symmetry

MDPI

Article

A Generative Model for Topic Discovery and Polysemy Embeddings on Directed Attributed Networks

Bianfang Chai [1,2], Xinyu Ji [2,3], Jianglin Guo [2], Lixiao Ma [2] and Yibo Zheng [1,*]

[1] Hebei Key Laboratory of Optoelectronic Information and Geo-Detection Technology, Hebei GEO University, Shijiazhuang 050031, China; chaibianfang@163.com
[2] Information Engineering College, Hebei GEO University, Shijiazhuang 050031, China; jxy033022@163.com (X.J.); gjlgeo@163.com (J.G.); malixiao@hgu.edu.cn (L.M.)
[3] Intelligent Sensor Network Engineering Research Center of Hebei Province, Hebei GEO University, Shijiazhuang 050031, China
* Correspondence: yibo_zheng@hgu.edu.cn

Abstract: Combining topic discovery with topic-specific word embeddings is a popular, powerful method for text mining in a small collection of documents. However, the existing researches purely modeled on the contents of documents and led to discovering noisy topics. This paper proposes a generative model, the skip-gram topical word-embedding model (simplified as steoLC) on asymmetric document link networks, where nodes correspond to documents and links refer to directed references between documents. It simultaneously improves the performance of topic discovery and polysemous word embeddings. Each skip-gram in a document is generated based on the topic distribution of the document and the two word embeddings in the skip-gram. Each directed link is generated based on the hidden topic distribution of the beginning document node. For a document, the skip-grams and links share a common topic distribution. Parameter estimation is inferred and an algorithm is designed to learn the model parameters by combining the expectation-maximization (EM) algorithm with the negative sampling method. Experimental results show that our method generates more useful topic-specific word embeddings and coherent latent topics than the state-of-the-art models.

Keywords: topic discovery; polysemous word embeddings; attributed network; EM algorithm

Citation: Chai, B.; Ji, X.; Guo, J.; Ma, L.; Zheng,Y. A Generative Model for Topic Discovery and Polysemy Embeddings on Directed Attributed Networks. *Symmetry* **2022**, *14*, 703. https://doi.org/10.3390/sym14040703

Academic Editor: László T. Kóczy

Received: 27 February 2022
Accepted: 28 March 2022
Published: 30 March 2022

1. Introduction

With the development of internet technology, an enormous number of online information service platforms generate more and more information with rich content and links, such as paper citation networks and hyperlink networks of the World Wide Web, etc. Many methods model these networked data as attributed networks. In the formalized representation, each node corresponds to a paper or a webpage, which is often associated with a rich set of text attributes formed by word sequence features. An edge represents whether there are associations between the two nodes. In this paper, we mainly focus on directed links, and an undirected link equals two directed links. Summarizing the semantic structures and topology structures hidden in these networked data can help people understand the networked data, which have been exploited in the areas of natural language processing and graph mining. In the former area, traditional topic discovery and word embedding technologies purely mine the data based on the node contents of networked data. While in the area of graph mining, research shows that the network topological structure and node attributes are often strongly correlated with each other. For example, two papers with high correlation topics would have a reference link. Semantic community detection methods are always used, which jointly exploit the two information sources to enhance the learning performance. The following will analyze the two kinds of methods.

Topic discovery is a popular tool for detecting the topic structure of the contents of networked data, such as Probabilistic Latent Semantic Analysis (PLSA) [1] and Latent

Dirichlet Allocation (LDA) [2]. Topic models can summarize documents as a mixture of the topics, and the topics can be summarized as a distribution on the large vocabularies of the document sets. However, they suffer from the coherence of discovered topics due to one hot word representation. In parallel with the development of traditional topic discovery, word embedding algorithms represent words as dense distributed vectors [3] to avoid the semantic gaps and sparsity caused by one hot representation. However, the embedding methods just learn local word co-ocurrence information, which loses the global semantic information of the words. Recently, some studies integrate topic discovery with word embedding. They aim to learn more comprehensive word embedding and discover more accurate topics. There are three main kinds of methods, whose advantages and disadvantages are compared in Table 1. The first kind makes use of the pre-trained word embeddings to improve the performance of topic discovery. For example, TopicVec modeled the generative process of words in each document given the topic embedding and the embeddings of the word and its contexts [4]. The GaussianLDA [5,6] learned pre-trained word embeddings, and modeled a multivariate Gaussian distribution with the topic embedding as its mean to generate word embeddings. The second kind utilizes topic discovery to aid word embeddings. For example, Topic Word Embedding (TWE) first detected topics using the LDA model, and then treated each topic as a pseudo-word to learn topic embedding, which was concentrated with word embeddings to get the final word embeddings [7]. Briakou et al. [8] first learned topics from a large corpus and then learned the topic-specific word embeddings spanned by anchor words. The aforementioned two approaches are both two-step processes, and they cannot model the mutual influence of topic discovery and word embeddings. Besides these two kinds of methods, the third method integrates the advantages of topic discovery with word embedding and models their mutual interactions [9–12]. The Collaborative Language Model (CLM) applied a nonnegative matrix factorization to model both topic discovery and word embeddings [9]. The Joint Topic Word-embedding (JTW) model provided a deep generative model by combining a variational autoencoder with the topic model [10]. Topic Modeling boosted with Sparse Autoencoder (TMSA) modeled the mutual influence of topic discovery and word embedding based topic modeling and an autoencoder [11]. The skip-gram Topical word Embedding (STE) model extended the skip-gram model by considering topic discovery and learned topic-specific word embeddings to solve the problem of polysemy [12]. Compared with other similar methods, the skip-gram embedding (STE) model has two advantages. It not only learns word embeddings and topics in a unified framework, but also explicitly obtains topic-specific word embeddings, thus solving polysemy problems. However, the STE model purely models the contents of documents, which always captains inconsistent top words in the detected topics. This inaccuracy of topic results would further mislead word embedding learning. In the area of graph mining, research has shown that semantic community detection methods on document link networks were able to improve the performance of topic discovery by either contents or links [13,14].

Table 1. The SOTA methods for topic discovery and word embeddings.

Word Embedding-Based Topic Discovery	
Classical Models	TopicVec [4], GaussianLDA [5], LF-LDA [6]
Advantages	Word embeddings supplement topic discovery
Disadvantages	Wrong word embeddings limit topic discovery
Topic Discovery-Based Word Embeddings	
Classical Models	TWE [7]
Advantages	Pre-trained topics improve word embeddings
Disadvantages	Unaccuracy topics make word embeddings worse

Table 1. *Cont.*

Mutual Model for Topic Discovery and Word Embeddings	
Classical Models	CTM [9], TMSA [11], STE [12]
Advantages	Topic discovery and word embeddings improve mutually
Disadvantages	Rich Links are not modeled

Several semantic community detection methods combine links and contents of the attributed networks to detect communities with common topics [13–17]. Different from community detection methods on the topology network, they are devoted to detecting semantic communities considering the text features of the nodes. Different from topic discovery, they make use of links to improve the performance of traditional topic discovery and discover topics with link patterns [18]. However, these existing methods largely use one hot encoding to represent words and documents. This representation loses many semantics of the documents and words due to the semantic gaps and the sparsity, and also increases the complexity of the algorithm due to the high dimension vector. Community-Enhanced Topic Embedding (CeTe) was the first model that integrated the contents and links for topic embedding and word embedding on attributed networks [19]. The CeTe model represented each word by just one word embedding vector, which did not explicitly model topic-specific word embeddings. In addition, the community detection on links was used in the preprocessing stage, which was not unified with topic discovery on contents and do not fully combine contents and links for topic discovery. This motivates us to design an integrated model for topic-specific embedding learning and semantic community discovery based on links and content, which uses links to improve the integrated models of word embeddings and topic discovery methods.

All in all, the STE model integrated word embeddings with topic discovery to improve the two tasks on the word sequences of documents. However, it ignored the rich links between documents, which were able to complement the semantic fuzziness problem of content-driven topic discovery. The CeTe model was an example that improved topic-specific word embedding on document contents by community detection. However, the community detection on links was not integrated with topic discovery and each word was not represented based on topics. Our model is designed to solve the above problems.

To learn better word embeddings and topics based on attributed networks, we propose a joint probabilistic model named the steoLC model for word embedding learning and topic discovery on document link networks. This model assumes that the topic distributions are decided by both content and links. Topic discovery and word embeddings influence each other. Each skip-gram word pair and each link are generated based on the given topic distributions. The topic distributions are decided by the topic distributions of skip-grams and links, and the more accurate topics are estimated according to links and content. Then word embedding is improved by the topic distribution. This model alleviates the semantic ambiguity of word embeddings caused by the ambiguity of topics and learns more accurate word embeddings. An algorithm for the steoLC model is designed based on the EM algorithm, which iteratively estimates the hidden topic distributions and model parameters. In each iteration, the skip-gram negative sampling method is used to learn the word embeddings.

The contributions of this work are as follows:

- First, we present a joint probabilistic generative model, called steoLC, for word embedding learning and topic discovery based on directed document link networks. It models the generative process of both the document word pairs and links by word embedding distributions on topics, the topic distributions of documents, the skip-gram distributions on topics, and the link distributions on topics;
- Second, an algorithm is inferred and designed to estimate the parameters of the steoLC model, which combines the EM algorithm framework with a negative sampling algorithm;

- Finally, the performance of the steoLC model is tested on four aspects, including the visualization of word embedding on different topics, document classification, computing nearest words, and the evaluation of topic consistency.

The remainder of the paper is organized as follows. Section 2 introduces the generative process of our model in detail. The algorithm for the estimation of model parameters is inferred and designed in Section 3. Experimental results on three data sets are presented in Section 4. The paper summarizes our model and looks forward to future work in Section 5.

2. A Generative Model for Topic Discovery and Word Embeddings on Attributed Networks

We aim to design a joint probabilistic model for topic discovery and word embedding considering the links and contents of an attributed network. It integrates the contents and links to improve the performance of topic discovery, which is further used to learn word embeddings, degrading the representation fuzziness of polysemy. In this section, we present the steoLC model in detail. The graphical representation of the steoLC model is shown in Figure 1.

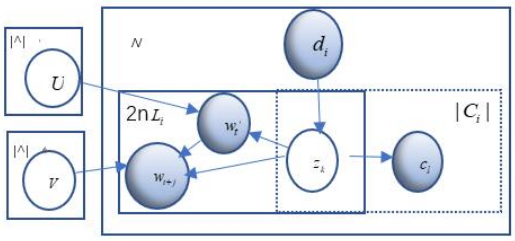

Figure 1. Graphical representation of the steoLC model. Part 1 in the box outlined by dotted lines denotes the component of generating document contents. Part 2 in the box outlined by solid lines denotes the component of generating document links.

The steoLC model is composed of two parts. One part generates the skip-grams in a document given the topic distribution, which is similar to the STE model [12]. To get a more accurate topic distribution, our model extends the STE model by combining the links between documents with contents. The other part models the generative process of each link of a document given a particular topic. The following gives a detailed introduction of the two parts.

Component of generating document contents. There are N documents, noted as a set of documents $D = \{d_1, \ldots, d_N\}$. The i-th document d_i has L_i words, and its content information is denoted as a word sequence $d_i = \{w_1, \ldots, w_{L_i}\}$. The generative process of document content models for each skip-gram in a document, i.e., contexts $w_{t+j}, (j \in [-n, n], j \neq 0)$, given each central word w_t. For a central word w_t, its context words include n words that appear before w_t and the n words that follow w_t. Each pair of (w_{t+j}, w_t) is a skip-gram of the document d_i. As in the STE model, the probability of (w_{t+j}, w_t) depends on the topic z of the central word w_t and the embeddings of w_t as a central word and w_{t+j} as a context word. We assume the document set includes K topics, and each document has a topic distribution on the K topics, noted as $p(z|d)$.

The probability of each skip-gram (w_{t+j}, w_t) in a document d is computed in the following.

$$p(w_{t+j} \mid w_t, d) = \sum_z p(z \mid d) p(w_{t+j} \mid w_t, z) \tag{1}$$

The probability of skip-gram $< w_{t+j}, w_t >$, given the topic z, is evaluated by

$$p(w_{t+j} \mid w_t, z) = \frac{\exp(V_{w_{t+j},z} \cdot U_{w_t,z})}{\sum_{w' \in \wedge} \exp(V_{w',z} \cdot U_{w_t,z})}, \tag{2}$$

where U_w and V_w are the word embedding matrix with $K \times S$ dimension as the central word and a context, respectively, S is the dimension of the embedding space, and $\wedge$ is the vocabulary number of the document set.

Component of generating document links. This section describes the generative process of links between two documents. Link set C is the set of all possible links between every two documents in the corpus, $C = \{C_1, C_2, \ldots, C_N\}$. C_i is the link set of the document d_i. It is assumed that a document d connects to c since they exist the same topic z. The document d has a topic distribution $p(z|d)$ in terms of its links and contents. The probability from d to c, given all the topics, is computed as:

$$p(c \mid d) = \sum_{z} p(z \mid d) p(c \mid z). \tag{3}$$

After finishing the definition of the two components, a complete generative process of links and contents in an attributed network is shown in the following.

For each document $d_i (i \in \{1, \ldots, N\})$:

(a) Draw a topic $z (z \in \{1, \ldots, K\})$ according to $p(z|d_i)$;

(b) Draw a link $c_l (l \in 1, \ldots, |C_i|)$ from documents d_i with probability $p(c_l|z)$;

(c) For each central word, $w_t (t \in \{L_1, \ldots, L_i\})$ in d_i:

Draw a topic z according to $p(z|d_i)$;

Draw each context $w_{t+j} \sim p(w_{t+j}|w_t, z)$.

Optimizating objection. According to the generative process, the likelihood of generating links and contents on an attributed network is defined in the following.

$$\begin{aligned} p(D) &= \prod_{i=1}^{N} p(d_i) p(C_i) \\ &= \prod_{i=1}^{N} \{ (\prod_{t=1}^{L_i} \prod_{\substack{j=-n \\ ,j \neq 0}}^{n} p(w_{t+j}|w_t, d_i)) \times \prod_{l=1}^{|C_i|} p(c_l|d_i) \} \\ &= \prod_{i=1}^{N} (\prod_{t=1}^{L_i} \prod_{\substack{j=-n \\ ,j \neq 0}}^{n} \sum_{z=1}^{K} p(z \mid d_i) p(w_{t+j} \mid w_t, z)) \times \prod_{l=1}^{|C_i|} \sum_{z=1}^{K} p(z \mid d_i) p(c_l \mid z) \end{aligned} \tag{4}$$

where $p(z|d_i)$ is shared by link modeling and content modeling, which integrates the document contents and document links by a common probabilistic distribution for more accurate topic discovery.

The above function is difficult to compute, and is always transformed as the log-likelihood as follows:

$$\log p(D) = \sum_{i=1}^{N} (\sum_{t=1}^{L_i} \sum_{\substack{j=-n \\ ,j \neq 0}}^{n} \log p(w_{t+j}|w_t, d_i)) + \sum_{i=1}^{N} \sum_{l=1}^{|C_i|} \log p(c_l|d_i) \tag{5}$$

In real applications, content components and link components have different importance. The loglikelihoods of the two components in Equation (5) are assigned with a weight α and a weight $1 - \alpha$, respectively, as follows:

$$L = \alpha \sum_{i=1}^{N} \sum_{t=1}^{L_i} \sum_{\substack{j=-n \\ ,j \neq 0}}^{n} \log p(w_{t+j} \mid w_t, d_i) + (1 - \alpha) \sum_{i=1}^{N} \sum_{l=1}^{|C_i|} \log p(c_l \mid d_i). \tag{6}$$

3. Parameter Estimating Algorithm

In this section, we introduce how to learn the values of model parameters given a corpus, including the word embeddings U, V, as well as the topic distribution of documents

$p(z|d)$, the link distribution on topics $p(c|z)$, and the skip-gram distribution on topics $p(w_{t+j}|w_t, z)$. Next, the algorithm of estimating parameters is described and its complexity is analyzed.

3.1. EM Algorithm with Negative Sampling for the setoLC Model

Since the parameters of the steoLC model contain hidden variables z, the EM algorithm is used to maximize the likelihood of Equation (6). In addition, to estimate parameters U, V in Equation (2), the negative sampling algorithm is used. The algorithm framework of the steoLC model combines the EM algorithm with the negative sampling algorithm. It is summarized in Algorithm 1.

Algorithm 1. The algorithm for the steoLC model.

Input: word sequence set D **and link set** C
Output: $U, V, p(z|d), p(c|z)$

Initialize $U, V, p(z|d), p(c|z)$.
for iter=1 to Max_iteration **do**
  for each document d_i in D **do**
    for each skip-gram $< w_{t+j}, w_t >$ in d_i **do**
      Sample negative instances from distribution P.
      Update $p(w_{t+j} \mid w_t, z)$, $p(z \mid d_i, w_t, w_{t+j})$ by Equations (13) and (7).
      Update U, V using the gradient decent method with Equations (14) and (15).
    for each link c_l in C_i **do**
      Update $p(z \mid d_i, c_l)$ by Equation (8).
  for each document d_i in D **do**
    Update $p(z \mid d_i)$ using Equation (11).
    for each link c_l in d_i **do**
      Update $p(c_l \mid z)$ using Equation (12)

The EM algorithm is an iterative algorithm, and each iteration includes an E step and an M step. In the E step, the distributions of hidden variables z, given each skip-gram $< w_{t+j}, w_t >$ and given each link $< d, c >$, are estimated. At the same time, the expectation of the log-likelihood is computed in the E step. In the M step, by maximizing the expectation of the likelihood, the algorithm updates the word embeddings U, V, as well as $p(z|d)$ and $p(c|z)$. To update the word embedding matrices, the algorithm iterates over each skip-gram and several negative sampling instances with the gradient descent method. The algorithm is inferred in detail.

In the E-step, the posterior probability distribution of hidden variables given each skip-gram $< w_{t+j}, w_t >$ in d_i is evaluated by the Bayes rule as follows:

$$p(z \mid d_i, w_t, w_{t+j}) = \frac{p(w_{t+j} \mid w_t, z)p(z \mid d_i)}{\sum_{z=1}^{K} p(w_{t+j} \mid w_t, z)p(z \mid d_i)}. \tag{7}$$

Further, the posterior probability distribution of hidden variables given each link $< d_i, c_l >$ from the document d_i is computed as:

$$p(z \mid d_i, c_l) = \frac{p(c_l \mid z)p(z \mid d_i)}{\sum_{z=1}^{|K|} p(c_l \mid z,)p(z \mid d_i)}. \tag{8}$$

Next, the hidden variable posterior distributions obtained from Equations (7) and (8) are used to compute the expectation of the log-likelihood, which is defined as:

$$\begin{aligned} Q =& \alpha \sum_{i=1}^{N} \sum_{t=1}^{L_i} \sum_{\substack{j=-n \\ ,j\neq 0}}^{n} \sum_{z=1}^{K} p(z \mid d_i, w_t, w_{t+j}) \times \log(p(w_{t+j} \mid w_t, z) \times p(z \mid d_i)) \\ &+ (1-\alpha) \sum_{i=1}^{N} \sum_{l=1}^{|C_i|} \sum_{z=1}^{K} p(z \mid d_i, c_l) \times \log(p(c_l \mid z) \times p(z \mid d_i)) \end{aligned} \tag{9}$$

The topic distribution given a document satisfies the constraint $\sum_{z=1}^{K} p(z \mid d_i) = 1$. The skip-gram distribution given a topic satisfies the constraint $\sum_{\substack{j=-n \\ ,j\neq 0}}^{n} p(w_{t+j} \mid w_t, z) = 1$. The link distribution given a topic satisfies the constraints $\sum_{l=1}^{|C_i|} p(c_l \mid z) = 1$.

The Lagrange multiplier method is used to estimate model parameters. The Lagrange function combines the expectation of the log-likelihood function in Equation (9) with the three constraint conditions defined as:

$$\begin{aligned} F =& \alpha \sum_{i=1}^{N} \sum_{t=1}^{L_i} \sum_{\substack{j=-n \\ ,j\neq 0}}^{n} \sum_{z=1}^{K} p(z \mid d, w_t, w_{t+j}) \log(p(w_{t+j} \mid w_t, z) p(z \mid d_i)) \\ &+ (1-\alpha) \sum_{i=1}^{N} \sum_{l=1}^{|C_i|} \sum_{z=1}^{K} p(z \mid d_i, c_l) \log(p(c_l \mid z,) p(z \mid d_i)) \\ &+ \sum_{i=1}^{N} \beta_1 (1 - \sum_{z=1}^{K} p(z \mid d_i)) + \sum_{z=1}^{|K|} \beta_2 (1 - \sum_{l=1}^{|C_i|} p(c_l \mid z)) \\ &+ \sum_{z=1}^{K} \sum_{t=1}^{L_i} \beta_3 (1 - \sum_{\substack{j=-n \\ ,j\neq 0}}^{n} p(w_{t+j} \mid w_t, z)) \end{aligned} \tag{10}$$

In order to maximize the Lagrange function, the partial derivatives with respect to $p(z \mid d_i), p(c_l \mid z)$ are computed. The updating equations of the topic distribution of a document and the link distribution for each topic are obtained by making partial derivatives equal to zero. The equations are as follows:

$$p(z \mid d_i) = \frac{\alpha \sum_{t=1}^{L_i} \sum_{\substack{j=-n \\ ,j\neq 0}}^{n} p(z \mid d_i, w_t, w_{t+j}) + (1-\alpha) \sum_{l=1}^{|C_i|} p(z \mid d_i, c_l)}{\sum_{z=1}^{|K|} (\alpha \sum_{t=1}^{L_i} \sum_{\substack{j=-n \\ ,j\neq 0}}^{n} p(z \mid d_i, w_t, w_{t+j}) + (1-\alpha) \sum_{l=1}^{|C_i|} p(z \mid d_i, c_l))} \tag{11}$$

$$p(c_l \mid z) = \frac{\sum_{i=1}^{N} p(z \mid d_i, c_l)}{\sum_{i=1}^{N} \sum_{l=1}^{|C_i|} p(z \mid d_i, c_l)} \tag{12}$$

To update the word embeddings, U, V, the immediate method is to compute the partial derivatives of U, V in Equation (2). However, $\sum_{w \in \wedge} exp(V_{w_{t+j}, z} \cdot U_{w_t, z})$ is intractible. The negative sampling method is used to compute U, V, in which the probability distribution of $p(w_{t+j} | w_t, z)$ is computed as follows:

$$\log p(w_{t+j} \mid w_t, z) \propto \log \sigma(V_{w_{t+j},z} \cdot U_{w_t,z}) + \sum_{i=1}^{m} E_{w_i \sim P}(\log \sigma(-V_{w_i,z} \cdot U_{w_t,z})) \tag{13}$$

where σ represents the sigmoid function, $\sigma(x) = 1/(1+\exp(-x))$, and w_i is a negative instance which is sampled from the distribution $P()$. $P(w)$ is a unigram distribution Unigram(w) raised to the $\frac{3}{4}rd$ power [12].

The negative sampling algorithm is a gradient algorithm. The gradient of the objective function concerning U and V is as follows:

$$\begin{aligned} \frac{\partial L}{\partial U_{w_t,z}} &= \sum_{\substack{w' \in \{w_{t+j}\} \\ \cup \{W_{neg}\}}} (-(\xi_{w'w_t} - \sigma(V_{w',z} \cdot U_{w_t,z})) \cdot V_{w',z}) \times p(z \mid d_i, w_t, w_{t+j}) \\ U_{w_t,z} &= U_{w_t,z} + d \times \frac{\partial L}{\partial U_{w_t,z}} \end{aligned} \tag{14}$$

$$\begin{aligned} \frac{\partial L}{\partial V_{w',z}} &= -(\xi_{w'w_t} - \sigma(V_{w',z} \cdot U_{w_t,z}) \cdot U_{w_t,z}) \times p(z \mid d_i, w_t, w_{t+j}), \\ V_{w',z} &= V_{w',z} + d' \times \frac{\partial L}{\partial V_{w',z}}, \end{aligned} \tag{15}$$

where W_{neg} is the negative instances corresponding to w_t, d,d' are step lengths, and

$$\xi_{w'w_t} = \begin{cases} 1, \text{ if } w' \text{ is a word in the context window of } w_t. \\ 0, \text{ otherwise.} \end{cases} \tag{16}$$

3.2. Complexity Analysis of the Algorithm

The process of steoLC is shown in Algorithm 1. In each iteration, the algorithm first computes the distributions of hidden variables $p(z|d_i, w_{t+j}, w_t)$, $p(z|d_i, c_l)$ in line 3–9. Since $p(w_{t+j}|w_t, z, d_i)$ affects $p(z|d_i, w_{t+j}, w_t)$ in Equation (7), it is computed before $p(z|d_i, w_{t+j}, w_t)$ in line 7. At the same time, the parameters of word embedding matrices U and V are updated by a gradient decent method in line 7, and the number of the iterations is set as a constant. The number of document words is $L_1 + L_2 + \cdots + L_N$, which is simplified as L. Next, all the skip-grams are $N \times L \times n \times 2$. The time complexity of line 3–7 is $O(N \times L \times n \times Neg \times K)$, where n is the length of left or right window sampling contexts, Neg represents the number of negative samples, and K is the number of topics. The time complexity of $P(z_k|d_i, c_l)$ in line 8–9 is $O(C \times K)$, where C is the number of links among the documents. The time complexities of $p(z|d_i)$ and $p(c_l|z)$ are, respectively, $O(N \times K)$ and $O(C \times K)$. To summarized, the time complexity of the whole algorithm is $O(Max\_iteration \times (N \times L \times n \times Neg \times K + C \times K))$, where $Max\_iteration$ is the iteration times.

4. Results

In this section, we present a detailed analysis of the performance of our method. We first introduce several experimental settings, including the dataset, parameter settings, and the baselines used in the experiment. Next, document classification is used to evaluate the quality of document representations on the topic distribution estimated by the steoLC model. Subsequently, we explain whether word embedding results are able to model polysemous words on topics by qualitative analysis. Finally, we evaluate the topic coherence of the steoLC model.

4.1. Experiment Settings

Dataset description. To verify the performance of the steoLC model, we experiment on three public datasets as the CeTe model [19], including the DBLP dataset [20] and two different scale hep-th datasets. The DBLP dataset includes many papers on the computer field, and the five largest categories are used to construct attributed networks. The titles and abstracts are represented as the contents of papers, and the citation relationships correspond to the links of papers. The hep-th includes a large corpus of physics-related papers. The four largest categories form one subdataset, named large-hep. The three smaller categories form the other subdataset were called small-hep. The dataset details are shown in Table 2.

Table 2. Dataset information.

Dataset	Documents	Edges	Words	Categories
DBLP	6936	12,353	506,269	5
Small-hep	397	812	18,718	3
Large-hep	11,752	134,956	622,642	4

Baselines. The algorithm of the steoLC model is compared with three state-of-the-art methods. The first model is a traditional topic model, i.e., Latent Dirichlet Allocation (LDA) [2], which models topic discovery on one hot word space. The STE model [12] integrates topic discovery with word embedding based on texts of documents and attempts to solve polysemy. The CeTe model also combines topic discovery with word embedding. In addition, it improves the performance by utilizing community detection on links as a pre-processing step of topic embedding modeling. The CeTe model has been compared with three kinds of state-of-the-art methods for specific-topic embedding modeling. If the performance of our algorithm is better than the CeTe model, our algorithm is superior to the three kinds of methods.

Parameter settings and Hardware specifications. In the experiments, the number of topics K is set as 10, such that topical specific word embedding on a fine-grained topic is possible. The iteration number of the document set and the number of gradient descent for U, V are both set as 15. The dimension of the embedding space is 400. The size n of the context window equals 10. For each skip-gram, eight negative instances are sampled. These settings are similar to the ste model. The steoLC model aims to learn word embeddings, document embeddings, and topic embeddings. Thus, the component of content is more important, and the α is set as 0.75 experimentally. We compare the accuracy of detected topics for classification at the small-hep data set by varying α from 0 to 1, and the accuracy on the test is shown in Figure 2. Computer configurations for the experiments are a CPU-Intel i7, with 16GB of internal storage and 1T of Hard Disk, etc.

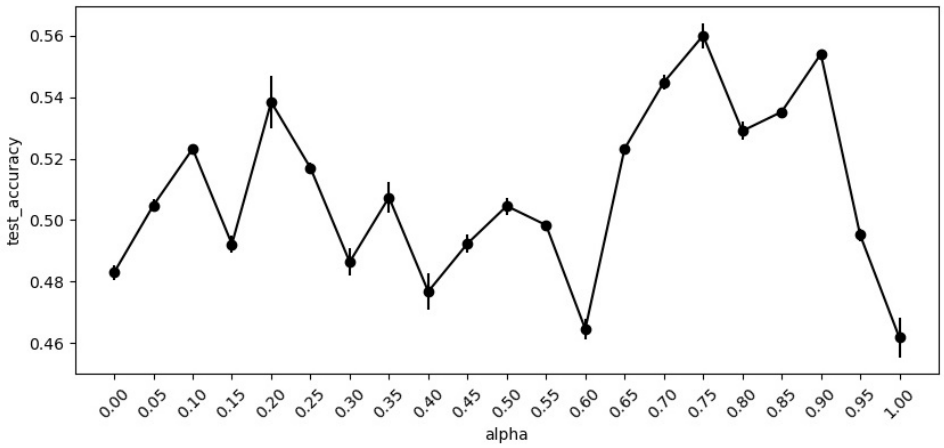

Figure 2. Classification accuracy on the small-hep data set for the steoLC model ($0 \leq \alpha \leq 1$).

4.2. Document Classification

To test whether our algorithm has learned accurate semantics for documents, a classification task is used. The document set is randomly divided into 80% as the training set and 20% as the test set [19]. The document representation learned by all the algorithms by Equation (11) is the input of the classifier. We use the $l-1$ regularized linear SVM one-vs-all classifier. We compare the steoLC model with the LDA model and the STE model. All the models represent a document by the topic distribution of the document. The performance is measured by accuracy, recall, and the F1-Score.

Each model is run ten times, and the mean and standard deviation of the results are computed and shown in Table 3. Our steoLC model performs the best among the three models. Compared with the distributed representation of the steoLC model, the LDA model discovers topics and learns word embeddings using only the contents represented by the one hot word vector, which suffers from sparsity and semantic gaps. Different from our steoLC model combining contents and links, the STE model only considers content information, although it represents words as dense embeddings. This method may captain vague topics. The CeTe model utilizes community detection on links to degrade the fuzziness of topic discovery, but the community detection is not integrated with topic discovery in a unified framework. Our steoLC model constructs a joint model for contents and links in a generative process, which discovers more accurate semantic topics. In addition, it combines topic modeling with word embedding modeling, such that the two tasks are improve each other. We also find that the results on the large-hep dataset are worse than the ones on the small-hep dataset. This is because each category in the large-hep dataset has more documents and is more noisy than each category in the small-hep dataset, which increases the difficulty of distinguishing the pattern of each category on the former dataset compared with the latter one.

Table 3. Document classification effect of different methods.

Dataset	Model	Accuracy	Recall	F1
small-hep	LDA	0.287 ± 0.002	0.287 ± 0.001	0.281 ± 0.001
	STE	0.668 ± 0.021	0.647 ± 0.019	0.649 ± 0.008
	CeTe	0.671 ± 0.007	0.650 ± 0.006	0.654 ± 0.002
	steoLC	**0.698** ± 0.002	**0.675** ± 0.002	**0.682** ± 0.001
large-hep	LDA	0.414 ± 0.002	0.411 ± 0.029	0.401 ± 0.003
	STE	0.432 ± 0.013	0.432 ± 0.008	0.415 ± 0.005
	CeTe	0.437 ± 0.005	0.433 ± 0.001	0.417 ± 0.002
	steoLC	**0.469** ± 0.003	**0.462** ± 0.003	**0.446** ± 0.001
DBLP	LDA	0.700 ± 0.011	0.457 ± 0.005	0.404 ± 0.003
	STE	0.756 ± 0.033	0.729 ± 0.02	0.725 ± 0.009
	CeTe	0.790 ± 0.008	0.736 ± 0.005	0.731 ± 0.003
	steoLC	**0.812** ± 0.002	**0.756** ± 0.019	**0.748** ± 0.005

4.3. Qalitative Analysis

To illustrate the performance of integrating word embedding with topic discovery in our model, we use the t-SNE algorithm [21] to visualize the vectors of the 500 most frequent words that are learned from the steoLC and STE models in the small-hep dataset in Figure 3. Similar to the STE model [12], the number of topics K is set as 10. The number of outer iterations and inner iterations are both set to 15. The dimension of the embedding vectors is set as 400. For each skip-gram, we set the window size to 10 and sample eight negative instances.

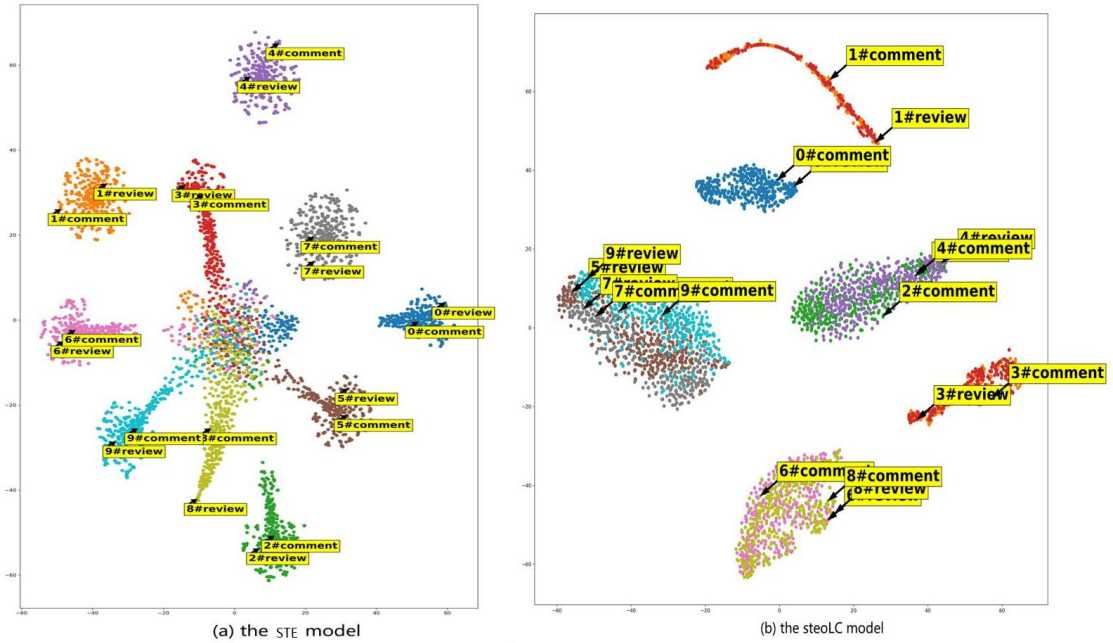

Figure 3. Visualization of the word embeddings learned by the ste and steoLC models. The polysemous word "review" and the monosemous word "comment" are highlighted for comparison.

Each node in Figure 3 denotes a topic-specific word vector. The algorithm of the steoLC model divides the whole space into 10 subspaces, and each subspace represents a topic. To illustrate the advantages of the steoLC model in dealing with polysemous words, we show labels for the word "review", as an example of a polysemous word and the word "comment" as an example of a monosemous word. The labels show both the word and topic index, separated by "#". In Figure 3b, we find that the steoLC model divides the whole space into six disjoint subspaces, and each subspace corresponds to a topic. We set the topic number as 10, and our model detects just six topics clearly. This illustrates that several topics are difficult to divide, such as Topic 7,5,9, Topics 6,8, and Topics 2,4. This guides us to select a small topic number. Additionally, "review" and "comment" are close to each other in some subspaces, but far apart in others. For example, under Topic 1, the word "review" is far away from the word "comment", which indicates that the word "review" does not mean "comment" under Topic 1. Under Topic 0, the word "review" is closer to the word "comment", which indicates that the word "review" means "comment". On the other hand, Figure 3a illustrates that some word embeddings of different topics from the STE model are mixed together, which means that different topics are difficult to discriminate. However, the steoLC model can discover topics more accurately by combining the contents and links, such that the word embeddings in the same topic are similar and word embeddings in different topics are able to be distinguished.

Table 4 shows the nearest neighbors of some polysemous words according to the skip-gram model, the STE model, and the steoLC model. Cosine similarity is used to compute the similarity between polysemous words and other words. We observe that these similar words, according to word embeddings learned by skip-gram, mix different senses of the given words. For example, the nearest neighbors of "present" are "show" and "recently", which indicates that the skip-gram model cannot distinguish different meanings of words. In contrast, the STE and steoLC models can distinguish polysemy due to the consideration of topic modeling. Under Topic 1, the most similar words of "present" are "show", "description", and "action", which clearly corresponds to the meaning of appearance. Under Topic 2, they are "recent" and "recently", clearly referring to 'lately'. In addition, Table 4 shows that similar words from the steoLC model are better than the ones from the STE model. For example, the most similar word to "argue" obtained by the

steoLC model under Topic 1, namely "argue# 1", has more meaning than the ones obtained by the STE model. The reason is that the steoLC model makes use of links to obtain more accurate topic distributions, which further aids in learning accurate word embeddings.

Table 4. The nearest neighbors of some polysemous words.

Model	Words	Similar Words
Word2vec	argue	demonstrate review consider
STE	argue#1 argue#2	prove gauge evidence introduce consider present
steoLC	argue#1 argue#2	demonstrate prove conclude conjectured consider suggest
Word2vec	present	show action recently
STE	present#1 present#2	show description presence recently recent reviewed
steoLC	present#1 present#2	show reveals exhibit recently resent conventional

4.4. Topic Coherence Evaluation

Table 5 compares the top words produced by the STE and steoLC models on three topics detected from the small-hep dataset. In Table 5, Topic 1 is about mathematics, Topic 2 is about cosmology, and Topic 3 is about physics. Both the STE and steoLC models produce words with similar themes. However, the STE model discovers fewer meaningful words related to these three topics. Some words from the STE model are not coherent. For example, for Topic 2, STE produces "conference" and "transition", which are less related to cosmology, while the hot words produced by steoLC are all related to cosmology. This shows that the steoLC model is able to discover more coherent topics than the STE model.

Table 5. The top words on three topics for the STE and steoLC models.

Topic	Method	Word
Topic1	STE steoLC	model formulae surface negative boundary model formulae surface matrix relation
Topic2	STE steoLC	time cosmological polyhedra conference transition time cosmological polyhedra infrared domain
Topic3	STE steoLC	quantization supergravity entropy electric create quantization supergravity entropy geometry calculation

In addition, a topic coherence measure is used to evaluate the models objectively. Coherence of a set of words measures the hanging and fitting together of single words or subsets of them. The method presented in [22] can cover all existing measures and construct new measures, which demonstrates that the coherence measure (C_v) has the best performance among PMI, NPMI, UMass, etc. The higher the value, the better the coherence. C_v combines the indirect cosine measure with the NPMI and the boolean sliding window. For each top word w_i of a topic word set, its vector v_i is calculated based on context words, and the l-element v_{il} is the NMPI value with the l-context word w_l is as follows:

$$v_{il} = NPMI(w_i, w_l) = \frac{\log \frac{p(w_i,w_l)+\epsilon}{p(w_i)p(w_l)}}{-\log(P(w_i, w_l)+\epsilon)}.$$

Probability, $P(w_i, w_l)$, is estimated based on word co-occurrence counts derived from virtual documents by a sliding window over Wikipedia. Each window position defines such a document.

Next, topic coherence C_v is averaged on the cosine similarity summation of any two top words w_i and w_j as $C_v = avg(\sum_{1 \le i \le j \le N} cos(v_i, v_j))$, where N is the number of top words of a topic, and $avg()$ and $cos()$ are the average function and the cosine similarity function, respectively. C_v is realized by tool gensim.

Table 6 shows the coherence values of C_v for the STE and steoLC models on the small-hep datasets. We can see that our steoLC model generally improves the coherence of the learned topics. Compared with the STE model, our model incorporates link information to improve the quality of the detected topics.

Table 6. Topic coherence evaluation with different numbers of top words.

Model	T = 5	T = 10	T = 15
STE	0.510	0.456	0.463
steoLC	0.521	0496	0.496

5. Discussion and Conclusions

A novel joint probabilistic model, the steoLC model, is first provided for topic discovery and polysemy word embedding based on document link networks. An algorithm is then designed to learn the model parameters by the EM algorithm with a negative sampling algorithm. The algorithm of the steoLC model learns the topic distribution on documents, and the embedding distribution on topics for words as central words and context words, as well as the link distribution on topics. This method not only mines topics and word embeddings by integrating contents with links, but also solves the polysemous problem, reducing the fuzzy semantic of topics. Compared with the state-of-the-art methods on several tasks, the experimental results demonstrate the superiority of our model. However, the proposed algorithm is difficult to apply on large-scale data due to the complexity of our probabilistic model. It may be helpful to use a python library for probabilistic models, such as ZhuSuan. In addition, the steoLC model does not model multi-granularity texts in the same representation space, which will affect the performance and explainability of topic discovery and text representations. In the future, we will further use the graph convolutional neural network or the depth generation model to address these problems.

Author Contributions: Methodology, B.C., X.J. and J.G.; investigation, B.C. and L.M.; data curation, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Natural Science Foundation of Hebei Province (No.F2019403070) and the science and technology research project for universities of Hebei (ZD2020175, ZD2020344) and the Science and Technology Innovation Team Project of Hebei GEO University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is downloaded from https://dblp.uni-trier.de/xml/ and https://www.cs.cornell.edu/projects/kddcup/datasets.html, accessed on 26 February 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hofmann, T. Probabilistic Latent Semantic Analysis. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99), Stockholm, Sweden, 30 July–1 August 1999; Laskey, K.B., Prade, H., Eds.; Morgan Kaufmann: Burlington, MA, USA, 1999; pp. 289–296.
2. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
3. Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.

4. Li, S.; Chua, T.S.; Zhu, J.; Miao, C. Generative Topic Embedding: A Continuous Representation of Documents. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–13 August 2016; pp. 666–675.
5. Das, R.; Zaheer, M.; Dyer, C. Gaussian LDA for Topic Models with Word Embeddings. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015.
6. Nguyen, D.Q.; Billingsley, R.; Du, L.; Johnson, M. Improving Topic Models with Latent Feature Word Representations. *Trans. Assoc. Comput. Linguist.* **2015**, *3*, 299–313. [CrossRef]
7. Liu, Y.; Liu, Z.; Chua, T.; Sun, M. Topical Word Embeddings. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Bonet, B., Koenig, S., Eds.; AAAI Press: Cambridge, MA, USA, 2015; pp. 2418–2424.
8. Briakou, E.; Athanasiou, N.; Potamianos, A. Cross-Topic Distributional Semantic Representations Via Unsupervised Mappings. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; (Long and Short Papers); Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 1052–1061. [CrossRef]
9. Xun, G.; Li, Y.; Gao, J.; Zhang, A. Collaboratively Improving Topic Discovery and Word Embeddings by Coordinating Global and Local Contexts. In Proceedings of the 23rd ACM SIGKDD International Conference, Halifax, NS, Canada, 13–17 August 2017.
10. Zhu, L.; He, Y.; Zhou, D. A Neural Generative Model for Joint Learning Topics and Topic-Specific Word Embeddings. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 471–485. [CrossRef]
11. Li, D.; Zhang, J.; Li, P. TMSA: A Mutual Learning Model for Topic Discovery and Word Embedding. In Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, AB, Canada, 2–4 May 2019; Berger-Wolf, T.Y., Chawla, N.V., Eds.; SIAM: Philadelphia, PA, USA, 2019; pp. 684–692. [CrossRef]
12. Shi, B.; Lam, W.; Jameel, S.; Schockaert, S.; Lai, K.P. Jointly learning word embeddings and latent topics. In Proceedings of the SIGIR 2017 International Conference, Tokyo, Japan, 7–11 August 2017.
13. Sun, H.; He, F.; Huang, J.; Sun, Y.; Li, Y.; Wang, C.; He, L.; Sun, Z.; Jia, X. Network Embedding for Community Detection in Attributed Networks. *ACM Trans. Knowl. Discov. Data* **2020**, *14*, 36:1–36:25. [CrossRef]
14. Chunaev, P. Community detection in node-attributed social networks: A survey. *Comput. Sci. Rev.* **2020**, *37*, 100286. [CrossRef]
15. Bothorel, C.; Cruz, J.D.; Magnani, M.; Micenková, B. Clustering Attributed Graphs: Models, Measures and Methods. *arXiv* **2015**, arXiv:1501.01676.
16. Liu, X.; Wang, W.; He, D.; Jiao, P.; Jin, D.; Cannistraci, C.V. Semi-supervised community detection based on non-negative matrix factorization with node popularity. *Inf. Sci.* **2017**, *381*, 304–321. [CrossRef]
17. Falih, I.; Grozavu, N.; Kanawati, R.; Bennani, Y. Community detection in Attributed Network. In Proceedings of the Companion of the The Web Conference 2018 on The Web Conference 2018, Lyon, France, 23–27 April 2018; Champin, P., Gandon, F.L., Lalmas, M., Ipeirotis, P.G., Eds.; ACM: New York, NY, USA, 2018; pp. 1299–1306. [CrossRef]
18. Yang, T.; Jin, R.; Chi, Y.; Zhu, S. Combining Link and Content for Community Detection. In Proceedings of the Acm Sigkdd International Conference on Knowledge Discovery Data Mining, Paris, France, 28 June–1 July 2009.
19. Jin, D.; Huang, J.; Jiao, P.; Yang, L.; He, D.; Fogelman-Soulié, F.; Huang, Y. A Novel Generative Topic Embedding Model by Introducing Network Communities. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; ACM: New York, NY, USA, 2019; pp. 2886–2892. [CrossRef]
20. Tang, J.; Zhang, J.; Yao, L. ArnetMiner: Extraction and Mining of Academic Social Networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data, Las Vegas, NV, USA, 24–27 August 2008; pp. 990–998.
21. Laurens, V.D.M.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
22. RoDer, M.; Both, A.; Hinneburg, A. *Exploring the Space of Topic Coherence Measures*; ACM: New York, NY, USA, 2015; pp. 399–408.