Inductive Document Network Embedding with Topic-Word Attention

Robin Brochier[1,2(✉)], Adrien Guille[1], and Julien Velcin[1]

[1] Université de Lyon, Lyon 2 ERIC EA3083, Lyon, France
{robin.brochier,adrien.guille,julien.velcin}@univ-lyon2.fr
[2] Digital Scientific Research Technology, Lyon, France

Abstract. Document network embedding aims at learning representations for a structured text corpus *i.e.* when documents are linked to each other. Recent algorithms extend network embedding approaches by incorporating the text content associated with the nodes in their formulations. In most cases, it is hard to interpret the learned representations. Moreover, little importance is given to the generalization to new documents that are not observed within the network. In this paper, we propose an interpretable and inductive document network embedding method. We introduce a novel mechanism, the Topic-Word Attention (TWA), that generates document representations based on the interplay between word and topic representations. We train these word and topic vectors through our general model, Inductive Document Network Embedding (IDNE), by leveraging the connections in the document network. Quantitative evaluations show that our approach achieves state-of-the-art performance on various networks and we qualitatively show that our model produces meaningful and interpretable representations of the words, topics and documents.

Keywords: Document network embedding · Interpretability · Attention mechanism

1 Introduction

Document networks, *e.g.* social media, question-and-answer websites, the scientific literature, are ubiquitous. Because these networks keep growing larger and larger, navigating efficiently through them becomes increasingly difficult. Modern information retrieval systems rely on machine learning algorithms to support users. The performance of these systems heavily depends on the quality of the document representations. Learning good features for documents is still challenging, in particular when they are structured in a network.

Recent methods learn the representations in an unsupervised manner by combining structural and textual information. Text-Associated DeepWalk (TADW) [28] incorporates text features into the low-rank factorization of a matrix describing the network. Graph2Gauss [2] learns a deep encoder, guided by the network,

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12035, pp. 326–340, 2020.
https://doi.org/10.1007/978-3-030-45439-5_22

that maps the nodes' attributes to embeddings. GVNR-t [3] factorizes a random walk based matrix of node co-occurrences and integrates word vectors of the documents in its formulation. CANE [25] introduces a mutual attention mechanism that builds representations of a document contextually to each of its direct neighbors in the network.

Apart from Graph2gauss, these methods are not intended to generate representations for documents with no connection to other documents and thus cannot induce *a posteriori* representations for new documents. Moreover, they provide little to no possibility to interpret the learned representations. CANE is a notable exception since its attention mechanism produces interpretable weights that highlight the words explaining the links between documents. Nevertheless, it lacks the ability to explain the representations for each document independently.

In this paper, we describe and evaluate an inductive and interpretable method that learns word, topic and document representations in a single vector space, based on a new attention mechanism. Our contributions are the following:

- we present a novel attention mechanism, Topic-Word Attention (TWA), that produces representations of a text where latent topic vectors attend to the word vectors of a document;
- we explain how to train the parameters of TWA by leveraging the links of the network. Our method, Inductive Document Network Embedding (IDNE), is able to produce representations for previously unseen documents, without network information;
- we quantitatively assess the performance of IDNE on several networks and show that our method performs better than recent methods in various settings, including when new documents, not part of the network, are inductively represented by the algorithms. To our knowledge, we are the first to evaluate this kind of inductive setting in the context of document network embedding;
- we qualitatively show that our model learns meaningful word and topic vectors and produces interpretable document representations.

The rest of the paper is organized as follows. In Sect. 2 we survey related works. We present in details our attention mechanism and show how to train it on networks of documents in Sect. 3. Next, in Sect. 4, we present a thorough experimental study, where we assess the performance of our model following the usual evaluation protocol on node classification and further evaluating its capacity of inducting representations for text documents with no connection to the network. In Sect. 5, we study the ability of our method to provide interpretable representations. Lastly, we conclude this paper and provide future directions in Sect. 6. The code for our model, the datasets and the evaluation procedure are made publicly available[1].

2 Related Work

Network embedding (NE) provides an efficient approach to represent nodes in a low dimensional vector space, suitable for solving various machine learning

[1] https://github.com/brochier/idne.

tasks. Recent techniques extend NE for document networks, showing that text and graph information can be combined to improve the resolution of classification and prediction tasks. In this section, we first cover important works in document NE and then relate recent advances in attention mechanisms.

2.1 Document Network Embedding

DeepWalk [22] and node2vec [9] are the most well-known NE algorithms. They train dense embedding vectors by predicting nodes co-occurrences through random walks by adapting the Skip-Gram model initially designed for word embedding [19]. VERSE [24] propose an efficient algorithm that can handle any type of similarity over the nodes.

Text-Associated DeepWalk (TADW) [28] extends DeepWalk to deal with textual attributes. Yang *et al.* prove, following the work in [17], that Skip-Gram with hierarchical softmax can be equivalently formulated as a matrix factorization problem. TADW then consists in constraining the factorization problem with a pre-computed representation of the documents T by using Latent Semantic Analysis (LSA) [6]. The task is to optimize the objective:

$$\text{argmin}_{W,H} ||M - W^\intercal HT||_F^2. \tag{1}$$

where $M = (A + A^2)/2$ is a normalized second-order adjacency matrix of the network, W is a matrix of one-hot node embeddings and H a feature transformation matrix. Final document embeddings are the concatenation of W and HT. Graph2Gauss (G2G) [2] is an approach that embeds each node as a Gaussian distribution instead of a vector. The algorithm is trained by passing node attributes through a non-linear transformation via a deep neural network (encoder). GVNR-t [3] is a matrix factorization approach for document network embedding, inspired by GloVe [21], that simultaneously learns word, node and document representations. In practice, the following least-square objective is optimized:

$$\underset{U,W}{\text{argmin}} \sum_{i=1}^{n_d} \sum_{j=1}^{n_d} \left(u_i \cdot \frac{\delta_j \, W}{|\delta_j|_1} - \log(1 + x_{ij})\right)^2. \tag{2}$$

where x_{ij} is the number of co-occurrences of nodes i and j, u_i is a one-hot encoding of node i and $\frac{\delta_j \, W}{|\delta_j|_1}$ is the average of the word embeddings of document j. Context-Aware Network Embedding (CANE) [25] consists in a mutual attention mechanism trained on a document network. It learns several embeddings for a document according to its different contextual documents, represented by its neighbors in the network. The attention mechanism selects meaningful features from text information in pairs of documents that explain their relatedness in the graph. A similar approach is presented in [4] where the links between pairs of documents are predicted by computing the mutual contribution of their word embeddings.

In this work, we aim at constructing representations of documents that reflect their connections in a network. A key motivation behind our approach is to be able to predict a document's neighborhood given only its textual content. This allows our model to inductively produce embeddings for new documents for which no existing link is known. To that extend, Graph2Gauss is a similar approach. On the contrary, TADW and GVNR-t are not primarily designed for this purpose as they both learn one-hot embeddings for each node in the document network. Note that if some methods like GraphSage [10], SDNE [27] and GAE [13] also enable induction on new nodes, they cannot deal with nodes that have no known connection. Also, our approach differs from CANE since this latter needs the neighbors of a document to generate its representation. IDNE learns to produce a single interpretable vector for each document in the network. In the next section, we review recent works in attention mechanisms for natural language processing (NLP) that inspired the conception of our method.

2.2 Attention Mechanism

An attention mechanism uses a contextual representation to highlight or hide some parts of input data. Attention is an essential element of state-of-the-art neural machine translation (NMT) algorithms [18] by providing a powerful way to capture dependencies between words.

The Transformer [26] introduces a formalism of attention mechanisms for NMT. Given a query vector q, a set of key vectors K and a set of value vectors V, an attention vector is produced with the following formula:

$$v_a = \omega(qK^T)V. \tag{3}$$

qK^T measures the similarity between the query and each key k of K. ω is a normalization function such that all attention weights are positive and sum to 1. v_a is then the weighted sum of the values V according to the attention weights. Multiple attention vectors can be generated by using a set of queries Q.

In CANE, as for various NLP tasks [7], an attention mechanism generates attention weights that represent the strengths of relation between pairs of input words. However, in this paper, we do not seek to learn dependencies between pairs of words, but rather between words and some global topics. In this direction, the Set Transformer [16] constitutes a computationally efficient attention mechanism where the queries are replaced with a fixed-size set of learnable global inducing points. This model is originally not intended for NLP tasks, therefore we will explore the capacity of such inducing points to play the role of topic representations when applied to textual data.

Even if we introduce the concept of topic vectors, the aim of this work is not to propose another topic model [5,23]. We hypothesize that the introduction of global topic vectors in an attention mechanism can (1) lead to useful representations of documents for different tasks and (2) bring an interpretable sight on the patterns learned by the model. Interpretability can help both machine learning practitioners to better refine their models and end users to understand automated recommendations.

3 Method

We are interested in finding low dimensional vector space representations of a set of n_d documents organized in a network, described by a document-term matrix $X \in \mathbb{N}^{n_d \times n_w}$ and an adjacency matrix $A \in \mathbb{N}^{n_d \times n_d}$, where n_w stands for the number of words in our vocabulary. The method we propose, Inductive Document Network Embedding (IDNE), learns to represent the words and topics underlying the corpus in a single vector space. The document representations are computed by combining words and topics through an attention mechanism.

In the following, we first describe how to derive the document vectors from known word and topic vectors through a novel attention mechanism, the Topic-Word Attention (TWA). Next, we show how to estimate the word and topic vectors, guided by the links connecting the documents of the network.

3.1 Representing Documents with Topic-Aware Attention

We assume a p-dimensional vector space in which both words and topics are represented. We note $W \in \mathbb{R}^{n_w \times p}$ the matrix that contain the n_w word embedding vectors and $T \in \mathbb{R}^{n_t \times p}$ the matrix of n_t topic vectors. Figure 1 shows the matrix computation of the attention weights.

Topic-Word Attention. Given a document i and its bag-of-word encoding $X_i \in \mathbb{N}^{+n_w}$, we measure the attention weights between topics and words, $Z^i \in \mathbb{R}^{n_t \times n_w}$, as follows:

$$Z^i = \mathrm{g}\big(TW^\intercal \mathrm{diag}(X_i)\big). \tag{4}$$

The activation function g must satisfy two requirements: (1) all the weights are non-negative and (2) columns of Z^i sum to one. The intuition behind the first requirement is that enforcing non-negativity should lead to sparse and interpretable topics. The second requirement transforms the raw weights into word-wise relative attention weights, which can be read as probabilities similarly to what is done in neural topic models [23]. An obvious choice would be column-wise softmax, however, we empirically find that ReLU followed by a column-wise normalization performs best.

Document Representation. Given Z^i, we are able to calculate topic-specific representations of the document i. From the perspective of topic k, the p-dimensional representation of document i is:

$$D^i_k = \frac{Z^i_k \mathrm{diag}(X_i) W}{|X_i|_1}. \tag{5}$$

Similarly to Eq. 3, each topic vector, akin to a query, attends to the word vectors that play the role of keys to generate Z^i. The topic-specific representations are then the weighted sum of the values, also played by the word vectors. The final document vector is obtained by simple summation of all the topic-specific representations, which leads to $d^i = \sum_k D^i_k$. Scaling by $\frac{1}{|X_i|_1}$ in Eq. 5 ensures that the document vectors have the same order of magnitude as the word vectors.

3.2 Learning from the Network

Since the corpus is organized in a network, we propose to estimate the parameters, W and T, by leveraging the links between the documents. We posit that the representations of documents connected by a short path in the network should be more similar in the vector space than those that are far apart. Thus, we learn W and T in a supervised manner, through the training of a discriminative model.

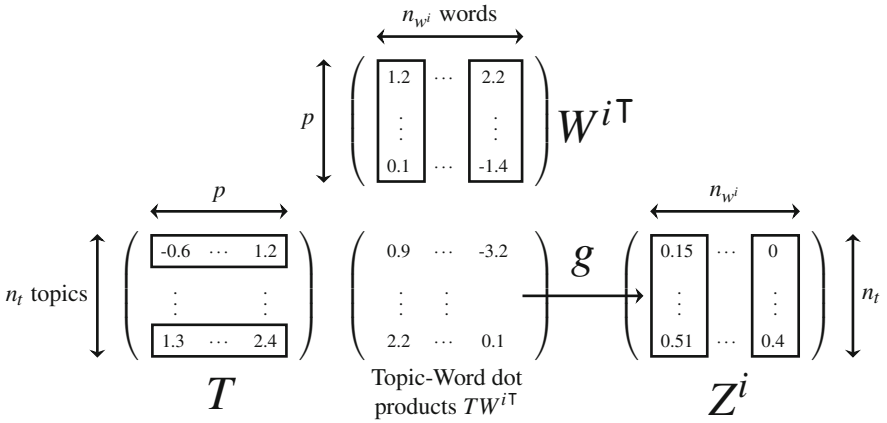

Fig. 1. Matrix computation of the attention weights. Here W^i is the compact view of $\text{diag}(X_i)W$ where zero-columns are removed since they do not impact on the result. n_{w^i} denotes the number of distinct words in document i. Each element z_{jk} of Z^i is the column-normalized rectified scalar product between the topic vector $\boldsymbol{t}_j$ and the word embedding $\boldsymbol{w^i}_k$ and represents the strength of association between the topic j and the word k in document i. The final document representation is then the sum of the topic-specific representations $D^i = \frac{Z^i W^i}{|X_i|_1}$.

Let $\Delta \in \{0,1\}^{n_d \times n_d}$ be a binary matrix, so that $\delta_{ij} = 1$ if document j is reachable from document i and $\delta_{ij} = 0$ otherwise. We model the probability of a pair of documents to be connected, given their representations, in terms of the sigmoid of the dot-product of d_i and d_j:

$$P(Y = 1|d_i, d_j; W, T) = \sigma(d_i \cdot d_j). \tag{6}$$

Assuming the document representations are i.i.d, we can express the log-likelihood of Δ given W and T:

$$\ell(W,T) = \sum_{i=1}^{n_d}\sum_{j=1}^{n_d} \log P(Y = \delta_{ij}|d_i, d_j; W, T)$$
$$= \sum_{i=1}^{n_d}\sum_{j=1}^{n_d} \delta_{ij} \log \sigma(d_i \cdot d_j) + (1 - \delta_{ij}) \log \sigma(-d_i \cdot d_j). \quad (7)$$

Through the maximization of this log-likelihood via a first-order optimization technique, we back-propagate the gradient and thus learn the word and topic vectors that lead to the document representations that best reconstruct Δ.

4 Quantitative Evaluation

Common tasks in document network embedding are classification and link prediction. We assess the quality of the representations learned with IDNE for these tasks in two different settings: (1) a traditional setting where all links and documents are observed and (2) an inductive setting where only a fraction of the links and documents is observed during training.

The first setting corresponds to a scenario where the goal is to propagate labels associated with a small portion of the documents. The second represents a scenario where we want to predict labels and links for new documents that have no network information, once the algorithm is already trained. This is common setting in real world applications. As an example, when a new user asks a new question on a Q&A website, we would like to suggest tags for its question and to recommend potential similar questions. In this case, the only information available to the algorithm is the textual content of the question.

4.1 Experimental Setup

We detail here the setup we use to train IDNE.

Computing the Δ Matrix. We consider paths of length up to 2 and compute the Δ matrix in the following manner:

$$\delta_{ij} = \begin{cases} 1 & \text{if}(A + A^2)_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

This means that two documents are considered close in the network if they are direct neighbors or share at least one neighbor. Note that this matrix is the binarized version of the matrix TADW factorizes.

Optimizing the Log-Likelihood. We perform mini-batch SGD with the ADAM [12] update rule. Because most document networks are sparse, rather than uniformly sampling entries of Δ, we sample 5000 balanced mini-batches in order to favor convergence. We sample 16 positive examples ($\delta_{ij} = 1$) and 16 negative ones ($\delta_{ij} = 0$) per mini-batch. Positive pairs of documents are drawn according to the number of paths of length 1 or 2 linking them. Negative samples are uniformly drawn. The impact of the number of steps is detailed in Sect. 4.6.

4.2 Networks

We consider 4 networks of documents of various nature:

- A well-known scientific citation network extracted from Cora[2]. Each document is an article labelled with a conference.
- New York Times (NYT) titles of articles from January 2007. Articles are linked according to common tags (*e.g.* business, arts, technology) and are labeled with the section they appear in (*e.g.* opinion, news). This network is particularly dense and documents have a short length.
- Two networks of the Q&A website Stack Exchange (SE)[3] from June 2019, namely gaming.stackexchange.com and travel.stackexchange.com. We only keep questions with at least 10 user votes and that have at least one answer with 10 user votes or more. We build the network by linking questions with their answers and by linking questions and answers of the same user. The labels are the tags associated with each question (Table 1).

Table 1. General properties of the studied networks.

	# docs	# links	# labels	Vocab size	# words per doc	Density	Multi-label
Cora	2,211	4,771	7	4,333	67 ± 32	0.20%	No
NYT	5,135	3,050,513	4	5,748	24 ± 17	23.14%	No
Gaming	22,872	400,664	40	15,760	53 ± 74	0.15%	Yes
Travel	15,087	465,696	60	14,539	70 ± 73	0.41%	Yes

4.3 Tasks and Evaluation Metrics

For each network, we consider a traditional classification tasks, an inductive classification task and an inductive link prediction task.

- the traditional task refers to a setting where the model is trained on the entire network and the learned representations are used as features for a one-vs-all linear classifier with a training set of labelled documents ranging from 2% to 10% for multi-class networks and from 10% to 50% for multi-label networks.
- the inductive tasks refer to a setting where 10% of the documents are removed from the network and the model is trained on the resulting sub-network. For the classification task, a linear classifier is trained with the representations and the labels of the observed documents. Representations for hidden documents are then generated in an inductive manner, using their textual content only. Classifications and link predictions are then performed on these induced representations.

[2] https://linqs.soe.ucsc.edu/data.
[3] https://archive.org/details/stackexchange.

To classify the learned representations, we use the LIBLINEAR [8] logistic regression [14] algorithm and we cross validate the regularization parameter for each dataset and each model. Every experiment is repeated 10 times and we report the micro average of the area under the ROC curve (AUC). The AUC uses the probabilities of the logistic regression for all classes and evaluates the quality of the resulting ranking given the true labels. This metric is thus suitable for information retrieval tasks where we want to penalize wrong predictions depending on their ranks. For link prediction, we rank pairs of documents according to the cosine similarity between their representations.

4.4 Compared Representations

For all document networks, we process the documents by tokenizing text into words, discarding punctuation, stop words and words that appear less than 5 times or in more than 25% of the documents. We create document-term matrices that are used as input for 6 algorithms. Our baselines are representative of the different approaches for document NE. TADW and GVNR-t are based on matrix factorization whereas CANE and G2G are deep learning models. For each of them, we used the implementations of the authors:

- LSA: we use a 256-dimensional SVD decomposition of the tf-idf vectors as a text-only baseline;
- TADW: we follow the guidelines of the original paper by using 20 iterations and a penalty term $\lambda = 0.2$. For induction, we generate a document vector by computing the textual component HT in Eq. 1;
- Graph2gauss (G2G): we make sure the loss function converges before the maximum number of iterations;
- GVNR-t: we use $\gamma = 10$ random walks of length $t = 40$, a sliding window of size $l = 5$ and a threshold $x_{\min} = 5$ with 1 iteration. For induction, we compute $\frac{\delta_j}{|\delta_j|_1} W$ in Eq. 2;
- CANE: we use the same parameters as in the original paper;
- IDNE: we run all experiments with $n_t = 32$ topic vectors. The effect of n_t is discussed in Sect. 4.6.

4.5 Results Analysis

Tables 2 and 3 detail the AUC scores on the traditional classification task. We report the results for CANE only for Cora since the algorithm did not terminate within 10 h for the other networks. In comparison, our method takes about 5 min to run on each network on a regular laptop. The classifier performs well on the representations we learned, achieving similar or better results than the baseline algorithms on Cora, Gaming and Travel Stack Exchange. However, regarding the New York Times network, GVNR-t and TADW have a slight advantage. Because of its high density, the links in this network are little informative which may explain the relative good scores of the LSA representations. We hypothesize that (1) TADW benefits from its input LSA features and that (2) GVNR-t benefits

both from its random walk based matrix of node co-occurrences [20], which captures more precisely the proximities of the nodes in such dense network, and from the short length of the documents making the word embedding averaging efficient [1,15].

Table 4 shows the AUC scores in the inductive settings. For link prediction IDNE performs best on three networks, showing its capacity to learn meaningful word and topic representations according to the network structure. For classification, LSA and GVNR-t achieve the best results while IDNE reaches similar but slightly lower scores on all datasets. On the contrary, TADW and Graph2gauss show weaknesses on NYT and Gaming SE.

In summary, IDNE shows constant performances across all settings where other methods lack of robustness against the type of network or the type of task. A surprising result is the good scores of GVNR-t for inductive classification which we didn't expect given that its textual component only is used for this setting. However, for the traditional classification, GVNR-t has difficulties to handle networks with longer documents. IDNE does not suffer the same problem because TWA carefully select discriminative words before averaging them. In Sect. 5, we further show that IDNE learns meaningful representations of words and topics and builds interpretable document representations.

4.6 Impact of the Number of Topics and Convergence Speed

Figure 2 shows the impact of the number of topic vectors n_t and of the number of steps (mini-batches) on the AUC scores obtained in traditional classification with Cora. Note that we observe a similar behavior on the other networks. We see that the scores improve from 1 to 16 topics and tend to stagnate for upper values. In a similar manner, performances improve up to 5000 iterations after which no increase is observed.

Table 2. Micro AUC scores on Cora and NYT

	Cora					NYT				
	2%	4%	6%	8%	10%	2%	4%	6%	8%	10%
LSA	67.54	81.76	88.63	89.68	91.43	79.90	82.06	81.18	83.99	86.06
TADW	65.17	74.11	80.27	83.04	86.56	85.28	**88.91**	87.49	89.39	88.72
G2G	91.12	92.38	91.98	93.79	94.09	79.74	81.41	80.91	82.37	81.42
CANE	**94.40**	**95.86**	95.90	96.37	95.88	NA	NA	NA	NA	NA
GVNR-t	87.13	92.54	94.37	95.21	95.83	**85.83**	87.67	**88.76**	**90.39**	**89.90**
IDNE	93.34	94.93	**95.98**	**96.77**	**96.68**	82.40	84.60	86.16	86.72	87.98

Table 3. Micro AUC scores on Stack Exchange networks

	Gaming					Travel				
	10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
LSA	86.73	88.51	89.51	90.25	90.18	80.18	83.77	83.40	84.12	84.60
TADW	88.05	90.34	91.64	93.18	93.29	78.69	84.33	85.05	83.60	84.62
G2G	82.12	84.42	85.14	86.10	87.84	66.04	67.48	69.67	70.94	71.58
GVNR-t	89.09	92.60	94.14	**94.79**	95.24	79.47	83.47	85.06	85.85	86.58
IDNE	**92.75**	**93.53**	**94.72**	94.61	**95.57**	**86.83**	**88.86**	**89.24**	**89.31**	**89.26**

5 Qualitative Evaluation

We first show in Sect. 5.1 that IDNE is capable of learning meaningful word and topic vectors. Then, we provide visualizations of documents that highlight the ability of the topic-word attention to reveal topics of interest. For all experiments, we set the number of topics to $n_t = 6$.

Table 4. Micro AUC scores for inductive classification and inductive link prediction

	Inductive classification				Inductive Link Prediction			
	Cora	NYT	Gaming	Travel	Cora	NYT	Gaming	Travel
LSA	97.02	**89.45**	90.70	85.88	88.10	60.71	58.99	58.97
TADW	96.23	86.06	93.16	91.35	84.82	69.10	57.00	57.91
G2G	94.04	85.44	89.81	80.71	81.58	74.22	58.18	**59.50**
GVNR-t	**97.60**	88.47	**96.09**	**91.54**	82.27	71.15	59.71	58.39
IDNE	96.58	88.21	95.22	90.78	**91.66**	**77.90**	**62.82**	58.43

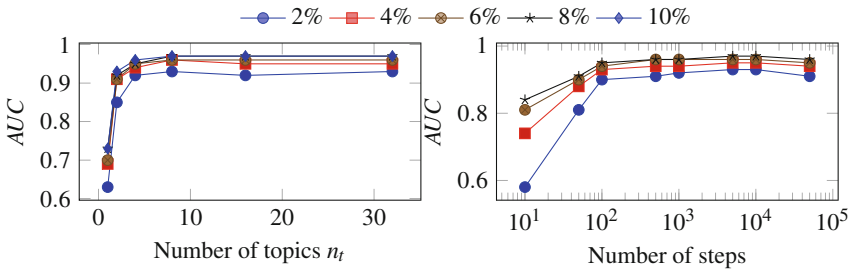

Fig. 2. Impact of the number of topics and of the number of steps on the traditional classification task on Cora with IDNE.

5.1 Word and Topic Vectors

Table 5 shows the closest words to each topic, computed as the dot product between their respective vectors, learned on Cora. Word and topic vectors are trained to predict the proximity of the nodes in a network, meaningless words are thus always dissimilar to the topic vectors, since they do not help to predict a link. This can be verified by observing the words that have the largest and the smallest norms, also reported in Table 5. Even though the topics are learned in an unsupervised manner, we notice that, when we set the number of topics close to the number of classes, each topic seems to capture the semantics of one particular class.

Table 5. Topics with their closest words produced by IDNE on Cora and words whose vector L_2 norms are the largest (resp. the smallest) reported in parenthesis. The labels in this dataset are: Case Based, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning and Theory.

Topic 1	Casebased, reasoning, reinforcement, knowledge, system, learning, decision
Topic 2	Chain, belief, probabilistic, length, inference, distributions, markov
Topic 3	Search, ilp, problem, optimal, algorithms, heuristic, decision
Topic 4	Genetic, algorithm, fitness, evolutionary, population, algorithms, trees
Topic 5	Bayesian, statistical, error, data, linear, accuracy, distribution
Topic 6	Accuracy, induction, classification, features, feature, domains, inductive
Largest	Genetic (8.80), network (8.07), neural (7.43), networks (6.94), reasoning (6.16)
Smallest	Calculus (0.34), instability (0.34), acquiring (0.34), tested (0.34), le (0.34)

5.2 Topic Attention Weights Visualization

To further highlight the ability of our model to bring interpretability, we show in Fig. 3 the topics that most likely generated the words of a document according to TWA. The document is the abstract of this paper whose weights are inductively calculated with IDNE previously trained on Cora. We compute its attention weights Z^i and associate each word k to the maximum value of its column Z^i_k. We then colorize and underline each word associated to the two most represented topics in the document, if its weight is higher than $\frac{1}{2}$. We see that the major topic (green and single underline), that accounts for 32% of the weights, deals with the type of data, here document networks. The second topic (blue and double underline), which represents 18% of the weights, relates to text modeling, with words like “interpretable” and “topics”.

Document network embedding aims at learning representations for a structured text corpus i.e when documents are linked to each other. Recent algorithms extend network embedding approaches by incorporating the text content associated with the nodes in their formulation. In most cases, it is hard to interpret the learned representations. Moreover, little importance is given to the generalization to new documents that are not observed within the network. In this paper, we propose an interpretable and inductive document network embedding method. We introduce a novel mechanism, the Topic-Word Attention (TWA), that generates document representations based on the interplay between word and topic representations. We train these word and topic vectors through our general model, Inductive Document Network Embedding (IDNE), by leveraging the connections in the document network. Quantitative evaluations show that our approach achieves state-of-the-art performance on various networks and we qualitatively show that our model produces meaningful and interpretable representations of the words, topics and documents.

Fig. 3. Topics provided by IDNE in the abstract of this very paper trained on Cora.

6 Discussion and Future Work

In this paper, we presented IDNE, an inductive document network embedding algorithm that learns word and latent topic representations via TWA, a topic-word attention mechanism able to produce interpretable document representations. We showed that IDNE performs state-of-the-art results on various network in different settings. Moreover, we showed that our attention mechanism provides an efficient way of interpreting the learned representations. In future work, we would like to study the effect of the sampling of the documents on the learned topics. In particular, the matrix Δ could capture other types of similarities between documents such as SimRank [11] which measures structural relatedness between nodes instead of proximities. This could reveal complementary topics underlying a document network and could provide interpretable explanations of the roles played by documents in networks.

References

1. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: International Conference on Learning Representations (2017)
2. Bojchevski, A., Günnemann, S.: Deep Gaussian embedding of graphs: unsupervised inductive learning via ranking. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=r1ZdKJ-0W
3. Brochier, R., Guille, A., Velcin, J.: Global vectors for node representations. In: The World Wide Web Conference, pp. 2587–2593. ACM (2019)
4. Brochier, R., Guille, A., Velcin, J.: Link prediction with mutual attention for text-attributed networks. In: Companion Proceedings of the 2019 World Wide Web Conference, pp. 283–284. ACM (2019)
5. Chang, J., Blei, D.: Relational topic models for document networks. In: Artificial Intelligence and Statistics, pp. 81–88 (2009)
6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)

7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
9. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864. ACM (2016)
10. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, pp. 1024–1034 (2017)
11. Jeh, G., Widom, J.: SimRank: a measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 538–543. ACM (2002)
12. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2014)
13. Kipf, T.N., Welling, M.: Variational graph auto-encoders. In: NIPS Workshop on Bayesian Deep Learning (2016)
14. Kleinbaum, D.G., Klein, M.: Logistic Regression. Statistics for Biology and Health. Springer, New York (2002). https://doi.org/10.1007/b97379
15. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
16. Lee, J., Lee, Y., Kim, J., Kosiorek, A.R., Choi, S., Teh, Y.W.: Set transformer (2019)
17. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, pp. 2177–2185 (2014)
18. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015, pp. 1412–1421 (2015). https://www.aclweb.org/anthology/D15-1166/
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
20. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab (1999)
21. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
22. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710. ACM (2014)
23. Srivastava, A., Sutton, C.: Autoencoding variational inference for topic models. In: Proceedings of International Conference on Learning Representations (ICLR) (2017)
24. Tsitsulin, A., Mottin, D., Karras, P., Müller, E.: VERSE: versatile graph embeddings from similarity measures. In: Proceedings of the 2018 World Wide Web Conference, pp. 539–548. International World Wide Web Conferences Steering Committee (2018)

25. Tu, C., Liu, H., Liu, Z., Sun, M.: CANE: context-aware network embedding for relation modeling. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1722–1731 (2017)
26. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
27. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234. ACM (2016)
28. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.: Network representation learning with rich text information. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)