Unsupervised Topic Modeling for Short Texts Using Distributed Representations of Words

Vivek Kumar Rangarajan Sridhar[*]
AT&T Labs
1 AT&T Way, Bedminster, NJ 07920

Abstract

We present an unsupervised topic model for short texts that performs soft clustering over distributed representations of words. We model the low-dimensional semantic vector space represented by the dense distributed representations of words using Gaussian mixture models (GMMs) whose components capture the notion of *latent topics*. While conventional topic modeling schemes such as probabilistic latent semantic analysis (pLSA) and latent Dirichlet allocation (LDA) need aggregation of short messages to avoid data sparsity in short documents, our framework works on large amounts of raw short texts (billions of words). In contrast with other topic modeling frameworks that use word co-occurrence statistics, our framework uses a vector space model that overcomes the issue of sparse word co-occurrence patterns. We demonstrate that our framework outperforms LDA on short texts through both subjective and objective evaluation. We also show the utility of our framework in learning topics and classifying short texts on Twitter data for English, Spanish, French, Portuguese and Russian.

1 Introduction

A popular way to infer semantics in an unsupervised manner is to model a document as a mixture of latent *topics*. Several schemes such as latent semantic analysis (Deerwester et al., 1990), probabilistic latent semantic analysis (pLSA) (Hofmann, 1999) and latent Dirichlet allocation (LDA) (Blei et al., 2003) have been used to good success in inferring the high level meaning of documents through a set of representative words (*topics*). However, the notion of a document has changed immensely over the last decade. Users have embraced new communication and information medium such as short messaging service (SMS), chats, Twitter, Facebook posts, Instagram and user comments on news pages/blogs in place of emails and conventional news websites. Document sizes have been reduced from a few hundred words to few hundred characters[1] while the amount of data has increased exponentially.

Conventional topic models such as pLSA and LDA learn latent topics in a corpus by exploiting document-level word co-ocurrences. Hence, these models typically suffer from data sparsity (estimating reliable word co-occurrence statistics) when applied to short documents. A popular strategy to overcome this bottleneck is to aggregate short texts into longer documents based on user information, title category, etc. (Weng et al., 2010; Hong and Davison, 2010). However, these schemes are heuristic and highly dependent on the data. Furthermore, such metadata may not be available for short texts such as news titles, advertisements or image captions.

In this work, we present an unsupervised topic model that uses soft clustering over distributed representations of words. The distributed word representations are obtained by using a log-linear model and we model the low-dimensional semantic vector space represented by the dense word vectors using Gaussian mixture models (GMMs). The K components of the Gaussian mixture model can be considered as the latent topics that are captured by the model. Unlike long documents, these short messages do not have long distance syntactic or semantic dependencies and we find that the distributed representations learned over limited context windows is sufficient in capturing the distributional similarity of words within a message. In comparison with previous approaches to topic modeling, we completely ignore the distribution over documents and consider the entire corpus, thereby eliminating the need for aggregation over short messages. The framework presented here is

[*] The author is currently with Apple, Inc., and can be contacted at vrangarajansridh@apple.com.

[1] Twitter currently imposes a limit of 140 characters for each message

Proceedings of NAACL-HLT 2015, pages 192–200,
Denver, Colorado, May 31 – June 5, 2015.

unsupervised, language agnostic and scalable.

2 Related Work

In the *tf-idf* scheme (Salton and McGill, 1986), a collection of documents is represented as a $V \times D$ matrix where the rows denote the terms (words) and the columns contain *tf-idf* values for the chosen terms (words). However, the approach reveals little about the underlying semantic structure of the documents. Latent semantic analysis (LSA) (Deerwester et al., 1990) addressed the limitations of the *tf-idf* scheme by performing singular value decomposition (SVD) on the $V \times D$ matrix. The LSA features are linear combinations of the *tf-idf* features in a lower dimensional subspace and can capture linguistic notions such as polysemy and synonymy.

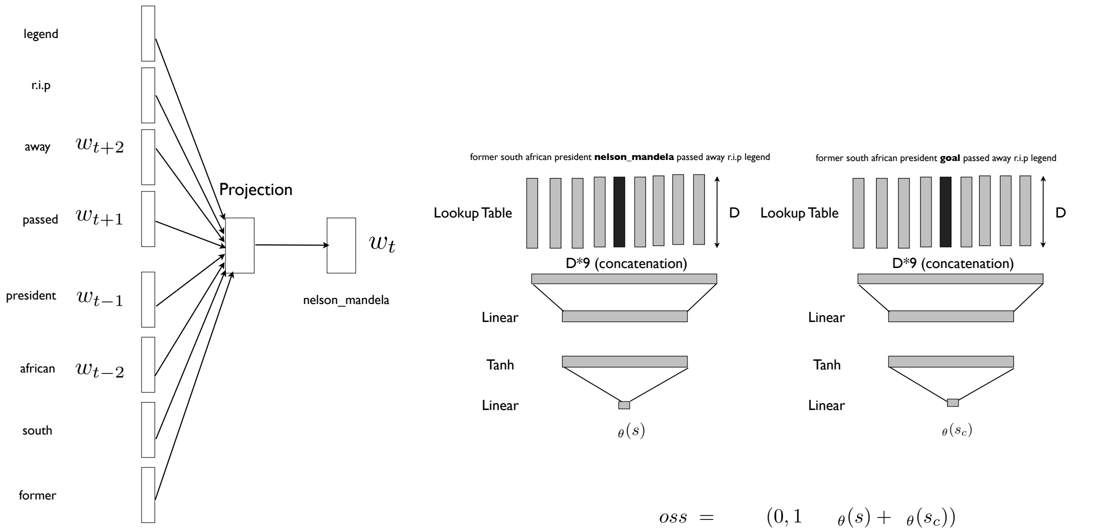
Probabilistic latent semantic analysis (pLSA) (Hofmann, 1999) improved on LSA by modeling each word as a sample from a mixture model, the components of which are multinomial random variables (*topics*). One of the main drawbacks of pLSA is that the topic distributions are learned for particular documents seen in training and consequently, the model is difficult to use on unseen documents. Moreover, the model size grows linearly with the size of the corpus and hence is prone to overfitting. Latent Dirichlet allocation (LDA) (Blei et al., 2003) is a generative model that overcomes some of the limitations of pLSI by using a Dirichlet prior on the topic distribution. The model can hence be used on unseen data and the parameters of the model do not grow with the size of training corpus.

LSA, pLSA and LDA have all been conventionally used on collection of documents that are typically at least a few hundred words. With the recent popularity of communication media such as SMS, Twitter, Facebook, Instagram, etc., many efforts (Weng et al., 2010; Hong and Davison, 2010) have addressed the application of topic models to short texts. (Weng et al., 2010) addressed the problem of identifying influential users on Twitter using a modified PageRank algorithm. They used LDA for inducing topics on user aggregated messages, i.e., a document is a collection of tweets from a single user. The work in (Hong and Davison, 2010) also experimented with different aggregation strategies to apply LDA for inducing topics. In (Ramage et al., 2010), a supervised version of LDA was used to model individual messages. However, such a scheme is not completely unsupervised and hence not desirable for large amounts of data than can span extremely large number of topics (billions of tweets, Facebook posts, image captions, etc.).

In contrast with previous approaches that have either modified LDA or the input to LDA (by aggregating short messages), our approach works on the entire corpus (e.g., billions of tweets or SMS messages) without any aggregation strategy and is completely unsupervised. We learn distributed representations of words over sufficiently long context windows and subsequently use Gaussian mixture models to parameterize the vector space represented by the distributed representations. Our framework is inspired by use of bottleneck features obtained from neural networks in hidden Markov model (HMM) based speech recognition (Grezl and Fousek, 2008). We can potentially use all the optimization and parallelization techniques used in HMM-based speech recognition to scale to large text data sets. The closest approach to that proposed in this work is the biterm topic model (BTM) (Yan et al., 2013) that learns topics over an entire corpus of short texts by directly modeling unordered word-pair co-occurrences (biterms) over the corpus. In our approach, the distributed representations capture longer word contexts, i.e., each word is projected into a vector that represents similarity between words within the contextual window. Hence, our approach can potentially capture context beyond unordered word-pair co-occurrences. Furthermore, since we use dense vectors to represent terms, our approach does not suffer from data sparsity issues typically encountered in co-occurrence statistics based topic models.

3 Distributed Word Representations

Distributed representation of words (also called word embeddings or continuous space representation of words) has become a popular way for capturing distributional similarity (lexical, semantic or even syntactic) between words. The basic idea is to represent each word in vocabulary V with a real-valued vector of some fixed dimension D, i.e., $w_i \in \mathbb{R}^D \;\; \forall \;\; i = 1, \cdots, V$. The idea of representing words in vector space was originally proposed in (Rumelhart et al., 1986; Elman, 1991). However, improved training techniques and tools


legend
r.i.p
away
passed
president
african
south
former
w_{t+2}
w_{t+1}
w_{t-1}
w_{t-2}
Projection
w_t
nelson_mandela
former south african president **nelson_mandela** passed away r.i.p legend
former south african president **goal** passed away r.i.p legend
Lookup Table
D
D*9 (concatenation)
Linear
Tanh
Linear
$_\theta(s)$
$_\theta(s_c)$
$oss = \quad (0,1 \quad _\theta(s) + \ _\theta(s_c))$

(a) Continuous Bag-of-Words Architecture

(b) Deep Neural Network Architecture

Figure 1: Illustration of obtaining distributed representations of words using two different approaches. An entire tweet can be captured with sufficient context. For Figure 1(b), θ denotes the parameters of the neural network while s and s_c denote the correct and corrupt windows, respectively.

in the recent past have made it possible to obtain such representations for large vocabularies.

Distributed representations can be induced for a given vocabulary V in several ways. While they are typically induced in the context of a deep neural network framework for a given task (Bengio et al., 2003; Collobert and Weston, 2008; Bengio et al., 2009; Turian et al., 2010; Mikolov et al., 2010), recent work in (Mikolov et al., 2013) has also shown that they can also be induced by using simple log-linear models.

Figure 1 shows two different architectures for inducing distributed representations. On the left side, the architecture for the "continuous bag-of-words" model (Mikolov et al., 2013) is shown while the deep learning architecture for inducing distributed representations in language models (Collobert and Weston, 2008) is shown on the right. Both these frameworks essentially perform a similar function in that the word representations are created based on contextual similarity. Since, the average sentence length for text media such as Twitter messages, SMS messages, Facebook posts, etc., is between 12-16 words, inducing distributed representations over similar length windows can capture the semantic similarity between the words in a message. In the next section, we demonstrate how this property can be exploited to perform topic modeling for short messages.

4 Gaussian Mixture Topic Model

We use a log-linear model for inducing the distributed representations using the continuous-bag-of-words architecture proposed in (Mikolov et al., 2013). The continuous-bag-of-words model is similar to the neural network language model (Bengio et al., 2003) with the non-linear layer replaced by a sum pooling layer, i.e., the model uses a bag of surrounding words to predict the center word. Since the implementation of this architecture was readily available through the word2vec tool[2], we used it for inducing the representations. We used hierarchical sampling for reducing the vocabulary during training and used a minimum count of 5 occurrences for each word. One can also use a deep neural network approach (Collobert and Weston, 2008) for inducing the representations. However, the training of these networks is extremely time consuming and we decided to use the simple log-linear model in this work. The framework presented here can work with distributed representations obtained with any methodology (latent semantic indexing, log-linear models, feedforward neural networks, convolutional neural networks, recurrent neural networks, etc.).

We use the continuous-bag-of-words

[2] https://code.google.com/p/word2vec/

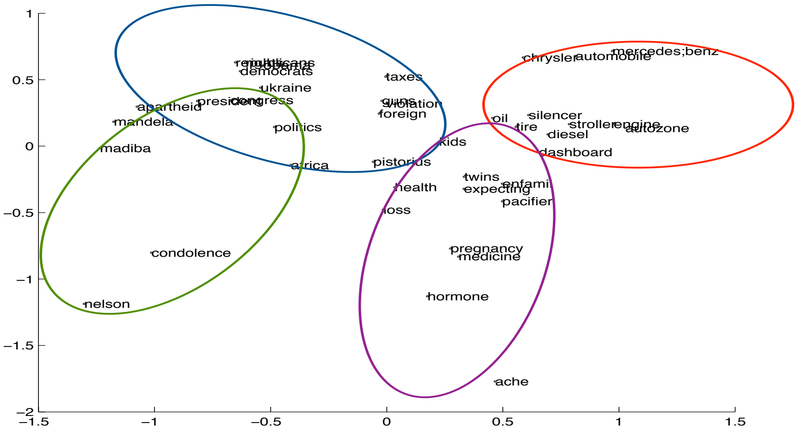


Figure 2: Illustration of fitting a Gaussian mixture model to distributed representations. The dimension of the distributed representations was reduced from 100 to 2 using principal component analysis and 4 GMM components were used.

model (Mikolov et al., 2013) to process all windows of length $wlen$ in a corpus and output a D-dimensional vector $\mathbf{d}_i$ for each word w_i in the vocabulary V. $wlen$ in our work is an odd number, i.e., $wlen = 11$ implies a left and right context of 5 words. Once we obtain the set of word embeddings $w_i \mapsto \mathbf{d}_i, \forall i \in V$, we use a Gaussian mixture model (GMM) to learn a parametric model for the distributed representations. Our idea is inspired from the use of bottleneck features obtained using neural networks for training HMM-based speech recognition systems (Grezl and Fousek, 2008). Our conjecture is that the Gaussian mixture model can learn the latent topics by clustering over the distributed representations that are already trained with a semantic similarity objective (positional and contextual similarity). The distributed representations for the vocabulary V can be represented as an $V \times D$ matrix where each row represents a word w_i in the vocabulary. If we choose to model this data with K Gaussian components, we need to estimate $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, p(k|w_i) \forall k \in K$, $w_i \in V$, namely the means, covariances and mixture weights. We denote the parameters for the k^{th} component by θ_k. We can use the standard Expectation-Maximization (EM) algorithm for Gaussian mixture models to estimate the parameters[3] (Hastie et al., 2001). The EM algorithm was initialized with k-means clustering. We use diagonal covariance matrix approximation in this work, i.e., $\boldsymbol{\Sigma}_k, \forall k \in K$ are diagonal.

Given a new sentence $s' = \{w'_1, \cdots, w'_N\}$, we can perform decoding in the following way to assign the sentence to a particular topic k or a collection of topics since one can obtain the posterior distribution over the topics for each sentence.

$$k^* = \arg\max_{\theta_k} p(k|w'_1, \cdots, w'_N) \quad (1)$$

$$= \arg\max_{\theta_k} p(w'_1, \cdots, w'_N|k)p(k) \quad (2)$$

$$k^* = \arg\max_{\theta_k} p(k) \prod_{i=1}^{N} p(w'_i|k) \quad (3)$$

where $p(k)$ and $p(w'_i|k)$ are obtained from the Gaussian mixture model. The notion of latent topics in this model is represented by the K components of the GMM. Figure 2 shows an example of fitting a GMM to distributed representation of words.

The key difference between our approach and previous approaches to topic modeling is that we start with a dense vector representation for each word in place of a multinomial distribution that is typically learned as part of the topic modeling framework. Second, we do not use the notion of a document since the distributed representations are learned over windows over the entire corpus.

5 Data

We acquired a 10% random sample of Twitter firehose data for 2 weeks across all languages. As a first step, we filtered the tweets by language code. Since the language code is a property set in the

[3] The computation can be parallelized by chunking the $V \times D$ matrix, computing sufficient statistics over the chunks and finally accumulating the statistics.

	Language									
	en		**es**		**fr**		**pt**		**ru**	
Corpus	#voc	#sents	#voc	#sents	#voc	#sents	#voc	#sents	#voc	#sents
Twitter	8371078	178770137	5820863	74784082	1697619	14383118	1816744	22031792	2410668	16025128

Table 1: Statistics of the data used to induce distributed representation in each language. **en**: English, **es**: Spanish, **fr**: French, **pt**: Portuguese. #voc stands for the vocabulary and #sents denotes number of sentences.

user profile, the language code does not guarantee that all tweets are in the same language. We used a simple frequency threshold for language identification based on language specific word lists obtained from Wikitionary[4]. Subsequently, we performed some basic clean-up such as replacing usernames, hashtags, web addresses and numerals with generic symbols such as *_user_*, *_hashtags_*, *_url_* and *_number_*. Finally, we removed all punctuations from the strings and lowercased the text. In this work, we perform our experiments on English, Spanish, French, Portuguese and Russian.

We also formed a stop word list to eliminate extremely common as well as rare words from our topic models. For English, the stop word list comprised of words with frequency greater than 5 million or less than 5 in the training data. For Spanish, French, Portuguese and Russian, the stop list comprised of words with frequency greater than 25000 or less than 5 in the respective training data.

6 Experiments

First, we randomly replaced low frequency words (less than 4 occurrences) with an _UNK_ token to keep the vocabulary open and subsequently used the stop word list to filter the training data. Distributed representations using the continuous-bag-of-words log-linear model was used to obtain $w_i \mapsto \mathbf{d}_i, \forall i \in V$ in each language. We experimented with different dimensions of distributed representations as well as mixture components. Figure 3 shows some topics learned by the model and the terms that comprise the topics for a model learned with D=100 and K=200 on English Twitter data. The terms are ranked by probability.

Unsupervised topic modeling schemes are inherently difficult to evaluate quantitatively. Perplexity of trained models on a held-out set is typically used to objectively evaluate topic models (Blei et al., 2003). However, our scheme does not model the generation process of short text documents. Hence, we use a variety of subjective and objective topic coherence measures to evaluate our framework. We also present a comparison with a state-of-the-art technique for modeling short texts, namely, biterm topic model (BTM) (Yan et al., 2013).

We perform unsupervised topic modeling experiments on the phrasified English Twitter corpus using three schemes. We use LDA as a baseline and treat each tweet as an independent document without any aggregation. We also use the BTM topic model that has been proven to be a suitable fit for short texts. For LDA, we used the open-source implementation GibbsLDA++[5] and for BTM, we used the implementation associated with (Yan et al., 2013). [6]. All three schemes used identical data. We set the parameters $\alpha = 0.05$ and $\beta = 0.01$ for LDA and $\alpha = \frac{50}{K}$ and $\beta = 0.01$ for BTM. The parameters for LDA and BTM were optimized on held-out set with line search using topic coherence metric described in Eq 4. We performed training using our framework for varying window lengths ($wlen$), vector space dimension (D) and number of clusters (K). Specifically, we trained GMMs with the following parameters, $wlen = \{11, 13, 15, 17\}$, $D = \{50, 100\}$ and $K = \{50, 100, 200\}$.

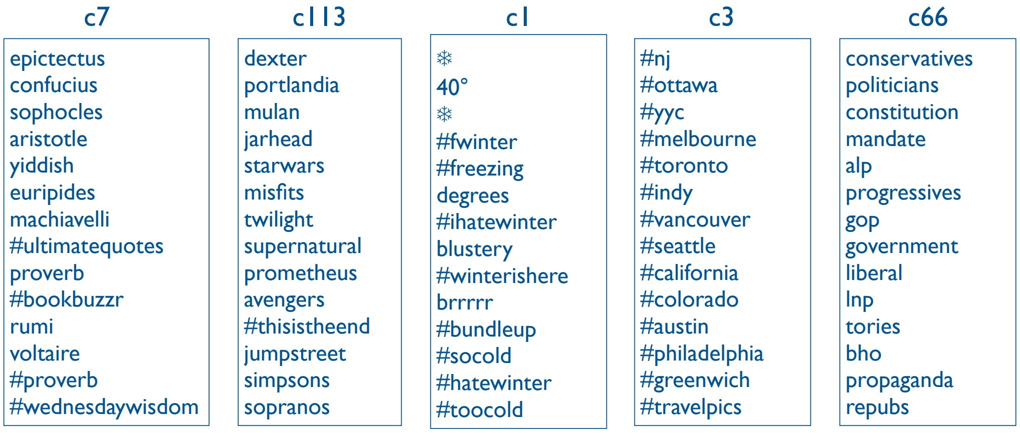
First, we manually inspected the topics obtained by our unsupervised distributed representation framework. A sample of the topics is shown in Figure 3. Manual inspection of many of the topic clusters (top ranked words in each cluster) indicated promising results [7]. As a subsequent step, we asked three professional speech transcribers (also NLP annotators) to subjectively rate the utility of each topic (by displaying the top 50 words) on a 1-3 Likert scale. A rating of *1* indicates completely useless topic cluster while *3* indicates useful topic cluster. *Useful* was defined as a collection

[4] http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists

[5] http://gibbslda.sourceforge.net

[6] http://code.google.com/p/btm/

[7] The topic clusters for all languages can be obtained from https://github.com/annontopicmodel/unsupervised_topic_modeling/. We are not able to share the sentence clusters due to Twitter's data policy.



c7	c113	c1	c3	c66
epictectus	dexter	❄	#nj	conservatives
confucius	portlandia	40°	#ottawa	politicians
sophocles	mulan	❄	#yyc	constitution
aristotle	jarhead	#fwinter	#melbourne	mandate
yiddish	starwars	#freezing	#toronto	alp
euripides	misfits	degrees	#indy	progressives
machiavelli	twilight	#ihatewinter	#vancouver	gop
#ultimatequotes	supernatural	blustery	#seattle	government
proverb	prometheus	#winterishere	#california	liberal
#bookbuzzr	avengers	brrrrr	#colorado	lnp
rumi	#thisistheend	#bundleup	#austin	tories
voltaire	jumpstreet	#socold	#philadelphia	bho
#proverb	simpsons	#hatewinter	#greenwich	propaganda
#wednesdaywisdom	sopranos	#toocold	#travelpics	repubs

Figure 3: Terms with the highest probability for sample latent topics over the entire English Twitter corpus. The topics were obtained by using $wlen = 15$, D=100 and K=200.

Context	K	Fleiss' κ	Mean rating	Median rating
$wlen = 11$	50	0.89	2.2 ± 0.78	2
	100	0.81	2.11 ± 0.79	2
	200	0.82	2.15 ± 0.85	2
$wlen = 15$	50	0.70	2.24 ± 0.77	2
	100	0.80	2.17 ± 0.88	2.5
	200	0.78	2.11 ± 0.89	2.5
$wlen = 17$	50	0.79	2.18 ± 0.82	2
	100	0.68	2.3 ± 0.89	3
	200	0.54	2.18 ± 0.87	2
LDA	100	0.80	1.97 ± 1.01	2
BTM	100	0.78	1.84 ± 1.15	2

Table 2: Subjective evaluation of topic coherence across three annotators ($D = 50$)

of terms that indicated some meaningful semantic property (e.g., movie names, politics, headlines, superlatives, sad emoticons/words, etc.) that could be used for a categorization task. In cases of ambiguity, we asked the labelers to confer a rating of *2*.

We computed the inter annotator agreement between the three labelers using Fleiss' kappa metric (Fleiss, 1971). The results are presented in Table 2. The inter-annotator agreement is quite high for the topic clusters induced with context windows $wlen$ of 11 and 15 words. The agreement is lower for model trained with longer context window perhaps indicating that a window of length 11 or 15 words is sufficient for tweets. The mean ratings are mostly higher than *2* and the median rating for $wlen = 15, K = \{50, 100\}$ are above 2. The subjective ratings are significantly better than LDA and BTM. Hence, subjective evaluation of topics learned using our framework are of consistently high quality.

In order to objectively measure the quality of topics, we also used *coherence score* (Mimno et al., 2011). Given a topic z and a set of top N words (ranked by likelihood) in z, $S^z = \{w_1^z, \cdots, w_N^z\}$, the coherence score is defined as:

$$C(z; S^z) = \sum_{n=2}^{N} \sum_{l=1}^{n-1} log \frac{D(w_n^z, w_l^z) + 1}{D(w_l^z)} \quad (4)$$

where $D(w)$ is the document frequency of word w and $D(w', w)$ is the co-document frequency of words w and w'. The coherence score was then averaged across all topics to obtain the mean coherence score for each scheme, i.e., we computed $\frac{1}{K} \sum_{k=1}^{K} C(z_k; S^{z_k})$. A high coherence score indicates a good topic cluster. Figure 4 shows the average topic coherence score over top N words across varying $wlen$ by fixing $D = 50$ and $K = 50$. The topic clusters are more coherent for $wlen = 11$ at lower values of N but for higher values of N, the model with $wlen = 13$ performs better. Since our vector space GMM model learns topic distributions across the entire corpus, many clusters have a large number of terms with high likelihoods. As a result, it is more appropriate to choose a model with high topic coherence for large values of N.

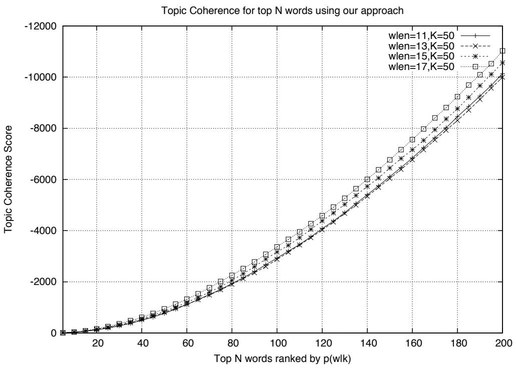


Figure 4: Topic coherence versus top N words in each topic for different values of $wlen$

Next, we analyze the effect of dimension of the vector space model on the topic modeling framework. Figure 5 plots the average topic coherence for varying D. We find that for $D = 100$, the model with lower K achieves better topic coherence. In contrast, for $D = 50$, the model with $K = 200$ is objectively better than the models with $K = \{50, 100\}$. In the former case, the number of topics is smaller and hence a higher dimension is separating the vectors in a better fashion while in the latter case, the increased number of topics achieves better separation even with smaller dimension vectors. One can balance the choice of K and D based on the size of data and desired clusters to be learned.

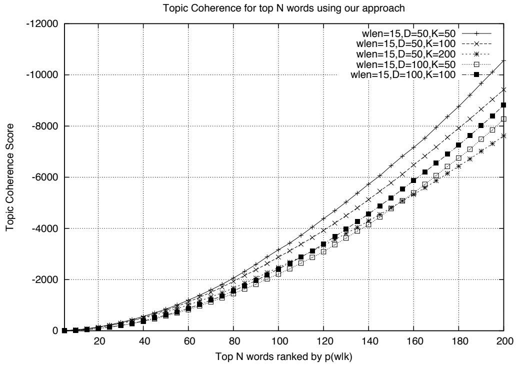


Figure 5: Topic coherence versus top N words in each topic for different values of D

In Figure 6, we plot the topic coherence score for different cluster sizes. The plot shows that for a given N, the best coherence score is obtained for $wlen = 15, D = 50, K = 200$. In general for a large dataset with millions of tweets, $K = 200$ results in better clustering since there are many topics in the data. The model with $wlen = 15$ is interesting since the context window is about the same as the average length of a tweet. The topic coherence scores for $D = 100, K = 200$ were consistently lower than that of the above presented results. It may again be due to the balance needed in the separation of topics due to vector space dimension versus the total number of GMM components. Finally, Figure 7 plots the topic coherence score for our approach, BTM and LDA. The results clearly indicate that our framework performs extremely well on short texts. While previous results using the BTM approach was only performed on a few million tweets, our experiments are performed on 178M tweets for English. The performance of LDA and BTM are very similar while our approach achieves significantly higher topic coherence scores. Finally, Figure 8 shows the topic coherence for Spanish, French, Portuguese and Russian. Our proposed scheme clearly outperforms LDA on large collections of short texts across languages.

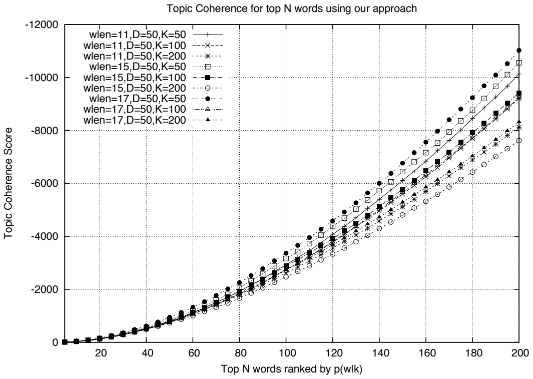


Figure 6: Topic coherence versus top N words in each topic for different values of K

7 Discussion

Conventional topic modeling schemes such as pLSA and LDA need to make modifications when applied on short texts and messages through aggregation strategies. We are not confounded with such a problem since our framework works on large amounts of raw short texts without the need for any aggregation strategy. For media such as Twitter, Facebook or SMS, aggregation over users

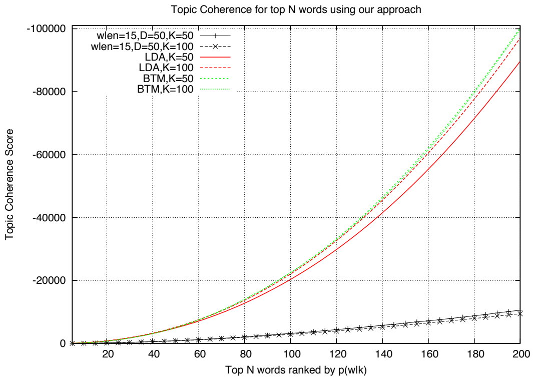


Figure 7: Topic coherence versus top N words in each topic for our scheme, LDA and BTM for English tweet data

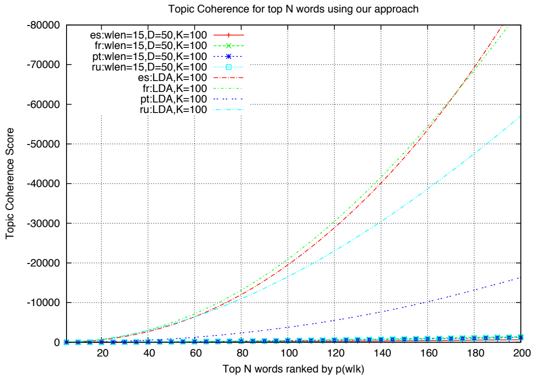


Figure 8: Topic coherence versus top N words in each topic for our scheme and LDA across languages

or location is not a good strategy since the interests of users is diverse and can change quickly. Besides such information is not available for news titles, image captions, etc. Our conjecture is that even for longer documents (emails, news pages, etc.), applying our scheme at the sentence level can be used to accrue topics over the document. The bottleneck is mainly due to the span of windows that one can use to learn reliable distributed representations of words.

We used a log-linear model for learning distributed representations of words in this work. However, our scheme can work with distributed representations obtained by neural networks or latent semantic indexing. The key requirement for distributed representations to work with our GMM framework is that they need to represent good partitioning of semantic concepts in the vector space $\mathbb{R}^D$, where D is the dimensionality of the vector space.

The GMM estimation in this work was simplified due to the assumption of diagonal covariance matrices for the components. We conjecture that the performance can be further improved with full covariance matrices at the cost of computational overhead involved in the Cholesky decomposition. However, the diagonal covariance assumption improves training time as the GMM parameter estimation can be parallelized.

For short texts, the likelihood of a message containing more than 2 or 3 topics is quite low. The decoding scheme presented in this work can obtain a complete posterior distribution over all topics (GMM components) for each message. However, we found that the a large proportion of messages (over 80%) contained only one topic, i.e., the posterior distribution peaks for a particular GMM component. Our scheme can potentially be used for a variety of monitoring tasks such as detection of offensive posts, removal of adult content, advertisement detection, targeted advertising (retail, entertainment, sports), sentiment classification, etc., since such posts are all clustered together.

8 Conclusion

We presented a novel unsupervised topic modeling framework for short texts that uses distributed representations of words and phrases. Our framework models the low-dimensional semantic vector space represented by the dense word vectors using Gaussian mixture models. By learning representations over sufficiently long context windows, we find that one can learn robust word embeddings that can be exploited to learn the semantics of entire short messages. The work presented here was inspired by the use of bottleneck features in HMM-based speech recognition and one can potentially use all the optimization techniques used to estimate GMMs over large datasets (thousands of hours of speech) for modeling large amounts of text. Our experimental results indicate that our scheme can reliably learn latent topics and can be used to categorize short messages with high fidelity in comparison with LDA and biterm topic model. Our scheme is language agnostic and we demonstrated the utility of our scheme in English, Spanish, French, Portuguese and Russian tweets.

References

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. 2009. Curriculum learning. In *Proceedings of ICML*.

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

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of ICML*.

C. De Marcken. 1996. *Unsupervised Language Acquisitiong*. Ph.D. thesis, Massachusetts Institute of Technology.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41.

J. L. Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7(2-3):195–225.

J. L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378382.

F. Grezl and P. Fousek. 2008. Optimizing bottle-neck features for LVCSR. In *Proceedings of ICASSP*, pages 4729–4732.

T. Hastie, R. Tibshirani, and J. Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Liangjie Hong and Brian D. Davison. 2010. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*.

C. Kit and Y. Wilks. 1999. Unsupervised Learning of Word Boundary with Description Length Gain. In *Proceedings of Workshop on Computational Natural Language Learning CoNLL*.

T. Mikolov, S. Kopecký, L. Burget, J. Černocký, and S. Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech*.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

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*, EMNLP '11.

D. Ramage, S. Dumais, and D. Liebling. 2010. Characterizing microblogs with topic models. In *International AAAI Conference on Weblogs and Social Media*.

J. Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14:465471.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362.

G. Salton and M. J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

C. E. Shannon. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379423.

S. Sista, R. Schwartz, T. R. Leek, and J. Makhoul. 2002. An algorithm for unsupervised topic discovery from broadcast news stories. In *Proceedings of HLT*, pages 110–114.

J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*.

Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. Twitterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13.