Self-Discriminative Learning for Unsupervised Document Embedding

Hong-You Chen[*1], **Chin-Hua Hu**[*1], **Leila Wehbe**[2], **Shou-De Lin**[1]
[1]Department of Computer Science and Information Engineering, National Taiwan University
[2]Machine Learning Department, Carnegie Mellon University
{b03902128, r07922028}@ntu.edu.tw,
lwehbe@cmu.edu, sdlin@csie.ntu.edu.tw

Abstract

Unsupervised document representation learning is an important task providing pre-trained features for NLP applications. Unlike most previous work which learn the embedding based on self-prediction of the surface of text, we explicitly exploit the inter-document information and directly model the relations of documents in embedding space with a discriminative network and a novel objective. Extensive experiments on both small and large public datasets show the competitiveness of the proposed method. In evaluations on standard document classification, our model has errors that are relatively 5 to 13% lower than state-of-the-art unsupervised embedding models. The reduction in error is even more pronounced in scarce label setting.

1 Introduction

Rapid advance in deep methods for natural language processing has contributed to a growing need for vector representation of documents as input features. Applications for such vector representations include machine translation (Sutskever et al., 2014), text classification (Dai and Le, 2015), image captioning (Mao et al., 2015), multi-lingual document matching (Pham et al., 2015), question answering (Rajpurkar et al., 2016), and more. This work studies *unsupervised* training for encoders that can efficiently encode long paragraph of text into compact vectors to be used as pre-trained features. Existing solutions are mostly based on the assumption that a good document embedding can be learned through modeling the **intra**-document information by predicting the occurrence of terms inside the document itself. We argue that such an assumption might not be sufficient to obtain meaningful a document embedding as they do not consider **inter**-document relationships.

Traditional document representation models such as Bag-of-words (BoW) and TF-IDF show competitive performance in some tasks (Wang and Manning, 2012). However, these models treat words as flat tokens which may neglect other useful information such as word order and semantic distance. This in turn can limit the models effectiveness on more complex tasks that require deeper level of understanding. Further, BoW models suffer from high dimensionality and sparsity. This is likely to prevent them from being used as input features for downstream NLP tasks.

Continuous vector representations for documents are being developed. A successful thread of work is based on the distributional hypothesis, and use contextual information for context-word predictions. Similar to Word2Vec (Mikolov et al., 2013), PV (Le and Mikolov, 2014) is optimized by predicting the next words given their contexts in a document, but it is conditioned on a unique document vector. Word2Vec-based methods for computing document embeddings achieve state-of-the-art performance on document embedding. Such methods rely on one strong underlying assumption: it is necessary to train the document embedding to optimize the prediction of the target words in the document. In other words, the objective requires the model to learn to predict the target words in surface text. We argue that there are several concerns with such a self-prediction assumption.

The strategy of predicting target words therefore only exploits in-document information, and do not explicitly model the inter-document distances. We believe an ideal embedding space should also infer the relations among training documents. For example, if all documents in the corpus are about *machine learning*, then the con-

[*] Equally contribution.

Proceedings of NAACL-HLT 2019, pages 2465–2474
Minneapolis, Minnesota, June 2 - June 7, 2019.

cept of *machine learning* becomes less critical in the embedding. However, if the corpus contains documents from different areas of computer science, then the concept of *machine learning* should be encoded in any document relevant to it. We therefore claim that *the embedding of a document should not only depend on the document itself but also the other documents in the corpus*, even though previous work seldom makes this consideration.

In addition, accurate predictions at the lexicon or word level do not necessarily reflect that the "true semantics" have been learned. For example, in IMDB dataset review No.10007:

$$"...the\ father\ did\ such\ a\ good\ job."$$

Obviously, *good* can be replaced with synonyms like *nice* without significantly altering the meaning of the sentence. However, since the synonyms are treated as independent tokens in PV and Doc2VecC, the lexicon *good* must be predicted exactly. Moreover, to accurately predict the final word *job*, the embedding probably only needs to know that *did a good job* is a very common phrase, without having to understand the true meaning of *job*. This example shows that in order to accurately predict a local lexicon, the embedding might opt to encode the syntactic relationship instead of true semantics. Enforcing document embeddings to make predictions at the word level could be too strong of an objective. More specifically, we argue that the true semantics should not only depend on a small context, but also the relations with other training documents at document level.

To address the above concerns we propose a novel model for learning document embedding unsupervisedly. In contrast with previous work (PV and Doc2Vec), we model documents according to two aspects.

First, we abandon the concept of context word prediction when training an embedding model. Instead we propose a self-supervision learning framework to model inter-document information. Conceptually, we use the embedding to determine whether a sentence belongs to a document. Our encoder is equipped with a discriminator to classify whether a sentence embedding is derived from a document given that document's embedding. This explicitly enforces documents to be spread reasonably in the embedding space without any labels so that they can be discriminated. To the best of our knowledge, this is the first deep embedding work to explicitly model the inter-document relationship.

Second, in our approach the predictions are inferred at the sentence level. This avoids the effect of only predicting the surface meaning in word level (e.g. good vs. nice). Unlike previous work, our model is explicitly optimized to represent documents as combinations of sequence embedding beyond words seen in training.

Below we summarize the key contributions:

- We present a deep and general framework and a novel objective for learning document representation unsupervisedly. Our models are end-to-end, easy to implement, and flexible to extend.
- We perform experiments through sentiment analysis and topic classification to show that our model, referred to as *self-discriminative document embedding (SDDE)*, is competitive to the state-of-the-art solutions based on traditional context-prediction objectives.
- Our extensive experiments quantitatively and qualitatively show that SDDE learns more effective features that capture more document-level information. To the best of our knowledge, SDDE is the first deep network to model inter-instance information at document level.
- We further propose to evaluate unsupervised document embedding models in weakly-supervised classification. That is, lots of unlabeled documents with only few labels attached to some of them, which is a realistic scenario that unsupervised embedding could be particularly useful.

2 Related Work

Here we give an overview of other related methods on learning unsupervised text representations. Besides BoW and TF-IDF, Latent Dirichlet Allocation models (Deerwester et al., 1990; Blei et al., 2003) leverage the orthogonality of high-dimensional BoW features by clustering a probabilistic BoW for latent topics.

Several models extend from Word2Vec (Mikolov et al., 2013), using context-word predictions for training document embedding end-to-end. PV (Le and Mikolov, 2014) keeps

a document embedding matrix in memory and is jointly trained. The required training parameters are linear to the number of documents and thus prohibit PV from being trained on a large corpus. Moreover, expensive inference for new documents is required during testing. To address the above concerns, Doc2VecC (Chen, 2017) combines PV and a denoising autoencoder (DEA) (Chen et al., 2012) with BoW vectors as global document information instead. The final document embedding are then produced by simply averaging the jointly trained word embedding.

Another thread of work uses two-stage pipelines to construct sentence/document embedding from pre-trained word embedding. Arora et al. (2017) propose post-processing weighting strategies on top of word embedding to build sentence representations. WME (Wu et al., 2018) propose a random feature kernel method based on distance between pairs of words, which also shows inter-document information helps. However, the cost scales with the size of training samples such that it is hard to be applied on large-scale dataset.

There have been more embedding work on sentences compared to documents. These approaches mostly learn the sentence embedding by modeling the sentence-level (Kiros et al., 2015; Tang et al., 2017b,a; Logeswaran and Lee, 2018) or word-level (Pagliardini et al., 2018; Kenter et al., 2016; Arora et al., 2018) distribution hypothesis (Harris, 1954; Polajnar et al., 2015) in a large ordered corpus. We note that the main difference between learning embedding for sentences and documents is that documents are not ordered in a corpus. Some other work model sentences with RNN autoencoders (Hill et al., 2016a; Gan et al., 2017). Documents often refer to long-length text containing multiple sentences, which might be hard to model with RNNs (Pascanu et al., 2013; Jing et al., 2017) and time-consuming on large corpus.

3 Model and Design Rationale

To facilitate downstream tasks, a document embedding is required to compress useful features into a compact representation. It is not an easy task to learn discriminable features unsupervisedly since validation information is not accessible for training. We first introduce some notations:

- $\mathcal{V}$: the training corpus vocabulary of size $|\mathcal{V}|$;
- $\mathcal{X} = \{X_1, \cdots, X_n\}$: a training corpus of document size $n = |\mathcal{X}|$, in which each document X_i is a set of sentences $\mathcal{S}_i$;
- $\mathcal{S}_i = \{\mathbf{s}_i^1, \cdots, \mathbf{s}_i^{|x_i|}\}$: a document divided into a set of sentences, of set size $|\mathcal{S}_i|$, in which each sentence $\mathbf{s}_i^j \in \mathcal{R}^{|\mathcal{V}| \times T_j}$ contains a sequence of variable length T_j of word one-hot vectors $\mathbf{w}_j^1, \cdots, \mathbf{w}_j^{T_j}$, each in $\mathcal{R}^{|\mathcal{V}| \times 1}$. $\mathcal{S}$ is the set of total sentences $\bigcup_{i=1}^{n} \mathcal{S}_i$ in the training corpus, of size $m = |\mathcal{S}|$;
- h_w: the size of the word embedding and $\mathbf{U} \in \mathcal{R}^{h_w \times |\mathcal{V}|}$: the word embedding projection matrix. We use $\mathbf{u}_w$ to denote the column in $\mathbf{U}$ for word $\mathbf{w}$
- h_s: the size of the sentence embedding and $\mathbf{e_s} \in \mathcal{R}^{h_s}$: the embedding of sentence $\mathbf{s}$.
- $\mathbf{d}_i \in \mathcal{R}^{h_s}$: document X_i's embedding.

Our goal is to learn a function $F : \mathcal{X} \to \mathcal{R}^{h_s \times n}$ that maps document X_i to $\mathbf{d}_i$ unsupervisedly.

Next, we formulate how SDDE represents a document, then introduce our self-discriminative learning procedure we use to train it.

3.1 Document Representation

We consider a document as **mean of sentences**, i.e., breaking a document into several subsequences. We demonstrate several benefits of the design in SDDE. First, decomposing long documents into shorter but reasonable semantic unit (e.g., sentences) makes encoding easier and faster since they can be processed in parallel. Similar concepts of modeling documents hierarchically have shown benefits in some supervised tasks such as text classification (Yang et al., 2016). It also makes the model insensitive to document length, which is important because length varies greatly in real documents (see Table 1).

In training, we further propose to represent a document *during training* using the average of only a subset of its sentences. This special **sentence-level dropout** is beneficial for training by creating up to $\binom{|X_i|}{q}$ combinations for each document, where q is the number of sentences to keep. This enforces the local sentence representations to capture global information of a document by representing it with a subset of sentences in it. The word embedding is used as globally shared building blocks for sentence embedding.

For a document $X_i = \{\mathbf{s}_i^j\}$, or $\mathcal{S}_i$, the embedding is derived from averaging the respective representations of subsequences. Noted as:

$$\mathbf{d}_i = \frac{1}{q} \sum_{\substack{j=1, \\ \mathbf{s} \sim P_{\mathcal{S}_i}(\mathbf{s})}}^{q} \mathbf{e}_{\mathbf{s}}^j, \tag{1}$$

where

$$\mathbf{e}_{\mathbf{s}} = E(\mathbf{s}), \tag{2}$$

where a sentence encoder E is introduced to produce sentence embedding for $\mathbf{s}_i^j$. In practice, sentences can be obtained by simply segmenting documents with punctuation. In testing, the document embedding is obtained by averaging all the sentences in which:

$$\mathbf{d} = \frac{1}{|\mathcal{S}_i|} \sum_{\forall \mathbf{s} \in \mathcal{S}_i} \mathbf{e}_{\mathbf{s}}. \tag{3}$$

We note that averaging subsequences differs from averaging of words in two aspects. First, each sentence is encoded individually before being averaged, allowing incorporation of word order into design rationale at least in a reasonable range. Second, subsequences may have different lengths that reveal syntactic information. To illustrate, BoW/mean-of-word models suffer from ambiguously modeling two different documents which are similar in word distributions but differ in some aspects of interest. Mean-of-sentence model avoids such concern by modeling documents at the sentence level. It could be expected that it is much less likely to find two documents with similar sentence distribution than similar word distribution. Mean-of-sentences formulation can be smoothly reduced to mean-of-word models (by treating each word as a sentence) or pure sequence models (by treating each document as a very long sentence).

3.2 Self-Discriminative Learning

Unlike PV or Doc2VecC which emphasize modeling distributional information within individual documents, we model relations across documents. The basic idea is that we hope to learn an embedding for each sentence in the document as well as a discriminator that determines whether a sentence belongs to a document. Self-discriminative learning uses a discriminator network D to determine whether a sentence belongs to a document. The aim is to learn a suitable embedding and a good discriminator to determine if a sentence belongs to a document. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Self-Discriminative Learning for Unsupervised Document Embedding

Input: Documents $\mathcal{X} = \{X_i\}_1^n$, p, k, h_w, h_s.
Output: Function $F : \mathcal{X} \rightarrow \mathcal{R}^{h_s}$ that maps text of a document to an embedding $\mathbf{d} \in \mathcal{R}^{h_s}$.

Compute $\mathcal{S} = \bigcup_{i=1}^{n} \mathcal{S}_i$ from set $\mathcal{S}_i$ of each X_i.
Create and initialize D, $\mathbf{U}$.
Create and initialize E as in Eq. 5 or Eq. 6.
while not converge **do**:
  for $i = 1, \ldots, n$ **do**
    Sample $\mathbf{s}$ from $\mathcal{S}_i$.
    Get $\mathbf{e}_{\mathbf{s}}^p \leftarrow E(\mathbf{s})$ as a positive sample.
    Sample $\mathbf{s}_i^1, \ldots, \mathbf{s}_i^q$ from $\mathcal{S}_i \setminus \{\mathbf{s}\}$.
    for $j = 1, \ldots, q$ **do**
      Get $\mathbf{e}_{\mathbf{s}}^j \leftarrow E(\mathbf{s}_i^j)$
    end for
    Get $\mathbf{d}_i$ with Eq. 1 given $\{\mathbf{e}_{\mathbf{s}}^j\}_{j=1}^q$.
    for $\ell = 1, \ldots, k$ **do**
      Sample $\mathbf{s}'_\ell$ from $\mathcal{S} \setminus \mathcal{S}_i$.
      Get $\mathbf{e}_{\mathbf{s}'}^\ell \leftarrow E(\mathbf{s}'_\ell)$ as a negative sample.
    end for
    Compute Eq. 4 given $\mathbf{d}_i$, $\mathbf{e}_{\mathbf{s}}^p$, $\{\mathbf{e}_{\mathbf{s}'}^\ell\}_{\ell=1}^k$.
    Backprop and update for E, D, $\mathbf{U}$.
  end for
end while
Return $F(E, \mathbf{U})$,

We propose an objective that explicitly optimizes SDDE towards representing a document with mean of (encoded) sentences. To optimize the discriminator D, we formulate it as a binary classifier that takes pairs of document embedding $\mathbf{d}$ of a document X_i and a sentence embedding $\mathbf{e}_{\mathbf{s}}$, $(\mathbf{d}, \mathbf{e}_{\mathbf{s}})$, as inputs. The discriminator is asked to discriminate using $\mathbf{d}$ whether the sentence $\mathbf{s}$ belongs to the document X_i or the other documents $X' \in \mathcal{X} \setminus \{X_i\}$. The loss then becomes:

$$log(1 - D(\mathbf{d}, \mathbf{e}_{\mathbf{s}}^p)) + \sum_{\ell=1}^{k} \mathop{\mathbb{E}}_{\substack{\mathbf{s}' \sim P_{\mathcal{S}}, \\ \mathbf{s}' \notin \mathcal{S}_i}} \left[log(D(\mathbf{d}, \mathbf{e}_{\mathbf{s}'}^\ell)) \right], \tag{4}$$

for each document with one positive sample $\mathbf{e}_{\mathbf{s}}^p$ and k negative samples of sentences $\mathbf{e}_{\mathbf{s}'}$, where $\mathbf{s}'$ are not in the sentence set $\mathcal{S}_i$ of X_i, as $\mathbf{s}' \notin \mathcal{S}_i$. Note that $\mathbf{e}_{\mathbf{s}}^p$ is not used for $\mathbf{d}$ otherwise it would be trivial to be solved by the discriminator.

The spirits of self-discriminative learning can

be understood as unsupervisedly mimicking supervised inference without knowledge of any label information by treating sentences from other documents as fake/negative samples. One main concern that it is possible to find similar sentences in two different documents. Our discriminator particularly addresses this issue by optimizing for the most discriminative sentences rather than similar ones that might not be critical to shape the embedding. To minimize the loss, the encoder would tend to preserve the most essential feature to facilitate the discriminator to push away any two documents, which should encourage the embedding points spread even more widely across the space. This in turn should result in more ease in downstream tasks: for example in learning a decision hyperplane in a classification task.

3.3 Sentence Encoder

Next, we narrow down to sentence encoder E. Given a sequence of word one-hot vectors as a sentence $\mathbf{s} = [\mathbf{w}^1, \dots, \mathbf{w}^T]$, we project them into an embedding layer $\mathbf{U}$ to retrieve their corresponding word embedding. Note that the word embedding are trained jointly.

Our first method uses:

$$E(\mathbf{s}) = \phi(ReLU(G(\mathbf{U}\mathbf{s}^t))), \tag{5}$$

where G is a single-layer RNN encoder using GRU cells to process the word embedding sequences and ϕ is a linear transform for dimension h_s of sentence embedding.

Our second method, we use a schema of mean-of-word for advantage of fast generation, we average the word embedding $\mathbf{w}$ within a sentence $\mathbf{s}$ along time axis as AVG encoder:

$$E(\mathbf{s}) = \phi(ReLU(\frac{1}{|\mathbf{s}|}\sum_{i=1}^{|\mathbf{s}|}\mathbf{U}\mathbf{w}_i^t)), \tag{6}$$

Let us stress that the role of encoder E is to extract local feature from every sentence, and the overall objective encourages SDDE to represent documents as mean of sentence embedding.

3.4 Discriminator

An undesired pitfall comes from a learned weak encoder with a powerful discriminator causing the embedding produced by the encoder useless for downstream tasks. To avoid such a pitfall, we adopt lightweight network structures for discriminators. For the IMDB datasets, we find inner product $(\mathbf{d}\mathbf{V})^t E(\mathbf{s})$ with a learnable matrix $\mathbf{V}$ sufficient. For the other datasets in Table 1, two fully-connected layers with ReLU activations in latent are used.

Dataset	#Class	#Train / #Test	Doc Length	Sent Length
IMDB	2	75k / 25k	124.6±8,856.7	11.6±105.5
Yelp P.	2	560k / 38k	70.0±4,117.8	8.2±48.7
AG's News	4	120k / 7.6k	27.2±66.1	11.8±66.7
DBPedia	14	560k / 70k	32.8±231.3	9.7±68.6

Table 1: Statistics of the datasets. Length of document and sentence in words (mean±variance), which could be high for real-world scenarios such as online reviews.

Dataset	k	p	h_w	h_s
IMDB	1	3	100	100
Yelp P.	1	3	300	500
AG's News	3	1	100	100
DBPedia	4	2	300	500

Table 2: Hyperparameters used in experiments. The document embedding trained with the same hyperparameters are used for all the evaluations without task-specific tuning.

3.5 Sampling Sentences

We relate our method to the Negative Sampling (Mikolov et al., 2013) technique which is a simplified objective of softmax approximation (Mikolov et al., 2013; Mnih and Teh, 2012; Zoph et al., 2016). Negative sampling has been used as an efficient and effective technique in learning word embedding. We reformulate it to train document embedding by sampling in sentence level, which is easy to implement and efficient to train just like Word2Vec (Mikolov et al., 2013).

In practice, when training with mini-batches the documents for negative samples are from the same mini-batch, which requires small extra computation efforts.

SDDE requires a similar number of parameters as Doc2VecC does, but much less than PV. In addition, the sentence encoder is flexible and can incorporate other techniques of text processing such as attention methods.

4 Experiments

4.1 Setup

Public datasets on sentiment analysis and topic classification across diverse domains including online reviews, news, and Wiki pages are used including IMDB dataset (Maas et al., 2011) and the others from Zhang et al. (2015). Table 1 provides

a summary. Only the training splits are used in training embedding with subsampled training split for cross-validations.

We preprocess the datasets by normalizing the text to lower class and replacing words appearing less than 10 times with a *UNK* token. Out-of-vocabulary words in testing set are also replaced. All our baseline models use the same input data. To define the sentences for experiment, we utilize the sentence tokenizer from NLTK. For the documents containing only one sentence we simply divide it into multiple subsequences for sampling. We use RMSProp method for optimization. Dropout 50% of input to the discriminator. Weights are random-uniformly initialized between [-1, 1].

The other hyperparameters are summarized in Table 2. All the models use the same embedding size for fair comparison. The trained document embedding are used for all the evaluations without specific tuning.

4.2 Evaluation

Generally, it is not easy to evaluate an unsupervised embedding model. In Section 4.3 and 4.4, we evaluate the performance on standard document classification following the common practice used by previous work (Chen, 2017; Wu et al., 2018; Arora et al., 2017): a classification task with a linear SVM (Fan et al., 2008) trained on the labels in each dataset. Next, we study unsupervised document embedding on two novel aspects. In Section 4.5, we study a weakly-supervised classification setting that fits the realistic scenario of using unsupervised embedding with only a few labels. In Section 4.6, we provide a metric to evaluate the effectiveness of modeling inter-document information.

4.3 Sentiment Analysis with IMDB Dataset

We first compare our models with the others state-of-the-art competitors. RNN-LM (Mikolov et al., 2010) and Skip-thought (Kiros et al., 2015) are RNN-based. SIF (Arora et al., 2017), W2V-AVG (Mikolov et al., 2013), and WME (Wu et al., 2018) are two-stage approach that post-processing on word embedding. We collect the results reported on the widely-used benchmark sentiment classification dataset IMDB. For PV, we use Gensim implementation (Řehůřek and Sojka, 2010); versions of both Distributed Memory (DM) and Distributed Bag of Words (DBoW) are reported. For different encoders in SDDE, AVG is for averaging word embedding and RNN is for the RNN encoder.

https://www.nltk.org/

Model	Error%
Skip-thought* (Kiros et al., 2015)	17.4
SIF (GloVe) (Arora et al., 2017)	15.0
RNN-LM* (Mikolov et al., 2010)	13.6
W2V-AVG* (Mikolov et al., 2013)	12.7
DEA* (Chen et al., 2012)	12.5
PV-DM	20.0
PV-DBoW	12.0
WME (Wu et al., 2018)	11.5
Doc2VecC* (Chen, 2017)	11.7
SDDE-AVG	**10.6**
SDDE-RNN	**10.2**

Table 3: Sentiment Classification on IMDB Benchmark. *Results are collected from Chen (2017).

Pred	SDDE-RNN		Doc2VecC	
	True	False	True	False
#Sent	14.8	14.7	14.5	14.8

Table 4: Mean of #sentence per document in IMDB dataset, in groups of classification correctness.

Self-Discriminative Learning is Effective From the experiment result in Table 3, we can see that our self-discriminative learning is effective and superior on the document embedding models for both AVG and RNN versions. SDDE-RNN achieves best accuracy on IMDB dataset 1.5% margin against Doc2VecC.

Study the Property of SDDE Unlike previous work modeling documents on the word or short context level, SDDE operates on the sentence level. We study the false and true predictions output by SVM upon SDDE in comparison with Doc2VecC. Table 5 show some examples that have the largest difference. We observed SDDE can better capture contradicting or contrasting opinions. We observe some wrong predictions (Row 3) are due to the ambivalent reviews. SDDE is insensitive to the number of sentences; we found the effect of the number of sentences per document was trivial as shown in Table 4.

4.4 Large-Scale Document Classification on More Dataset

Next, we borrow some public large-scale dataset in Table 1 to further validate the effectiveness of SDDE compared to the other models. For Doc2vecC and SIF, we use the code from the au-

Label: 1 SDDE: 0.89 Doc2VecC: 0.27	i don t even like watching those late night talk shows , but i found this one really interesting . i imagine it s probably close to the truth — it feels like an honest account , if that means anything . kinda feel for the people somewhat when you watch it . a nice movie for a saturday night .
Label: 0 SDDE: 0.12 Doc2VecC: 0.75	i m a boorman fan but this is arguably his least successful film . comedy has never been his strong suit , and here his attempts at screwball farce are clumsily done . still , it s almost worth seeing for boorman s eye for talent : this is one of uma thurman s first starring roles , and as always she is ravishing to watch . on a sad side note boorman wrote the script with his daughter , <UNK> who died a couple years ago .
Label: 0 SDDE: 0.77 Doc2VecC: 0.10	michael dudikoff stars as joe armstrong a martial artist who fights ninjas who are stealing weapons from the u s army , in this entertaining yet admittedly brainless martial arts actioner , which is hampered by too many long pauses without action , but helped by some high energy action <UNK> as well as steve james performance .

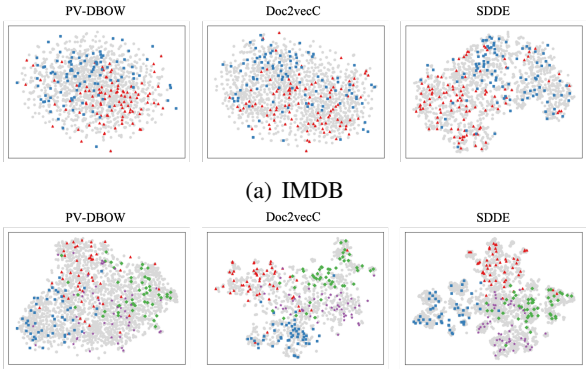
Table 5: IMDB Examples with scores 0 to 1 for negative to positive assigned by SVM.

Model	Yelp P.	AG	DBP.
PV-DM	17.6	39.6	21.6
PV-DBoW	12.1	16.8	10.4
Word2Vec AVG	8.0	14.2	2.7
SIF	8.4	13.8	2.9
Doc2VecC	7.4	10.4	2.0
SDDE-RNN	8.7	12.7	8.6
SDDE-AVG	**6.7**	**9.8**	**1.8**

Table 6: Testing error rate (%) with standard classification on public datasets. Bold text indicates passing hypothesis test with p-value < 0.05 with different random initialization.

Model	IMDB	Yelp P.	AG	DBP.
Doc2VecC	12.5	11.5	13.9	3.6
PV-DBoW	13.7	18.6	13.9	13.1
SDDE	**11.7**	**10.0**	**10.0**	**2.9**

Table 7: Testing error (%) in weakly-supervised setting. Only 1k labeled data per class were used. 4.5).



(a) IMDB

(b) AG's news

Figure 1: t-SNE embedding Visualizations of weakly-supervised learning experiments in Section 4.5. Colored points are labeled training data in the experiments and gray points are the unlabeled testing documents.

thors. We use SIF to generate document embedding with Word2Vec trained on each dataset as its inputs.

Results are shown in Table 6. SDDE-AVG performs slightly better across different dataset. We hypothesis SDDE gets larger improvement on IMDB dataset since SDDE can handle longer documents better by exploiting sentence embeddings. On the other hand, the RNN version of SDDE performs significantly worse than the word-averaging version. We may remind the reader that state-of-the-art unsupervised document embedding models are not RNN-based. The effects of word order are still unclear. Wieting and Gimpel (2017) provides a study of sentence embedding. We hypothesize that it may be difficult for an RNN encoder to learn to incorporate multi-domain information in datasets with many classes (e.g., DBpedia) unsupervisedly. This would be our future work.

4.5 Classification with Few Labeled Data

Next, we consider a more real-world weakly-supervised learning scenario: classification on the datasets we have used in previous experiments, but this time only when very few labels are available.

We hypothesize that SDDE is particularly useful for classification with few labels since the self-discriminative learning has exploited the possible features to map the text onto the embedding space properly during the representation learning phase. The embedding is expected to be more discriminable to facilitate finding the classification decision hyperplanes with fewer labeled data.

We randomly sample equal number of instances from each class to train a SVM and verify with the whole testing set. PV-DBoW, Doc2VecC, and SDDE-AVG are examined in this experiment. We use the same pre-trained document embedding as in the previous experiments. We repeat the whole procedure 30 times and report the means and tune the penalty parameter C to find the best value for each model. Results in Figure 2 and Table 7 show SDDEs outperform PV and Doc2VecC. We visualize the training points with t-SNE. As shown in Figure 1, SDDE seems to be able to spread the embedded data more widely, which eventually leads to better usage of scarce data for classification.

https://github.com/mchen24/iclr2017
https://github.com/PrincetonML/SIF

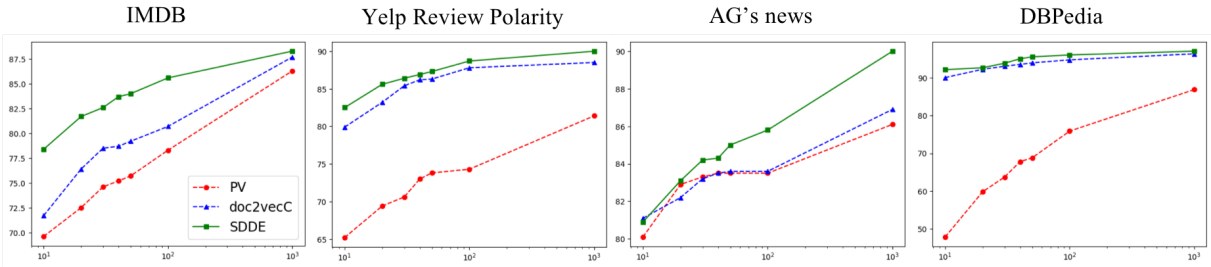


Figure 2: Weakly-supervised learning on datasets in Table 1. Training with [10, 20, 30, 40, 50, 100, 1000] instances per class (X-axis) and computing the accuracy on the whole testing set (Y-axis).

Model	IMDB			Yelp P.			AG			DBP.		
Metric	A	E	**A-E**	A	E	**A-E**	A	E	**A-E**	A	E	**A-E**
Doc2VecC	.78	.76	.02	.52	.49	.03	.57	.46	.11	.60	.40	.20
Word2Vec AVG	.86	.85	.01	.54	.51	.03	.57	.44	.13	.75	.61	.14
PV-DBoW	.28	.27	.01	.11	.10	.01	.71	.60	.09	.45	.36	.09
SDDE	.36	.27	**.09**	.14	.08	**.06**	.20	.01	**.19**	.58	-.03	**.61**

Table 8: Distances of Intra & Inter-class cosine similarity. A for IntraCos and E for InterCos, note that they cannot be compared across different models. Instead, distance A-E defined in Equation 7 is reported to study a method's effectiveness of modeling inter-document features. The higher the number the better.

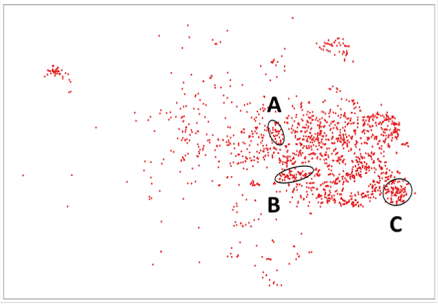
4.6 Distance of Intra&Inter-Class Pairwise Cosine Similarity

Definition We examine our assumption of the ability of SDDE to model inter-document feature. Similar to (Hill et al., 2016b), we consider pairwise cosine similarity between documents with topic labels, this allows us to quantitatively evaluate unsupervised document embedding at inter-document level. Our assumption is that: if pairwise similarity between documents is calculated based on different kinds of embedding, the better embedding results should comply with the properties of both high similarities between those documents within the same underlying class, denoted as $IntraCos(d, d')$ and low similarities between document pairs from different classes, or $InterCos(d, \tilde{d})$. The mean distance:

$$mean(IntraCos) - mean(InterCos), \quad (7)$$

is considered as our metric to avoid simply maximizing IntraCos or minimizing InterCos.

SDDE Provides High Separation Table 8 shows the evaluation. The distances (Eq. 7) for the baseline models are small, which support our assumption that these methods are not able to model inter-document features properly. On the other hand, distances for SDDE are significantly larger. With the classification experiments, we believe SDDE better preserves meaningful inter-document features. Figure 3 shows some meaningful clusters in SDDE in the World class as cohesive sub-classes.



A: Earthquake hits Indonesian island
Ship hits Japan breakwater
Morocco mosque collapse kills 10
Deadly fire sweeps through China mines

B: Cheney: Kerry 'Wrong Choice' for President
P. Diddy Takes Vote Drive to Swing States
Parties call for postponement of elections
Confident Bush Outlines Ambitious Plan for 2nd Term

C: Iraq's Sadr Orders Fighters to Lay Down Weapons
Afghan Forces Arrest 2 Taliban Leaders
Two US soldiers killed
Death toll climbs in Baghdad blast

Figure 3: SDDE-AVG t-SNE visualization of class "World" in AG News dataset testing set. Titles of documents are shown. Meaningful clusters such as (A) disaster, (B) election, and (C) war.

5 Conclusion

Compared to mainstream unsupervised document embedding models (trained to perform predictions on the lexicon level) SDDE embeddings capture information at the inter-document level, as they are trained to maximize the distance between a sentence and a corresponding document. We hope the underlying idea of SDDE offers the document-embedding community a new investigation direction. Self-discriminative learning shows potential

for real-world scarcely-labeled scenarios, and our future work will focus on joint training of representations for semi-supervised learning.

Acknowledgements

This material is based upon work supported by Microsoft Research Asia (MSRA) grant, and by Taiwan Ministry of Science and Technology (MOST) under grant number 108-2634-F-002 -019.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*.

Sanjeev Arora, Yingyu Liang, Tengyu Ma, Mikhail Khodak, Nikunj Saunshi, and Brandon Stewart. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 12–22.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Minmin Chen. 2017. Efficient vector representation for documents through corruption. In *ICLR*.

Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *ICML*.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *NIPS*, pages 3079–3087.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *JMLR*, pages 1871–1874.

Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. 2017. Learning generic sentence representations using convolutional neural networks. In *EMNLP*.

Zellig S. Harris. 1954. Distributional structure. *WORD*, 10(2-3):146–162.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016a. Learning distributed representations of sentences from unlabelled data. In *NAACL HLT*, pages 1367–1377.

Felix Hill, KyungHyun Cho, Anna Korhonen, and Yoshua Bengio. 2016b. Learning to understand phrases by embedding the dictionary. *TACL*, 4:17–30.

Li Jing, Caglar Gulcehre, John Peurifoy, Yichen Shen, Max Tegmark, Marin Soljacic, and Yoshua Bengio. 2017. Gated orthogonal recurrent units: On learning to forget. *CoRR*, abs/1706.02761.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese CBOW: optimizing word embeddings for sentence representations. In *ACL*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*, pages 3294–3302.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *ICLR*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2015. Deep captioning with multimodal recurrent neural networks (m-rnn). In *ICLR*.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*

NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 528–540.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318.

Hieu Pham, Minh-Thang Luong, and Christopher D. Manning. 2015. Learning distributed representations for multilingual text sequences. In *NAACL-HLT*, pages 88–94.

Tamara Polajnar, Laura Rimell, and Stephen Clark. 2015. An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 1–11. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Shuai Tang, Hailin Jin, Chen Fang, Zhaowen Wang, and Virginia R. de Sa. 2017a. Rethinking skip-thought: A neighborhood based approach. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 211–218.

Shuai Tang, Hailin Jin, Chen Fang, Zhaowen Wang, and Virginia R. de Sa. 2017b. Trimming and improving skip-thought vectors. *CoRR*, abs/1706.03148.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 90–94.

John Wieting and Kevin Gimpel. 2017. Revisiting recurrent networks for paraphrastic sentence embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 2078–2088.

Lingfei Wu, Ian En-Hsu Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J. Witbrock. 2018. Word mover's embedding: From word2vec to document embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4524–4534.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL HLT*, pages 1480–1489.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*, pages 649–657.

Barret Zoph, Ashish Vaswani, Jonathan May, and Kevin Knight. 2016. Simple, fast noise-contrastive estimation for large RNN vocabularies. In *NAACL HLT*, pages 1217–1222.