Information Sciences 430–431 (2018) 216–227

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Entity mention aware document representation

Hongliang Dai, Siliang Tang*, Fei Wu, Yueting Zhuang

Zhejiang University, Hangzhou, Zhejiang, China

ARTICLE INFO

Article history:
Received 19 February 2017
Revised 10 October 2017
Accepted 17 November 2017
Available online 20 November 2017

Keywords:
Distributed representation
Text clustering
Text classification
Entity linking

ABSTRACT

Representing variable length texts (e.g., sentences, documents) with low-dimensional continuous vectors has been a topic of recent interest due to its successful applications in various NLP tasks. During the learning process, most of existing methods tend to treat all the words equally regardless of their possibly different intrinsic nature. We believe that for some types of documents (e.g., news articles), entity mentions are more informative than ordinary words and it can be beneficial for certain tasks if they are properly utilized. In this paper, we propose a novel approach for learning low-dimensional vector representations of documents. The learned representations captures information of not only the words in documents, but also the entity mentions in documents and the connections between different entities. Experimental results demonstrate that our approach is able to significantly improve text clustering, text classification performance and outperform previous studies on the TAC-KBP entity linking benchmark.

1. Introduction

Learning low-dimensional vector representations for documents can be an effective approach in many NLP tasks and has the potential to outperform traditional representation methods like bag-of-words (BoW) [9,15,19]. However, despite the fact that different words have different properties and are of different importance to the whole document, such methods always tend to treat all words equally.

In particular, existing document representation learning methods do not distinguish named entity mentions (e.g., the mention of “Hillary Clinton” in a document) with ordinary words. Entity mentions occur frequently in various types of documents and are usually more informative than most ordinary words. In news articles, mentioned entities such as persons, organizations and locations provide essential information about *who* and *where* of the reported events. Two documents are more likely to have similar content if some of the entities mentioned by them are the same. Sometimes, knowing what entities are mentioned, we can even infer the possible topics of a document. For example, if a news article mentioned both “Hillary Clinton” and “Donald Trump”, then we know there is a high probability that this article is about the US presidential election of 2016. Thus entity mention information can help to better capture the semantic similarities between documents while learning document representations. Moreover, different entities may be related with each other. For example, a person has connections with many other persons and is related to many locations and organizations. Documents that mention different but related entities are also more likely to have similar content. In order to leverage this property, the relatedness between different entities should also to be considered while learning representations for documents.

* Corresponding author.
E-mail addresses: hldai@zju.edu.cn (H. Dai), siliang@zju.edu.cn (S. Tang), wufei@zju.edu.cn (F. Wu), yzhuang@zju.edu.cn (Y. Zhuang).

https://doi.org/10.1016/j.ins.2017.11.032

Therefore, we believe that it is possible to improve the quality of document representations by capturing the entity mention information of documents and the relatedness between different entities. Document representations learned with such information may achieve better performance when applied to tasks such as text clustering and text classification. They are also very suitable for the task of entity linking, which aims to map the mentions in a document to their referred entities in the referent knowledge base, since existing research [6,10,21,30] has already shown that while performing entity linking for a mention, other mentions in the same context can be particularly helpful for the inference.

In this paper, we propose a novel approach to learn distributed representations of documents that are aware of the entity mentions in documents. We name our approach EMADR (Entity Mention Aware Document Representations). EMADR generalizes the PV-DBOW model proposed by Le and Mikolov [19] to make it possible to incorporate multiple types of related information into document representations. The learned document representations captures three types of information: what words are used in each document, what entities are mentioned in each document and the relatedness between different entities.

The main contributions of this paper are:

- We propose EMADR, which to the best of our knowledge, is the first document representation learning method that leverages entity mention information.
- We apply EMADR to entity linking with a neural network model. Compare with some existing neural network methods [12,33] for this task, it has the advantage of only requiring a small amount of training data.
- We study the performance of EMADR by conducting experiments on text clustering, text classification and entity linking. We find that EMADR is able to significantly improve text clustering and classification performance. Its application in entity linking also beats previous studies on the TAC-KBP entity linking benchmark.

The rest of this paper is structured as follows: In Section 2 we discuss the technical details of learning document representations with our approach. Section 3 shows how we apply the learned representations to the task of entity linking. In Section 4, we conduct a series of experiments on text clustering, text classification and entity linking. Finally, we introduce some related works in Section 5 and dummyTXdummy- concludes our work in Section 6.

2. Entity mention aware document representations

As previously mentioned in the introduction, for each document, we want its representation to capture both what words are used and what entities are mentioned. We also aim to capture the relatedness between different entities so that the representations of documents with different but related entities may also be similar. In order to do this, we generalize the well-known document representation learning model PV-DBOW [27] by introducing the concept of *prediction lists*. Then we employ this idea to learn document representations based on constructing three prediction lists.

2.1. Embedding method based on prediction lists

We start by introducing the PV-DBOW model. PV-DBOW represents each document with a dense vector that is trained to model the distribution of words in the document. Given a set of documents D and a set of words W, It uses a softmax to model the probability of observing word w in document d:

$$p(w|d) = \frac{\exp(v_w'^{\top} v_d)}{\sum_{\hat{w}\in W} \exp(v_{\hat{w}}'^{\top} v_d)}, \tag{1}$$

where v_d is the vector representation of d, v_w' is the weight vector with respect to w. Eq. (1) has the property that if a word w occurs frequently in document d, then $v_w'^{\top} v_d$ should be large, which usually means v_d will be similar with v_w'.

Let $w_1^d, w_2^d, \ldots, w_{N(d)}^d$ be the sequence of words in document $d \in D$, where $N(d)$ is the number of words in d. The objective of PV-DBOW is to maximize the log probability

$$I = \sum_{d\in D} \sum_{i=1}^{N(d)} \log p(w_i^d|d). \tag{2}$$

The document representations learned with Eqs. (1) and (2) will preserve second-order proximity [34], which means that if two document use similar words, then their corresponding representations will also be similar.

In order to generalize this model, we use $\langle x, y\rangle$ to denote a positive sample of observing y given x, which means $p(\langle x, y\rangle) = p(y|x)$. For each document $d \in D$, we add $\langle d, w_1^d\rangle, \langle d, w_2^d\rangle, \ldots, \langle d, w_{N(d)}^d\rangle$ in a list L, then maximizing Eq. (2) equals to maximizing the log probability of observing all the samples in L. We call $\langle x, y\rangle$ a prediction sample and L a prediction list. This shows that we can get the same objective as the PV-DBOW model based on a prediction list constructed from the documents.

Moreover, we can also train the model based on the constructed list. Suppose we randomly draw T samples from L and denote them as $\langle \hat{d}_1, \hat{w}_1\rangle, \langle \hat{d}_2, \hat{w}_2\rangle, \ldots, \langle \hat{d}_T, \hat{w}_T\rangle$. Then it can be easily shown that when T is sufficiently large,

$$\sum_{i=1}^{T} \log p(\langle \hat{d}_i, \hat{w}_i\rangle) \approx \frac{T}{\sum_d N(d)} \cdot \sum_{d\in D} \sum_{i=1}^{N(d)} \log p(w_i^d|d). \tag{3}$$

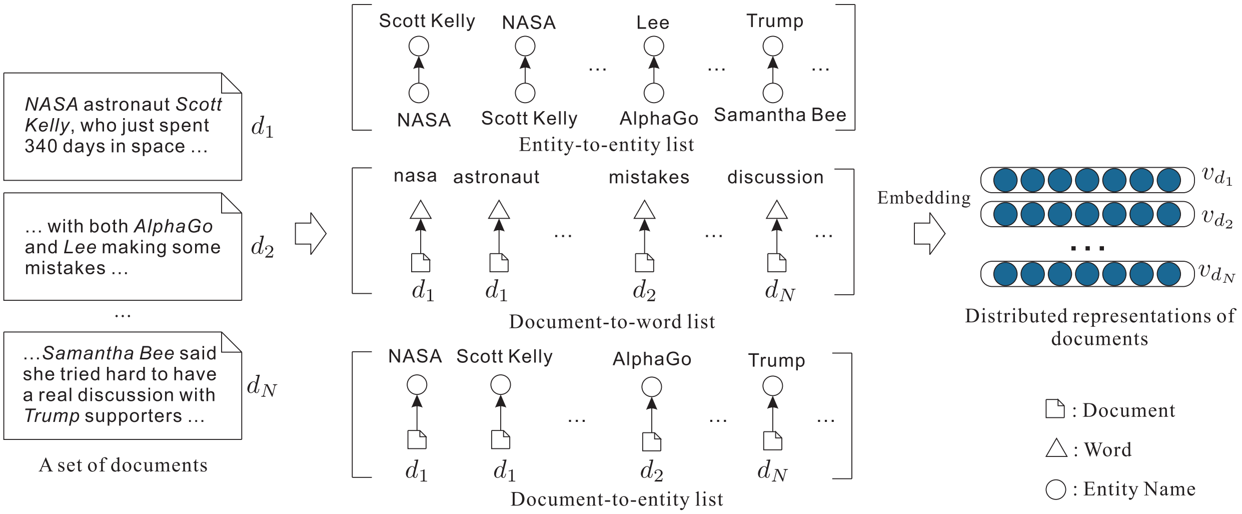


Fig. 1. Learning representations for a set of documents $d_1, d_2, \ldots d_N$. With these documents, a *document-to-word* list, a *document-to-entity* list and a *entity-to-entity* list are constructed. The document representations $v_{d_1}, v_{d_2}, \ldots v_{d_N}$ are learned based the lists.

This means we can train the model by drawing a sufficient number of samples from L randomly and treating them as training samples.

Furthermore, if there are multiple types of information that can be expressed in such a form. We may construct a prediction list for each of them. The final objective can be the weighted sum of the objectives with respect to each list. A joint training procedure can be achieved by drawing a sufficient number of samples from all of the constructed lists randomly and treating them as training samples.

To apply this idea, given a set of documents, we first construct three prediction lists to capture three types of information: (1) a document-to-word list to capture what words are used in each document, (2) a document-to-entity list to capture information of the entity mentions in each document, (3) an entity-to-entity list to capture the relatedness between entities. After these lists are built, we then learn document representations based on them. Fig. 1 illustrates how we learn representations for documents with this procedure. Next, we show in detail how the three lists are constructed and how the document representations are learned.

2.2. Prediction list construction

First, note that constructing the document-to-entity list and the entity-to-entity list requires to find entities in documents. We apply an NER tagger to find entity mentions in documents and then use different entity names to represent different entities. It is possible to identify which entity each mention actually refers to with an entity linking system. But this will introduce extra complexity and slow down the process.

Let D be the given set of documents; let W and E be the set of words and the set of entity names that occurs in the documents in D respectively. We construct the three lists for D as follows:

Document-to-word list. For each document $d \in D$, assume that $w_1, w_2, \ldots, w_r$ are the sequence of words in d, then add $\langle d, w_1\rangle, \langle d, w_2\rangle, \ldots, \langle d, w_r\rangle$ into the document-to-word prediction list L_{dw}.

Document-to-entity list. For each document $d \in D$, assume that $\epsilon_1, \epsilon_2, \ldots, \epsilon_s$ are the sequence of entity names that occurred in d, then add $\langle d, \epsilon_1\rangle, \langle d, \epsilon_2\rangle, \ldots, \langle d, \epsilon_s\rangle$ into the document-to-entity prediction list L_{de}. Entity mentions can be extracted with named entity recognition (NER). There are many existing NER methods [7,17,22,23], some neural network approaches based on bidirectional LSTMs are state-of-the-art [17,23]. However, we find in our experiments that the influence to the learned document representations caused by the small performance differences of the used NER method can almost be neglected. Thus we believe the choice of NER method is not important for our approach.

Entity-to-entity list. Two entities that are mentioned in a same sentence are likely to be related with each other. Thus every time two entity names ϵ_i and ϵ_j occur in a same sentence, add $\langle \epsilon_i, \epsilon_j\rangle$ into the entity-to-entity prediction list L_{ee}.

Note that these three lists are possibly unbalanced. The document-to-word list usually has a stronger influence to the learned representations than the other two lists. To resolve this problem, for the entity-to-entity list, since it aims to model the relatedness between different entities and is not directly related with the documents, it is possible to use such a list constructed from a different set of documents that is much larger than D. For the unbalance between the document-to-word list and the document-to-entity list, different weights can be assigned to them in the final objective function for training the document representations, thus makes it possible to control how much awareness of the entities in documents and finally reduce the influence of document-to-word list to the learned representations.

2.3. Document representation learning

With the three lists constructed, for each of them, an objective function that is the log probability of observing all the samples in the list can be defined. We define our final objective function as the weighted sum of these three objectives together with a regularization term:

$$J = \lambda_1 \sum_{\langle d,w\rangle \in L_{dw}} \log p(w|d) + \lambda_2 \sum_{\langle d,\epsilon\rangle \in L_{de}} \log p(\epsilon|d) + \lambda_3 \sum_{\langle \epsilon_0,\epsilon_1\rangle \in L_{dw}} \log p(\epsilon_1|\epsilon_0) - \gamma \cdot \|\Omega\|_2^2, \tag{4}$$

where λ_1, λ_2 and λ_3 are weights that are used to control how much each of the three parts influence the learned document representations. Ω is the set of all parameters to be trained. For convenience, we denote $V^{(W)}$ and $V^{(E)}$ as the sets of weight vectors of all the words and all the entity names respectively, i.e., $V^{(W)} = \{v'_w : w \in W\}$ and $V^{(E)} = \{v'_\epsilon : \epsilon \in E\}$, also, define function

$$q(v, v', V) = \frac{\exp(v'^{\top} v)}{\sum_{\hat{v} \in V} \exp(\hat{v}^{\top} v)}. \tag{5}$$

Then back to Eq. (4), we have $p(w|d) = q(v_d, v'_w, V^{(W)})$, $p(\epsilon|d) = q(v_d, v'_\epsilon, V^{(E)})$ and $p(\epsilon_1|\epsilon_0) = q(v_{\epsilon_0}, v'_{\epsilon_1}, V^{(E)})$. Note that the document vectors for the modeling of $p(w|d)$ and $p(\epsilon|d)$ are shared, thus the learned representations incorporates information about both words and entity mentions in documents. Moreover, the weight vectors in the softmax functions $q(v_d, v'_\epsilon, V^{(E)})$ and $q(v_{\epsilon_0}, v'_{\epsilon_1}, V^{(E)})$ are also shared, so that documents mention different but closely related entities will also get similar representations.

Training. To train this model, we first employ negative sampling (NEG) [27] to transform the objective J. For the document-to-word part of Eq. (4), first expand it with $p(w|d) = q(v_d, v'_w, V^{(W)})$, then replace every $q(v_d, v'_w, V^{(W)})$ term with

$$\log \sigma(v'^{\top}_w v_d) + \sum_{i=1}^{t} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-v'^{\top}_{w_i} v_d) \right],$$

where $\sigma(x) = 1/(1 + \exp(-x))$, t is the number of negative samples to use. The negative samples are drawn from the noise distribution $P_n(w)$, which is set as $P_n(w) \propto N_w^{3/4}$, where N_w is the number of times word w occurred in the document set. The transformations for the document-to-entity part and the entity-to-entity part are similar. Afterwards, we draw a sufficient number of prediction samples from the three lists randomly, every sample is given the same probability to be drawn. Then, treat the drawn samples as training samples and update the parameters with stochastic gradient decent according to the transformed objective function.

2.4. Learning the representation of a new document

The above procedure takes a fixed set of documents as input and perform the training off-line. After that, we need a different strategy if we want to learn the representation of a new document that is not in the trained document set.

After training on a large set of documents D, apart from the document representations, we also save all the weight vectors in $V^{(W)}$ and $V^{(E)}$. Let the new document be d_n, we use the methods described in 2.2 to construct a document-to-word list L^n_{dw} and a document-to-entity list L^n_{de} for d_n. Then the objective for training the representation of d_n is

$$J_n = \lambda_1 \sum_{\langle d,w\rangle \in L^n_{dw}} q(v_d, v'_w, V^{(W)}) + \lambda_2 \sum_{\langle d,\epsilon\rangle \in L^n_{de}} q(v_d, v'_\epsilon, V^{(E)}) - \gamma \cdot \|\Omega_n\|_2^2, \tag{6}$$

where Ω_n is the set of all parameters. The training process is similar to the training procedure with Eq. (4), but the weight vectors in $V^{(W)}$ and $V^{(E)}$ are initiated with the values as those trained on D and are kept fixed. It is reasonable to keep the weight vectors fixed, since after training on a large set of documents, they are already proper representations of the corresponding words and entity names.

3. Application to entity linking

Entity linking is the task of mapping mentions in documents to their referred entities in a knowledge base. It is required by many applications, such as relation extraction [36], entity relationship explaining [35], etc. We introduce, in this section, how we apply our approach to this task.

Given a mention within a document, a typical approach for entity linking first finds the candidate entities that this mention may refer to, then rank them and take the one with the highest score as the linking result. The key to an entity linking system is to measure how well a candidate entity matches the context of the mention semantically.

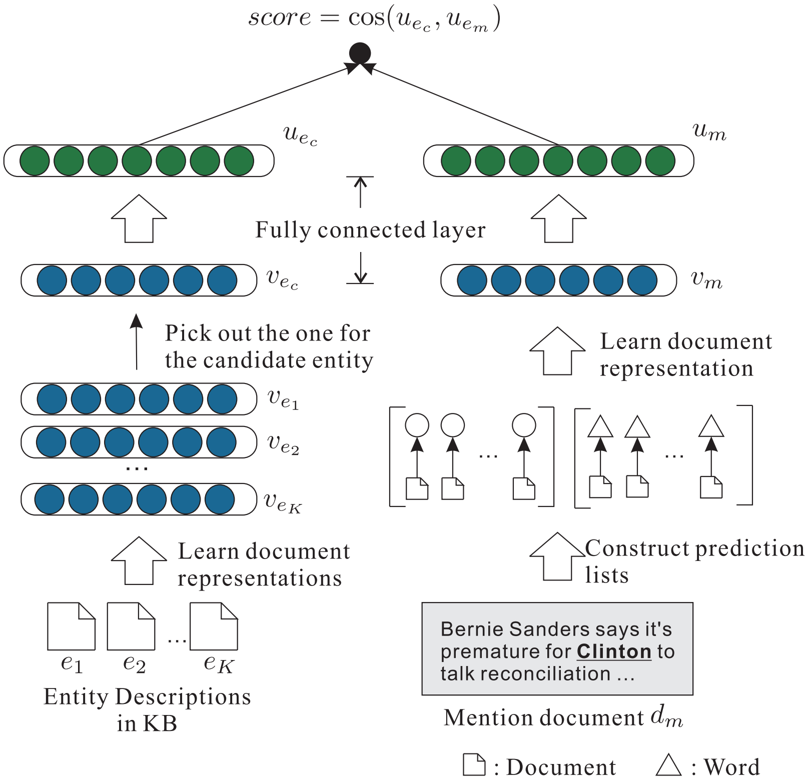

$score = \cos(u_{e_c}, u_{e_m})$
u_{e_c}
u_m
Fully connected layer
v_{e_c}
v_m
Pick out the one for the candidate entity
Learn document representation
v_{e_1}
v_{e_2}
...
v_{e_K}
Learn document representations
Construct prediction lists
e_1 e_2 ... e_K
Entity Descriptions in KB
Bernie Sanders says it's premature for **Clinton** to talk reconciliation ...
Mention document d_m
: Document
: Word

Fig. 2. Measuring how well a candidate entity matches the mention context. $e_1, e_2, \ldots, e_K$ are entities in KB, each of them has a text description that can be treated as a document, $v_{e_1}, v_{e_2}, \ldots, v_{e_K}$ are the learned representations for these text descriptions. v_{e_c} is the representation that corresponds to the candidate entity e_c, v_m is the learned representation for the current mention document.

Fig. 2 illustrates how we measure the compatibility between a mention m and one of its candidate entities e_c. The text descriptions for entities $e_1, e_2, \ldots, e_K$ in the knowledge base can be treated as a set of documents. We apply our approach to learn their document representations $v_{e_1}, v_{e_2}, \ldots, v_{e_K}$ beforehand (i.e. their representations are learned during the construction of the entity linking system). Afterwards, the method described in 2.4 can be employed to learn the representation of the mention document d_m, which is denoted as v_m. The document representation corresponds to the candidate entity e_c is denoted as v_{e_c}.

Since the learning of v_m and v_{e_c} is unsupervised, they may not fully match the requirement of entity linking. Thus we further apply a fully connected layer on top of each of them to get corresponding feature vectors u_{e_c} and u_m that will better suit the task. That is,

$$u_{e_c} = g(v_{e_c} \cdot W), \tag{7}$$

$$u_m = g(v_m \cdot W), \tag{8}$$

where W is a matrix, g is a non-linear function that applies element-wise to a vector. Finally, if e_c is the correct entity that m refers to, then usually its description in KB will have common features with the mention document d_m, e.g., they may have some common topics, mention some common entities, etc. We expect that u_{e_c} and u_m will capture these features and therefore become more similar with each other when e_c is the referred entity. Thus we employ cosine similarity and define

$$s(m, e_c) = \cos(u_m, u_{e_c}) \tag{9}$$

to be the score that represents how well e_c matches m semantically.

To train W, we use the contrastive max-margin criterion [3]. The loss function is defined as

$$L = \sum_{m \in M} \max(0, 1 - s(m, e_g) + s(m, e_f)) + \lambda \|W\|_2^2, \tag{10}$$

where e_g is the correct entity that mention m refers to, e_f is an incorrect one chosen from the candidates, M is the set of all mentions used for training. This loss function tries to create margins between the scores of correct entity candidates and the scores of incorrect entity candidates, giving the correct ones larger scores.

Table 1
Statistics of 20 Newsgroups and NYT. #Documents means number of documents. #Categories means number of categories. #Words means number of words in the vocabulary. #Entities means number of different entity names found in the documents. #CooccurEntityPairs means the number of entity pairs that are mentioned in a same sentence. #MentionsPerDoc means number of mentions per document.

Dataset	20 Newsgroups	NYT		
		Business	Sports	World
#Documents	18,846	14,263	36,915	36,959
#Categories	20	4	11	5
#Words	9590	12,331	20,329	20,413
#Entities	23,229	33,027	51,212	50,825
#CooccurEntityPairs	116,886	527,093	1,754,357	1,479,472
#MentionsPerDoc	10.9	41.9	48.3	41.5

We name this approach as EMADR-EL. Since in the entity linking part, the only parameters that need training is the shared matrix W of Eqs. 7 and 8. Since the size of W is not big, this approach only requires a small amount of training data.

4. Experiments

We conduct experiments on text clustering, text classification and entity linking to validate the effectiveness of EMADR and EMADR-EL.

The Stanford NER tool[1] is used to find entity mentions in documents throughout the experiments, it classifies the mentions in 4 classes: Location, Person, Organization and Misc.

4.1. Text clustering and classification

In this section, we present the setup and results of text clustering and classification experiments. These experiments are to validate whether EMADR could learn better document representations that are more informative and discriminative.

4.1.1. Datasets

Two datasets are used for both text clustering and classification: 20 Newsgroups and NYT.

20 Newsgroups. The 20 Newsgroups[2] dataset contains approximately 20,000 newsgroup documents, which are partitioned evenly across 20 different categories. 60% documents are treated as the training set and 40% documents are treated as the testing set. We further take 20% documents in the training set as the validation set to tune hyperparameters.

NYT. It is a set of news articles that are collected from the New York Times, which covers a time period from January 1, 2010 to December 31, 2014. These news articles are from three main categories: business, sports, world. Each main category contains several subcategories. We treat these three main categories as three different datasets and name them NYT-BIZ, NYT-SPORTS and NYT-WORLD respectively. For documents in each of these three main categories, we split them randomly and use 1/3 for training, 1/3 for validation, 1/3 for testing.

Some statistics of these two datasets are summarized in Table 1.

4.1.2. Compared methods

There are currently two typical approaches to learn continuous low dimensional document representations: topic models and neural network approaches. Topic models represent each document with a topic distribution vector. Neural network approaches represent each document with a vector of parameters in the neural network model used. We compare EMADR with two topic models: LDA [1] and LFTM [28]; three neural network approaches: RSM [13], PV-DM and PV-DBOW [19]. Descriptions of these methods and the reasons why we choose to compare with them are listed below.

LDA. Latent Dirichlet allocation assigns a topic distribution to each document. Each topic is modeled by a multinomial distribution over words in a fixed vocabulary. LDA is the most popular topic model.

LFTM. This method incorporates word vector representations trained on large corpora to extend Dirichlet multinomial topic models. It also generates a topic distribution vector for each document. We compare with this method since it is a topic model that also takes the advantage of neural network methods.

[1] http://nlp.stanford.edu/software/CRF-NER.shtml.
[2] http://qwone.com/~jason/20Newsgroups.

Table 2
Experimental results of text clustering. Best performances are highlighted in bold.

	NYT-BIZ		NYT-SPORTS		NYT-WORLD		20 Newsgroups	
Method	NMI	Purity	NMI	Purity	NMI	Purity	NMI	Purity
LDA	0.458	0.769	0.506	0.582	0.069	0.367	0.320	0.282
LFTM	0.454	0.768	0.290	0.402	0.061	0.392	0.383	0.368
RSM	0.466	0.783	0.675	0.724	0.047	0.377	0.561	0.546
PV-DM	0.456	0.782	0.612	0.677	0.059	0.365	0.413	0.369
PV-DBOW	**0.477**	0.788	0.691	0.750	0.058	0.396	0.631	0.644
EMADR	0.472	**0.790**	**0.761**	**0.800**	**0.372**	**0.660**	**0.676**	**0.700**

RSM. Replicated softmax (RSM) is a generative model of word counts that employs restricted Boltzmann machine. Thus although it is also a neural network approach, it is quite different from EMADR. RSM is also a well-known model and it inspired several later works [18,37].

PV-DM and PV-DBOW. The distributed memory version (PV-DM) and the distributed bag of words version (PV-DBOW) of the *Paragraph Vector* model proposed by [19]. PV-DM takes the order of words in document into consideration while PV-DBOW does not. We compare with them since *Paragraph Vector* is currently highly representative in learning distributed representations for documents and PV-DBOW is the most closely related model to EMADR.

4.1.3. Parameter settings

During the training of EMADR, 10 negative samples are drawn for negative sampling. For text clustering, the weights λ_1, λ_2 and λ_3 in Eq. (4) are set to 0.15, 1.0 and 1.0 respectively to balance the influence of common words and entity mentions to the learned document representations. For text classification, λ_1, λ_2 and λ_3 are tuned according to the performance on the validation set. The regularization factor γ is set to 0.01.

For topic models LDA and LFTM, we set the hyperparameters α to 0.1 and β to 0.01 [28]. While conducting text clustering experiments, each document is assigned the topic with the highest probability given the document [28,38]. While conducting text classification experiments, we set their number of topics to be 100 and treat the topic distributions as features.

The training of RSM was carried out using Contrastive Divergence by starting with one full Gibbs step and gradually increasing to 10 steps [13], the mini-batch size was set to 10.

PV-DM and PV-DBOW are trained with hierarchical softmax. Their initial learning rates are both set to 0.01 and are decreased linearly during the training.

For all neural network based methods EMADR, RSM, PV-DM and PV-DBOW, the dimension of document vector representations are all set to 100. Except for topic models LDA and LFTM, we employ k-means on the document representations to apply them to text clustering. K-means is repeated for 20 times in each run and the result with the lowest loss is used. For text classification, we use LIBSVM[3] to train SVM classifiers with the learned document representations. Parameters for training the SVM are tuned with the validation sets.

4.1.4. Results

We will first present the results of all the text clustering and classification experiments conducted, and then analyze them together. First, we evaluate the text clustering performance with the number of clusters set to be the same with the number of categories. The clustering performance is evaluated with NMI (Normalized Mutual Information) [32] and Purity [24]. Table 2 presents the results.

Next, we compare the clustering performance of different methods under different number of clusters with NYT-WORLD. For each method, we cluster the documents into 5, 10, 15, 20 clusters and evaluate the performance. The result is shown in Fig. 3.

Finally, the text classification results are listed in Table 3. The performance is evaluated with accuracy and macro-averaging F1.

In the above experimental results, EMADR has shown great advantage. From Table 2, Fig. 3 and Table 3, we see that EMADR almost always outperforms all the competing approaches. Some of the improvements are very significant, e.g., the text clustering performance on NYT-SPORTS and NYT-WORLD in Table 2, the text classification performance on NYT-WORLD and 20 Newsgroups in Table 3. EMADR also consistently outperforms all the other methods in Fig. 3, which compares the clustering performance with respect to different number of clusters.

We can also see from the experimental results that neural network based methods RSM, PV-DM and PV-DBOW achieves either better or comparable performance than topic models LDA and LFTM. Without the constraint of representing one topic with each dimension in the document vector, the flexibility of the neural network based methods allows them to

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

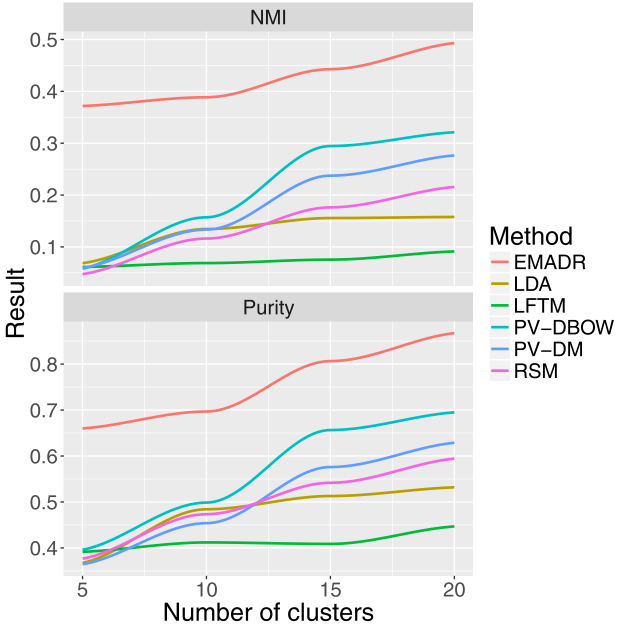


Fig. 3. Clustering performance with respect to different numbers of clusters. The curves are smoothed with LOESS [5].

Table 3
Experimental results of text classification. The F1 here is macro-averaging F1, Acc means accuracy. Best performances are highlighted in bold.

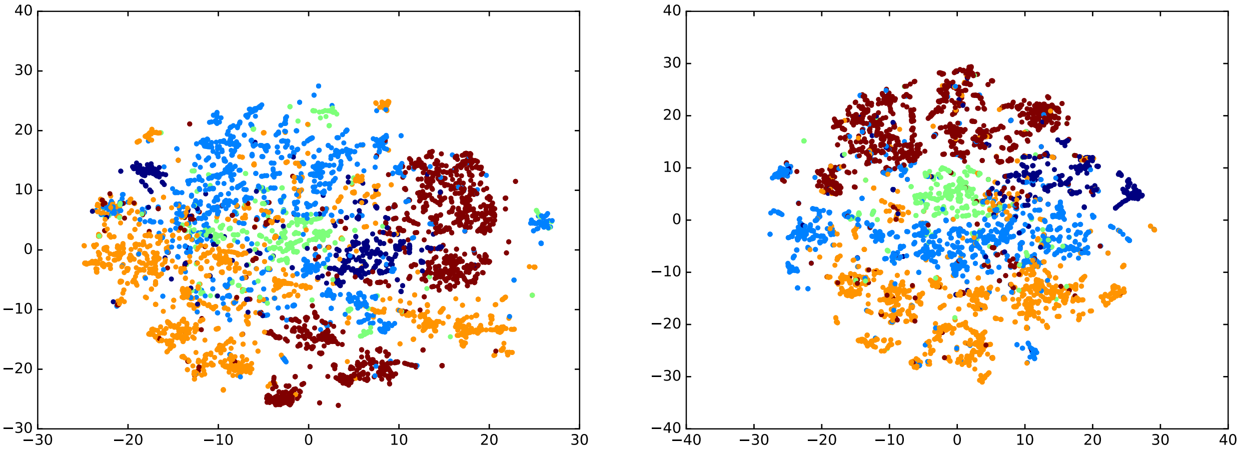
	NYT-BIZ		NYT-SPORTS		NYT-WORLD		20 Newsgroups	
Method	Acc	F1	Acc	F1	Acc	F1	Acc	F1
LDA	0.834	0.634	0.921	0.908	0.827	0.781	0.498	0.476
LFTM	0.833	0.626	0.914	0.904	0.852	0.804	0.670	0.659
RSM	0.843	0.647	0.941	0.930	0.871	0.848	0.719	0.708
PV-DM	0.837	0.633	0.954	0.947	0.875	0.855	0.665	0.653
PV-DBOW	0.838	0.635	0.965	0.960	0.896	0.883	0.768	0.757
EMADR	**0.854**	**0.715**	**0.970**	**0.966**	**0.921**	**0.915**	**0.823**	**0.815**

learn better document representations than the topic models. Among the three compared neural network based methods, PV-DBOW often yields the best performance and is quite robust.

It’s also important to compare EMADR with PV-DBOW since EMADR is a generalization of PV-DBOW with the entity mention information incorporated. When compared with PV-DBOW, in the text clustering results of Table 2, EMADR gains 0.106 and 0.093 average improvement on NMI and purity respectively; in the text classification results of Table 3, it gains 0.026 and 0.044 average improvement on accuracy and macro-averaging F1 respectively. This indicates that incorporating entity mention information with our approach makes the learned document representations more informative and more discriminative, which helps to improve the clustering and classification performance. On the other hand, since most ordinary words are less representative than entity mentions for documents, the equal treating of all words by PV-DBOW leads it to inferior performance.

Qualitative evaluation. We also evaluate the learned document representations qualitatively. The visualization of the document representations learned with PV-DBOW and EMADR is illustrated in Fig. 4. 5000 news articles that cover a continuous time period are used to plot the figures. We choose to compare with PV-DBOW because it is the second best performing method in the above experiments and the most closely related model to EMADR. We can see that with PV-DBOW, the categories are not clearly separated. In contract, EMADR is better at revealing the natural categories in the data.

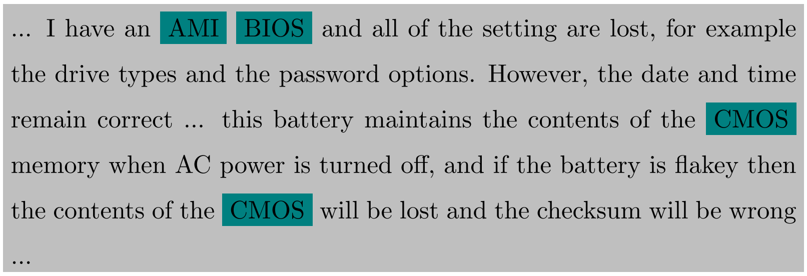
Example. We show a snippet of a document that belongs to the *comp. sys. ibm. pc. hardware* category of 20 Newsgroups in Fig. 5. In our text classification experiments, this document is correctly classified with EMADR, however with the document representations learned by PV-DBOW, it is incorrectly classified to the *comp.sys.mac.hardware* category. Since documents from both *comp. sys. ibm. pc. hardware* and *comp. sys. mac. hardware* are about computer hardware, they are likely to use some similar words such as “battery” and “power”, which makes it difficult for the classifier to distinguish with PV-DBOW document representations. One the other hand, EMADR does not have this problem since it is aware of the entity mentions such as “AMI”, “BIOS” and “CMOS”, which are more related to PC hardware.



(a) PV-DBOW

(b) EMADR

Fig. 4. Visualization of the document representations with t-SNE. Each point represents an article, the color of a point represent the true category the article belongs to.



... I have an AMI BIOS and all of the setting are lost, for example the drive types and the password options. However, the date and time remain correct ... this battery maintains the contents of the CMOS memory when AC power is turned off, and if the battery is flakey then the contents of the CMOS will be lost and the checksum will be wrong ...

Fig. 5. A snippet of a document from the comp.sys.ibm.pc.hardware category of 20 Newsgroups. Entity mentions are highlighted.

4.2. Entity linking

We also conduct experiments to evaluate the performance of EMADR applied to entity linking. We use the articles in an English Wikipedia dump[4] to train the representations of entity text descriptions. We get approximately 4M articles after filtering out some that are not describing entities (e.g. "History of Computing Hardware", "List of Presidents of the United States", etc.). The dimension of the vector representation for each document is set to 100. The weights λ_1, λ_2 and λ_3 in Eq. (4) are all set to 1.0, the regularization factor γ is set to 0.01. We draw 10 negative samples for negative sampling.

We find candidate entities for a mention from 5 sources: (1) the "also known as" fields of freebase, (2) the titles of Wikipedia pages, (3) the redirect pages of Wikipedia, (4) the anchor texts of Wikipedia, (5) the disambiguation pages of Wikipedia. The last four are frequently used in former entity linking research works. The candidate entities for each mention are sorted with respect to commonness [25] and only the top 30 ones are kept. Commonness estimates how likely a mention refers to a particular entity without considering the context with Wikipedia anchor text. Then a linear combination of commonness and the score obtained with Eq. (9) are used to rank the candidate entities.

Two entity linking datasets are used: TAC-KBP 2009 and TAC-KBP 2010. The training set of TAC-KBP 2010 is used as the validation set to tune hyperparameters. While evaluating with TAC-KBP 2009, the evaluation set of TAC-KBP 2010 is used for training, and vice versa. Excluding the NIL mentions (those whose referred entities are not in the referent KB) in the datasets, there are 1675 mentions in TAC-KBP 2009, 1500 mentions in TAC-KBP 2010 training set, 1020 mentions in TAC-KBP 2010 evaluation set. Note that these mentions are the manually labeled mentions used for entity linking, they are only a small part compared with the mentions we find with the NER tool for learning document representations. As for the mentions we found with the NER tool, for TAC-KBP 2009, TAC-KBP 2010 training set and TAC-KBP 2010 evaluation set respectively, there are on average 48.1, 54.1, 53.1 mentions per document and 2.0, 1.7, 1.4 mentions per sentence.

The dimensions of W as 100×200, which means u_m and u_{e_c} in Eq. (9) are of dimension 200. tanh is used as the non-linear function g in Eq. (7) and (8). We choose up to 5 incorrect candidate entities for each mention to serve the training

[4] https://dumps.wikimedia.org/enwiki/.

Table 4
Experimental results of entity linking on TAC-KBP 2009 and TAC-KBP 2010. Best performances are highlighted in bold.

Method	TAC-KBP 2009	TAC-KBP 2010
Rank 1	77.25	80.59
SDA-Based [12]	-	80.97
CNN-Based [33]	82.26	83.92
EMADR-EL	**82.45**	**86.27**

Table 5
Typical unsupervised document representation learning approaches. Note that this table does not summarize all document representation learning approaches. It is also possible for an approach to belong to multiple categories.

Category	Examples	Description
Neural Network	*Paragraph Vector* [19], DocNADE [18], DRBM [37]	Using the parameters of trained neural networks as document representations.
Topic Model	LDA [1], Yang et al. [39] DRBM [37]	Represent each document as a vector of topic mixing proportions.
Topic Model + Word Embedding	Li et al. [20], LFTM [28]	Combine the advantages of topic modeling and word embedding.

with the objective given by Eq. (10). Micro-averaging accuracy, which is the number of correctly linked mentions divided by the number of all mentions, is used to evaluate the performances.

Table 4 reports the evaluation results. Rank 1 is the top ranked systems when TAC-KBP 2009 and TAC-KBP 2010 was held. SDA-Based [12] and CNN-Based [33] are two neural network approaches, both focused on measuring how well a candidate entity matches the mention context. SDA-Based [12] first employs Stacked Denoising Auto-encoder to learn initial document representations, then optimize the representation with a fine-tuning stage. CNN-Based [33] is a state-of-the-art neural network approach that models variable length context with convolutional neural network, it also employs neural tensor network to model the semantic interactions between context and mention. Both SDA-Based [12] and CNN-Based [33] collected large sets of training data with anchor text in Wikipedia to train their model.

It can be seen from the table that SDA-Based [12] and CNN-Based [33] both yields better performance than the then top ranked systems. Meanwhile, our method EMADR-EL outperforms all the other methods. Especially on TAC-KBP 2010, EMADR-EL is better than the second best performing approach for more than 2 percentage. This demonstrates the advantage of our learned document representations over this task, since the application of EMADR on entity linking is rather straightforward compared with the elaborately designed competing approaches.

5. Related work

Our work is mainly related to varied length text representation learning and entity linking. We will introduce the related work on both topics.

5.1. Varied length text representation learning

Due to the evident disadvantages of BoW, many approaches have been proposed to learn continuous vector representations for short texts or documents. These approaches are either unsupervised and learns representations that are of general purpose or supervised and learns representations for specific tasks.

We list three types of unsupervised approaches in Table 5, along with a brief description and some examples for each of them. Note that Table 5 does not summarize all document representation learning approaches, for example, Nikolentzos et al. [29] proposed an approach that obtains document representations by modeling each document as a multivariate Gaussian distribution based on distributed word representations, which does not belong to any of the listed categories in Table 5.

Next, we introduce some unsupervised neural network approaches in detail since they are more related to our proposed method. Neural network approaches typically employs models such as multilayer perceptron [19] and restricted Boltzmann machine (RBM) [13,37] to perform document modeling. Then the learned parameters in the used neural network models can be used as document representations. Replicated Softmax (RSM) is a well known model proposed by Hinton and Salakhutdinov [13] that models documents with RBM's. The hidden variables of RSM are viewed as topic features and can be used to represent documents. Inspired by RSM, Larochelle and Lauly [18] proposed DocNADE, which employs neural autoregressive distribution estimator to model documents, the hidden layer parameters are used as document representations. Also based on RBM, DRBM [37] diversifies the hidden units to cover the topics in the long-tail region. *Paragraph Vector* is inspired by the word embedding method proposed by Mikolov et al. [27], Its PV-DBOW version models the distribution of words in each document with a neural network. PV-DBOW can be viewed as a special case of our model that learns document representations with a single document-to-word prediction list. Among the above approaches and the other unsupervised approaches listed in Table 5, none took advantage of entity mention information.

Document representations can also be learned with supervision to be task specific. Kim [16] and Blunsom et al. [2] learn sentence vector representations with convolutional neural network for tasks such as sentiment classification and question classification. Tang et al. [34] learns representations for text through embedding a heterogeneous text network. Supervised approaches may assign different importance to words or phrases according to the tasks, but background information such as the relatedness between entities may still be useful and are omitted by those studies. In addition, since the learned representations are task specific, they are not suitable for tasks such as clustering and entity linking.

5.2. Entity linking

An unavoidable problem for entity linking systems is to measure how well a candidate entity matches the context of the mention. This has also been the focus of most research works on this task. A frequently used strategy to address this problem is to estimate the similarity between the mention context and the text description of the candidate entity in knowledge base, which is often achieved by measuring the similarity between two feature vectors, one from each of the two sides. Traditional methods may use features like BoW [26], the topic proportions learned with Latent Dirichlet allocation [40]. He et al. [12] and Sun et al. [33] proposed two neural network approaches. He et al. [12] applied stacked denoising auto-encoders to get vector representations for both the mention document and the text description of the candidate entity, then optimize the representations with a fine-tuning stage to use as features. Sun et al. [33] used convolutional neural network and neural tensor network in their model to learn features for both sides. Neither He et al. [12] nor Sun et al. [33] attempted to utilize information about other entity mentions in the context.

Some entity linking methods [8,10,14] also take into consideration the interdependence of entities mentioned in a document. These methods are called collective entity linking, they perform disambiguation for multiple mentions in a same context simultaneously, trying to make the joint assignment coherent, i.e., the linked entities should have a high probability of being mentioned together. Usually, the coherence of a set of entities is measured by a score based on the relatedness between two different entities [4,11,31]. Since EMADR captures entity mention and entity relatedness information, its application in entity linking also possesses the advantage of collective approaches.

6. Conclusion and future work

In this paper, we demonstrate that better document representations can be learned by exploiting entity mention information. We propose EMADR, a document representation approach that generalizes the PV-DBOW model. It is different from existing approaches in that it captures information of the entity mentions and the relatedness of different entities into the learned document representations. Empirical results show that EMADR achieves significant performance gains in text clustering and text classification.

We also apply EMADR to entity linking. Compared to some existing deep neural network methods for this task, the resulting approach has the advantage of requiring much less training data for learning entity linking features. It also provides state-of-the-art performance on TAC-KBP 2009 and 2010 datasets.

In our proposed approach, the relatedness information of different entities can be viewed as a kind of background knowledge that helps to get a better understanding of each document. We believe that more background knowledge has the potential to be leveraged for document representation learning. For example, the type of each mentioned entity, the relationship between two mentioned person, etc. For future work, we are considering to incorporate more of such knowledge to get better representations.

Acknowledgments

This work was supported in part by the NSFC (No. U1611461, No. 61402401), the China Knowledge Centre for Engineering Sciences and Technology (CKCEST), Qianjiang Talents Program of Zhejiang Province 2015.

References

[1] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[2] P. Blunsom, E. Grefenstette, N. Kalchbrenner, A convolutional neural network for modelling sentences, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.
[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems, 2013, pp. 2787–2795.
[4] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, S. Trani, Learning relatedness measures for entity linking, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, ACM, 2013, pp. 139–148.
[5] W.S. Cleveland, Robust locally weighted regression and smoothing scatterplots, J. Am. Stat. Assoc. 74 (368) (1979) 829–836.
[6] S. Cucerzan, Large-scale named entity disambiguation based on wikipedia data., in: EMNLP-CoNLL, 7, 2007, pp. 708–716.
[7] J.R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by Gibbs sampling, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2005, pp. 363–370.
[8] A. Globerson, N. Lazic, S. Chakrabarti, A. Subramanya, M. Ringgaard, F. Pereira, Collective entity resolution with multi-focal attention, in: Proceedings of The 54th Annual Meeting of the Association for Computational Linguistics (ACL), 2016.

[9] E. Grefenstette, M. Sadrzadeh, Multi- step regression learning for compositional distributional semantics, in: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013), Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013), 2013.

[10] Z. Guo, D. Barbosa, Robust entity linking via random walks, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM, 2014, pp. 499–508.

[11] X. Han, L. Sun, J. Zhao, Collective entity linking in web text: a graph-based method, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2011, pp. 765–774.

[12] Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, H. Wang, Learning entity representation for entity disambiguation., in: ACL (2), 2013, pp. 30–34.

[13] G.E. Hinton, R.R. Salakhutdinov, Replicated softmax: an undirected topic model, in: Advances in Neural Information Processing Systems, 2009, pp. 1607–1614.

[14] Z. Hu, P. Huang, Y. Deng, Y. Gao, E.P. Xing, Entity hierarchy embedding, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP), 1, 2015, pp. 1292–1300.

[15] H.K. Kim, H. Kim, S. Cho, Bag-of-concepts: comprehending document representation through clustering words in distributed representation, Neurocomputing (2017).

[16] Y. Kim, Convolutional neural networks for sentence classification, in: EMNLP, 2014, pp. 1746–1751.

[17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of NAACL-HLT, 2016, pp. 260–270.

[18] H. Larochelle, S. Lauly, A neural autoregressive topic model, in: Advances in Neural Information Processing Systems, 2012, pp. 2708–2716.

[19] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 1188–1196.

[20] S. Li, T.-S. Chua, J. Zhu, C. Miao, Generative topic embedding: a continuous representation of documents, in: Proceedings of The 54th Annual Meeting of the Association for Computational Linguistics (ACL), 2016.

[21] Y. Lin, C.-Y. Lin, H. Ji, List-only entity linking, in: Proceedings of The 55th Annual Meeting of the Association for Computational Linguistics (ACL), 2017.

[22] G. Luo, X. Huang, C.-Y. Lin, Z. Nie, Joint named entity recognition and disambiguation, in: Proc. EMNLP, 2015.

[23] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1064–1074.

[24] C.D. Manning, P. Raghavan, H. Schütze, et al., Introduction to information retrieval, 1, Cambridge University Press Cambridge, 2008.

[25] O. Medelyan, C. Legg, Integrating cyc and wikipedia: Folksonomy meets rigorously defined common-sense, in: In Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI), 2008.

[26] R. Mihalcea, A. Csomai, Wikify!: linking documents to encyclopedic knowledge, in: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, ACM, 2007, pp. 233–242.

[27] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[28] D.Q. Nguyen, R. Billingsley, L. Du, M. Johnson, Improving topic models with latent feature word representations, Trans. Assoc. Comput. Linguist. 3 (2015) 299–313.

[29] G. Nikolentzos, P. Meladianos, F. Rousseau, M. Vazirgiannis, Y. Stavrakas, Multivariate Gaussian document representation from word embeddings for text categorization, EACL 2017 (2017) 450.

[30] X. Pan, T. Cassidy, U. Hermjakob, H. Ji, K. Knight, Unsupervised entity linking with abstract meaning representation, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics–Human Language Technologies, 2015.

[31] L. Ratinov, D. Roth, D. Downey, M. Anderson, Local and global algorithms for disambiguation to wikipedia, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 1375–1384.

[32] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2003) 583–617.

[33] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, X. Wang, Modeling mention, context and entity with neural networks for entity disambiguation, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), 2015, pp. 1333–1339.

[34] J. Tang, M. Qu, Q. Mei, PTE: predictive text embedding through large-scale heterogeneous text networks, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1165–1174.

[35] N. Voskarides, E. Meij, M. Tsagkias, M. de Rijke, W. Weerkamp, Learning to explain entity relationships in knowledge graphs, Proceedings of ACL, Beijing, China (2015) 11.

[36] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph and text jointly embedding., in: EMNLP, Citeseer, 2014, pp. 1591–1601.

[37] P. Xie, Y. Deng, E. Xing, Diversifying restricted boltzmann machine for document modeling, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1315–1324.

[38] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the 22nd International Conference on World Wide Web, ACM, 2013, pp. 1445–1456.

[39] W. Yang, J. Boyd-Graber, P. Resnik, A discriminative topic model using document network structure, Association for Computational Linguistics, 2016.

[40] W. Zhang, Y.C. Sim, J. Su, C.L. Tan, Entity linking with effective acronym expansion, instance selection, and topic modeling., in: IJCAI, 2011, 2011, pp. 1909–1914.