A Topical Word Embeddings for Text Classification

João Marcos Carvalho Lima[1] **José Everardo Bessa Maia**[1]

[1] *Universidade Estadual do Ceará (UECE)*
Centro de Ciências e Tecnologia
Fortaleza, Brasil

joaomarcosdeveloper@gmail.com, jose.maia@uece.br

Abstract. *This paper presents an approach that uses topic models based on LDA to represent documents in text categorization problems. The document representation is achieved through the cosine similarity between document embeddings and embeddings of topic words, creating a Bag-of-Topics (BoT) variant. The performance of this approach is compared against those of two other representations: BoW (Bag-of-Words) and Topic Model, both based on standard tf-idf. Also, to reveal the effect of the classifier, we compared the performance of the nonlinear classifier SVM against that of the linear classifier Naive Bayes, taken as baseline. To evaluate the approach we use two bases, one multi-label (RCV-1) and another single-label (20 Newsgroup). The model presents significant results with low dimensionality when compared to the state of the art.*

1. Introduction

Automatic text classification on a set of predefined labels is a problem applied to various scenarios such as online news classification, book-by-topic classification and documents classification. For efficiency in this task, the document representation and the classification algorithm are the essential elements.

A document is never rendered in its original format. Thus, document representation is an important task in sorting and other word processing tasks. The process of creating low-dimensional representations that capture semantic and contextual information becomes increasingly crucial to the efficiency of algorithms in text-sorting tasks.

One of the most common forms of document representation is the Bag-of-Words (BoW) model, where a document is represented by a vector in the terms space d_i = $[w_{i1}, ..., w_{in}]$, where n is the dimension that corresponds to the number of terms in the dictionary of terms, w_{ij} is the weight of the jth term in the ith document. However, this model suffers with high dimensions, as well as low semantic relationships between dimensions. These weights are calculated using the *tf-idf* (Term frequency-inverse document frequency) statistics [Schütze et al. 2008]. The assumptions here are of unambiguous terms and that the frequency of terms in a document is representative of its semantics.

Various approaches have appeared as an alternative to the Bag-of-Words model, and the word embeddings representation model is one of the most effective. In accordance with the word2vec algorithm[Mikolov et al. 2013], the word is represented by a vector of dimension n which depends on the frequency of the contexts of use of the word in a large base of colloquial use of the language. These dimensions capture different semantic relationships, where they most often represent characteristics such as gender, degree and

number of a given word. Note, however, that it is not straightforward to obtain document representation from the embedding representation of words.

In addition to these approaches, we find in the literature several papers that seek to represent documents through topic models. In this model, a topic is defined by a statistical distribution over words while a document is represented by a statistical distribution on the topics[Liu et al. 2015]. This is an unsupervised model so it does not give meaning to topics. But there is research effort in this direction[Lau et al. 2011, Rubin et al. 2012, Ramage et al. 2009].

An attempt to search for interpretive document vectors is the approach seek to represent documents through concepts, known as Bag-of-Concepts (BoC) [Mouriño-garcía et al. 2015][Kim et al. 2017]. Interpretable vectors can provide deeper understanding of documents and of the data set and thus enhance the design of models with greater semantics. Bag-of-Concepts is, in general, based on semantic annotation, which extracts concepts that define a document.

This paper proposes a document representation approach combining topics extracted via LDA *(Latent Dirichlet Allocation)* and words embeddings and applies to the task of categorizing text. With the use of topic model it is expected to obtain a strong reduction of dimensionality retaining the significant part of the semantics of the documents. Using the embedding representation of the words, it is expected to reduce the effects of polysemy and synonymy. The combination of these two approaches allows to obtain low dimensions and to capture semantic relations between documents and topics. The evaluated experiment combines several topic sizes, which determines the size of the input that is passed to the classifiers. The results obtained in text categorization tasks with low dimensionality are significant when compared with the state of the art.

Two classifiers are used in this work. SVM is a principle rooted algorithm for classification and regression (SVR) capable of generating non-linear decision boundaries. On the other hand, Naive Bayes is one of the most popular linear classification algorithms for its ability to combine programming simplicity and performance. These two algorithms are used in the experiments to reveal the influence of the classification algorithm on the task of categorizing news texts.

The remainder of this paper is organized as follows. Section 2 presents the related works, Section 3 describes fundamentals of extraction of topics, generating word embeddings, as well as algorithms SVM and Naive Bayes. Section 4 describes the task of classification, the data and the evaluation criteria. Section 5 presents the proposed approach. Section 6 presents the results and discussion, and the article is concluded in Section 7.

2. Related Works

Several previous papers have developed document representation using a combination of topic model with embeddings and this representation is applied to several tasks including single-label and multi-label classification. Others papers simply use the representation of topics to represent documents. This section presents a brief review of papers directly related to the proposed approach here.

In [Sriurai 2011], the author applied two feature selection algorithms in two exper-

iments. The first using the BoW model for document representation, and the second using topic models, where a document was represented as a probability vector for each topic. The number of topics extracted was 200. After obtaining this representation the author applied three classification algorithms: Naïve Bayes (NB), Decision tree (Dtree) and Support Vector Machines (SVM). The best result was using the approach with topic models applying the IG (Information gain) feature selection algorithm with the SVM algorithm reaching 79% of F1 measurement.

Some papers seek to represent documents through concepts (BoC). In [Mouriño-garcía et al. 2015], authors represent concepts in a document through external resources. They used Wikipedia as their source for semantic annotations. In the evaluation of the experiment, they used the SVM algorithm and compared the performance of the BoC model with the traditional BoW. The BoC model was superior to BoW reaching close 70% of F1 measurement in dataset Reuters, the authors did not inform the dimensions of the evaluated models.

In [Kim et al. 2017], the authors find disadvantages in BoC models, as well as doc2vec. Doc2vec is a variant of word2vec that represents documents in a space of n dimensions. They point to interpretability in the meaning of each dimension of doc2vec and represent each document based on concepts, using the frequency of clusters obtained in representation of words embeddings. The authors evaluate the model in classification tasks, comparing them with the doc2vec, BoW, LSA and average words embeddings models. They reached 82.86% of F1 measurement with 100 dimensions in R52 dataset.

Since words embeddings capture semantic relationships, a number of papers have appeared exploring this approach. In [Li et al. 2016] combine topics and words embeddings in model TopicVec. They add an embedding link function to model the word distribution in a topic replace the categorical distribution in LDA. This link function is used because the semantic relatedness is already encoded as the cosine distance in the embedding space. They reached 92.2% of F1 measurement with 111 dimensions in Reuters dataset.

In the work of [Liu et al. 2015] the authors propose an approach called Topical Word Embeddings (TWE), which combines word embeddings and topic models and simple form to represent embeddings of topics by words. They obtained the mean of the words embeddings of the words of each topic, assigning that average as the embedding of the topic. They reached 80.6% of F1 measurement with 400 dimensions in 20NewsGroup dataset.

3. Background

In this section we briefly describe the methods used to create the document representation approach and classification. They are: LDA, word2vec (Embeddings) and the classification algorithms SVM (Support Vector Machine) and NB (Naive Bayes). The reader interested in detailed presentations should search for the indicated references.

3.1. Topic Model

The topic model used in this work is the Latent Dirichlet Allocation (LDA) [Blei et al. 2003]. LDA is used to abstract topics from a corpus. A topic is a probability distribution on terms of a vocabulary. These topics may mean the representation

of an underlying semantic theme. Each document is represented as a finite mixture of an underlying set of topics. LDA is not supervised and not based on context. As it uses Dirichlet distribution to draw words over topics and draw topics over documents.

Formally, given the set of documents D, and W the set of terms (vocabulary) and T the set of topics, where T is the result of the statistical inference on the set of terms W, LDA models the generation of documents within a corpus as the following process: 1) A mixture of k topics, θ , is sampled from a Dirichlet prior, which is parameterized by α; 2) A topic z_n is sampled from the multinomial distribution, $p(\theta, \alpha)$, which models $p(z_n = i|\theta)$; 3) A word, w_n, is then sampled (given the topic z_n) via the multinomial distribution $p(w|z_n)$. Given a corpus of M documents $D = \{w_1, ..., w_M\}$, the EM algorithm is usually used to learn the parameters of an LDA model.

3.2. Words Embeddings

Briefly, representation by words embedding is one of the ways to solve the problems found in BoW models, such as high dimensionality and semantic problems. There are several models for word embeddings, in this work Word2Vec [Mikolov et al. 2013] is used. Word2vec is a model trained by a neural network to find representations of words in a space of n dimensions. It is based on the distributional hypothesis model, where words occurring in a similar context have a similar meaning.The neural network consists of two layers that are trained to construct context vectors of words.

3.3. SVM

In the Support Vector Machines (SVM) algorithm, the classification is based on the separation margin of the classes. Thus, the objective of the SVM training is to find an optimal separator hyperplane, where the separation distance between classes is maximum, called maximum margin hyperplane. The samples that are located on the margins are the most informative for the creation of the limit of decision of the classification and are called support vectors.

The classification can be performed both in the original attribute space and in a feature space designed through a kernel function. Thus, problems that are not linearly separable in the original space can become linearly separable in the feature space. As the size of the feature space increases, so does the likelihood of such a problem becoming linearly separable. The ability to separate data with non-linearly separable distribution depends on the choice of the kernel function, and must be analyzed according to the problem domain. The most used kernels are: Linear, Polynomial and RBF (Radial Base Function). In this work, Linear was used.

3.4. Naive Bayes

There are several algorithms based on the naive Bayes assumption, as Bayesian Network, multi-variate Bernoulli model and multinomial model. In this paper we used a multinomial model, that is a uni-gram language model with integer word counts.

This model captures word frequency information. Each document is composed of a sequence of words from the same vocabulary V. The probability of each word event in a document is independent of the word's context and position in the document an each document di is extracted a multinomial distribution of words. The independence between words (features) is what characterizes the algorithm to be an NB.

4. Task and Evaluation

In this section we briefly describe the task of classification, as well as the datasets used and the evaluation method.

4.1. Task

A variant of the classification problem is the multi-label classification [Rubin et al. 2012]. For a formal description of the multi-label classification problem, let X = $(x_1, ..., x_n)$ be a finite set of instances (documents, news) and let Y = $(y_1, ..., y_m$ a finite set of labels (categories) such that each instance $x_i \in X$ is associated with multiple labels of class Y_i, where $Y_i \subseteq Y$. Given a set of training examples S = $\{(x_1, Y_1), ..., (x_n, Y_n)\}$ the machine learning task is to construct, from S, a classifier $f : X \rightarrow 2^Y$ capable of estimating an unknown target function $\varphi : X \rightarrow 2^Y$. In this context, the power set 2^Y represents all possible multi-label categorizations for a news. The problem of multiclassing with a single-label is a special case of the multi-label problem in which each instance is assigned only one label

To the multi-label problem, we apply the One-Against-All strategy - which constructs —C— binary prediction models, where —C— is the number of classes. Each model is trained to separate one of the classes from the others. Therefore, the ith model is trained considering that all examples of the ith class belong to the positive class, while the examples of the other classes belong to the negative class. When an example of an unknown class is presented, it is sorted by each of the — C — models and receives the class relative to the model that obtains the best result.

4.2. Datasets

The experiment was performed in two text classification datasets. The description of each dataset and the pre-processing done is detailed below:

- **RCV-1 (Reuters Corpus Volume 1):** It is a collection that contains 800,000 news from the Reuters site arranged in 103 categories (Economy, Science and Technology, Sports, Corporate, among others). It is a multi-label dataset, where each news item is associated with more than one category. For reasons of computational limitation, we used only 33.149 news items divided into 101 categories.
- **20 Newsgroups:** It is a collection that contains 18.821 documents arranged in 20 categories (Technology, Politics, Religion, Sport among others). It is a single-label dataset, where each document is associated with only one category. We used all dataset in the experiment.

Preprocessing is the same for both datasets. First we apply the Tokenization process (removal of spaces). In the next step are removed the stopwords (terms like: articles, prepositions, numbers, among others). Lastly, the Stemming process (radicalization of words).

4.3. Metrics

We used the Precision, Recall and F1-score metrics to evaluate the experiments. The value of Precision is defined as: $P = TP/(TP + FP)$ and Recall $R = TP/(TP + FN)$, where TP is the number of positive true, TN true negative, FP false positive and FN false negative.

F1-score is a weighted measure of Precision and Recall defined by: *F1-score* $= (2 * P * R)/(P + R)$.

5. Approach

In this section we present the document representation approach based on embeddings of topic words extracted via LDA. We call our approach Topical Word Embedding Space Model (TWESM). The implementation of LDA and word2vec provided by the gensim Python library was used in the experiments of this article.

Given a collection of documents D = $\{d_1,...,d_n\}$ and a sets of words W = $\{w_1,...,w_n\}$, where each document is represented by a set $T \in W$. The LDA model receives the collection of documents as input, and extracts n topics. Each topic is represented by T_i = $\{(w_1, p_1),...,(w_i, p_i)\}$, where w_i is a word belonging to corpus W and p_i is the probability of that word in the topic T_i.

We used word2vec with dimension 100 to obtain the embedding representation of the set of words W. The size of the diemension was chosen by the default value of gensim For the representation of each document, we obtain the average of the words embeddings belonging to a document d_i, forming a matrix DE of size MxN, where M represents the number of documents and N is the size of embeddings.

The embedding representation of each word of each topic T_i, forming a matrix TE of size MxN, where M represents the number of topic words and N is the dimension of embeddings. Because a word may appear in more than one topic, we removed the words embeddings repeated in the matrix TE.

In order to compare each document with the terms obtained through LDA, we calculated the cosine similarity, according to the Equation 1 of each element of the matrix DE with the elements of the matrix TE obtaining a matrix H of size MxN, where M represents the number of documents and N the number of terms:

$$cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||}. \quad (1)$$

The Algorithm 1 presents the pseudocode for obtaining document representation based on embeddings of topics.

Algorithm 1 The algorithm for generate the TWESM

Input: D, a document colection; n, a number of topics; N, a number of documents. M, a number of topic words.
Output: D_{twe}, a TWE Space Model for D.

$topic\_words$ ←LDA(D, n) topic_words_embeddings ←Word2Vec($topic\_words$)
$documents\_embeddings$ ←Mean_words_embeddings(D)
for $i \leftarrow 1$ **to** N **do**
  for $j \leftarrow 1$ **to** M **do**
    $m[i][j]$ ←cosine($documents\_embeddings[i], topic\_words\_embeddings[j]$);
  end for
end for

The Figure 1 presents the approach for obtaining document representation based on embeddings of topics.

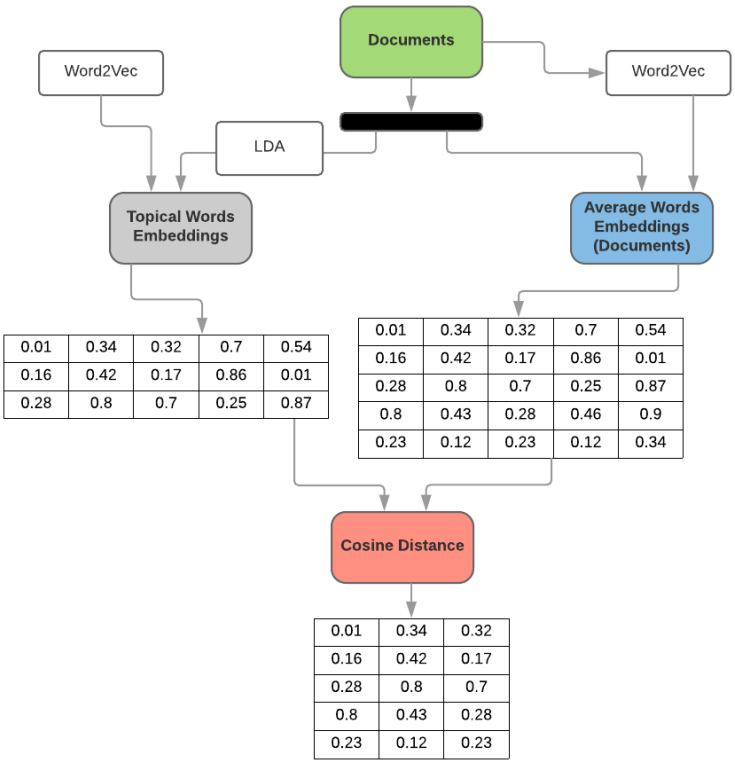

Figure 1. Approach

6. Results and Discussion

The experiment was performed with variations of the number of topics extracted in the LDA process, where n topic words were obtained. For the RCV-1 datasets 70% was used for the set and training and 30% for the test set. For the 20newsgroup dataset, we also used 70% for the training set and 30% for the test set.

Tables 1 to 4 and the graphs of Figures 2 and 3 show the results of experiments performed to evaluate the approach. The Tables 1 to 4 present the evaluation of the approach together with the comparative models in datasets with the respective classifier.

We highlight the best result in each model and we observe that our approach was superior to the BoT model when the SVM classifier is used in both datasets. When NB classifier is used, we observed that the BoT model has the best result. We also observed that even with a smaller size than the BoT model, our approach presents the same result. As we can see in Table 1, the BoT model with 500 topics extracted and dimension 500 obtained 66% of *F1-score*, already our approach with 50 topics extracted and dimension 265 obtained the same 66% of *F1-score*.

When compared to the BoW model, we can observe that the TWESM model presented a nearby result with a reduced size in 95%.

Method	Topics	Dimension	Precision	Recall	F1-score
TWESM	10	58	0.69	0.50	0.55
	50	265	0.77	0.60	0.66
	100	532	0.77	0.62	0.68
	200	1091	0.79	0.64	0.69
	500	**2714**	0.79	0.66	**0.71**
BoT	10	10	0.33	0.28	0.30
	50	50	0.58	0.46	0.50
	100	100	0.62	0.49	0.53
	200	200	0.74	0.57	0.63
	500	**500**	0.81	0.59	**0.66**
BoW	0	**56269**	0.91	0.77	**0.82**

Table 1. RCV-1 - SVM.

Method	Topics	Dimension	Precision	Recall	F1-score
TWESM	10	58	0.18	0.09	0.12
	50	265	0.27	0.25	0.26
	100	532	0.32	0.35	0.30
	200	1091	0.33	0.50	0.35
	500	**2714**	0.30	0.66	**0.36**
BoT	10	10	0.31	0.22	0.25
	50	50	0.53	0.29	0.35
	100	100	0.56	0.28	0.34
	200	**200**	0.68	0.33	**0.40**
	500	500	0.70	0.32	0.39
BoW	0	**56269**	0.69	0.33	**0.39**

Table 2. RCV-1 - NB.

Method	Topics	Dimension	Precision	Recall	F1-score
TWESM	10	40	0.45	0.45	0.44
	50	141	0.49	0.49	0.48
	100	277	0.50	0.51	0.50
	200	666	0.51	0.52	0.51
	500	**2114**	0.51	0.53	**0.52**
BoT	10	10	0.2	0.21	0.16
	50	50	0.31	0.33	0.30
	100	100	0.34	0.35	0.33
	200	200	0.35	0.35	0.34
	500	**500**	0.36	0.36	**0.36**
BoW	0	**98507**	0.76	0.76	**0.76**

Table 3. 20 Newsgroup - SVM.

Method	Topics	Dimension	Precision	Recall	F1-score
TWESM	10	40	0.25	0.28	0.22
	50	141	0.31	0.35	0.31
	100	277	0.31	0.34	0.32
	200	666	0.34	0.35	0.34
	500	**2114**	0.35	0.35	**0.34**
BoT	10	10	0.19	0.23	0.19
	50	50	0.31	0.33	0.30
	100	100	0.34	0.35	0.33
	200	200	0.36	0.37	0.35
	500	**500**	0.36	0.37	**0.36**
BoW	0	**98507**	0.79	0.73	**0.72**

Table 4. 20 Newsgroup - NB.

The graphs show a comparative between model Bot and model TWESM, and we observe for both datasets and for both classifiers, dimensionality reduction with 100 topics is sufficient to obtain stable results in all cases. There is no gain in increasing the number of topics beyond 100.

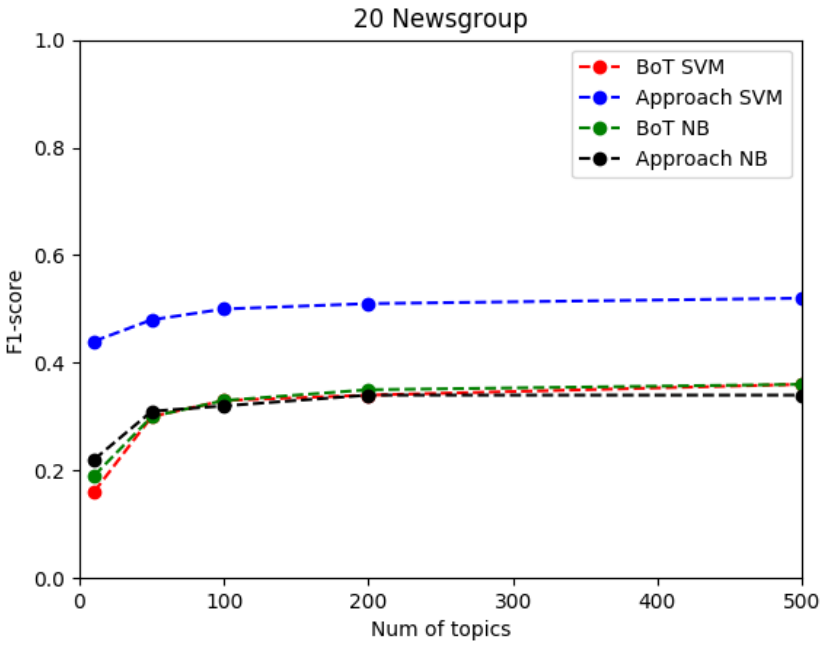

Figure 2. Results 20 Newsgroup

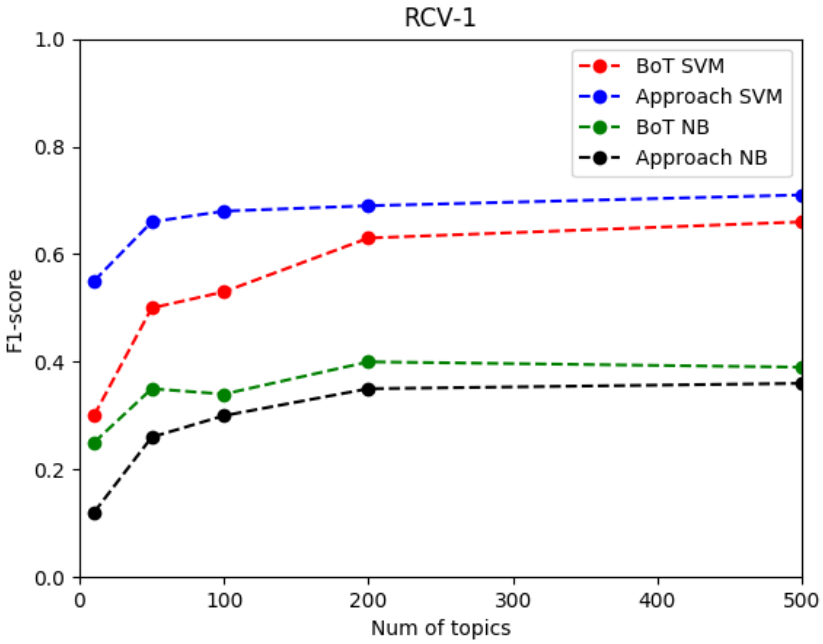

Figure 3. Results RCV-1

7. Conclusion

We apply a combination of two document representation methodologies to obtain a new representation. The proposed model presents acceptable results in comparison with traditional document representation approaches.

We observed that in some experiments in low dimension our approach reached the same result when compared to the BoT model with a higher dimension. We can conclude that in addition to our approach having the strengths of Topic Model and Word2vec, it has been able to achieve good results with low dimensions.

In future works, we will evaluate the representation approach in other datasets, as well as investigate other models similar to LDA for topic extraction.

References

Blei, D. M., Edu, B. B., Ng, A. Y., Edu, A. S., Jordan, M. I., and Edu, J. B. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Kim, H. K., Kim, H., and Cho, S. (2017). Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, 266:336–352.

Lau, J. H., Grieser, K., Newman, D., and Baldwin, T. (2011). Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1536–1545. Association for Computational Linguistics.

Li, S., Chua, T.-S., Zhu, J., and Miao, C. (2016). Generative Topic Embedding: a Continuous Representation of Documents (Extended Version with Proofs).

Liu, Y., Liu, Z., Chua, T.-S., and Sun, M. (2015). Topical Word Embeddings. *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, 2(C):2418–2424.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Mouriño-garcía, M., Pérez-rodríguez, R., and Anido-rifón, L. (2015). Bag-of-Concepts Document Representation for Textual News Classification (PDF Download Available).pdf. 6(1):173–188.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics.

Rubin, T. N., Chambers, A., Smyth, P., and Steyvers, M. (2012). Statistical topic models for multi-label document classification. *Machine learning*, 88(1-2):157–208.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press.

Sriurai, W. (2011). Improving Text Categorization by using a Topic Model. *Advanced Computing*, 2(6):21–27.