 Calculate y using Equation 4 ;
2 Calculate gradient gr using stochastic gradient descent ;
3 Update document vector $\mathcal{D}_d$, topic vectors $\mathcal{T}_{T_d}$ and word vectors $\mathcal{W}_{C_w}$ using gr;

Algorithm 2: *LearnTopic* – algorithm for learning topic distribution.

input : Document id, d
N-gram or context id, g
parameter: Distributed representations $\mathcal{D}$, $\mathcal{W}$ and $\mathcal{T}$
Weight matrices W_1 and W_2
output : Updated weight matrices

1 Calculate $ls(g, d)$ using equations for lt and ld ;
2 Determine error in output node with respect to the ideal value:
$\delta^{(3)} = ls(g, d) - 1$;
3 Compute the error in n-gram-topic hidden node:
$\delta_1^{(2)} = (\delta^{(3)} \times ld(d)) \cdot (lt(g) \cdot (1 - lt(g)))$;
4 Update W_2: $W_2 = W_2 + \alpha[\delta_1^{(2)} \times \mathcal{W}_g + \lambda \times W_2]$;
5 Compute error in the document-topic hidden node:
$new\_ld(d) = ld(d) + \alpha[\delta^{(3)} \times lt(g) + \lambda \times ld(d)]$;
6 $\delta_2^{(2)} = new\_ld(d) - ld(d)$;
7 Update W_1: $W_1 = W_1 + \alpha[\delta_2^{(2)} \times \mathcal{D}_d + \lambda \times W_1]$;

a similar position in the vector space. To obtain a document vector, a document is thought of as another word. The only change in the model is in Equation 3, where h is constructed using $\mathcal{W}$ and $\mathcal{D}$.

Inclusion of further labels, for example, authors, topic, and tags can be done the same way document vectors are added. Our focus in this paper is to incorporate topics instead of additional labels. Incorporation of topic vectors is challenging because the topics are not given and rather should be generated using the documents and words. For the time being, let us assume that topic is just a given label that comes with the data. In contrast to the word vector matrix $\mathcal{W}$ that is shared across all the documents, a topic vector can be shared only across the documents which contain that particular topic. Considering topic vectors along with the vectors for words and documents, Equation 3 is modified to:

$$y = b + Uh(w_{t-k}, \ldots, w_{t+k}, d_q, t_{r_1}, t_{r_2}, \ldots, t_{r_s}; \mathcal{W}, \mathcal{D}, \mathcal{T}) \tag{4}$$

For the training purpose, we use sampling of variable-length contexts using a sliding window over each document. Such a sliding window is commonly referred to as n-gram. We use n-grams instead of single words (unigrams) since n-grams produce representative contexts around each word [18]. A procedure for training this generic network for topic, documents, and words is explained in Algorithm 1.

4.2 Estimating Topic Labels of Documents

As stated earlier, the generic model described in Section 4.1 requires topic as labels of each document. This section focuses on a topic modeling technique that can generate topic labels taking document vectors and word vectors into account. For effective and efficient generation of topic vectors, the topic modeling technique must synchronize with the iterations of the distributed vector generation part. Several topic modeling techniques have been proposed in the literature to find topic distribution of documents of such unlabeled datasets.

In a general topic model, each document is seen as a mixture of topics, and each topic is represented as a probability distribution over the vocabulary of the entire corpus. The conditional probability $p(w|d)$ of a word and a document is computed from word-topic distribution and topic-document distribution as $p(w|d) = \sum_{i=1}^{K} p(w|t_i)p(t_i|d)$, where K is the number of topics and t_i is a latent topic. This equation can be re-written as

$$p(w|d) = \phi(w) \times \theta^T(d) \tag{5}$$

where $\phi(w) = [p(w|t_1), p(w|t_2), \ldots, p(w|t_K)]$ is the conditional probabilities of w with all the topics and $\theta(d) = [p(t_1|d), p(t_2|d), \ldots, p(t_K|d)]$ is the topic distribution of d.

We can view topic models from a neural network perspective considering the formation of Equation 5. Let us consider a neural network with two input nodes for sliding window with n-gram g and document d, two hidden nodes lt (representing $\phi(g)$) and ld (representing $\theta(d)$), and one output node ls producing the conditional probability $p(g|d)$. The topic-document node $ld \in \mathbb{R}^{1\times K}$ computes the topic distribution of a document (similar to θ in topic models) using the weight matrix $W_1 \in \mathbb{R}^{L\times K}$. It is computed by the equation $ld(d) = softmax(\mathcal{D}_d \times W_1)$ which uses a softmax function to maintain the probabilistic constraint on topic distribution that all the topic probabilities of a document must sum up to 1.

The n-gram-topic node $lt \in \mathbb{R}^{1\times K}$ stands for the topic representation of the input n-grams, and calculated as $lt(g) = sigmoid(\mathcal{W}_g \times W_2)$ where $W_2 \in \mathbb{R}^{L\times K}$ denotes the weight matrix between the n-gram input node and the n-gram-topic node. This vector follows a probabilistic form similar to ϕ in topic models.

The output node $ls \in \mathbb{R}$ gives the matching score of an n-gram g and a document d by computing the dot product of $lt(g)$ and $ld(d)$. The outputted score $ls(g, d) = lt(g) \times ld(d)^T$ is a value between 0 and 1, similar to the conditional probability of $p(g|d)$.

The n-gram-document probability $p(g|d)$, which initially is expected to be very different from the ideal value, is estimated by performing a forward propagation in the network. Algorithm 2 describes the training procedure for the neural topic model part of our proposed model. For each n-gram-document pair (g, d) the expected output value is 1 due to the fact that g is taken from document d. The weights are updated using backpropagation to mitigate that error (Steps 3 to 7 in Algorithm 2).

4.3 Concurrent Training

The training process runs concurrently for both topic modeling and distributed vector generation. Fig. 1 shows the proposed combination of two networks. Notice the training is simultaneous unlike NTM [6] where already trained word vectors are used for topic modeling. All the weights (W_1 and W_2 matrices) and vectors ($\mathcal{W}$, $\mathcal{D}$ and $\mathcal{T}$ matrices) in both the networks are initialized with random values (Step 1 and 2 of Algorithm 3). As shown in the loop at Step 3 of Algorithm 3, the combined framework reads each document in sequence of n words

(context) using a continuous window. For a particular document, the topic modeling network gives its topic distribution as the output of the hidden node ld. We select k most probable topics from this distribution – with an assumption that a document is made up of k number of topics – and provide them as input to the distributed vector generation network. The call to the method `LearnTopics` in Step 7 of Algorithm 3 accomplishes this task. The corresponding word, document and topic vectors are updated using method `LearnDistRep` in Step 8 of Algorithm Algorithm 3. Method `LearnTopics` and `LearnDistRep` are explained in Algorithms 2 and 1, respectively.

Notice that the document and word vectors of context (n-gram) generated by Algorithm 1 are provided as input to the topic modeling network of Algorithm 2. Also the top k topics generated for each document using Algorithm 2 are provided to the distributed vector generation part (Algorithm 1). Algorithm 3 combines all these steps.

5 Complexity Analysis

Although both the neural networks in our proposed framework are concurrently trained, we analyze their complexities separately for simplicity. For every example during the training of the distributed vector generation network, there are P words (context length), k topics and one document as input resulting in $I = P+k+1$ input nodes. These inputs are projected into a L dimensional space. Although there are $V = N+M+K$ output nodes, this part of the network needs to update only $O(\log V)$ nodes using the gradient vector since the model uses hierarchical softmax. I input nodes get updated during backpropagation making the complexity for training a single example, $C_{dr} = I \times L + O(\log V) \times L$.

The topic modeling network takes the same document and input words. Calculating $\mathcal{W}_g$ from the words in n-gram g takes $O(P \times L)$ time. Calculating each

Algorithm 3: *ConcurrentTrain* – algorithm for simultaneous training of both networks

input : Document collection D
parameter: Distributed representations $\mathcal{D}$, $\mathcal{W}$ and $\mathcal{T}$
Weight matrices W_1 and W_2 of topic modeling network
output : $\mathcal{D}$, $\mathcal{W}$ and $\mathcal{T}$

1 Randomly initialize $\mathcal{D}$, $\mathcal{W}$ and $\mathcal{T}$;
2 Randomly initialize W_1 and W_2 ;
3 **for** *each document* $d \in D$ **do**
4 Topics in d, $T_d \leftarrow$ top k topics from $ld(d)$;
5 **for** *each word* w *of* d **do**
6 $C_w \leftarrow$ context of w ;
7 LearnTopics(d, C_w) ;
8 LearnDistRep(d, T_d, w, C_w) ;
9 **end**
10 **end**

of ld and lt takes $O(L \times K)$ operations and ls requires $O(K)$ operations. Back-propagation (step 3 to 7 of Algorithm 2) runs in $O(L \times K)$ time incurring a total cost of $C_{tm} = O(P \times L) + O(L \times K) + O(K) + O(L \times K)$, or $C_{tm} = O(L \times K)$ given $K > P$, for every example. Therefore, the cost of training the combined network for each example is $C = C_{tm} + C_{dr}$.

6 Evaluation

We use a number of metrics to evaluate the quality of our results. Some of these metrics are generally used to evaluate clustering results when ground truth labels are not available. Two such evaluations are the Dunn Index (DI) [10] and the Average Silhouette Coefficient (ASC) [21]. DI measures the separation between groups of vectors and larger values are better. ASC is a measure that takes both cohesion and separation of groups into account (higher values are better). In our experiments, we utilize ASC and DI together to evaluate the final topic assignments of the documents. Topics are analogous to clusters in those evaluations. ASC and DI give us an idea about how crisply the topics are distributed across the documents.

In the presence of ground truth labels, we evaluated the assigned topics using Normalized Mutual Information (NMI) [7], Adjusted Rand Index (ARI) [24], and the hypergeometric distribution-based enrichment. Both NMI and ARI estimates the agreement between two topic assignments, irrespective of permutations. Higher values are better for NMI and ARI. Hypergeometric enrichment [23] maps topics to available ground truth labels. This allows us to measure a significance based on hypergeometric distribution of the topic assignments over the already known labels. Higher number of enriched topics is better.

Our proposed model is able to generate topic and document vectors in the same hyperspace. In an ideal case, all angles between a topic vector and each document vector assigned to this topic should be similar and the standard deviation of those angles should be small. We use this concept to compute alignment between a topic vector and a given set of document vectors. Given a topic vector $\mathcal{T}_i$ of topic t_i, and a set of document vectors $\mathcal{D}^{t_j}$ that are assigned a topic t_j, we compute alignment using the following formula:

$$A(\mathcal{T}_i, \mathcal{D}^{t_j}) = \sqrt{\frac{1}{|\mathcal{D}^{t_j}|} \sum_{m=1}^{|\mathcal{D}^{t_j}|} \left(\frac{\mathcal{T}_i . \mathcal{D}_m^{t_j}}{\|\mathcal{T}_i\| \|\mathcal{D}_m^{t_j}\|} - \mu \right)^2} \tag{6}$$

where $\mathcal{D}_m^{t_j}$ refers to the document vector of mth document in topic t_j, and

$$\mu = \frac{1}{|\mathcal{D}^{t_j}|} \sum_{m=1}^{|\mathcal{D}^{t_j}|} \frac{\mathcal{T}_i . \mathcal{D}_m^{t_j}}{\|\mathcal{T}_i\| \|\mathcal{D}_m^{t_j}\|} \tag{7}$$

Notice that Equation 6 is the standard deviation between the cosine angles between the topic vectors and the document vectors. Lower values are expected when $t_i = t_j$ and higher values are expected when $t_i \neq t_j$.

7 Experiments

In this section, we seek to answer the following questions to justify the capabilities and correctness of the proposed model.

1. Can our framework establish relationships between distributed representations of topics and documents? (Section 7.1)
2. Are the generated topic vectors expressive enough to capture similarity between topics and to distinguish difference between them? (Section 7.2)
3. How do our topic modeling results compare with the results produced by other topic modeling algorithms? (Section 7.3)
4. Do the generated topics bring documents with similar domain-specific themes together? (Section 7.4)
5. How does the runtime of the proposed framework scale with the size of the distributed representations, increasing number of documents, and increasing number of topics? (Section 7.5)

We used seven different text datasets[1] with different number of documents and words. The datasets are listed in Table 1. Some of these datasets are widely used in the text processing literature (e.g., Reuters , WebKB, and 20Newsgroups datasets), while we have collected most of the other corpora from the public domain. The PubMed dataset is collected from publicly available citation databases for biomedical literature provided by the US National Library of Medicine. The PubMed dataset contains abstracts of *cancer*-related publications. The *Spanish news* dataset was collected as a part of the *EMBERS* [20] project. The articles covered news stories from 207 countries around the world.

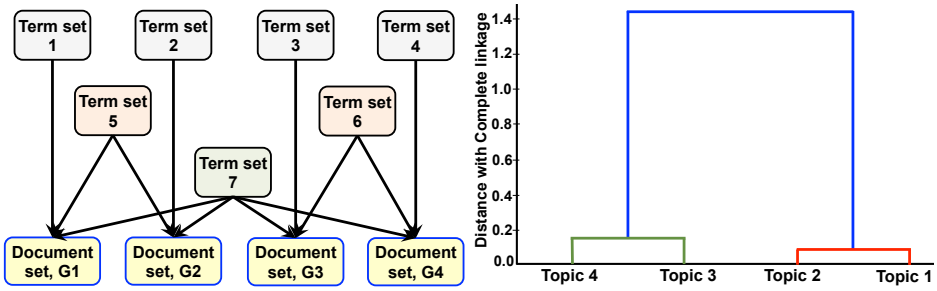
7.1 Analysis of Distributed Representations of Topics and Documents

The topic and document vectors generated by the proposed framework maintain consistent relationships that can be leveraged in many applications to study the topics of a stream of unseen documents. To be able to develop such applications, a relationship between a topic vector $\mathcal{T}_i$ and any of its document vectors $\mathcal{D}_p^{t_i}$ should be different than the relationship between another topic $\mathcal{T}_j$ and a document vector $\mathcal{D}_q^{t_j}$.

Table 1. Summary of the datasets.

Dataset	**#Docs**	**#Words**	**Additional information**
Synthetic	400	40,000	Four lower and two upper level groups.
20 Newsgroups	18,821	2,654,769	20 categories in seven groups.
Reuters R8	7,674	495,226	Eight category labels.
Reuters R52	9,100	624,456	52 groups.
WebKB	4,199	559,984	Four overlapping categories
PubMed	1.3 million	220 million	Publication abstracts related to cancer.
Spanish news	3.7 million	3 billion	News articles from 2013 and 2014.

[1]Data and software source codes are provided here: `http://dal.cs.utep.edu/projects/tvec/`.



(a) The synthetic dataset has four groups of documents.

(b) Dendrogram generated using topic vectors.

Fig. 3. Experiment with a synthetic dataset. (a) Sets of terms used to prepare the synthetic text corpus, (b) Dendrogram generated from the topic vectors.

In contrast, such topic-document relationships should be similar for two documents of the same topic. Each plot of Fig. 2 shows a heat map of alignment between a topic vector $\mathcal{T}_i$ of topic t_i and all document vectors $\mathcal{D}^{t_j}$ of topic t_j using Equation 6. Fig. 2 shows the heat map with four topics of the synthetic dataset. In this heat map, lower alignment values result in darker cells depicting stronger topic-document alignment for topic and document vectors of the same topic, whereas weaker alignments are exhibited when document vectors are chosen from a different topic. This indicates that our proposed framework captures topical structures as well as it models relationships between topics and documents in the same hyperspace.

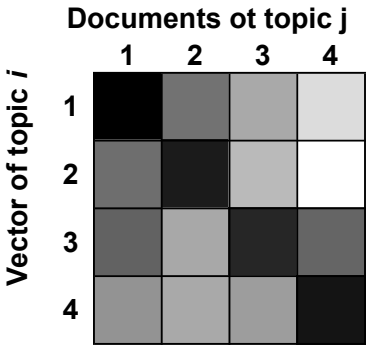


Fig. 2. Heat map of variance of cosine similarity between topic vector i and all documents of topic j.

7.2 Expressiveness of Topic Vectors

As described in Section 4.3, k-best topics generated by the topic modeling part of the proposed model are selected as input to the distributed representation generation part. We set $k = 1$ for all our experiments including the ones described in this subsection . To examine how expressive our distributed topic vectors are, we prepared a synthetic corpus containing documents with term from seven sets as illustrated by Fig. 3(a). Four groups of documents contains terms specific to each group. The same dataset can be divided into two groups of documents because each group contains terms from a specific group set of words. Additionally, all sets of documents share a common set of terms. We generated topic, document, and words vectors using our proposed framework. A dendrogram for the generated four topic vectors is shown in Fig. 3(b). As expected, the dendrogram

exhibits the topical structure where two topic vectors separately and then those two groups merge at the top of the hierarchy. The dendrogram of topic vectors reflects the grouping mechanism we used to create the dataset.

In a second experiment in this space, we used a dataset that already has category labels (20 Newsgroups) to verify how intuitive the topic vectors are in bringing similar categories together. To be able to generate distributed vectors for existing categories along with document and word vectors, we directly provided the known labels to the distributed representation generation part of the model as an inputs as opposed to providing topics generated by the topic modeling network. The official site for the 20 News Groups dataset reports that some of the newsgroups are very closely related to each other (e.g. *comp.sys.ibm.pc.hardware* and *comp.sys.mac.hardware*), while others may be highly unrelated (e.g *misc.forsale* and *soc.religion.christian*). Our target is to verify if the generated category vectors can provide insights about how the topics should be merged. Fig. 4 shows the dendrogram prepared for the 20 category vectors of 20 Newsgroups dataset. There are some differences between the official grouping and the grouping we have discovered using the category vectors, for example, sci.electronics is grouped with *comp.sys.mac.hardware* and *comp.sys.ibm.pc.hardware*. The label *sci.electronics* is far away from *sci.space* even though they have a common prefix "sci". Our observation is that *sci.electronics* has many documents containing hardware related discussions. As a result, *sci.electronics* has greater similarity with hardware than *sci.space*. Similar evidences are found for the rec.* groups. For example, rec.sport.* groups are different from rec.motorcycles and rec.autos but the latter two groups are closely related, as evident in the dendrogram.

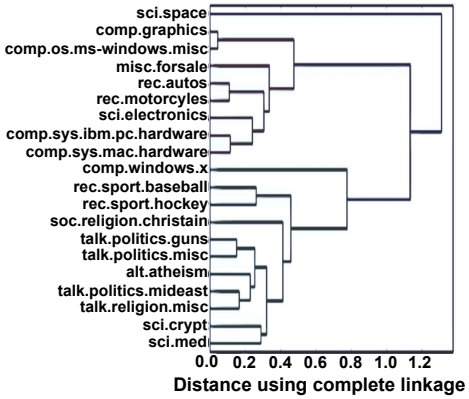
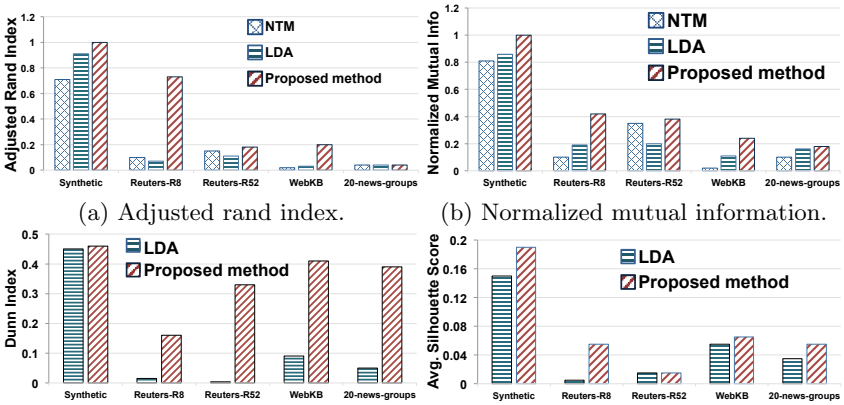


Fig. 4. Dendrogram prepared with the 20 category vectors of 20 Newsgroups dataset.

7.3 Comparison of Quality of Generated Topics

Fig. 5 shows a comparison of results generated by our framework and two other topic modeling methods, LDA and NTM, when applied on four classification datasets — synthetic, Reuters-R8, Reuters-R52, WebKB, and 20 Newsgroups. Fig. 5 (a) and (b) use adjusted Rand index (ARI) and normalized mutual information (NMI) to compare the topic assignments of the documents with the expected classes. ARI and NMI are larger for the proposed methods for all the datasets. This implies that our framework realizes the expected themes of the collections better than LDA and NTM. Not only the expected categories better match with the topic assignments, but also the generated topics are local in the corresponding space of our framework. Higher Dunn index and higher average



(a) Adjusted rand index. (b) Normalized mutual information.

(c) Dunn index. (d) Average Silhouette score.

Fig. 5. Evaluation using benchmark labels (a & b) and locality of the topics (c & d).

silhouette coefficient for all the datasets, as depicted in Fig. 5(c) and (d), imply that our model provides high quality local topics. Notice that Fig. 5(c) and (d) do not have NTM. This is because Dunn index and average silhouette coefficient require document vectors, but NTM [6] does not directly use any document vector; rather, it uses precomputed word vectors only.

We also used a hypergeometric distribution based procedure to map each topic to a class label. Fig. 6 shows that the topic assignments suing our framework have higher number of enriched topics than any other method. This indicates that the topics generated by our methods has higher thematic resemblance with the benchmark labels.

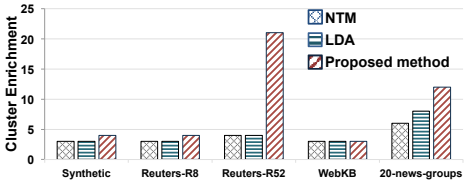


Fig. 6. Comparison of numbers of topics enriched by hypergeometric distribution.

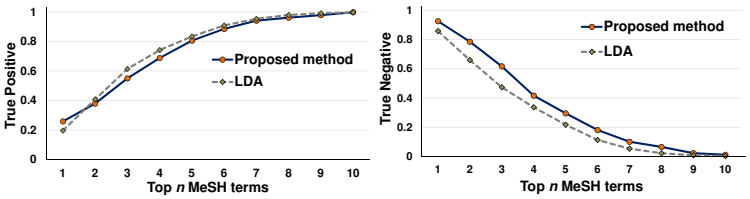
All these datasets described so far, in this subsection are labeled and are widely used a ground truths in many data mining and machine learning evaluations. In addition to these datasets, we used our EMBERS data containing around 3.7 million news articles to compare locality of the topics with other methods. Table 2 shows that our method produces topics with greater Dunn index and average silhouette score than other methods. This indicates that our method performs even better when the datasets are very large.

Table 2. Evaluation using the EMBERS news article dataset.

Method	Evaluation metric	
	Dunn index	Silhouette score
NTM	0.04	0.01
LDA	0.01	-0.015
Proposed method	**0.1**	**0.05**

7.4 Evaluation using Domain Specific Information

In this experiment, we used the PubMed dataset to compute overlap of domain specific information for documents in the same topic (i.e., true positive) and



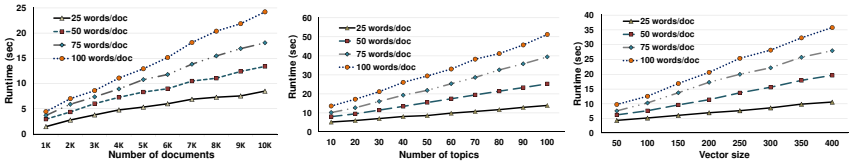
(a) Ratio of true positives. (b) Ratio of true negatives.

Fig. 7. Comparison of our method and LDA using MeSH terms associated with the PubMed abstracts.

lack of such overlap for a pair of documents from two different topics (i.e., true negative). In the PubMed dataset, each abstract is provided with some major Medical Subject Header (MeSH) terms which come from a predefined ontology. We used these MeSH terms as domain specific information to evaluate the topics. It is expected that the sets of MeSH terms of two documents of the same topic will have some common entries, where as the sets of MeSH terms of two documents from two different topics will have lesser or no overlapping records. For each abstract, we ordered the MeSH terms based on Jaccard similarity between a MeSH terms and the abstract. Notice that if we pick up n best MeSH terms for two documents from the same topic the chance that these two sets of n best MeSH terms have common entries increases with larger n. This trend is observed in Fig. 7(a) for both our framework and LDA. The true positive ratio quickly becomes around 80% with only five best MeSH terms for each pair of documents. Now, the top n MeSH terms of two documents from two different topics should have higher absence of overlapping terms with smaller n since the topical similarity of these two documents is minimal. As n increases the true negative ratio will decrease due to inclusion of more general entries in the lists of n best MeSH terms. Fig. 7(b) shows the expected trend for both LDA and our framework. We selected random 5,000 pairs of documents from same topics and another 5,000 pairs from different topics for the two plots, Fig. 7(a) and (b) respectively. Fig. 7(a) and (b) demonstrate that our method follows an expected trend of sharing domain specific information. Although the true positive values are slightly lower than LDA in our method in some cases, the true negative values are always greater than LDA. This indicates that our model generates topics containing similar biological themes while documents of different topics, as expected, have lesser similarity in domain specific information.

7.5 Runtime Characteristics

Fig. 8 depicts the runtime behavior of our proposed framework with varying number of documents, topics, and vector size. The runtime increases almost linearly with each of these variables. This indicates our proposed framework is scalable with large amount of data. The experiments in this space were done using synthetic data with different number of words in each document as depicted by multiple lines in each of the plots of Fig. 8.



(a) (b) (c)

Fig. 8. Execution time with varying (a) number of documents, (b) number of topics, and (c) vector size.

8 Conclusion

We have presented a framework to generate distributed vectors for elements in a corpus as well as the underlying latent topics. All types of vectors — topics, documents, and words — share the same space allowing the framework to compute relationships between all types of elements. Our results show that the framework can efficiently discover latent topics and generate distributed vectors simultaneously. The proposed framework is expressive and able to capture domain specific information in a lower-dimensional space. In future, we will investigate how one can study the information genealogy of a document collection with temporal signatures using the proposed framework. We are inspired by the fact that we can train the distributed vector generation network in a sequence as found in the temporal signatures associated with the documents and observe the shift of the word probabilities at the output of the network. We can also observe how the probability distributions of the topic generation network change over the given time sequence. This would help identify how one topic influence and transcend another and how the topical vocabulary shifts over time.

Acknowledgments. This work is supported in part by M. S. Hossain's startup grant at UTEP, University Research Institute (URI, Office of Research and and Sponsored Projects, UTEP), and the Intelligence Advanced Research Projects Activity (IARPA) via DoI/NBC contract number D12PC000337. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon.

References

1. L. AlSumait, D. Barbará, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM'08*, pages 3–12, 2008.
2. Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Machine Learning Research*, 3:1137–1155, 2003.
3. D. Blei and J. Lafferty. Correlated topic models. *Advances in Neural Information Processing Systems*, 18:147, 2006.

4. D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML'06*, pages 113–120, 2006.
5. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Machine Learning Research*, 3:993–1022, 2003.
6. Z. Cao, S. Li, Y. Liu, W. Li, and H. Ji. A novel neural topic model and its supervised extension. In *AAAI'15*, 2015.
7. G. J. Chaitin. *Algorithmic information theory*. Wiley Online Library, 1982.
8. D. J. Chalmers. Syntactic transformations on distributed representations. In *Connectionist Natural Language Processing*, pages 46–55. Springer, 1992.
9. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *American Society for Information Science*, 41(6):391–407, 1990.
10. J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.
11. T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum. Integrating topics and syntax. In *NIPS'04*, pages 537–544, 2004.
12. G. E. Hinton. Learning distributed representations of concepts. In *CogSci'86*, volume 1, page 12, 1986.
13. T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR'99*, pages 50–57. ACM, 1999.
14. J. E. Hummel and K. J. Holyoak. Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, 104(3):427, 1997.
15. H. Larochelle and S. Lauly. A neural autoregressive topic model. In *NIPS'12*, pages 2708–2716, 2012.
16. Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML'14*, pages 1188–1196, 2014.
17. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
18. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS'13*, pages 3111–3119, 2013.
19. J. B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1):77–105, 1990.
20. N. Ramakrishnan et al. 'Beating the news' with EMBERS: Forecasting civil unrest using open source indicators. In *SIGKDD'14*, pages 1799–1808, 2014.
21. P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20(0):53 – 65, 1987.
22. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive Modeling*, 5, 1988.
23. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Enrichment or depletion of a go category within a class of genes: which test? *Bioinformatics*, 23(4):401–407, 2007.
24. D. Steinley. Properties of the hubert-arable adjusted rand index. *Psychological Methods*, 9(3):386, 2004.
25. H. M. Wallach. Topic modeling: beyond bag-of-words. In *ICML'06*, pages 977–984, 2006.
26. L. Wan, L. Zhu, and R. Fergus. A hybrid neural network-latent topic model. In *AISTATS'12*, pages 1287–1294, 2012.