The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Document Informed Neural Autoregressive Topic Models with Distributional Prior

Pankaj Gupta,[1,2] Yatin Chaudhary,[1] Florian Buettner,[1] Hinrich Schütze[2]
[1]Corporate Technology, Machine-Intelligence (MIC-DE), Siemens AG Munich, Germany
[2]CIS, University of Munich (LMU) Munich, Germany
{pankaj.gupta, yatin.chaudhary, buettner.florian}@siemens.com
pankaj.gupta@campus.lmu.de — inquiries@cislmu.org

Abstract

We address two challenges in topic models: (1) Context information around words helps in determining their actual meaning, e.g., "networks" used in the contexts *artificial neural networks* vs. *biological neuron networks*. Generative topic models infer topic-word distributions, taking no or only little context into account. Here, we extend a neural autoregressive topic model to exploit the full context information around words in a document in a language modeling fashion. The proposed model is named as *iDocNADE*. (2) Due to the small number of word occurrences (i.e., lack of context) in short text and data sparsity in a corpus of few documents, the application of topic models is challenging on such texts. Therefore, we propose a simple and efficient way of incorporating external knowledge into neural autoregressive topic models: we use embeddings as a distributional prior. The proposed variants are named as *DocNADEe* and *iDocNADEe*.

We present novel neural autoregressive topic model variants that consistently outperform state-of-the-art generative topic models in terms of generalization, interpretability (topic coherence) and applicability (retrieval and classification) over 7 long-text and 8 short-text datasets from diverse domains.

Introduction

Probabilistic topic models, such as LDA (Blei, Ng, and Jordan 2003), Replicated Softmax (RSM) (Salakhutdinov and Hinton 2009) and Document Autoregressive Neural Distribution Estimator (DocNADE) (Larochelle and Lauly 2012) are often used to extract topics from text collections and learn document representations to perform NLP tasks such as information retrieval (IR), document classification or summarization.

To motivate our first task of *incorporating full contextual information*, assume that we conduct topic analysis on a collection of research papers from NIPS conference, where one of the popular terms is "networks". However, without context information (nearby and/or distant words), its actual meaning is ambiguous since it can refer to such different concepts as *artificial neural networks* in *computer science* or *biological neural networks* in *neuroscience* or *Computer/data networks* in *telecommunications*. Given the context, one can determine the actual meaning of "networks", for instance, "Extracting rules from artificial neural networks with distributed representations", or "Spikes from the presynaptic neurons and postsynaptic neurons in small networks" or "Studies of neurons or networks under noise in artificial neural networks" or "Packet Routing in Dynamically Changing Networks".

Generative topic models such as LDA or DocNADE infer topic-word distributions that can be used to estimate a document likelihood. While basic models such as LDA do not account for context information when inferring these distributions, more recent approaches such as DocNADE achieve *amplified word and document likelihoods* by accounting for words preceding a word of interest in a document. More specifically, DocNADE (Larochelle and Lauly 2012; Zheng, Zhang, and Larochelle 2016) (Figure 1, Left) is a probabilistic graphical model that learns topics over sequences of words, corresponding to a language model (Manning and Schütze 1999; Bengio et al. 2003) that can be interpreted as a neural network with several parallel hidden layers. To predict the word v_i, each hidden layer $\mathbf{h}_i$ takes as input the sequence of preceding words $\mathbf{v}_{<i}$. However, it does *not* take into account the following words $\mathbf{v}_{>i}$ in the sequence. Inspired by bidirectional language models (Mousa and Schuller 2017) and recurrent neural networks (Elman 1990; Gupta, Schütze, and Andrassy 2016; Vu et al. 2016b; 2016a), trained to predict a word (or label) depending on its full left and right contexts, we extend DocNADE and incorporate full contextual information (all words around v_i) at each hidden layer $\mathbf{h}_i$ when predicting the word v_i in a language modeling fashion with neural topic modeling.

While this is a powerful approach for incorporating contextual information in particular for long texts and corpora with many documents, learning contextual information remains challenging in topic models with short texts and few documents, due to (1) limited word co-occurrences or little context and (2) significant word non-overlap in such short texts. However, distributional word representations (i.e. word embeddings) have shown to capture both the semantic and syntactic relatedness in words and demonstrated impressive performance in natural language processing (NLP) tasks. For example, assume that we conduct topic analysis over the two short text fragments: "*Goldman shares drop sharply downgrade*" and "*Falling market homes*

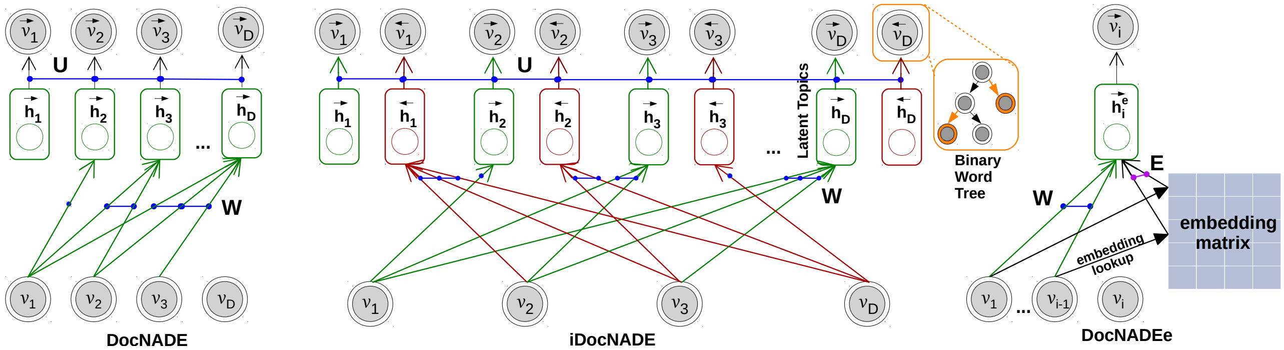


Figure 1: *DocNADE* (left), *iDocNADE* (middle) and DocNADEe (right) models. Blue colored lines signify the connections that share parameters. The observations (*double circle*) for each word v_i are multinomial. Hidden vectors in *green* and *red* colors identify the forward and backward network layers, respectively. Symbols $\overrightarrow{v}_i$ and $\overleftarrow{v}_i$ represent the autoregressive conditionals $p(v_i|\mathbf{v}_{<i})$ and $p(v_i|\mathbf{v}_{>i})$, respectively. Connections between each v_i and hidden units are shared, and each conditional $\overrightarrow{v}_i$ (or $\overleftarrow{v}_i$) is decomposed into a tree of binary logistic regressions, i.e. hierarchical softmax.

weaken economy". Traditional topic models will not be able to infer relatedness between word pairs across sentences such as (*economy*, *shares*) due to the lack of word-overlap between sentences. However, in embedding space, the word pairs (*economy*, *shares*), (*market*, *shares*) and (*falling*, *drop*) have cosine similarities of 0.65, 0.56 and 0.54.

Therefore, we *incorporate word embeddings* as fixed prior in neural topic models in order to introduce complementary information. The proposed neural architectures learn task specific word vectors in association with static embedding priors leading to better text representation for topic extraction, information retrieval, classification, etc.

The multi-fold **contributions** in this work are: **(1)** We propose an advancement in neural autoregressive topic model by incorporating full contextual information around words in a document to boost the likelihood of each word (and document). This enables learning better (*informed*) document representations that we quantify via *generalization* (perplexity), *interpretability* (topic coherence) and *applicability* (document retrieval and classification). We name the proposed topic model as *Document Informed Neural Autoregressive Distribution Estimator* (**iDocNADE**). **(2)** We propose a further extension of DocNADE-like models by incorporating complementary information via word embeddings, along with the standard sparse word representations (e.g., one-hot encoding). The resulting two DocNADE variants are named as *Document Neural Autoregressive Distribution Estimator with Embeddings* (**DocNADEe**) and *Document Informed Neural Autoregressive Distribution Estimator with Embeddings* (**iDocNADEe**). **(3)** We also investigate the two contributions above in the deep versions of topic models. **(4)** We apply our modeling approaches to 8 short-text and 7 long-text datasets from diverse domains. With the learned representations, we show a gain of 5.2% (404 vs 426) in perplexity, 11.1% (.60 vs .54) in precision at retrieval fraction 0.02 and 5.2% (.664 vs .631) in $F1$ for text categorization, compared to the DocNADE model (on average over 15 datasets). *Code* and *supplementary material* are available at `https://github.com/pgcool/iDocNADEe`.

Neural Autoregressive Topic Models

RSM (Salakhutdinov and Hinton 2009), a probabilistic undirected topic model, is a generalization of the energy-based Restricted Boltzmann Machines RBM (Hinton 2002) that can be used to model word counts. NADE (Larochelle and Murray 2011) decomposes the joint distribution of observations into autoregressive conditional distributions, modeled using non-linear functions. Unlike for RBM/RSM, this leads to tractable gradients of the data negative log-likelihood but can only be used to model binary observations.

DocNADE (Figure 1, Left) is a generative neural autoregressive topic model to account for word counts, inspired by RSM and NADE. For a document $\mathbf{v} = [v_1, ..., v_D]$ of size D, it models the joint distribution $p(\mathbf{v})$ of all words v_i, where $v_i \in \{1, ..., K\}$ is the index of the ith word in the dictionary of vocabulary size K. This is achieved by decomposing it as a product of conditional distributions i.e. $p(\mathbf{v}) = \prod_{i=1}^{D} p(v_i|\mathbf{v}_{<i})$ and computing each autoregressive conditional $p(v_i|\mathbf{v}_{<i})$ via a feed-forward neural network for $i \in \{1, ...D\}$,

$$\begin{aligned} \overrightarrow{\mathbf{h}}_i(\mathbf{v}_{<i}) &= g(\mathbf{c} + \textstyle\sum_{k<i} \mathbf{W}_{:,v_k}) \\ p(v_i = w|\mathbf{v}_{<i}) &= \frac{\exp(b_w + \mathbf{U}_{w,:}\overrightarrow{\mathbf{h}}_i(\mathbf{v}_{<i}))}{\sum_{w'} \exp(b_{w'} + \mathbf{U}_{w',:}\overrightarrow{\mathbf{h}}_i(\mathbf{v}_{<i}))} \end{aligned} \tag{1}$$

where $\mathbf{v}_{<i} \in \{v_1, ..., v_{i-1}\}$. $g(\cdot)$ is a non-linear activation function, $\mathbf{W} \in \mathbb{R}^{H \times K}$ and $\mathbf{U} \in \mathbb{R}^{K \times H}$ are weight matrices, $\mathbf{c} \in \mathbb{R}^H$ and $\mathbf{b} \in \mathbb{R}^K$ are bias parameter vectors. H is the number of hidden units (topics). $\mathbf{W}_{:,<i}$ is a matrix made of the $i-1$ first columns of $\mathbf{W}$. The probability of the word v_i is thus computed using a position-dependent hidden layer $\overrightarrow{\mathbf{h}}_i(\mathbf{v}_{<i})$ that learns a representation based on all previous words $\mathbf{v}_{<i}$; however it does *not* incorporate the following words $\mathbf{v}_{>i}$. Taken together, the log-likelihood of any document $\mathbf{v}$ of arbitrary length can be computed as:

$$\mathcal{L}^{DocNADE}(\mathbf{v}) = \sum_{i=1}^{D} \log p(v_i|\mathbf{v}_{<i}) \tag{2}$$

iDocNADE (Figure 1, Right), our *proposed* model, accounts for the full context information (both previous $\mathbf{v}_{<i}$ and following $\mathbf{v}_{>i}$ words) around each word v_i for a document $\mathbf{v}$. Therefore, the log-likelihood $\mathcal{L}^{iDocNADE}$ for a document $\mathbf{v}$ in *iDocNADE* is computed using forward and backward language models as:

$$\log p(\mathbf{v}) = \frac{1}{2} \sum_{i=1}^{D} \underbrace{\log p(v_i|\mathbf{v}_{<i})}_{\text{forward}} + \underbrace{\log p(v_i|\mathbf{v}_{>i})}_{\text{backward}} \tag{3}$$

i.e., the mean of the forward ($\overrightarrow{\mathcal{L}}$) and backward ($\overleftarrow{\mathcal{L}}$) log-likelihoods. This is achieved in a bi-directional language modeling and feed-forward fashion by computing position dependent *forward* ($\overrightarrow{\mathbf{h}}_i$) and *backward* ($\overleftarrow{\mathbf{h}}_i$) hidden layers for each word i, as:

$$\overrightarrow{\mathbf{h}}_i(\mathbf{v}_{<i}) = g(\overrightarrow{\mathbf{c}} + \sum_{k<i} \mathbf{W}_{:,v_k}) \tag{4}$$

$$\overleftarrow{\mathbf{h}}_i(\mathbf{v}_{>i}) = g(\overleftarrow{\mathbf{c}} + \sum_{k>i} \mathbf{W}_{:,v_k}) \tag{5}$$

where $\overrightarrow{\mathbf{c}} \in \mathbb{R}^H$ and $\overleftarrow{\mathbf{c}} \in \mathbb{R}^H$ are bias parameters in forward and backward passes, respectively. H is the number of hidden units (topics).

Two autoregressive conditionals are computed for each ith word using the forward and backward hidden vectors,

$$p(v_i = w|\mathbf{v}_{<i}) = \frac{\exp(\overrightarrow{b}_w + \mathbf{U}_{w,:}\overrightarrow{\mathbf{h}}_i(\mathbf{v}_{<i}))}{\sum_{w'} \exp(\overrightarrow{b}_{w'} + \mathbf{U}_{w',:}\overrightarrow{\mathbf{h}}_i(\mathbf{v}_{<i}))} \tag{6}$$

$$p(v_i = w|\mathbf{v}_{>i}) = \frac{\exp(\overleftarrow{b}_w + \mathbf{U}_{w,:}\overleftarrow{\mathbf{h}}_i(\mathbf{v}_{>i}))}{\sum_{w'} \exp(\overleftarrow{b}_{w'} + \mathbf{U}_{w',:}\overleftarrow{\mathbf{h}}_i(\mathbf{v}_{>i}))} \tag{7}$$

for $i \in [1, ..., D]$ where $\overrightarrow{\mathbf{b}} \in \mathbb{R}^K$ and $\overleftarrow{\mathbf{b}} \in \mathbb{R}^K$ are biases in forward and backward passes, respectively. Note that the parameters $\mathbf{W}$ and $\mathbf{U}$ are shared between the two networks.

DocNADEe and iDocNADEe with Embedding priors: We introduce additional semantic information for each word into DocNADE-like models via its pre-trained embedding vector, thereby enabling better textual representations and semantically more coherent topic distributions, in particular for short texts. In its simplest form, we extend DocNADE with word embedding aggregation at each autoregressive step k to generate a complementary textual representation, i.e., $\sum_{k<i} \mathbf{E}_{:,v_k}$. This mechanism utilizes prior knowledge encoded in a pre-trained embedding matrix $\mathbf{E} \in \mathbb{R}^{H\times K}$ when learning task-specific matrices $\mathbf{W}$ and latent representations in DocNADE-like models. The position dependent forward $\overrightarrow{\mathbf{h}}_i^e(\mathbf{v}_{<i})$ and (only in iDocNADEe) backward $\overleftarrow{\mathbf{h}}_i^e(\mathbf{v}_{>i})$ hidden layers for each word i now depend on $\mathbf{E}$ as:

$$\overrightarrow{\mathbf{h}}_i^e(\mathbf{v}_{<i}) = g(\overrightarrow{\mathbf{c}} + \sum_{k<i} \mathbf{W}_{:,v_k} + \lambda \sum_{k<i} \mathbf{E}_{:,v_k}) \tag{8}$$

$$\overleftarrow{\mathbf{h}}_i^e(\mathbf{v}_{>i}) = g(\overleftarrow{\mathbf{c}} + \sum_{k>i} \mathbf{W}_{:,v_k} + \lambda \sum_{k>i} \mathbf{E}_{:,v_k}) \tag{9}$$

where, λ is a mixture coefficient, determined using validation set. As in equations 6 and 7, the forward and backward autoregressive conditionals are computed via hidden vectors $\overrightarrow{\mathbf{h}}_i^e(\mathbf{v}_{<i})$ and $\overleftarrow{\mathbf{h}}_i^e(\mathbf{v}_{>i})$, respectively.

Algorithm 1 Computation of $\log p(\mathbf{v})$ in *iDocNADE* or *iDocNADEe* using *tree-softmax* or *full-softmax*

Input: A training document vector $\mathbf{v}$, Embedding matrix $\mathbf{E}$
Parameters: $\{\overrightarrow{\mathbf{b}}, \overleftarrow{\mathbf{b}}, \overrightarrow{\mathbf{c}}, \overleftarrow{\mathbf{c}}, \mathbf{W}, \mathbf{U}\}$
Output: $\log p(\mathbf{v})$
$\overrightarrow{\mathbf{a}} \leftarrow \overrightarrow{\mathbf{c}}$
if `iDocNADE` **then**
  $\overleftarrow{\mathbf{a}} \leftarrow \overleftarrow{\mathbf{c}} + \sum_{i>1} \mathbf{W}_{:,v_i}$
if `iDocNADEe` **then**
  $\overleftarrow{\mathbf{a}} \leftarrow \overleftarrow{\mathbf{c}} + \sum_{i>1} \mathbf{W}_{:,v_i} + \lambda \sum_{i>1} \mathbf{E}_{:,v_i}$
$q(\mathbf{v}) = 1$
for i from 1 to D **do**
  $\overrightarrow{\mathbf{h}}_i \leftarrow g(\overrightarrow{\mathbf{a}})$; $\overleftarrow{\mathbf{h}}_i \leftarrow g(\overleftarrow{\mathbf{a}})$
  if `tree-softmax` **then**
    $p(v_i|\mathbf{v}_{<i}) = 1$; $p(v_i|\mathbf{v}_{>i}) = 1$
    for m from 1 to $|\pi(v_i)|$ **do**
      $p(v_i|\mathbf{v}_{<i}) \leftarrow p(v_i|\mathbf{v}_{<i}) p(\pi(v_i)_m|\mathbf{v}_{<i})$
      $p(v_i|\mathbf{v}_{>i}) \leftarrow p(v_i|\mathbf{v}_{>i}) p(\pi(v_i)_m|\mathbf{v}_{>i})$
  if `full-softmax` **then**
    compute $p(v_i|\mathbf{v}_{<i})$ using equation 6
    compute $p(v_i|\mathbf{v}_{>i})$ using equation 7
  $q(\mathbf{v}) \leftarrow q(\mathbf{v}) p(v_i|\mathbf{v}_{<i}) p(v_i|\mathbf{v}_{>i})$
  if `iDocNADE` **then**
    $\overrightarrow{\mathbf{a}} \leftarrow \overrightarrow{\mathbf{a}} + \mathbf{W}_{:,v_i}$; $\overleftarrow{\mathbf{a}} \leftarrow \overleftarrow{\mathbf{a}} - \mathbf{W}_{:,v_i}$
  if `iDocNADEe` **then**
    $\overrightarrow{\mathbf{a}} \leftarrow \overrightarrow{\mathbf{a}} + \mathbf{W}_{:,v_i} + \lambda\, \mathbf{E}_{:,v_i}$
    $\overleftarrow{\mathbf{a}} \leftarrow \overleftarrow{\mathbf{a}} - \mathbf{W}_{:,v_i} - \lambda\, \mathbf{E}_{:,v_i}$
$\log p(\mathbf{v}) \leftarrow \frac{1}{2} \log q(\mathbf{v})$

Deep DocNADEs with/without Embedding Priors: DocNADE can be extended to a deep, multiple hidden layer architecture by adding new hidden layers as in a regular deep feed-forward neural network, allowing for improved performance (Lauly et al. 2017). In this deep version of DocNADE variants, the first hidden layers are computed in an analogous fashion to iDocNADE (eq. 4 and 5). Subsequent hidden layers are computed as:

$$\overrightarrow{\mathbf{h}}_i^{(d)}(\mathbf{v}_{<i}) = g(\overrightarrow{\mathbf{c}}^{(d)} + \mathbf{W}^{(d)} \cdot \overrightarrow{\mathbf{h}}_i^{(d-1)}(\mathbf{v}_{<i}))$$

and similarly, $\overleftarrow{\mathbf{h}}_i^{(d)}(\mathbf{v}_{>i})$ for $d = 2, ..., n$, where n is the total number of hidden layers. The exponent "(d)" is used as an index over the hidden layers and parameters in the deep feed-forward network. Forward and/or backward conditionals for each word i are modeled using the forward and backward hidden vectors at the last layer n. The deep DocNADE or iDocNADE variants without or with embeddings are named as *DeepDNE*, *iDeepDNE*, *DeepDNEe* and *iDeepDNEe*, respectively where $\mathbf{W}^{(1)}$ is the word representation matrix. However in *DeepDNEe* (or *iDeepDNEe*), we introduce embedding prior $\mathbf{E}$ in the first hidden layer, i.e.,

$$\overrightarrow{\mathbf{h}}_i^{e,(1)} = g(\overrightarrow{\mathbf{c}}^{(1)} + \sum_{k<i} \mathbf{W}_{:,v_k}^{(1)} + \lambda \sum_{k<i} \mathbf{E}_{:,v_k})$$

for each word i via embedding aggregation of its context $\mathbf{v}_{<i}$ (and $\mathbf{v}_{>i}$). Similarly, we compute $\overleftarrow{\mathbf{h}}_i^{e,(1)}$.

Learning: Similar to DocNADE, the conditionals $p(v_i = w|\mathbf{v}_{<i})$ and $p(v_i = w|\mathbf{v}_{>i})$ in DocNADEe, iDocNADE or iDocNADEe are computed by a neural network for each word v_i, allowing efficient learning of *informed* representations $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ (or $\overrightarrow{\mathbf{h}}_i^e(\mathbf{v}_{<i})$ and $\overleftarrow{\mathbf{h}}_i^e(\mathbf{v}_{>i})$), as it consists simply of a linear transformation followed by a non-linearity. Observe that the weight $\mathbf{W}$ (or prior embedding matrix $\mathbf{E}$) is the same across all conditionals and ties contextual observables (blue colored lines in Figure 1) by computing each $\overrightarrow{\mathbf{h}}_i$ or $\overleftarrow{\mathbf{h}}_i$ (or $\overrightarrow{\mathbf{h}}_i^e(\mathbf{v}_{<i})$ and $\overleftarrow{\mathbf{h}}_i^e(\mathbf{v}_{>i})$).

Binary word tree (`tree-softmax`) to compute conditionals: To compute the likelihood of a document, the autoregressive conditionals $p(v_i = w|\mathbf{v}_{<i})$ and $p(v_i = w|\mathbf{v}_{>i})$ have to be computed for each word $i \in [1, 2, ...D]$, requiring time linear in vocabulary size K. To reduce computational cost and achieve a complexity logarithmic in K we follow Larochelle and Lauly (2012) and decompose the computation of the conditionals using a probabilistic tree. All words in the documents are randomly assigned to a different leaf in a binary tree and the probability of a word is computed as the probability of reaching its associated leaf from the root. Each left/right transition probability is modeled using a binary logistic regressor with the hidden layer $\overrightarrow{\mathbf{h}}_i$ or $\overleftarrow{\mathbf{h}}_i$ ($\overrightarrow{\mathbf{h}}_i^e$ or $\overleftarrow{\mathbf{h}}_i^e$) as its input. In the binary tree, the probability of a given word is computed by multiplying each of the left/right transition probabilities along the tree path.

Algorithm 1 shows the computation of $\log p(\mathbf{v})$ using *iDocNADE* (or *iDocNADEe*) structure, where the autogressive conditionals (lines 14 and 15) for each word v_i are obtained from the forward and backward networks and modeled into a binary word tree, where $\pi(v_i)$ denotes the sequence of binary left/right choices at the internal nodes along the tree path and $\mathbf{l}(v_i)$ the sequence of tree nodes on that tree path. For instance, $l(v_i)_1$ will always be the root of the binary tree and $\pi(v_i)_1$ will be 0 if the word leaf v_i is in the left subtree or 1 otherwise. Therefore, each of the forward and backward conditionals are computed as:

$$p(v_i = w|\mathbf{v}_{<i}) = \prod_{m=1}^{|\pi(v_i)|} p(\pi(v_i)_m|\mathbf{v}_{<i})$$

$$p(v_i = w|\mathbf{v}_{>i}) = \prod_{m=1}^{|\pi(v_i)|} p(\pi(v_i)_m|\mathbf{v}_{>i})$$

$$p(\pi(v_i)_m|\mathbf{v}_{<i}) = g(\overrightarrow{b}_{l(v_i)_m} + \mathbf{U}_{l(v_i)_m,:}\overrightarrow{\mathbf{h}}(\mathbf{v}_{<i}))$$

$$p(\pi(v_i)_m|\mathbf{v}_{>i}) = g(\overleftarrow{b}_{l(v_i)_m} + \mathbf{U}_{l(v_i)_m,:}\overleftarrow{\mathbf{h}}(\mathbf{v}_{>i}))$$

where $\mathbf{U} \in \mathbb{R}^{T\times H}$ is the matrix of logistic regressions weights, T is the number of internal nodes in binary tree, and $\overrightarrow{\mathbf{b}}$ and $\overleftarrow{\mathbf{b}}$ are bias vectors.

Each of the forward and backward conditionals $p(v_i = w|\mathbf{v}_{<i})$ or $p(v_i = w|\mathbf{v}_{>i})$ requires the computation of its own hidden layers $\overrightarrow{\mathbf{h}}_i(\mathbf{v}_{<i})$ and $\overleftarrow{\mathbf{h}}_i(\mathbf{v}_{>i})$ (or $\overrightarrow{\mathbf{h}}_i^e(\mathbf{v}_{<i})$ and $\overleftarrow{\mathbf{h}}_i^e(\mathbf{v}_{>i})$), respectively. With H being the size of each hidden layer and D the number of words in $\mathbf{v}$, computing a single layer requires $O(HD)$, and since there are D hidden layers to compute, a naive approach for computing all hidden layers would be in $O(D^2H)$. However, since the weights in the matrix $\mathbf{W}$ are tied, the linear activations $\overrightarrow{\mathbf{a}}$ and $\overleftarrow{\mathbf{a}}$ (algorithm 1) can be re-used in every hidden layer and computational complexity reduces to $O(HD)$.

Algorithm 2 Computing gradients of $-\log p(\mathbf{v})$ in *iDocNADE* or *iDocNADEe* using *tree-softmax*

Input: A training document vector $\mathbf{v}$
Parameters: $\{\overrightarrow{\mathbf{b}}, \overleftarrow{\mathbf{b}}, \overrightarrow{\mathbf{c}}, \overleftarrow{\mathbf{c}}, \mathbf{W}, \mathbf{U}\}$
Output: $\delta\overrightarrow{\mathbf{b}}, \delta\overleftarrow{\mathbf{b}}, \delta\overrightarrow{\mathbf{c}}, \delta\overleftarrow{\mathbf{c}}, \delta\mathbf{W}, \delta\mathbf{U}$

$\overrightarrow{\mathbf{a}} \leftarrow 0$; $\overleftarrow{\mathbf{a}} \leftarrow 0$; $\overrightarrow{\mathbf{c}} \leftarrow 0$; $\overleftarrow{\mathbf{c}} \leftarrow 0$; $\overrightarrow{\mathbf{b}} \leftarrow 0$; $\overleftarrow{\mathbf{b}} \leftarrow 0$
for i from D to 1 **do**
  $\delta\overrightarrow{\mathbf{h}}_i \leftarrow 0$; $\delta\overleftarrow{\mathbf{h}}_i \leftarrow 0$
  for m from 1 to $|\pi(v_i)|$ **do**
    $\overrightarrow{b}_{l(v_i)_m} \leftarrow \overrightarrow{b}_{l(v_i)_m} + (p(\pi(v_i)_m|\mathbf{v}_{<i}) - \pi(v_i)_m)$
    $\overleftarrow{b}_{l(v_i)_m} \leftarrow \overleftarrow{b}_{l(v_i)_m} + (p(\pi(v_i)_m|\mathbf{v}_{>i}) - \pi(v_i)_m)$
    $\delta\overrightarrow{\mathbf{h}}_i \leftarrow \delta\overrightarrow{\mathbf{h}}_i + (p(\pi(v_i)_m|\mathbf{v}_{<i}) - \pi(v_i)_m)\mathbf{U}_{l(v_i)_m,:}$
    $\delta\overleftarrow{\mathbf{h}}_i \leftarrow \delta\overleftarrow{\mathbf{h}}_i + (p(\pi(v_i)_m|\mathbf{v}_{>i}) - \pi(v_i)_m)\mathbf{U}_{l(v_i)_m,:}$
    $\delta\mathbf{U}_{l(v_i)_m} \leftarrow \delta\mathbf{U}_{l(v_i)_m} + (p(\pi(v_i)_m|\mathbf{v}_{<i}) - \pi(v_i)_m)\overrightarrow{\mathbf{h}}_i^T + (p(\pi(v_i)_m|\mathbf{v}_{>i}) - \pi(v_i)_m)\overleftarrow{\mathbf{h}}_i^T$
  $\delta\overrightarrow{\mathbf{g}} \leftarrow \overrightarrow{\mathbf{h}}_i \circ (1 - \overrightarrow{\mathbf{h}}_i)$ # for sigmoid activation
  $\delta\overleftarrow{\mathbf{g}} \leftarrow \overleftarrow{\mathbf{h}}_i \circ (1 - \overleftarrow{\mathbf{h}}_i)$ # for sigmoid activation
  $\delta\overrightarrow{\mathbf{c}} \leftarrow \delta\overrightarrow{\mathbf{c}} + \delta\overrightarrow{\mathbf{h}}_i \circ \delta\overrightarrow{\mathbf{g}}$; $\delta\overleftarrow{\mathbf{c}} \leftarrow \delta\overleftarrow{\mathbf{c}} + \delta\overleftarrow{\mathbf{h}}_i \circ \delta\overleftarrow{\mathbf{g}}$
  $\delta\mathbf{W}_{:,v_i} \leftarrow \delta\mathbf{W}_{:,v_i} + \delta\overrightarrow{\mathbf{a}} + \delta\overleftarrow{\mathbf{a}}$
  $\delta\overrightarrow{\mathbf{a}} \leftarrow \delta\overrightarrow{\mathbf{a}} + \delta\overrightarrow{\mathbf{h}}_i \circ \delta\overrightarrow{\mathbf{g}}$; $\delta\overleftarrow{\mathbf{a}} \leftarrow \delta\overleftarrow{\mathbf{a}} + \delta\overleftarrow{\mathbf{h}}_i \circ \delta\overleftarrow{\mathbf{g}}$

With the trained *iDocNADEe* (or *DocNADE* variants), the representation ($\overleftrightarrow{\mathfrak{h}}^e \in \mathbb{R}^H$) for a new document $\mathbf{v}^*$ of size D^* is extracted by summing the hidden representations from the forward and backward networks to account for the context information around each word in the words' sequence, as

$$\overrightarrow{\mathbf{h}}^e(\mathbf{v}^*) = g(\overrightarrow{\mathbf{c}} + \sum_{k\leqslant D^*} \mathbf{W}_{:,v_k^*} + \lambda \sum_{k\leqslant D^*} \mathbf{E}_{:,v_k^*}) \quad (10)$$

$$\overleftarrow{\mathbf{h}}^e(\mathbf{v}^*) = g(\overleftarrow{\mathbf{c}} + \sum_{k\geqslant 1} \mathbf{W}_{:,v_k^*} + \lambda \sum_{k\geqslant 1} \mathbf{E}_{:,v_k^*}) \quad (11)$$

$$\text{Therefore; } \overleftrightarrow{\mathfrak{h}}^e = \overrightarrow{\mathbf{h}}^e(\mathbf{v}^*) + \overleftarrow{\mathbf{h}}^e(\mathbf{v}^*) \quad (12)$$

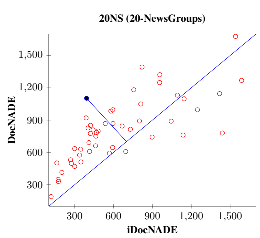
The DocNADE variants without embeddings compute the representation $\overleftrightarrow{\mathfrak{h}}$ excluding the embedding term $\mathbf{E}$. Parameters $\{\overrightarrow{\mathbf{b}}, \overleftarrow{\mathbf{b}}, \overrightarrow{\mathbf{c}}, \overleftarrow{\mathbf{c}}, \mathbf{W}, \mathbf{U}\}$ are learned by minimizing the average negative log-likelihood of the training documents using stochastic gradient descent (algorithm 2). In our proposed formulation of iDocNADE or its variants (Figure 1), we perform inference by computing $\mathcal{L}^{iDocNADE}(\mathbf{v})$ (Eq.3).

Evaluation

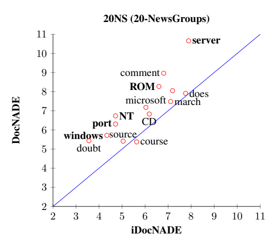
We perform evaluations on 15 (8 short-text and 7 long-text) datasets of varying size with single/multi-class labeled documents from public as well as industrial corpora. See the *supplementary material* for the data description, hyper-parameters and grid-search results for generalization and IR tasks. Table 1 shows the data statistics, where 20NS: 20NewsGroups and R21578: Reuters21578. Since, Gupta

Data	Train	Val	Test	K	L	C	Domain	Tree-Softmax(TS)				Full-Softmax (FS)							
								DocNADE		**iDocNADE**		**DocNADE**		**iDocNADE**		**DocNADEe**		**iDocNADEe**	
								PPL	IR	PPL	IR	PPL	IR	PPL	IR	PPL	IR	PPL	IR
20NSshort	1.3k	0.1k	0.5k	2k	13.5	20	News	894	.23	880	.30	646	.25	639	.26	638	.28	633	**.28**
TREC6	5.5k	0.5k	0.5k	2k	9.8	6	Q&A	42	.48	39	.55	64	.54	61	.56	62	.56	60	**.57**
R21578title†	7.3k	0.5k	3.0k	2k	7.3	90	News	298	.61	239	.63	193	.61	181	.62	179	.65	176	**.66**
Subjectivity	8.0k	.05k	2.0k	2k	23.1	2	Senti	303	.78	287	.81	371	.77	365	.80	362	.80	361	**.81**
Polarity	8.5k	.05k	2.1k	2k	21.0	2	Senti	311	.51	292	.54	358	.54	345	.56	341	.56	340	**.57**
TMNtitle	22.8k	2.0k	7.8k	2k	4.9	7	News	863	.57	823	.59	711	.44	670	.46	668	.54	664	**.55**
TMN	22.8k	2.0k	7.8k	2k	19	7	News	548	.64	536	.66	592	.60	560	.64	563	.64	561	**.66**
AGnewstitle	118k	2.0k	7.6k	5k	6.8	4	News	811	.59	793	.65	545	.62	516	.64	516	.66	514	**.68**
Avg (short)								509	.55	486	.59	435	.54	417	.57	416	.58	413	**.60**
20NSsmall	0.4k	0.2k	0.2k	2k	187	20	News	-	-	-	-	628	.30	592	.32	607	**.33**	590	**.33**
Reuters8	5.0k	0.5k	2.2k	2k	102	8	News	172	.88	152	.89	184	.83	178	.88	178	**.87**	178	**.87**
20NS	8.9k	2.2k	7.4k	2k	229	20	News	830	.27	812	.33	474	.20	463	.24	464	**.25**	463	**.25**
R21578†	7.3k	0.5k	3.0k	2k	128	90	News	215	.70	179	.74	297	.70	285	**.73**	286	.71	285	.72
RCV1V2†	23.0k	.05k	10.0k	2k	123	103	News	381	.81	364	.86	479	.86	463	**.89**	465	.87	462	.88
SiROBs†	27.0k	1.0k	10.5k	3k	39	22	Industry	398	.31	351	.35	399	.34	340	.34	343	**.37**	340	.36
AGNews	118k	2.0k	7.6k	5k	38	4	News	471	.72	441	.77	451	.71	439	.78	433	.76	438	**.79**
Avg (long)								417	.61	383	.65	416	.56	394	**.60**	396	**.60**	393	**.60**
Avg (all)								469	.57	442	.62	426	.54	406	.58	407	.59	404	**.60**

Table 1: *Data statistics* of short and long texts as well as small and large corpora from various domains. *State-of-the-art* comparison in terms of PPL and IR (i.e, IR-precision) for **short** and **long** text datasets. The symbols are- L: average text length in number of words, K:dictionary size, C: number of classes, Senti: Sentiment, Avg: average, 'k':thousand and †: multi-label data. PPL and IR (IR-precision) are computed over 200 ($T200$) topics at retrieval fraction = 0.02. For short-text, $L < 25$. The *underline* and **bold** numbers indicate the best scores in PPL and retrieval task, respectively in FS setting. See Larochelle and Lauly (2012) for LDA (Blei, Ng, and Jordan 2003) performance in terms of PPL, where DocNADE outperforms LDA.



(a) PPL: 20NS



(b) NLL of 14 Words

Figure 2: (a) PPL (T200) by iDocNADE and DocNADE for each of the 50 held-out documents of 20NS. The *filled circle* points to the document for which *PPL* differs by maximum. (b) NLL of each of the words in the document marked by the *filled circle* in (a), due to iDocNADE and DocNADE.

et al. (2018a) have shown that DocNADE outperforms gaussian-LDA (Das, Zaheer, and Dyer 2015), glove-LDA and glove-DMM (Nguyen et al. 2015) in terms of topic coherence, text retrieval and classification, therefore we adopt DocNADE as the strong *baseline*. We use the development (dev) sets of each of the datasets to perform a grid-search on mixture weights, $\lambda = [0.1, 0.5, 1.0]$.

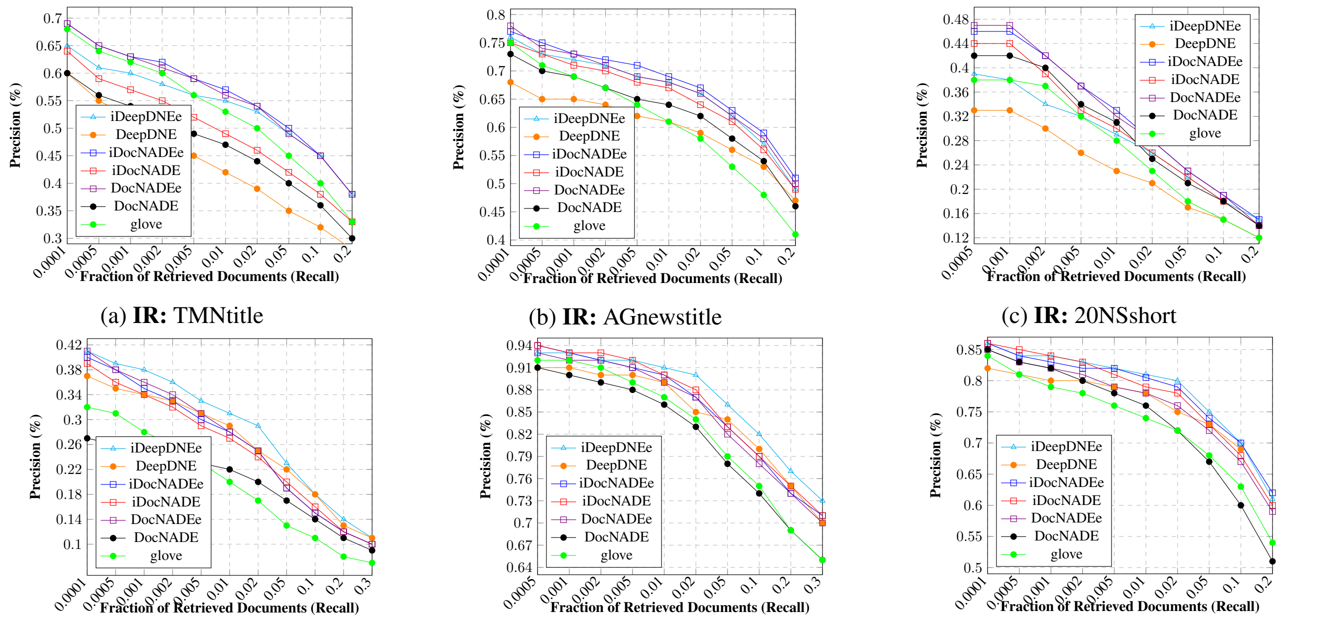
Generalization (Perplexity, PPL) We evaluate the topic models' generative performance as a generative model of documents by estimating log-probability for the test documents. During training, we initialize the proposed DocNADE extensions with DocNADE, i.e., $\mathbf{W}$ matrix. A comparison is made with the *baselines* (DocNADE and DeepDNE) and proposed variants (iDocNADE, DocNADEe, iDocNADEe, iDeepDNE, DeepDNEe and iDeepDNEe) using 50 (in *supplementary*) and 200 (T200) topics, set by the hidden layer size H.

Quantitative: Table 1 shows the average held-out perplexity (PPL) per word as, $PPL = \exp\big(-\frac{1}{N}\sum_{t=1}^{N}\frac{1}{|\mathbf{v}^t|}\log p(\mathbf{v}^t)\big)$ where N and $|\mathbf{v}^t|$ are the total number of documents and words in a document $\mathbf{v}^t$. To compute PPL, the log-likelihood of the document $\mathbf{v}^t$, i.e., $\log p(\mathbf{v}^t)$, is obtained by $\mathcal{L}^{DocNADE}$ (eqn. 2) in the DocNADE (forward only) variants, while we average PPL scores from the forward and backward networks of the iDocNADE variants.

Table 1 shows that the proposed models achieve lower perplexity for both the short-text (413 vs 435) and long-text (393 vs 416) datasets than *baseline* DocNADE with full-softmax (or tree-softmax). In total, we show a gain of 5.2% (404 vs 426) in PPL score on an average over the 15 datasets.

Table 2 illustrates the generalization performance of deep variants, where the proposed extensions outperform the DeepDNE for both short-text and long-text datasets. We report a gain of 10.7% (402 vs 450) in PPL due to iDeepDNEe over the baseline DeepDNE, on an average over 11 datasets.

Inspection: We quantify the use of context information in learning informed document representations. For 20NS dataset, we randomly select 50 held-out documents from its test set and compare (Figure 2a) the *PPL* for each of the held-out documents under the learned 200-dimensional DocNADE and iDocNADE. Observe that iDocNADE achieves lower *PPL* for the majority of the documents. The *filled* circle(s) points to the document for which



(a) **IR:** TMNtitle (b) **IR:** AGnewstitle (c) **IR:** 20NSshort

(d) **IR:** 20NS (e) **IR:** Reuters8 (f) **IR:** AGnews

Figure 3: Document retrieval performance (IR-precision) on 3 short-text and 3 long-text datasets at different retrieval fractions

PPL differs by a maximum between iDocNADE and DocNADE. We select the corresponding document and compute the negative log-likelihood (*NLL*) for every word. Figure 2b shows that the *NLL* for the majority of the words is lower (better) in iDocNADE than DocNADE. See the *supplementary material* for the raw text of the selected documents.

Interpretability (Topic Coherence) Beyond PPL, we compute topic coherence (Chang et al. 2009; Newman, Karimi, and Cavedon 2009; Das, Zaheer, and Dyer 2015; Gupta et al. 2018b) to assess the meaningfulness of the underlying topics captured. We choose the coherence measure proposed by Röder, Both, and Hinneburg (2015) that identifies context features for each topic word using a sliding window over the reference corpus. The higher scores imply more coherent topics.

Quantitative: We use gensim module (*coherence type* = c_v) to estimate coherence for each of the 200 topics (top 10 and 20 words). Table 3 shows average coherence over 200 topics using short-text and long-text datasets, where the high scores for long-text in iDocNADE (.636 vs .602) suggest that the contextual information helps in generating more coherent topics than DocNADE. On top, the introduction of embeddings, i.e., iDocNADEe for short-text boosts (.847 vs .839) topic coherence. **Qualitative:** Table 5 illustrates example topics each with a coherence score.

Applicability (Document Retrieval) To evaluate the quality of the learned representations, we perform a document retrieval task using the 15 datasets and their label information. We use the experimental setup similar to Lauly et al. (2017), where all test documents are treated as queries to retrieve a fraction of the closest documents in the original training set using cosine similarity measure between their representations (eqn. 12 in iDocNADE and $\overrightarrow{\mathbf{h}}_D$ in DocNADE). To compute retrieval precision for each fraction (e.g., 0.0001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, etc.), we average the number of retrieved training documents with the same label as the query. For multi-label datasets, we average the precision scores over multiple labels for each query. Since Salakhutdinov and Hinton (2009) and Lauly et al. (2017) showed that RSM and DocNADE strictly outperform LDA on this task, we only compare DocNADE and its proposed extensions.

Table 1 shows the IR-precision scores at retrieval fraction 0.02. Observe that the introduction of both pre-trained embedding priors and contextual information leads to im-

data	DeepDNE		iDeepDNE		DeepDNEe		iDeepDNEe	
	PPL	IR	PPL	IR	PPL	IR	PPL	IR
20NSshort	917	.21	841	.22	827	.25	830	**.26**
TREC6	114	.50	69	.52	69	**.55**	68	**.55**
R21578title	253	.50	231	.52	236	**.63**	230	.61
Subjectivity	428	.77	393	.77	392	.81	392	**.82**
Polarity	408	.51	385	.51	383	**.55**	387	.53
TMN	681	.60	624	.62	627	.63	623	**.66**
Avg (short)	467	.51	424	.53	422	**.57**	421	**.57**
Reuters8	216	.85	192	.89	191	.88	191	**.90**
20NS	551	.25	504	.28	504	**.29**	506	**.29**
R21578	318	.71	299	**.73**	297	.72	298	**.73**
AGNews	572	.75	441	.77	441	.75	440	**.80**
RCV1V2	489	.86	464	.88	466	**.89**	462	**.89**
Avg (long)	429	.68	380	.71	379	.71	379	**.72**
Avg (all)	450	.59	404	.61	403	.63	402	**.64**

Table 2: Deep Variants (+ Full-softmax) with T200: PPL and IR (i.e, IR-precision) for **short** and **long** text datasets.

model	DocNADE		iDocNADE		DocNADEe		iDocNADEe	
	W10	W20	W10	W20	W10	W20	W10	W20
20NSshort	.744	.849	.748	.852	.747	.851	.744	.849
TREC6	.746	.860	.748	.864	.753	.858	.752	.866
R21578title	.742	.845	.748	.855	.749	.859	.746	.856
Polarity	.730	.833	.732	.837	.734	.839	.738	.841
TMNtitle	.738	.840	.744	.848	.746	.850	.746	.850
TMN	.709	.811	.713	.814	.717	.818	.721	.822
Avg (short)	.734	.839	.739	.845	**.742**	.846	.741	**.847**
20NSsmall	.515	.629	.564	.669	.533	.641	.549	.661
Reuters8	.578	.665	.564	.657	.574	.655	.554	.641
20NS	.417	.496	.453	.531	.385	.458	.417	.490
R21578	.540	.570	.548	.640	.542	.596	.551	.663
AGnews	.718	.828	.721	.840	.677	.739	.696	.760
RCV1V2	.383	.426	.428	.480	.364	.392	.420	.463
Avg (long)	.525	.602	**.546**	**.636**	.513	.580	.531	.613

Table 3: Topic coherence with the top 10 (W10) and 20 (W20) words from topic models (T200). Since, (Gupta et al. 2018a) have shown that DocNADE outperforms both glove-DMM and glove-LDA, therefore DocNADE as the baseline.

data	glove		doc2vec		DocNADE		DocNADEe		iDocNADE		iDocNADEe	
	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc
20NSshort	.493	.520	.413	.457	.428	.474	.473	.529	.456	.491	.518	.535
TREC6	.798	.810	.400	.512	.804	.822	.854	.856	.808	.812	.842	.844
R21578title	.356	.695	.176	.505	.318	.653	.352	.693	.302	.665	.335	.700
Subjectivity	.882	.882	.763	.763	.872	.872	.886	.886	.871	.871	.886	.886
Polarity	.715	.715	.624	.624	.693	.693	.712	.712	.688	.688	.714	.714
TMNtitle	.693	.727	.582	.617	.624	.667	.697	.732	.632	.675	.696	.731
TMN	.736	.755	.720	.751	.740	.778	.765	.801	.751	.790	.771	.805
AGnewstitle	.814	.815	.513	.515	.812	.812	.829	.828	.819	.818	.829	.828
Avg (short)	.685	.739	.523	.593	.661	.721	.696	.755	.666	.726	**.700**	**.756**
Reuters8	.830	.950	.937	.852	.753	.931	.848	.956	.836	.957	.860	.960
20NS	.509	.525	.396	.409	.512	.535	.514	.540	.524	.548	.523	.544
R21578	.316	.703	.215	.622	.324	.716	.322	.721	.350	.710	.300	.722
AGnews	.870	.871	.713	.711	.873	.876	.880	.880	.880	.880	.886	.886
RCV1V2	.442	.368	.442	.341	.461	.438	.460	.457	.463	.452	.465	.454
Avg (long)	.593	.683	.540	.587	.584	.699	.605	.711	**.611**	.710	.607	**.713**
Avg (all)	.650	.718	.530	.590	.631	.712	.661	.738	.645	.720	**.664**	**.740**

Table 4: Text classification for short and long texts with T200 or word embedding dimension (Topic models with FS)

proved performance on the IR task for short-text and long-text datasets. We report a gain of 11.1% (.60 vs .54) in precision on an average over the 15 datasets, compared to DocNADE. On top, the deep variant i.e. iDeepDNEe (Table 2) demonstrates a gain of 8.5% (.64 vs .59) in precision over the 11 datasets, compared to DeepDNE. Figures (3a, 3b, 3c) and (3d, 3e and 3f) illustrate the average precision for the retrieval task on short-text and long-text datasets, respectively.

Applicability (Text Categorization) Beyond the document retrieval, we perform text categorization to measure the quality of word vectors learned in the topic models. We consider the same experimental setup as in the document retrieval task and extract the document representation (latent vector) of 200 dimension for each document (or text), learned during the training of DocNADE variants. To perform document categorization, we employ a logistic regression classifier with $L2$ regularization. We also compute document representations from pre-trained glove (Pennington, Socher, and Manning 2014) embedding matrix by summing the word vectors and compute classification performance. On top, we also extract document representation from doc2vec (Le and Mikolov 2014).

DocNADE	iDocNADE	DocNADEe
beliefs, muslims, forward, alt, islam, towards, atheism, christianity, hands, opinions	scripture, atheists, sin, religions, christianity, lord, bible, msg, heaven, jesus	atheists, christianity, belief, eternal, atheism, catholic, bible, arguments, islam, religions
0.44	0.46	<u>0.52</u>

Table 5: Topics (top 10 words) of 20NS with coherence

book			jesus			windows			gun		
neighbors	s_i	s_g	*neighbors*	s_i	s_g	*neighbors*	s_i	s_g	*neighbors*	s_i	s_g
books	.61	.84	christ	.86	.83	dos	.74	.34	guns	.72	.79
reference	.52	.51	god	.78	.63	files	.63	.36	firearms	.63	.63
published	.46	.74	christians	.74	.49	version	.59	.43	criminal	.63	.33
reading	.45	.54	faith	.71	.51	file	.59	.36	crime	.62	.42
author	.44	.77	bible	.71	.51	unix	.52	.47	police	.61	.43

Table 6: 20NS dataset: The five nearest neighbors by iDocNADE. s_i: Cosine similarity between the word vectors from iDocNADE, for instance vectors of *jesus* and *god*. s_g: Cosine similarity in embedding vectors from glove.

Table 4 shows that *glove* leads DocNADE in classification performance, suggesting a need for distributional priors. For short-text dataset, iDocNADEe (and DocNADEe) outperforms *glove* (.700 vs .685) and DocNADE (.700 vs .661) in F1. Overall, we report a gain of 5.2% (.664 vs .631) in F1 due to iDocNADEe over DocNADE for classification on an average over 13 datasets.

Inspection of Learned Representations: To analyze the meaningful semantics captured, we perform a qualitative inspection of the learned representations by the topic models. Table 5 shows topics for 20NS dataset that could be interpreted as *religion*, which are (sub)categories in the data, confirming that meaningful topics are captured. Observe that DocNADEe extracts a more coherent topic.

For word level inspection, we extract *word representations* using the columns $W_{:,v_i}$ as the vector (200 dimension) representation of each word v_i, learned by iDocNADE using 20NS dataset. Table 6 shows the five nearest neighbors of some selected words in this space and their corresponding similarity scores. We also compare similarity in word vectors from iDocNADE and glove embeddings, confirming that meaningful word representations are learned.

Conclusion

We show that leveraging contextual information and introducing distributional priors via pre-trained word embeddings in our proposed topic models result in learning better word/document representation for short and long documents, and improve generalization, interpretability of topics and their applicability in text retrieval and classification.

References

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

Blei, D.; Ng, A.; and Jordan, M. 2003. Latent dirichlet allocation. 993–1022.

Chang, J.; Boyd-Graber, J.; Wang, C.; Gerrish, S.; and Blei., D. M. 2009. Reading tea leaves: How humans interpret topic models. In *In Neural Information Processing Systems (NIPS)*.

Das, R.; Zaheer, M.; and Dyer, C. 2015. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics.

Elman, J. L. 1990. Finding structure in time. *Cognitive science* 14(2):179–211.

Gupta, P.; Chaudhary, Y.; Buettner, F.; and Schütze, H. 2018a. texttovec: Deep contextualized neural autoregressive models of language with distributed compositional prior. In *Preprint arxiv*.

Gupta, P.; Rajaram, S.; Schütze, H.; and Andrassy, B. 2018b. Deep temporal-recurrent-replicated-softmax for topical trends over time. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, 1079–1089. New Orleans, USA: Association of Computational Linguistics.

Gupta, P.; Schütze, H.; and Andrassy, B. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2537—2547.

Hinton, G. E. 2002. Training products of experts by minimizing contrastive divergence. In *Neural Computation*, 1771–1800.

Larochelle, H., and Lauly, S. 2012. A neural autoregressive topic model. In *Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012)*. NIPS.

Larochelle, H., and Murray, I. 2011. The neural autoregressive distribution estimato. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, 29—37. JMLR.

Lauly, S.; Zheng, Y.; Allauzen, A.; and Larochelle, H. 2017. Document neural autoregressive distribution estimation. *Journal of Machine Learning Research* 18(113):1–24.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. 1188–1196.

Manning, C. D., and Schütze, H. 1999. Foundations of statistical natural language processing. Cambridge MA: The MIT Press.

Mousa, A. E.-D., and Schuller, B. 2017. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 1023—1032. Association for Computational Linguistics.

Newman, D.; Karimi, S.; and Cavedon, L. 2009. External evaluation of topic models. In *Proceedings of the 14th Australasian Document Computing Symposium*.

Nguyen, D. Q.; Billingsley, R.; Du, L.; and Johnson, M. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics* 3:299–313.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1532–1543.

Röder, M.; Both, A.; and Hinneburg, A. 2015. Exploring the space of topic coherence measures. In *Proceedings of the WSDM*. ACM.

Salakhutdinov, R., and Hinton, G. 2009. Replicated softmax: an undirected topic model. In *Proceedings of the Advances in Neural Information Processing Systems 22 (NIPS 2009)*, 1607–1614. NIPS.

Vu, N. T.; Adel, H.; Gupta, P.; and Schütze, H. 2016a. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 534–539. San Diego, California USA: Association for Computational Linguistics.

Vu, N. T.; Gupta, P.; Adel, H.; and Schütze, H. 2016b. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *Proceedings of IEEE/ACM Trans. on Audio, Speech, and Language Processing (ICASSP)*. IEEE.

Zheng, Y.; Zhang, Y.-J.; and Larochelle, H. 2016. A deep and autoregressive approach for topic modeling of multimodal data. In *IEEE transactions on pattern analysis and machine intelligence*, 1056–1069. IEEE.