Augmenting word2vec with Latent Dirichlet Allocation within a Clinical Application

Akshay Budhkar
University of Toronto;
Vector Institute for Artificial Intelligence
abudhkar@cs.toronto.edu

Frank Rudzicz
University of Toronto;
St Michael's Hospital;
Surgical Safety Technologies Inc.;
Vector Institute for Artificial Intelligence
frank@cs.toronto.edu

Abstract

This paper presents three hybrid models that directly combine latent Dirichlet allocation and word embedding for distinguishing between speakers with and without Alzheimer's disease from transcripts of picture descriptions. Two of our models get F-scores over the current state-of-the-art using automatic methods on the DementiaBank dataset.

1 Introduction

Word embedding projects words into a lower-dimensional latent space that captures semantic and morphological information. Separately but related, the task of topic modelling also discovers latent semantic structures or *topics* in a corpus. Latent Dirichlet allocation (LDA) uses bag-of-words statistics to infer topics in an unsupervised manner. LDA considers each document to be a probability distribution over hidden topics, and each topic is a probability distribution over all words in the vocabulary, both with Dirichlet priors.

The inferred probabilities over learned latent topics of a given document (i.e., *topic vectors*) can be used along with a discriminative classifier, as in the work by Luo and Li (2014), but other approaches such as TF-IDF (Lan et al., 2005) easily outperform this model, like in the case of the *Reuters-21578* corpus (Lewis et al., 1987). Here, we hypothesize that creating a hybrid of LDA and word2vec (Mikolov et al., 2013b) models will produce discriminative features. We introduce three new variants of hybrid LDA-word2vec models, and investigate the effect of dropping the first component after principal component analysis (PCA). These models can be thought of as extending the conglomeration of topical embedding models. We incorporate topical information into our word2vec models by using the final state of the topic-word distribution in the LDA model during training.

1.1 Motivation and Related Work

Alzheimer's disease (AD) is a neurodegenerative disease that affects approximately 5.5 million Americans with annual costs of care up to $259B in the US, in 2017, alone (Alzheimer's Association et al., 2017). The existing state-of-the-art methods for detecting AD from speech used extensive feature engineering, some of which involved experienced clinicians. Fraser et al. (2016) investigated multiple linguistic and acoustic characteristics and obtained accuracies up to 81% with aggressive feature selection. Yancheva and Rudzicz (2016) use vector-space topic models, and achieved F-scores up to 74%. It is generally expensive to get sufficient labeled data for arbitrary pathological conditions.

In our experiments, we train our hybrid models on a normative dataset and apply them for classification on a clinical dataset. The goal of this project is to i) effectively augment word2vec with LDA for classification, and ii) to improve the accuracy of dementia detection using automatic methods.

2 Datasets

The **Wisconsin Longitudinal Study** (WLS) is a normative dataset where residents of Wisconsin perform the Cookie Theft picture description task (Goodglass and Barresi, 2000). The audio excerpts from the 2011 survey *(N = 1,366)* were converted to text using the Kaldi open source automatic speech recognition (ASR) engine (Povey et al., 2011), specifically using a bi-directional LSTM trained to the Fisher data set (Cieri et al., 2004).

DementiaBank (DB) is part of the TalkBank project (MacWhinney et al., 2011). Each participant was assigned to either the 'Dementia' group ($N = 167$) or the 'Control' group ($N = 97$). We use 240 samples from those in the 'Dementia'

Proceedings of NAACL-HLT 2019, pages 4095–4099
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Total
CT	55	56	40	40	50	241
AD	56	54	70	70	60	310

Table 1: DB test-data distribution

group, and 233 from those in the 'Control' group. Each speech sample was recorded and manually transcribed at the word level following the CHAT protocol (MacWhinney, 1992). We use a 5−fold group cross-validation (CV) to split this dataset while ensuring that a particular participant does not occur in both the train and test splits. Table 1 presents the distribution of Control and Dementia groups in the test split for each fold.

WLS is used to train our LDA, word2vec and hybrid models that are then used to generate feature vectors on the DB dataset. The feature vectors on the train set are used to train a discriminative classifier (e.g., SVM), that is then used to do the AD/CT binary classification on the feature vectors of the test set. During training we filter out spaCy's (Honnibal and Montani, 2017) list of stop words from our datasets. For our LDA models trained on ASR transcripts, we remove the *[UNK]* and *[NOISE]* tokens generated by Kaldi, as well as the *um* and *uh* tokens, as this improved downstream model performance.

3 Methods

3.1 Baselines

Once an **LDA** model is trained, it can be used to infer the topic distribution on a given document. We set the number of topics empirically to *K=5* and *K=25*.

We use a pre-trained **word2vec** model trained on the Google News Dataset (Mikolov et al., 2013a). We also train our own word vectors with 300 dimensions and window size of 2 to be consistent with the pre-trained variant. We represent a document by averaging the word embeddings for all the words in that document.

Third, **TF-IDF** is a common numerical statistic in information retrieval that measures the number of times a word occurs in a document, and through the entire corpus. We use a TF-IDF vector representation for each transcript for the top *1,000* words after preprocessing, learned on the train set.

Finally, since the goal of this paper is to create a hybrid of LDA and word2vec models, one of the simpler hybrid models – i.e., **concatenating**

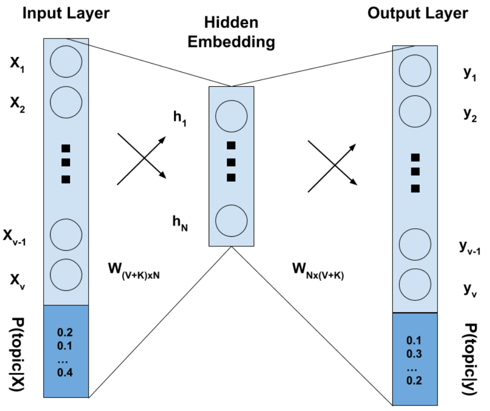

Figure 1: Neural representation of topical word2vec

LDA probabilities with average word2vec representations – is the fourth baseline model.

3.2 Topic Vectors

We represent a topic vector as the *weighted* combination of the word2vec vectors of the words in the vocabulary. This represents every inferred *topic* as a real-valued vector, with the same dimensions as the word embedding. A topic vector for a given topic is defined as:

$$topic\_vector_D = \frac{\sum_{i=1}^{V} p_i W_i}{V} \qquad (1)$$

where V is the vocabulary size of our corpus, p_i is the probability that a given word appears in the topic, from LDA, and W_i is the word2vec embedding of that word.

A document vector is given by:

$$avg\_topic\_vector_D = \frac{\sum_{i=1}^{K} p_i T_i}{K} \qquad (2)$$

where T_i is the topic vector defined in Equation 1, K is the number of topics of the LDA model, and p_i is the inferred probability that a given document contains topic i.

3.3 Topical Embedding

To generate topical embeddings, we use the $P(word\,|\,topic)$ from LDA training as the ground truth of how words and topics are related to each other. We normalize that distribution, so that $\sum_{topics} P(topic\,|\,word) = 1$. This gives a topical representation for every word in the vocabulary.

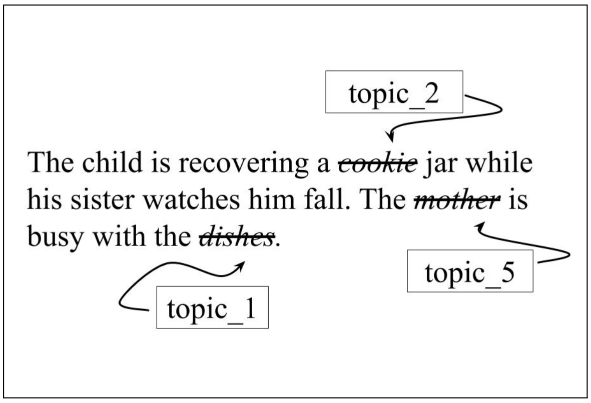
topic_2
The child is recovering a ~~cookie~~ jar while his sister watches him fall. The ~~mother~~ is busy with the ~~dishes~~.
topic_5
topic_1

Figure 2: Example topic induction in the WLS corpus.

We concatenate this representation to the one-hot encoding of a given word to train a skip-gram word2vec model. Figure 1 shows a single pass of the word2vec training with this added information. There, X and Y are the concatenated representations of the input-output words determined by a context window, and h is an N-dimensional hidden layer. All the words and the topics are mapped to an N-dimensional embedding during inference. Our algorithm also skips the softmax layer at the output of a standard word2vec model, as our vectors are now a combination of one-hot encoding and dense probability matrices.

To get document representations, we use the average these modified word2vec embeddings. We also propose a new way of representing documents as seen in Figure 3 where we concatenate the average word2vec with the word2vec representation of the most prevalent topic in the document following LDA inference.

3.4 Topic-induced word2vec

Our final model involves inducing topics into the corpus itself. We represent every topic with the string *topic_i* where i is its topic number; e.g., topic 1 is *topic_1*, and topic 25 is *topic_25*. We also create a *sunk* topic character (analogous to *UNK* in vocabulary space) and set it to *topic_(K+1)*, where K is the number of topics in the LDA model.

We normalize $P(word\,|\,topic)$ to get $P(topic\,|\,word)$ (Section 3.3). With a probability of 0.5, set empirically, we replace a given word with the topic string for $\max(P(topic\,|\,word))$, provided the max value is ≥ 0.2. If this max value is < 0.2, the word is replaced with the sunk topic for that model.

Figure 2 shows an example of topic induction on a snapshot of an ASR transcript of WLS. This process is repeated $N = 10$ times and this augmented corpus is now run through a standard skip-gram word2vec model with dimensions set to 400 to accommodate the bigger corpus. The intuition behind this approach is that it allows words to learn how they occur around *topics* in a corpus and vice versa. Document representations follow the same format as in Section 3.3 and in Figure 3.

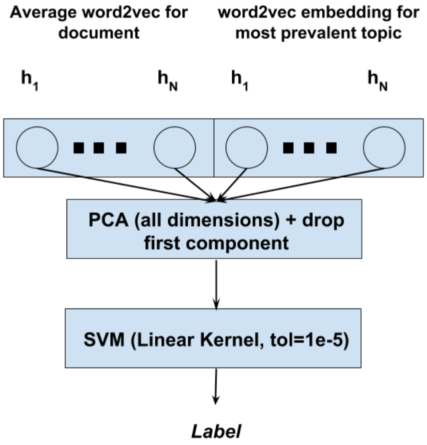
Average word2vec for document
word2vec embedding for most prevalent topic
h1
hN
h1
hN
PCA (all dimensions) + drop first component
SVM (Linear Kernel, tol=1e-5)
Label

Figure 3: Setup for classification using hybrid models. The PCA step exists for models applying work described in Section 3.4.1.

3.4.1 PCA Update

Inspired by the work of Arora et al. (2016), we transform the features of our models with PCA, drop the first component, and input the result to the classifier, as it improves accuracy empirically.

Apart from the ablation study, all experiments use an SVM classifier with a linear kernel and tolerance set to 10^{-5}.

4 Results

4.1 DB Classification

We report the average of the F1 micro and F1 macro scores for the 5-folds for all baseline and proposed models. These results are presented in two parts in Tables 2 and 3.

The TF-IDF model sets a very strong baseline with an accuracy of 74.95%, which is already better than the automatic models of Yancheva and Rudzicz (2016) on the same data.

The 25-topic topical embedding model outperforms the TF-IDF baseline and gives accuracies of 75.32% when using the average word2vec approach. All topic-induced models beat the topical

	LDA		**Pre-trained word2vec**		**Trained word2vec**		**TF-IDF**	**Concatenation**	**Topic Vectors**
	5 Topics	25 Topics		PCA Update		PCA Update			
F1 micro	55.70%	62.78%	66.97%	67.34%	71.50%	72.60%	**74.95%**	**74.22%**	56.27%
F1 macro	54.44%	62.46%	64.78%	65.10%	71.33%	72.25%	**74.49%**	**73.90%**	35.90%

Table 2: DB Classification results (Average 5-Fold F-scores): Part 1

	Topical word2vec	**Topical word2vec + topic**	**Topical word2vec**	**Topical word2vec + topic**	**Topic-Induced word2vec**	**Topic-Induced word2vec + topic**	**Topic-Induced word2vec**	**Topic-Induced word2vec + topic**	**Topic-Induced word2vec**	**Topic-Induced word2vec + topic**	**Topic-Induced word2vec**	**Topic-Induced word2vec + topic**
	25 topics		25 topics and PCA		5 topics		25 topics		5 topics and PCA		25 topics and PCA	
F1 micro	75.32%	75.32%	73.69%	71.14%	75.32%	75.68%	**77.50%**	76.40%	**77.10%**	74.59%	**76.77%**	75.31%
F1 macro	74.97%	75.01%	73.32%	70.70%	74.98%	75.36%	**77.19%**	76.09%	**76.86%**	72.27%	**76.48%**	75%

Table 3: DB Classification results (Average 5-Fold F-scores): Part 2

embedding model, with the 25-topics variant giving an average accuracy of 77.5%.

To check if accuracies are statistically significant, we calculate our test statistic (Z) as follows:

$$Z = \frac{p_1 - p_2}{\sqrt{2\bar{p}(1-\bar{p})/n}} \tag{3}$$

where (p_1, p_2) are the proportions of samples correctly classified by the two classifiers respectively, n is the number of samples (which in our case is 551) and $\bar{p} = \frac{p_1+p_2}{2}$.

Augmenting word2vec models with topic information significantly improves accuracy in the topic-induced word2vec model ($p = 0.0227$) when compared to the vanilla-trained word2vec model. This change is not significant in the topical embedding model ($p = 0.152$).

4.2 Evaluation of Different Classifiers

Using the the 25-topic topic-induced word2vec, we consider other discriminative classifiers. As seen in Table 4, the linear SVM model gives the best accuracy of 77.5%, though all other models perform similarly, with accuracies upwards of 70%. There is no statistically significant difference between using an SVM vs. a LR ($p = 0.569$) or a gradient boosting classifier ($p = 0.094$).

Discriminative Classifier	**F1 micro**	**F1 macro**
SVM w/ linear kernel	**77.50%**	**77.19%**
Logistic Regression (LR)	76.05%	75.51%
Random Forest	71.13%	69.97%
Gradient Boosting Classifier	73.14%	72.39%

Table 4: DB: Discriminative Classifiers on Topic-induced LDA-25 model

5 Discussions

Although the topic distributions of the LDA models were not distinctive enough in themselves, they capture subtle differences between the AD and CT patients *missed* by the vanilla word2vec models. Simple concatenation of this distribution to the document increases the accuracy by 2.72% ($p = 0.31$).

Topic vectors on their own do not provide much generative potential for this clinical data set, as representing a document as a single point in space, after going through two layers of contraction, removes information relevant to classification.

Our novel topic-induced model performs the best among our proposed models, with an accuracy of 77.5% on a 5-fold split of the DB dataset. To put this in perspective, Yancheva and Rudzicz (2016)'s automatic vector-space topic models achieved 74% on the same data set, albeit with a slightly different setup. Applying PCA to the features does not have a significant trend.

6 Limitations and Future Work

Both of our proposed topic-induced and topical embedding models could benefit from using corpus-level word probability priors during normalization, and we intend on experimenting with those in the future. Work needs to be done to directly compare the performance of our models to the topical models proposed by (Liu et al., 2015), given that both kinds of models fall in the same universe. Finally, while we get promising results on a clinical application, the generalizability of these methods needs to be studied on other text classification tasks.

7 Conclusions

In this paper, we show the utility of augmenting word2vec with LDA-induced topics. We present three models, two of which outperform vanilla word2vec and LDA models for a clinical binary text classification task. Going forward, we will test this model on other tasks, diagnostic and otherwise, to see its generalizability. This can provide a starting point for clinical classification problems where labeled data may be scarce.

Acknowledgements

The Wisconsin Longitudinal Study is sponsored by the National Institute on Aging (grant numbers R01AG009775, R01AG033285, and R01AG041868), and was conducted by the University of Wisconsin. Rudzicz is an Inaugural CIFAR Chair in artificial intelligence.

References

Alzheimer's Association et al. 2017. 2017 Alzheimer's disease facts and figures. *Alzheimer's & Dementia*, 13(4):325–373.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings.

Christopher Cieri, David Miller, and Kevin Walker. 2004. Fisher English training speech parts 1 and 2. *Philadelphia: Linguistic Data Consortium.*

Kathleen C Fraser, Jed A Meltzer, and Frank Rudzicz. 2016. Linguistic features identify Alzheimer's disease in narrative speech. *Journal of Alzheimer's Disease*, 49(2):407–422.

Harold Goodglass and Barbara Barresi. 2000. *Boston diagnostic aphasia examination: Short form record booklet*. Lippincott Williams & Wilkins.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear.*

Man Lan, Chew-Lim Tan, Hwee-Boon Low, and Sam-Yuan Sung. 2005. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1032–1033. ACM.

David Lewis et al. 1987. Reuters-21578. *Test Collections*, 1.

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

Le Luo and Li Li. 2014. Defining and evaluating classification algorithm for high-dimensional data based on latent topics. *PloS one*, 9(1):e82119.

Brian MacWhinney. 1992. The CHILDES project: Tools for analyzing talk. *Child Language Teaching and Therapy*, 8(2):217–218.

Brian MacWhinney, Davida Fromm, Margaret Forbes, and Audrey Holland. 2011. AphasiaBank: Methods for studying discourse. *Aphasiology*, 25(11):1286–1307.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Maria Yancheva and Frank Rudzicz. 2016. Vector-space topic models for detecting Alzheimer's disease. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2337–2346.