**textTOvec:** DEEP CONTEXTUALIZED NEURAL AUTOREGRESSIVE TOPIC MODELS OF LANGUAGE WITH DISTRIBUTED COMPOSITIONAL PRIOR

Pankaj Gupta1,2, Yatin Chaudhary1, Florian Buettner1, Hinrich Schütze2

1Corporate Technology, Machine-Intelligence (MIC-DE), Siemens AG Munich, Germany
2CIS, University of Munich (LMU) Munich, Germany
{pankaj.gupta, yatin.chaudhary, buettner.florian}@siemens.com

**ABSTRACT**

We address two challenges of probabilistic topic modelling in order to better estimate the probability of a word in a given context, i.e., $P(\text{word} | \text{context})$: (1) *No language structure in context:* Probabilistic topic models ignore word order by summarizing a given context as a “bag-of-word” and consequently the semantics of words in the context is lost. In this work, we incorporate language structure by combining a neural autoregressive topic model (TM) (e.g., DocNADE) with a LSTM based language model (LSTM-LM) in a single probabilistic framework. The LSTM-LM learns a vector-space representation of each word by accounting for word order in local collocation patterns, while the TM simultaneously learns a latent representation from the entire document. In addition, the LSTM-LM models complex characteristics of language (e.g., syntax and semantics), while the TM discovers the underlying thematic structure in a collection of documents. We unite two complementary paradigms of learning the meaning of word occurrences by combining a topic model and a language model in a unified probabilistic framework, named as ctx-DocNADE. (2) *Limited context and/or smaller training corpus of documents:* In settings with a small number of word occurrences (i.e., lack of context) in short text or data sparsity in a corpus of few documents, the application of TMs is challenging. We address this challenge by incorporating external knowledge into neural autoregressive topic models via a language modelling approach: we use word embeddings as input of a LSTM-LM with the aim to improve the word-topic mapping on a smaller and/or short-text corpus. The proposed DocNADE extension is named as ctx-DocNADEe.

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

1 INTRODUCTION

Probabilistic topic models, such as LDA (Blei et al., 2003), Replicated Softmax (RSM) (Salakhutdinov & Hinton, 2009) and Document Neural Autoregressive Distribution Estimator (DocNADE) variants (Larochelle & Lauly, 2012; Zheng et al., 2016; Lauly et al., 2017; Gupta et al., 2019) are often used to extract topics from text collections, and predict the probabilities of each word in a given document belonging to each topic. Subsequently, they learn latent document representations that can be used to perform natural language processing (NLP) tasks such as information retrieval (IR), document classification or summarization. However, such probabilistic topic models ignore word order and represent a given context as a bag of its words, thereby disregarding semantic information.

To motivate our first task of extending probabilistic topic models to incorporate word order and language structure, assume that we conduct topic analysis on the following two sentences:

Bear falls into market territory and Market falls into bear territory
When estimating the probability of a word in a given context (here: $P(\text{"bear"}|\text{context})$), traditional topic models do not account for language structure since they ignore word order within the context and are based on “bag-of-words” (BoWs) only. In this particular setting, the two sentences have the same unigram statistics, but are about different topics. On deciding which topic generated the word “bear” in the second sentence, the preceding words “market falls” make it more likely that it was generated by a topic that assigns a high probability to words related to stock market trading, where “bear territory” is a colloquial expression in the domain. In addition, the language structure (e.g., syntax and semantics) is also ignored. For instance, the word “bear” in the first sentence is a proper noun and subject while it is an object in the second. In practice, topic models also ignore functional words such as “into”, which may not be appropriate in some scenarios.

Recently, Peters et al. (2018) have shown that a deep contextualized LSTM-based language model (LSTM-LM) is able to capture different language concepts in a layer-wise fashion, e.g., the lowest layer captures language syntax and topmost layer captures semantics. However, in LSTM-LMs the probability of a word is a function of its sentence only and word occurrences are modeled in a fine granularity. Consequently, LSTM-LMs do not capture semantics at a document level. To this end, recent studies such as TDLM (Lau et al., 2017), Topic-RNN (Dieng et al., 2016) and TCNLM (Wang et al., 2018) have integrated the merits of latent topic and neural language models (LMs); however, they have focused on improving LMs with global (semantics) dependencies using latent topics.

Similarly, while bi-gram LDA based topic models (Wallach, 2006; Wang et al., 2007) and n-gram based topic learning (Lauly et al., 2017) can capture word order in short contexts, they are unable to capture long term dependencies and language concepts. In contrast, DocNADE variants (Larochelle & Lauly, 2012; Gupta et al., 2019) learns word occurrences across documents i.e., coarse granularity (in the sense that the topic assigned to a given word occurrence equally depends on all the other words appearing in the same document); however since it is based on the BoW assumption all language structure is ignored. In language modeling, Mikolov et al. (2010) have shown that recurrent neural networks result in a significant reduction of perplexity over standard n-gram models.

Contribution 1: We introduce language structure into neural autoregressive topic models via a LSTM-LM, thereby accounting for word ordering (or semantic regularities), language concepts and long-range dependencies. This allows for the accurate prediction of words, where the probability of each word is a function of global and local (semantics) contexts, modeled via DocNADE and LSTM-LM, respectively. The proposed neural topic model is named as contextualized-Document Neural Autoregressive Distribution Estimator (ctx-DocNADE) and offers learning complementary semantics by combining joint word and latent topic learning in a unified neural autoregressive framework. For instance, Figure 1 (left and middle) shows the complementary topic and word semantics, based on TM and LM representations of the term “fall”. Observe that the topic captures the usage of “fall” in the context of stock market trading, attributed to the global (semantic) view.

While this is a powerful approach for incorporating language structure and word order in particular for long texts and corpora with many documents, learning from contextual information remains challenging in settings with short texts and few documents, since (1) limited word co-occurrences or little context (2) significant word non-overlap in such short texts and (3) small training corpus of documents lead to little evidence for learning word co-occurrences. However, distributional word representations (i.e. word embeddings) (Pennington et al., 2014) have shown to capture both the semantic and syntactic relatedness in words and demonstrated impressive performance in NLP tasks.

For example, assume that we conduct topic analysis over the two short text fragments: Deal with stock index falls and Brace for market share drops. Traditional topic models
with "BoW" assumption will not be able to infer relatedness between word pairs such as (falls, 
drops) due to the lack of word-overlap and small context in the two phrases. However, in 
distributed embedding space, the word pairs are semantically related as shown in Figure 1 (left).

Related work such as Sahami & Heilman (2006) employed web search results to improve the 
information in short texts and Petterson et al. (2010) introduced word similarity via thesauri and 
dictionaries into LDA. Das et al. (2015) and Nguyen et al. (2015) integrated word embeddings into LDA 
and Dirichlet Multinomial Mixture (DMM) (Nigam et al., 2000) models. Recently, Gupta et al. (2019) 
extends DocNADE by introducing pre-trained word embeddings in topic learning. However, they 
ignore the underlying language structure, e.g., word ordering, syntax, etc. In addition, DocNADE 
and its extensions outperform LDA and RSM topic models in terms of perplexity and IR.

Contribution 2: We incorporate distributed compositional priors in DocNADE: we use pre-trained 
word embeddings via LSTM-LM to supplement the multinomial topic model (i.e., DocNADE) in 
learning latent topic and textual representations on a smaller corpus and/or short texts. Knowing 
similarities in a distributed space and integrating this complementary information via a LSTM-LM, 
a topic representation is much more likely and coherent.

Taken together, we combine the advantages of complementary learning and external knowledge, and 
couple topic- and language models with pre-trained word embeddings to model short and long text 
documents in a unified neural autoregressive framework, named as ctx-DocNADEe. Our approach 
learns better textual representations, which we quantify via generalizability (e.g., perplexity), inter-
pretability (e.g., topic extraction and coherence) and applicability (e.g., IR and classification).

To illustrate our two contributions, we apply our modeling approaches to 7 long-text and 8 short-text 
datasets from diverse domains and demonstrate that our approach consistently outperforms state-of-
the-art generative topic models. Our learned representations, result in a gain of: (1) 4.6% (.790 vs 
.755) in topic coherence, (2) 6.5% (.615 vs .577) in precision at retrieval fraction 0.02, and (3) 4.4% 
(.662 vs .634) in F1 for text classification, averaged over 6 long-text and 8 short-text datasets.

When applied to short-text and long-text documents, our proposed modeling approaches generate 
contextualized topic vectors, which we name textTOvec. The code is available at https://
github.com/pgcool/textTOvec

2 NeuRAL AutOREGRESSIVE TopIC MODELS

Generative models are based on estimating the probability distribution of multidimensional data, 
implicitly requiring modeling complex dependencies. Restricted Boltzmann Machine (RBM) (Hin-
ton et al., 2006) and its variants (Larochelle & Bengio, 2008) are probabilistic undirected models
of binary data. RSM [Salakhutdinov & Hinton 2009] and its variants (Gupta et al. 2018) are generalization of the RBM, that are used to model word counts. However, estimating the complex probability distribution of the underlying high-dimensional observations is intractable. To address this challenge, NADE (Larochelle & Murray 2011) decomposes the joint distribution of binary observations into autoregressive conditional distributions, each modeled using a feed-forward network. Unlike for RSM/RSM, this leads to tractable gradients of the data negative log-likelihood.

2.1 DOCUMENT NEURAL AUTOREGRESSIVE TOPIC MODEL (DocNADE)

An extension of NADE and RSM, DocNADE (Larochelle & Lauly 2012) models collections of documents as orderless bags of words (BoW approach), thereby disregarding any language structure. In other words, it is trained to learn word representations reflecting the underlying topics of the documents only, ignoring syntactical and semantic features as those encoded in word embeddings (Bengio et al., 2003; Mikolov et al., 2013; Pennington et al., 2014; Peters et al., 2018).

DocNADE (Lauly et al., 2017) represents a document by transforming its BoWs into a sequence \( \mathbf{v} = [v_1, ..., v_D] \) of size \( D \), where each element \( v_i \) corresponds to a multinomial observation (representing a word from a vocabulary of size \( K \)). Thus, \( v_i \) is the index in the vocabulary of the \( i \)th word of the document \( \mathbf{v} \). DocNADE models the joint distribution \( p(\mathbf{v}) \) of all words \( v_i \) by decomposing it as \( p(\mathbf{v}) = \prod_{i=1}^{D} p(v_i|\mathbf{v}_{<i}) \), where each autoregressive conditional \( p(v_i|\mathbf{v}_{<i}) \) for the word observation \( v_i \) is computed using the preceding observations \( \mathbf{v}_{<i} \) in a feed-forward neural network for \( i \in \{1, ..., D\} \),

\[
\mathbf{h}_{i}^{DN}(v_{<i}) = g(e + \sum_{k<i} \mathbf{W}_{i,k}) \quad \text{and} \quad p(v_i = w|\mathbf{v}_{<i}) = \frac{\exp(b_w + U_{w,i} \mathbf{h}_{i}^{DN}(v_{<i}))}{\sum_w \exp(b_w' + U_{w',i} \mathbf{h}_{i}^{DN}(v_{<i}))} \quad (1)
\]

where \( g(\cdot) \) is an activation function, \( U \in \mathbb{R}^{K \times H} \) is a weight matrix connecting hidden to output, \( e \in \mathbb{R}^{H} \) and \( b \in \mathbb{R}^{K} \) are bias vectors, \( \mathbf{W} \in \mathbb{R}^{H \times K} \) is a word representation matrix in which a column \( \mathbf{W}_{i} \) is a vector representation of the word \( v_i \) in the vocabulary, and \( H \) is the number of hidden units (topics). The log-likelihood of any document \( \mathbf{v} \) of arbitrary length is given by:

\[
\mathcal{L}^{DN}(\mathbf{v}) = \sum_{i=1}^{D} \log p(v_i|\mathbf{v}_{<i}).
\]

Note that the past word observations \( \mathbf{v}_{<i} \) are orderless due to BoWs, and may not correspond to the words preceding the \( i \)th word in the document itself.

Algorithm 1 Computation of \( \log p(\mathbf{v}) \)

| model | \( \mathbf{h}_{i} \) | \( p(v_i|\mathbf{v}_{<i}) \) |
|-------|-----------------|------------------|
| DocNADE | \( \mathbf{h}_{i}^{DN} \leftarrow g(e) \) | \( \mathbf{h}_{i} \leftarrow \mathbf{h}_{i}^{DN} \) \quad equation 1 |
| ctx-DocNADE | \( \mathbf{h}_{i}^{LM} \leftarrow \text{LSTM}(\mathbf{c}_i, \text{embedding} = \mathbf{W}) \) | \( \mathbf{h}_{i} \leftarrow \mathbf{h}_{i}^{DN} + \lambda \mathbf{h}_{i}^{LM} \) \quad equation 2 |
| ctx-DocNADEe | \( \mathbf{h}_{i}^{LM} \leftarrow \text{LSTM}(\mathbf{c}_i, \text{embedding} = \mathbf{W} + \mathbf{E}) \) | \( \mathbf{h}_{i} \leftarrow \mathbf{h}_{i}^{DN} + \lambda \mathbf{h}_{i}^{LM} \) \quad equation 2 |

Table 1: Computation of \( \mathbf{h}_{i} \) and \( p(v_i|\mathbf{v}_{<i}) \) in DocNADE, ctx-DocNADE and ctx-DocNADEe models, correspondingly used in estimating \( \log p(\mathbf{v}) \) (Algorithm 1).

2.2 DEEP CONTEXTUALIZED DOCNADE WITH DISTRIBUTIONAL SEMANTICS

We propose two extensions of the DocNADE model: (1) ctx-DocNADE: introducing language structure via LSTM-LM and (2) ctx-DocNADEe: incorporating external knowledge via pre-trained word embeddings \( \mathbf{E} \), to model short and long texts. The unified network(s) account for the ordering of words, syntactical and semantic structures in a language, long and short term dependencies, as well as external knowledge, thereby circumventing the major drawbacks of BoW-based representations.

Similar to DocNADE, ctx-DocNADE models each document \( \mathbf{v} \) as a sequence of multinomial observations. Let \( \{x_1,x_2, ..., x_N\} \) be a sequence of \( N \) words in a given document, where \( x_i \) is represented by an embedding vector of dimension, \( \text{dim} \). Further, for each element \( v_i \in \mathbf{v} \), let \( e_c = [x_1,x_2, ..., x_{i-1}] \) be the context (preceding words) of \( i \)th word in the document. Unlike in DocNADE, the conditional probability of the word \( v_i \) in ctx-DocNADE (or ctx-DocNADEe) is a function of two hidden vectors: \( \mathbf{h}_{i}^{DN}(v_{<i}) \) and \( \mathbf{h}_{i}^{LM}(e_i) \), stemming from the DocNADE-based and
LSTM-based components of ctx-DocNADE, respectively:

\[
 h_i(v_{<i}) = h_i^{DN}(v_{<i}) + \lambda h_i^{LM}(c_i) \text{ and } p(v_i = w|v_{<i}) = \frac{\exp(b_w + U_{w} h_i(v_{<i}))}{\sum_{w'} \exp(b_{w'} + U_{w'} h_i(v_{<i}))} \tag{2}
\]

where \( h_i^{DN}(v_{<i}) \) is computed as in DocNADE (equation 1) and \( \lambda \) is the mixture weight of the LM component, which can be optimized during training (e.g., based on the validation set). The second term \( h_i^{LM} \) is a context-dependent representation and output of an LSTM layer at position \( i \) over input sequence \( c_i \), trained to predict the next word \( v_i \). The LSTM offers history for the \( i \)th word via modeling temporal dependencies in the input sequence, \( c_i \). The conditional distribution for each word \( v_i \) is estimated by equation 2 where the unified network of DocNADE and LM combines global and context-dependent representations. Our model is jointly optimized to maximize the pseudo log likelihood, \( \log p(v) \approx \sum_{i=1}^{n} \log p(v_i|v_{<i}) \) with stochastic gradient descent. See Larochelle & Lauly (2012) for more details on training from bag of word counts.

In the weight matrix \( W \) of DocNADE (Larochelle & Lauly 2012), each row vector \( W_j \) encodes topic information for the \( j \)th hidden topic feature and each column vector \( W_{:,i} \) is a vector for the word \( v_i \). To obtain complementary semantics, we exploit this property and expose \( W \) to both global and local influences by sharing \( W \) in the DocNADE and LSTM-LM components. Thus, the embedding layer of LSTM-LM component represents the column vectors.

cxt-DocNADE, in this realization of the unified network the embedding layer in the LSTM component is randomly initialized. This extends DocNADE by accounting for the ordering of words and language concepts via context-dependent representations for each word in the document.

cxt-DocNADEe, the second version extends ctx-DocNADE with distributional priors, where the embedding layer in the LSTM component is initialized by the sum of a pre-trained embedding matrix \( E \) and the weight matrix \( W \). Note that \( W \) is a model parameter; however \( E \) is a static prior.

Algorithm 1 and Table 1 show the \( \log p(v) \) for a document \( v \) in three different settings: DocNADE, cxt-DocNADE and cxt-DocNADEe. In the DocNADE component, since the weights in the \( W \) are tied, the linear activation can be re-used in every hidden layer and computational complexity reduces to \( O(HD) \), where \( H \) is the size of each hidden layer. In every epoch, we run an LSTM over the sequence of words in the document and extract hidden vectors \( h_i^{LM} \), corresponding to \( c_i \) for every target word \( v_i \). Therefore, the computational complexity in cxt-DocNADE or cxt-DocNADEe is \( O(HD+n) \), where \( n \) is the total number of edges in the LSTM network (Hochreiter & Schmidhuber 1997; Sak et al. 2014). The trained models can be used to extract a textTOvec representation, i.e., \( h_i(v^*) = h_i^{DN}(v^*) + \lambda h_i^{LM}(e_{N+1}) \) for the text \( v^* \) of length \( D^* \), where \( h_i^{DN}(v^*) = g(e_i + \sum_{k \in D^*} W_{:,v_k}) \) and \( h_i^{LM}(e_{N+1}) = \text{LSTM}(e_{N+1}, \text{embedding} = W \text{ or } (W+E)) \).

cxt-DeepDNEe: DocNADE and LSTM 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. For the first hidden layer, the code is computed in an analogous fashion to DocNADE variants (equation 1 or 2). Subsequent hidden layers are computed as:

\[
 h_i^{DN}(v_{<i}) = g(e_i + W_{i,d} \cdot h_{i,d-1}(v_{<i})) \text{ or } h_i^{LM}(c_i) = \text{deepLSTM}(c_i, \text{depth} = d)
\]

for \( d = 2, \ldots, n \), where \( n \) is the total number of hidden layers (i.e., depth) in the deep feed-forward and LSTM networks. For \( d=1 \), the hidden vectors \( h_i^{DN} \) and \( h_i^{LM} \) correspond to equations 1 and 2.

The conditional \( p(v_i = w|v_{<i}) \) is computed using the last layer, i.e., \( h_i,n = h_i^{DN} + \lambda h_i^{LM} \).
3 Evaluation

We apply our modeling approaches (in improving topic models, i.e., DocNADE using language concepts from LSTM-LM) to 8 short-text and 7 long-text datasets of varying size with single/multi-class labeled documents from public as well as industrial corpora. We present four quantitative measures in evaluating topic models: generalization (perplexity), topic coherence, text retrieval and categorization. See the appendices for the data description and example texts. Table 2 shows the data statistics, where 20NS and R21578 signify 20NewsGroups and Reuters21578, respectively.

**Baselines:** While, we evaluate our multi-fold contributions on four tasks: generalization (perplexity), topic coherence, text retrieval and categorization, we compare performance of our proposed models with related baselines based on: (1) word representation: glove (Pennington et al., 2014), where a document is represented by summing the embedding vectors of it’s words, (2) document representation: doc2vec (Le & Mikolov, 2014), (3) LDA based BoW TMs: ProdLDA (Srivastava & Sutton, 2017) and SCHOLAR (Card et al., 2017), (4) neural BoW TMs: DocNADE and TMN (Cao et al., 2015) and. (5) TMs, including pre-trained word embeddings: Gauss-LDA (GaussianLDA) (Das et al., 2015), and glove-DMM, glove-LDA (Nguyen et al., 2015), (6) jointly trained topic and language models: TDLM (Lau et al., 2017), Topic-RNN (Ding et al., 2016) and TCMN (Wang et al., 2018).

**Table 3:** State-of-the-art comparison: IR (i.e., IR-precision at 0.02 fraction) and classification F1 for short texts, where Avg: average over the row values, the bold and underline: the maximum for IR and F1, respectively.

| Model | 20NSshort | TRC6 | R21578title | Subjectivity | Polarity | TMNtitle | TMN | AGnews | Avg
|-------|-----------|------|-------------|--------------|----------|----------|-----|--------|-----
| ProdLDA | 236.493 | 480.798 | 587.356 | 754.882 | 543.713 | 513.693 | 638.736 | 588.818 | 542.685
| glove(DV) | 236.488 | 480.785 | 595.356 | 775.901 | 553.724 | 545.736 | 643.813 | 612.870 | 554.704
| glove(DV) + DocNADE | 0.043 | 0.260 | 0.492 | 0.518 | 0.576 | 0.510 | 0.624 | 0.190 | 0.582
| Gauss-LDA | 0.080 | 1.180 | 2.020 | 2.067 | 0.505 | 0.511 | 0.408 | 0.472 | 0.516
| glove-DMM | 0.183 | 0.213 | 0.370 | 0.454 | 0.273 | 0.251 | 0.378 | 0.784 | 0.515
| glove-LDA | 0.160 | 0.320 | 0.380 | 0.360 | 0.257 | 0.236 | 0.260 | 0.426 | 0.247
| DocNADE(RV) | 2.900 | 0.440 | 0.550 | 0.804 | 0.657 | 0.313 | 0.820 | 0.889 | 0.360
| DocNADE(FV) | 2.900 | 0.440 | 0.550 | 0.804 | 0.657 | 0.313 | 0.820 | 0.889 | 0.360

**Table 4:** IR-precision at fraction 0.02 and classification F1 for long texts

**Table 5:** Generalization: PPL

<table>
<thead>
<tr>
<th>Model</th>
<th>PPL</th>
<th>Model</th>
<th>PPL</th>
<th>Model</th>
<th>PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProdLDA</td>
<td>968</td>
<td>DocNADE</td>
<td>968</td>
<td>DocNADE</td>
<td>968</td>
</tr>
<tr>
<td>20NS</td>
<td>846</td>
<td>20NS</td>
<td>846</td>
<td>20NS</td>
<td>846</td>
</tr>
<tr>
<td>Subjectivity</td>
<td>433</td>
<td>Subjectivity</td>
<td>433</td>
<td>Subjectivity</td>
<td>433</td>
</tr>
<tr>
<td>Polarity</td>
<td>870</td>
<td>Polarity</td>
<td>870</td>
<td>Polarity</td>
<td>870</td>
</tr>
<tr>
<td>TMN</td>
<td>501</td>
<td>TMN</td>
<td>501</td>
<td>TMN</td>
<td>501</td>
</tr>
<tr>
<td>AGnews</td>
<td>483</td>
<td>AGnews</td>
<td>483</td>
<td>AGnews</td>
<td>483</td>
</tr>
</tbody>
</table>
We run TDLM ([Lau et al., 2017]) for all the short-text datasets to evaluate the quality of representations learned in the spare data setting. For a fair comparison, we set 200 topics and hidden size, and initialize with the same pre-trained word embeddings (i.e., glove) as used in the ctx-DocNADEe.

### 3.1 Generalization: Perplexity (PPL)

To evaluate the generative performance of the topic models, we estimate the log-probabilities for the test documents and compute the average held-out perplexity (PPL) per word as,  

\[
PPL = \exp \left(-\frac{1}{N} \sum_{r=1}^{N} \log p(v^r) \right),
\]

where \(N\) and \(|v^r|\) are the total number of documents and words in a document \(v^r\). For DocNADE, the log-probability \(\log p(v^r)\) is computed using \(\mathcal{L}^{DN} (v);\) however, we ignore the mixture coefficient, i.e., \(\lambda = 0\) (equation 2) to compute the exact log-likelihood in ctx-DocNADE versions. The optimal \(\lambda\) is determined based on the validation set. Table 5 quantitatively shows the PPL scores, where the complementary learning with \(\lambda = 0.01\) (optimal) in ctx-DocNADE achieves lower perplexity than the baseline DocNADE for both short and long texts, e.g., (822 vs 846) and (1358 vs 1375) on AGnewstitle and 20NS datasets, respectively, in the FV setting.

### 3.2 Interpretability: Topic Coherence

We compute topic coherence ([Chang et al., 2009], [Newman et al., 2009], [Gupta et al., 2018]) to assess the meaningfulness of the underlying topics captured. We choose the coherence measure proposed by [Röder et al., 2015], which identifies context features for each topic word using a sliding window over the reference corpus. Higher scores imply more coherent topics.

We use the gensim module ([radimrehurek.com/gensim/models/coherencemodel.html], coherence type = c_v) to estimate coherence for each of the 200 topics (top 10 and 20 words). Table 6 shows average coherence over 200 topics, where the higher scores in ctx-DocNADE compared to DocNADE (772 vs .755) suggest that the contextual information and language structure help in generating more coherent topics. The introduction of embeddings in ctx-DocNADEe boosts the topic coherence, leading to a gain of 4.6% (.790 vs .755) on average over 11 datasets. Note that the proposed models also outperform the baselines methods glove-DMM and glove-LDA. Qualitatively, Table 7 illustrates an example topic from the 20NSShort text dataset for DocNADE, ctx-DocNADE and ctx-DocNADEe, where the inclusion of embeddings results in a more coherent topic.

**Additional Baselines:** We further compare our proposed models to other approaches that combining topic and language models, such as TDLM ([Lau et al., 2017]), Topic-RNN ([Dieng et al., 2016]) and TCNL ([Wang et al., 2018]). However, the related studies focus on improving language models using topic models: in contrast, the focus of our work is on improving topic models for textual representations (short-text or long-text documents) by incorporating language concepts (e.g., word ordering, syntax, semantics, etc.) and external knowledge (e.g., word embeddings) via neural language models, as discussed in section 3.

To this end, we follow the experimental setup of the most recent work, TCNL and quantitatively compare the performance of our models (i.e., ctx-DocNADE and ctx-DocNADEe) in terms of topic coherence (NPMI) on BNC dataset. Table 8 (left) shows NPMI scores of different models, where the results suggest that our contribution (i.e., ctx-DocNADE) of introducing language concepts into BoW topic model (i.e., DocNADE) improves topic coherence. The better performance for high-val-

---

2. PPL scores in (RV/FV) settings: DocNADE (665/1375) outperforms ProdLDA (1168/2097) on 200 topics
3. NPMI over (50/200) topics learned on 20NS by: ProdLDA (.24/.19) and DocNADE (.15/.12) in RV setting
with TDLM, our proposed models (ctx-DocNADE and ctx-DocNADEe) outperform it by a notable

text and 6 long-text datasets, ctx-DocNADEe
RV setting. Therefore, we opt for the FV in the proposed extensions. On an average over the 8 short-

texts. We also investigate topic modeling without pre-processing and filtering certain words, i.e. the

language/contextual information leads to improved performance on the IR task noticeably for short

0

ratively at retrieval fraction

Table 3 and 4 show the retrieval precision scores for the short-text and long-text datasets, respec-

& Hinton (2009) and Lauly et al. (2017) have shown that RSM and DocNADE strictly outperform

average the number of retrieved training documents with the same label as the query. For multi-label

training set using cosine similarity measure between their

textTOvec

2.2). To compute retrieval precision for each fraction (e.g.,

3.3 A

PPLICABILITY

us of λ illustrates the relevance of the LM component for topic coherence (DocNADE corresponds to

λ=0). Similarly, the inclusion of word embeddings (i.e., ctx-DocNADEe) results in more coherent

topics than the baseline DocNADE. Importantly, while ctx-DocNADEe is motivated by sparse

data settings, the BNC dataset is neither a collection of short-text nor a corpus of few documents.

Consequently, ctx-DocNADEe does not show improvements in topic coherence over ctx-DocNADE.

In Table 7 (right), we further qualitatively show the top 5 words of seven topics (topic name summa-

rized by Wang et al. [2018]) from TCNLM and our models. Observe that ctx-DocNADE captures a topic expression that is a collection of only verbs in the past participle. Since the BNC dataset is unlabeled, we are here restricted to comparing model performance in terms of topic coherence only.

3.3 Applicability: Text Retrieval and Categorization

Text Retrieval: We perform a document retrieval task using the short-text and long-text documents with label information. We follow 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 textTOvec representations (section 2.2). To compute retrieval precision for each fraction (e.g., 0.0001, 0.005, 0.01, 0.02, 0.05, 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 & Hinton (2009) and Lauly et al. (2017) have shown that RSM and DocNADE strictly outperform LDA on this task, we solely compare DocNADE with our proposed extensions.

Table 3 and 4 show the retrieval precision scores for the short-text and long-text datasets, respec-

tively at retrieval fraction 0.02. Observe that the introduction of both pre-trained embeddings and language/contextual information leads to improved performance on the IR task noticeably for short texts. We also investigate topic modeling without pre-processing and filtering certain words, i.e. the

FV setting and find that the DocNADE(FV) or glove(FV) improves IR precision over the baseline

RV setting. Therefore, we opt for the FV in the proposed extensions. On an average over the 8 short-
text and 6 long-text datasets, ctx-DocNADEe reports a gain of 7.1% (.630 vs .588) (Table 3) 6.0% (.601 vs .567) (Table 3), respectively in precision compared to DocNADE(RV). To further compare with TDLM, our proposed models (ctx-DocNADE and ctx-DocNADEe) outperform it by a notable margin for all the short-text datasets, i.e., a gain of 14.5% (.630 vs .550: ctx-DocNADEe vs TDLM)
which are (sub)categories in the data, confirming that meaningful topics are captured. Observe that DocNADE establishes itself as a strong neural topic model baseline. Cao et al. (2015), Figure 2) and SCHOLAR (results taken from Card et al. (2017), Table 2), the posed models, i.e., ctx-DocNADE and ctx-DocNADEe outperform both NTM (results taken from Hsu et al., 2017) in terms of classification accuracy on 20NS dataset, the scores are: DocNADE (0.734), ctx-DocNADE (0.744), ctx-DocNADEe (0.751), NTM (0.72) and SCHOLAR (0.71). While, our proposed models, i.e., ctx-DocNADE and ctx-DocNADEe outperform both NTM (results taken from Cao et al., 2015, Figure 2) and SCHOLAR (results taken from Card et al., 2017, Table 2), the DocNADE establishes itself as a strong neural topic model baseline.

To further interpret the topic models, we analyze the meaningful semantics captured via topic extraction. Table 8 shows a topic extracted using 20NS dataset that could be interpreted as computers, which are (sub)categories in the data, confirming that meaningful topics are captured. Observe that

\[
\text{Precision (%) vs Fraction of Retrieved Documents (Recall)}
\]

**Figure 3:** Retrieval performance (IR-precision) on 6 datasets at different fractions

in IR-precision. In addition, the deep variant (d=3) with embeddings, i.e., ctx-DeepDNEe shows competitive performance on TREC6 and Subjectivity datasets.

Figures (3a, 3b, 3c, 3d, 3e and 3f) illustrate the average precision for the retrieval task on 6 datasets. Observe that the ctx-DocNADEe outperforms DocNADE(RV) at all the fractions and demonstrates a gain of 6.5% (0.615 vs 0.577) in precision at fraction 0.02, averaged over 14 datasets. Additionally, our proposed models outperform TDLM and ProdLDA\(^6\) (for 20NS) by noticeable margins.

**Text Categorization:** We perform text categorization to measure the quality of our text2vec representations. We consider the same experimental setup as in the retrieval task and extract text2vec of 200 dimension for each document, learned during the training of ctx-DocNADE variants. To perform text categorization, we employ a logistic regression classifier with \(L_2\) regularization. While, ctx-DocNADE and ctx-DeepDNEe make use of glove embeddings, they are evaluated against the topic model baselines with embeddings. For the short texts (Table 3), the glove leads DocNADE in classification performance, suggesting a need for distributional priors in the topic model. Therefore, the ctx-DocNADEe reports a gain of 4.8% (0.705 vs 0.673) and 3.6% (0.618 vs 0.596) in \(F_1\), compared to DocNADE(RV) on an average over the short (Table 3) and long (Table 4) texts, respectively. In result, a gain of 4.4% (0.662 vs 0.634) overall.

In terms of classification accuracy on 20NS dataset, the scores are: DocNADE (0.734), ctx-DocNADE (0.744), ctx-DocNADEe (0.751), NTM (0.72) and SCHOLAR (0.71). While, our proposed models, i.e., ctx-DocNADE and ctx-DocNADEe outperform both NTM (results taken from Cao et al., 2015, Figure 2) and SCHOLAR (results taken from Card et al., 2017, Table 2), the DocNADE establishes itself as a strong neural topic model baseline.

### 3.4 Inspection of Learned Representations

To further interpret the topic models, we analyze the meaningful semantics captured via topic extraction. Table 8 shows a topic extracted using 20NS dataset that could be interpreted as computers, which are (sub)categories in the data, confirming that meaningful topics are captured. Observe that

\[
\text{Precision (%) vs Fraction of Retrieved Documents (Recall)}
\]
the ctx-DocNADEe extracts a more coherent topic due to embedding priors. To qualitatively inspect the contribution of word embeddings and textTOvec representations in topic models, we analyse the text retrieved for each query using the representations learned from DocNADE and ctx-DocNADEe models. Table 9 illustrates the retrieval of the top 3 texts for an input query, selected from TMNtitle dataset, where #match is YES if the query and retrievals have the same class label. Observe that ctx-DocNADEe retrieves the top 3 texts, each with no unigram overlap with the query.

<table>
<thead>
<tr>
<th>Query</th>
<th>“emerging economies move ahead nuclear plans”</th>
<th>#match</th>
</tr>
</thead>
<tbody>
<tr>
<td>#IR1 :: IMF signs lifting Japan yen</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>#IR2 :: Japan recovery takes hold debt downgrade looms</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>#IR3 :: Japan ministers confident treasuries move</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>#IR1 :: nuclear regulator back power plans</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>#IR2 :: defiant Iran plans big rise nuclear</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>#IR3 :: Japan banks billion nuclear operator sources</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: A topic of 20NS dataset with coherence

Table 9: Illustration of the top-3 retrievals for an input query

Additionally, we show the quality of representations learned at different fractions (20%, 40%, 60%, 80%, 100%) of training set from TMNtitle data and use the same experimental setup for the IR and classification tasks, as in section 3.3. In Figure 4, we quantify the quality of representations learned and demonstrate improvements due to the proposed models, i.e., ctx-DocNADE and ctx-DocNADEe over DocNADE at different fractions of the training data. Observe that the gains in both the tasks are large for smaller fractions of the datasets. For instance, one of the proposed models, i.e., ctx-DocNADEe (vs DocNADE) reports: (1) a precision (at 0.02 fraction) of 0.580 vs 0.444 at 20% and 0.595 vs 0.525 at 100% of the training set, and (2) an F1 of 0.711 vs 0.615 at 20% and 0.726 vs 0.688 at 100% of the training set. Therefore, the findings conform to our second contribution of improving topic models with word embeddings, especially in the sparse data setting.

3.5 CONCLUSION

In this work, we have shown that accounting for language concepts such as word ordering, syntactic and semantic information in neural autoregressive topic models helps to better estimate the probability of a word in a given context. To this end, we have combined a neural autoregressive topic- (i.e., DocNADE) and a neural language (e.g., LSTM-LM) model in a single probabilistic framework with an aim to introduce language concepts in each of the autoregressive steps of the topic model. This facilitates learning a latent representation from the entire document whilst accounting for the local dynamics of the collocation patterns, encoded in the internal states of LSTM-LM. We further augment this complementary learning with external knowledge by introducing word embeddings. Our experimental results show that our proposed modeling approaches consistently outperform state-of-the-art generative topic models, quantified by generalization (perplexity), topic interpretability (coherence), and applicability (text retrieval and categorization) on 15 datasets.
REFERENCES


A DATA DESCRIPTION

Table 10: SiROBs data: Example Documents (Requirement Objects) with their types (label).

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Search Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning rate</td>
<td>[0.001]</td>
</tr>
<tr>
<td>hidden units</td>
<td>[200]</td>
</tr>
<tr>
<td>iterations</td>
<td>[2000]</td>
</tr>
<tr>
<td>activation function</td>
<td>sigmoid</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>[1.0, 0.8, 0.5, 0.3, 0.1, 0.01, 0.001]</td>
</tr>
</tbody>
</table>

Table 11: Hyperparameters in Generalization in the DocNADE and ctx-DocNADE variants for 200 topics

The SiROBs is our industrial corpus, extracted from industrial tender documents. The documents contain requirement specifications for an industrial project for example, railway metro construction. There are 22 types of requirements i.e. class labels (multi-class), where a requirement is a paragraph or collection of paragraphs within a document. We name the requirement as Requirement Objects (ROBs). Some of the requirement types are project management, testing, legal, risk analysis, financial cost, technical requirement, etc. We need to classify the requirements in the tender documents and assign each ROB to a relevant department(s). Therefore, we analyze such documents to automate decision making, tender comparison, similar tender as well as ROB retrieval and assigning ROBs to a relevant department(s) to optimize/expedite tender analysis. See some examples of ROBs from SiROBs corpus in Table 10.

B Experimental Setup

B.1 Experimental Setup and Hyperparameters for Generalization Task

See Table 11 for hyperparameters used in generalization.

B.2 Experimental Setup and Hyperparameters for IR Task

We set the maximum number of training passes to 1000, topics to 200 and the learning rate to 0.001 with \( \tanh \) hidden activation. For model selection, we used the validation set as the query set and used the average precision at 0.02 retrieved documents as the performance measure. Note that the
<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Search Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieval fraction</td>
<td>[0.02]</td>
</tr>
<tr>
<td>learning rate</td>
<td>[0.001]</td>
</tr>
<tr>
<td>hidden units</td>
<td>[200]</td>
</tr>
<tr>
<td>activation function</td>
<td>tanh</td>
</tr>
<tr>
<td>iterations</td>
<td>[2000]</td>
</tr>
<tr>
<td>λ</td>
<td>[1.0, 0.8, 0.5, 0.3, 0.1, 0.01, 0.001]</td>
</tr>
</tbody>
</table>

Table 12: Hyperparameters in the Document Retrieval task.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>λ 1.0</th>
<th>λ 0.1</th>
<th>λ 0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>20NSshort</td>
<td>ctx-DocNADE</td>
<td>899.04</td>
<td>829.5</td>
<td>842.1</td>
</tr>
<tr>
<td></td>
<td>ctx-DocNADEe</td>
<td>890.3</td>
<td>828.8</td>
<td>832.4</td>
</tr>
<tr>
<td>Subjectivity</td>
<td>ctx-DocNADE</td>
<td>982.8</td>
<td>977.8</td>
<td>966.5</td>
</tr>
<tr>
<td></td>
<td>ctx-DocNADEe</td>
<td>977.1</td>
<td>975.0</td>
<td>964.2</td>
</tr>
<tr>
<td>TMNtitle</td>
<td>ctx-DocNADE</td>
<td>1898.1</td>
<td>1482.7</td>
<td>1487.1</td>
</tr>
<tr>
<td></td>
<td>ctx-DocNADEe</td>
<td>1877.7</td>
<td>1480.2</td>
<td>1484.7</td>
</tr>
<tr>
<td>AGnewstitle</td>
<td>ctx-DocNADE</td>
<td>1296.1</td>
<td>861.1</td>
<td>865.0</td>
</tr>
<tr>
<td></td>
<td>ctx-DocNADEe</td>
<td>1279.2</td>
<td>853.3</td>
<td>862.9</td>
</tr>
<tr>
<td>Reuters-8</td>
<td>ctx-DocNADE</td>
<td>336.1</td>
<td>313.2</td>
<td>311.9</td>
</tr>
<tr>
<td></td>
<td>ctx-DocNADEe</td>
<td>323.3</td>
<td>312.0</td>
<td>310.2</td>
</tr>
<tr>
<td>20NS</td>
<td>ctx-DocNADE</td>
<td>1282.1</td>
<td>1209.3</td>
<td>1207.2</td>
</tr>
<tr>
<td></td>
<td>ctx-DocNADEe</td>
<td>1247.1</td>
<td>1211.6</td>
<td>1206.1</td>
</tr>
</tbody>
</table>

Table 13: Perplexity scores for different λ in Generalization task: Ablation over validation set

labels are not used during training. The class labels are only used to check if the retrieved documents have the same class label as the query document. To perform document retrieval, we use the same train/development/test split of documents discussed in data statistics (experimental section) for all the datasets during learning.

See Table 12 for the hyperparameters in the document retrieval task.

B.3 EXPERIMENTAL SETUP FOR DOC2VEC MODEL

We used gensim (https://github.com/RaRe-Technologies/gensim) to train Doc2Vec models for 12 datasets. Models were trained with distributed bag of words, for 1000 iterations using a window size of 5 and a vector size of 500.

B.4 CLASSIFICATION TASK

We used the same split in training/development/test as for training the Doc2Vec models (also same split as in IR task) and trained a regularized logistic regression classifier on the inferred document vectors to predict class labels. In the case of multilabel datasets (R21578,R21578title, RCV1v2), we used a one-vs-all approach. Models were trained with a liblinear solver using L2 regularization and accuracy and macro-averaged F1 score were computed on the test set to quantify predictive power.

B.5 EXPERIMENTAL SETUP FOR GLOVE-DMM AND GLOVE-LDA MODELS

We used LFTM (https://github.com/datquocnguyen/LFTM) to train glove-DMM and glove-LDA models. Models were trained for 200 iterations with 2000 initial iterations using 200 topics. For short texts we set the hyperparameter beta to 0.1, for long texts to 0.01; the mixture parameter lambda was set to 0.6 for all datasets. The setup for the classification task was the same as for doc2vec; classification was performed using relative topic proportions as input (i.e. we inferred the topic distribution of the training and test documents and used the relative distribution as input.
for the logistic regression classifier). Similarly, for the IR task, similarities were computed based on the inferred relative topic distribution.

### B.6 Experimental Setup for ProdLDA

We run ProdLDA ([https://github.com/akashgit/autoencoding_vi_for_topic_models](https://github.com/akashgit/autoencoding_vi_for_topic_models)) on the short-text datasets in the FV setting to generate document vectors for IR-task. We use 200 topics for a fair comparison with other baselines used for the IR tasks. We infer topic distribution of the training and test documents and used the relative distribution as input for the IR task, similar to section 3.3.

To fairly compare PPL scores of ProdLDA and DocNADE in the RV setting, we take the pre-processed 20NS dataset released by ProdLDA and run DocNADE for 200 topics. To further compare them in the FV setting, we run ProdLDA ([https://github.com/akashgit/autoencoding_vi_for_topic_models](https://github.com/akashgit/autoencoding_vi_for_topic_models)) on the processed 20NS dataset for 200 topics used in this paper.

### C Ablation over the Mixture Weight $\lambda$

#### C.1 $\lambda$ for Generalization Task

See Table[13]

#### C.2 $\lambda$ for IR Task

See Table[14]
D ADDITIONAL BASELINES

D.1 DocNADE vs SCHOLAR

**PPL scores over 20 topics:** DocNADE (752) and SCHOLAR (921), i.e., DocNADE outperforms SCHOLAR in terms of generalization.

*Topic coherence (NPMI) using 20 topics:* DocNADE (.18) and SCHOLAR (.35), i.e., SCHOLAR (Card et al., 2017) generates more coherence topics than DocNADE, though worse in PPL and text classification (see section 3.3) than DocNADE, ctx-DocNADE and ctx-DocNADEe.

**IR tasks:** Since, SCHOLAR (Card et al., 2017) without meta-data equates to ProdLDA and we have shown in section 3.3 that ProdLDA is worse on IR tasks than our proposed models, therefore one can infer the performance of SCHOLAR on IR task.

The experimental results above suggest that the DocNADE is better than SCHOLAR in generating good representations for downstream tasks such as information retrieval or classification, however falls behind SCHOLAR in interpretability. The investigation opens up an interesting direction for future research.