Lifelong Learning of Topics and Domain-Specific Word Embeddings

Xiaorui Qin, Yuyin Lu, Yufu Chen, Yanghui Rao[*]
School of Computer Science and Engineering,
Sun Yat-sen University, Guangzhou, China
qinxr5@mail2.sysu.edu.cn, luyy37@mail2.sysu.edu.cn,
chenyf66@mail2.sysu.edu.cn, raoyangh@mail.sysu.edu.cn

Abstract

Lifelong topic models mainly focus on in-domain text streams in which each chunk only contains documents from a single domain. To overcome data diversity of the in-domain corpus, most of the existing methods exploit the information from limited sources in a separate and heuristic manner. In this study, we develop a lifelong collaborative model (LCM) based on non-negative matrix factorization to accurately learn topics and domain-specific word embeddings. LCM particularly investigates: (1) developing a knowledge graph based on the semantic relationships among words in the lifelong learning process, so as to accumulate global context information discovered by topic models and local context information reflected by context word embeddings from previous domains, and (2) developing a subword graph based on byte pair encoding and pairwise word relationships to exploit subword information of words in the current in-domain corpus. To the best of our knowledge, we are the first to collaboratively learn topics and word embeddings via lifelong learning. Experiments on real-world in-domain text streams validate the effectiveness of our method.

1 Introduction

Lifelong learning (Silver, 2011; Mitchell et al., 2015), which accumulates and maintains the past knowledge to help future learning in an endless manner, has attracted considerable attention in topic modeling (Chen et al., 2020b; Gupta et al., 2020). Most lifelong topic models (Chen and Liu, 2014b; Chen, 2015; Wang et al., 2016) focus on the corpus that only contains text from a single domain, dubbed the in-domain corpus (Xu et al., 2018). This is because in-domain corpora are widespread in real-world applications, such as breaking news and tweets related to a specific topic (domain). The key to the success of a lifelong topic model within in-domain corpora is based on a precondition that prior topical information of previous domains can be fully exploited to guide meaningful learning in the new coming domain (Chen and Liu, 2014b). However, because the in-domain corpus is typically of limited size (Xu et al., 2018), it is insufficient for the existing methods to train coherent topics.

To alleviate the lack of global context in a corpus, one simple solution for topic models is to incorporate general-purpose pre-trained word embeddings (Das et al., 2015; Xun et al., 2016, 2017b; Dieng et al., 2020). Although the general-purpose embeddings can provide some useful information for words within the in-domain corpus, their embedding representations may not be ideal for the target domain and in some cases they may even conflict with the meanings of the words in the task domain because words often have multiple senses or meanings (Xu et al., 2018). Another solution trains topics and word embeddings jointly in the one-shot learning scenario (Xun et al., 2017a; Dieng et al., 2020). Such a unified method prevents relying on the external embedding corpus that is not always closely aligned with the domain task, because the model can learn domain-specific word embeddings by itself. Unfortunately, the aforementioned models are conducted on collected documents without the guidance of any prior knowledge. Besides, they all treat words as atomic units, which may not perform well on the in-domain corpus with relatively few words.

In light of these considerations, we aim to generate coherent topics and domain-specific word embeddings jointly by a lifelong process. On the one hand, domain-specific word embeddings tend to offer more accurate complementary information to lifelong topic modeling than pre-trained embeddings. On the other hand, we alleviate the lack of global and local context information within in-

[*] The corresponding author.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2294–2309
August 1–6, 2021. ©2021 Association for Computational Linguistics

domain corpora by exploiting subwords (Pinter et al., 2017). Both the topical and subword information are leveraged in our knowledge-based learner to generate better domain-specific word embeddings. To achieve this, we propose a lifelong collaborative model (LCM)[1] by coordinating global context, local context, and subword information. First, our LCM maintains a knowledge graph based on word relationships to accumulate the past knowledge learned from previous domains, which exploits from both the global word-document matrix and the local word co-occurrence matrix. Second, we develop a subword graph from the current in-domain corpus to capture extra information of words. We use non-negative matrix factorization (NMF) as our framework, which is an effective method of mining latent text semantics with great flexibility in transforming prior knowledge into regulations (Lee and Seung, 1999; Chen et al., 2015, 2020b) and it gives sparseness to matrices with interpretability (Hoyer, 2004). The main contributions of this study can be summarized as follows:

- We propose a lifelong learning method to jointly generate topics and word embeddings over in-domain text streams. To the best of our knowledge, we are the first to collaboratively learn topics and domain-specific word embeddings through a lifelong process.
- We incorporate local context information and subword information into lifelong topic modeling, which can alleviate the lack of global context information when the target dataset is relatively small.
- In lifelong word embedding learning, we leverage the topical and subword information to help generate better domain-specific embeddings for down-stream learning tasks.

2 Related Work

Topic modeling (Deerwester et al., 1990; Hofmann, 1999; Blei et al., 2003) and word embedding learning (Mikolov et al., 2013a,b) are two of the most important tasks in natural language processing. The former task aims to discover the latent semantic structure of documents based on the global context, while the latter one follows the distributional hypothesis that words occurring in similar local contexts tend to have similar syntactic and semantic properties (Harris, 1954). The traditional topic and word embedding learning models are based on isolated learning, i.e., a one-shot task learning, thus they lack ability to continually learn from incrementally available data.

Lifelong Topic Modeling. A lifelong topic model (Chen and Liu, 2014b; Chen, 2015; Wang et al., 2016), as a typical example of lifelong machine learning, is gaining more and more research interests than traditional one-shot deal that conducts a topic model on collected documents just for once (Chen et al., 2020b). Lifelong topic models inherit three key characteristics in lifelong machine learning, i.e., continuous tasks, knowledge accumulation and maintenance, and a knowledge-based learner that can leverage the past knowledge to help future learning in a never-ending manner. Furthermore, lifelong topic modeling is mainly applied to in-domain corpora where each chunk only contains text from a single domain.

Based on NMF, Chen et al. (2020b) proposed a lifelong topic model named NMF-LTM. However, the above method only considers the most important 10 words under every topic while ignores other non-top words, i.e., most words in the vocabulary. This problem will be more serious if the vocabulary size is large. Besides, NMF-LTM may perform poorly for other downstream tasks, because it can only capture the information of limited words in sentences. Finally, NMF-LTM only mines word relationships from the perspective of global (topical) information, which is inadequate within in-domain corpora. Considering the limited global context in the new coming corpus, Gupta et al. (2020) incorporated general-purpose pre-trained word embeddings as complementary to topics into the knowledge base for lifelong learning. Unfortunately, the above method required that the dimension of word embeddings being equal to the number of topics and each dimension of word embeddings corresponding to a topic. This violates the complementary but different points of view, i.e., the global viewpoint and the local viewpoint, for topic models and word embedding models (Xun et al., 2017a). This model optimizes a topic-word matrix, in which each row represents the word distribution of a topic and each column represents the embedding of a word. Each dimension of the word embedding learned by this model implied the possibility of the word occurring in the corresponding topic. However, word embeddings contain many other features that cannot be captured by global (context) information, e.g., the syntactic feature.

[1] https://github.com/XiaoruiQ/LCM

Word Embedding Learning. Lifelong learning has also been adopted to train domain-specific word embeddings, which fills the gap between general-purpose embeddings trained on large-scale corpora and the topic (domain) of the down-stream task. For example, Xu et al. (2018) first developed a meta-learner to expand the new in-domain corpus by measuring the content similarity of past domains and the new domain. Then, they generated word embeddings for the new domain using the combined data. However, this method only considered the local context information from past domains, which is inadequate to capture the polysemous nature of words. As an illustration, *apple* is one of polysemous words that is topically contextualized by several domains, i.e., product line, operating system, and fruit (Gupta et al., 2020).

3 Lifelong Collaborative Model

In this section, we detail the proposed LCM for jointly learning topics and domain-specific word embeddings in a lifelong process. The topical information and local context from previous domains, and a subword graph constructed from the current in-domain corpus are exploited in LCM to guide future tasks.

3.1 Problem Formalization

Given a stream of document chunks $\left\{\mathcal{DOC}_t\right\}_{t=1}^{T}$ accumulated in an endless manner ($T = +\infty$), we aim to jointly generate topics and domain-specific word embeddings when each chunk only contains text from a single domain. At any time point, our LCM deals with the current document chunk, e.g., $\mathcal{DOC}_t$, by leveraging the past knowledge learned from the previous document chunks, i.e., $\mathcal{DOC}_1$, $\mathcal{DOC}_2$, …, $\mathcal{DOC}_{t-1}$. Table 1 lists the notations used in this paper. We use bold uppercase letters such as $\boldsymbol{D}_t$ to represent matrices, regular uppercase letters such as M to represent scalar constants, and regular lowercase letters such as λ_v to represent scalar variables.

Notation	Description
$\boldsymbol{D}_t \in R^{M \times N}$	Word-document matrix at the current moment
$\boldsymbol{U}_t \in R^{M \times K}$	Word-topic matrix at the current moment
$\boldsymbol{V}_t \in R^{K \times N}$	Topic-document matrix at the current moment
$\boldsymbol{X}_t \in R^{M \times M}$	Word co-occurrence matrix at the current moment
$\boldsymbol{B}_t \in R^{M \times E}$	Word embedding matrix at the current moment
$\boldsymbol{C}_t \in R^{M \times E}$	Context word embedding matrix at the current moment
M	The number of words
N	The number of documents
K	The number of topics
E	The dimension of word embeddings

Table 1: Frequently used notations.

3.2 Objective Function

Figure 1 illustrates the architecture of our LCM, which processes in-domain text streams through a knowledge-based learner. Formally, the objective function of LCM is defined as follows:

$$
\begin{aligned}
L = \|\boldsymbol{D}_t - \boldsymbol{U}_t\boldsymbol{V}_t\|_F^2 + \left\|\boldsymbol{X}_t - \boldsymbol{B}_t\boldsymbol{C}_t^T\right\|_F^2 \\
+ \Upsilon(\boldsymbol{V}_t) + \Psi(\boldsymbol{U}_t) + \Phi(\boldsymbol{C}_t) + \Omega(\boldsymbol{B}_t), \\
s.t. \quad \boldsymbol{U}_t \geq 0, \boldsymbol{V}_t \geq 0, \boldsymbol{B}_t \geq 0, \boldsymbol{C}_t \geq 0. \quad (1)
\end{aligned}
$$

It is noteworthy that we constrain the non-negativity of $\boldsymbol{B}_t$ and $\boldsymbol{C}_t$ to learn sparse interpretable word embeddings (Murphy et al., 2012; Luo et al., 2015), so as to capture the polysemous nature of words (Panigrahi et al., 2019). With non-negativity constraints, words are represented by limited dimensions (Murphy et al., 2012). All words that have positive values under specific dimensions may share a common characteristic, which enhances the interpretability of word embeddings and helps capture the polysemous nature.

The first term of our objective function aims to factorize the global word-document matrix $\boldsymbol{D}_t$ into the word-topic matrix $\boldsymbol{U}_t$ and the topic-document matrix $\boldsymbol{V}_t$, and the interpretability of $\boldsymbol{U}_t$ and $\boldsymbol{V}_t$ is ensured by their non-negativity. For the local context information, Levy and Goldberg (2014) have proved that the Skip-Gram model with negative sampling (SGNS) is implicitly factorizing a positive pointwise mutual information word co-occurrence matrix shifted by a constant offset. Accordingly, we use the shifted positive pointwise mutual information matrix as our word co-occurrence matrix $\boldsymbol{X}_t$ and decompose it into the word embedding matrix $\boldsymbol{B}_t$ and the context word embedding matrix $\boldsymbol{C}_t$, as presented in the second term. Given a hyperparameter λ_v, the sparsity constraint on $\boldsymbol{V}_t$ is introduced as the third term $\Upsilon(\boldsymbol{V}_t) = \lambda_v \|\boldsymbol{V}_t\|_1$. This ensures that each document covers limited topics (Chen et al., 2020b). The sparsity of topics encourages interpretable topics (Card et al., 2018), which corresponds with the tuition that a document usually focuses on several salient topics instead of covering a wide variety of topics (Lin et al., 2019). Although NMF has given sparseness to V_t, a more direct control over such properties of the representation is still needed (Hoyer, 2004). The rest terms $\Psi(\boldsymbol{U}_t)$, $\Phi(\boldsymbol{C}_t)$, and $\Omega(\boldsymbol{B}_t)$ are the constraints on matrices $\boldsymbol{U}_t$, $\boldsymbol{C}_t$, and $\boldsymbol{B}_t$, which will be described in sections 3.2.3-3.2.5, respectively.

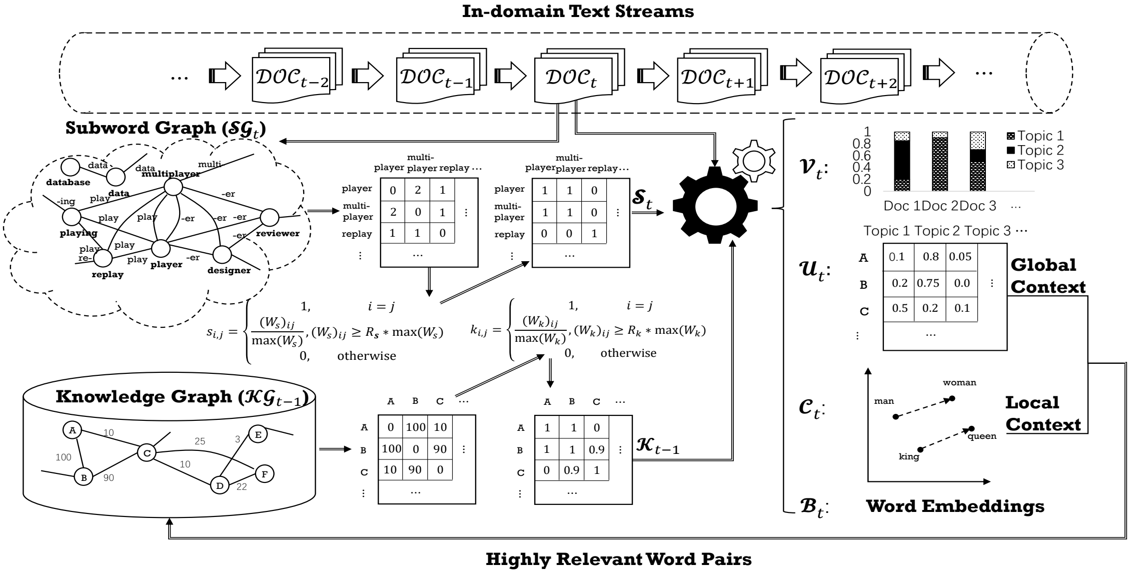

Figure 1: Lifelong Collaborative Model (LCM)

3.2.1 Knowledge Graph ($\mathcal{KG}$)

LCM uses relationships between words as the representations of our $\mathcal{KG}$ to maintain knowledge of past domains to help with the current in-domain task. $\mathcal{KG}$ accumulates the knowledge of past domains from two sources of information, i.e., the global context information mined by topic models and the local context information reflected by context word embeddings. As shown in Figure 1, the output of LCM contains the word-topic matrix $\boldsymbol{U}_t$, the topic-document matrix $\boldsymbol{V}_t$, the word embedding matrix $\boldsymbol{B}_t$, and the context word embedding matrix $\boldsymbol{C}_t$. $\mathcal{KG}$ fuels the global context and local context information with the help of $\boldsymbol{U}_t$ and $\boldsymbol{C}_t$, as follows.

For the global context information, we use the inner product to measure similarities between topic distributions of words in $\boldsymbol{U}_t$. For each word w_i in the current vocabulary of $\boldsymbol{D}_t$, we find $topT$ words w_j $(j = 1, 2, ..., T)$, whose topic distributions of all topics are most similar to w_i. Each w_j and w_i are seen as word pairs that reflect the relationship from the global context information, i.e., the topical information. After finding $topT$ related words of each word, all the word pairs are accumulated and de-duplicated. Following (Chen et al., 2020b), we set the weight of each word pair (w_i, w_j) to 1.

Regarding the local context information, we use the inner product to measure similarities between context word embeddings of words in $\boldsymbol{C}_t$. For each word w_i in the current vocabulary of $\boldsymbol{D}_t$, we find $topT$ words whose context word embeddings are most similar to this word. All word pairs represent the relationship from the perspective of local context information and their weights are set to η after de-duplication, where η adjusts the weight relationship between global context information and local context information.

Then, we accumulate the word pairs from these two sources to fuel the information of global context and local context. It is worth noting that if a word pair (w_i, w_j) appears simultaneously in the two kinds of word pairs, its weight is recorded as $1 + \eta$. De-duplication is not required here, because the relationship between two words related in both global context and local context is closer than that of two words only related in one kind of source information. We use $\mathcal{J}_t$ to denote all the related word pairs in the current in-domain corpus, and $\mathcal{J}_t$ is defined as:

$$\mathcal{J}_t = \{(W_k)_{ij}\}, \tag{2}$$

where $(W_k)_{ij}$ represents the weight of the word pair (w_i, w_j) in $\mathcal{KG}$. Finally, $\mathcal{KG}$ is updated as follows:

$$\mathcal{KG}_t = \mathcal{KG}_{t-1} + \mathcal{J}_t. \tag{3}$$

3.2.2 Subword Graph ($\mathcal{SG}$)

To incorporate subword information, LCM uses a subword graph ($\mathcal{SG}$) to store relationships between words in $\boldsymbol{D}_t$ from the perspective of subword information. The motivation of introducing $\mathcal{SG}$ is to capture more information from the structures of words themselves as the complement for global and local contexts (Bojanowski et al., 2017; Pinter et al., 2017). Many in-domain text streams contain a high proportion of rare words with low word frequencies, e.g., proper nouns in specific domains, which cannot be adequately reflected by context due to the low frequencies. Note that $\mathcal{SG}$ only mines the subword information of the current in-domain corpus since subwords are only related to domain-independent structures of words.

A typical word in English is composed by three kinds of subword units, i.e., the word root, the prefix, and the suffix[2]. Word roots and prefixes determine the meaning of words, while suffixes determine the syntactic-related part of speech. We adopt byte pair encoding (BPE) (Sennrich et al., 2016), which can implicitly match these morpheme boundaries, to conduct subword segmentation. We also compare this segmentation method with character n-gram features (Bojanowski et al., 2017) in experiments. For every word pair, the number of shared subword units between them are recorded as the weight. In the current in-domain corpus, $\mathcal{SG}_t$ is defined as:

$$\mathcal{SG}_t = \{(W_s)_{ij}\}, \tag{4}$$

where $(W_s)_{ij}$ represents the weight of the word pair (w_i, w_j) in $\mathcal{SG}_t$.

3.2.3 Constraint on $\boldsymbol{U}_t$

Before introducing $\Psi(\boldsymbol{U}_t)$, we first construct the word-word relationship matrix $\boldsymbol{K}_{t-1} \in R^{M\times M}$ from $\mathcal{KG}_{t-1}$ to represent the closeness of relationships between words. In $\mathcal{KG}_{t-1}$, we select all of the word pairs in which both of the two words occurred in the current vocabulary of $\boldsymbol{D}_t$. Only these words contribute to the current in-domain task on $\boldsymbol{D}_t$, and all diagonal elements of $\boldsymbol{K}_{t-1}$ are 1. R_k, which represents the threshold ratio for $\mathcal{KG}$, is used to select the "close" relationships between words. For two words w_i and w_j, if the corresponding pair (w_i, w_j) occurred in the selected word pairs in $\boldsymbol{D}_t$ mentioned above, the value of k_{ij} will be determined by the threshold ratio R_k. If the weight of (w_i, w_j), i.e., $(W_k)_{ij}$, is greater than or equal to the max weight of all the word pairs in $\boldsymbol{D}_t$ multiplied by R_k, then $k_{ij} = \frac{(W_k)_{ij}}{max(W_k)}$. If it is less than the max weight multiplied by R_k or w_i and w_j are not connected in $\mathcal{KG}_{t-1}$, then $k_{ij} = 0$. In the above, R_k helps to select word pairs with relatively large weights. Although wrong connections between some word pairs are kept in our $\mathcal{KG}$, the weights of them cannot be large enough, because the max weight of $\mathcal{KG}$ becomes larger and larger over time. These pairs will not be chosen to participate in constraints of matrices. In summary, $\boldsymbol{K}_{t-1}$ is calculated as follows:

$$k_{ij} = \begin{cases} 1, & i = j \\ \dfrac{(W_k)_{ij}}{max(W_k)}, & (W_k)_{ij} \geq R_k max(W_k) \\ 0, & otherwise. \end{cases} \tag{5}$$

The word-word relationship regularization based on $\mathcal{KG}$ for $\Psi(\boldsymbol{U}_t)$ holds that the topic distributions of words that are closely related in $\mathcal{KG}$ would be more similar than those have no connection in $\mathcal{KG}$. We use the Graph Laplacian (Dai et al., 2020) as the first part of $\Psi(\boldsymbol{U}_t)$ to depict that under each topic in $\boldsymbol{U}_t$, the closer two words are connected in $\mathcal{KG}$, the closer their probabilities are. The second part of $\Psi(\boldsymbol{U}_t)$ is the diversity regularization to reduce the overlapping of topics, i.e., to improve the topic uniqueness (Nan et al., 2019). Accordingly, $\Psi(\boldsymbol{U}_t)$ is defined as follows:

$$\begin{aligned}\Psi(\boldsymbol{U}_t) =& \lambda_{u1} tr(\boldsymbol{U}_t^T \boldsymbol{H}_{t-1} \boldsymbol{U}_t) \\ &+ \lambda_{u2} \left\| \boldsymbol{U}_t^T \boldsymbol{U}_t - \boldsymbol{I}_K \right\|_F^2 .\end{aligned} \tag{6}$$

In the above, λ_{u1} and λ_{u2} are hyperparameters. $\boldsymbol{H}_{t-1} = diag(\boldsymbol{K}_{t-1} \cdot \mathbf{1}) - \boldsymbol{K}_{t-1}$ represents the Graph Laplacian of $\boldsymbol{K}_{t-1}$, where $\mathbf{1}$ represents a column vector in which all of the elements are 1, and $diag(\boldsymbol{K}_{t-1} \cdot \mathbf{1})$ represents the matrix with the vector $\boldsymbol{K}_{t-1} \cdot \mathbf{1}$ as diagonal elements. $\boldsymbol{I}_K$ is an identity matrix of order $K \times K$.

3.2.4 Constraint on $\boldsymbol{C}_t$

$\mathcal{KG}$, which fuels the information of global context and local context from previous domains, is constructed with the help of $\boldsymbol{U}_t$ and $\boldsymbol{C}_t$. It also contributes to both the two matrices on their constraints. For $\Phi(\boldsymbol{C}_t)$, we use $\boldsymbol{K}_{t-1}$ to introduce the word-word relationship regularization. It depicts that context embeddings of words that are closely related in $\mathcal{KG}$ would be more similar than those have less connection in $\mathcal{KG}$. Specifically, under each dimension in $\boldsymbol{C}_t$, the closer two words are connected in $\mathcal{KG}$, the closer their representations are. $\Phi(\boldsymbol{C}_t)$ is calculated as follows:

$$\Phi(\boldsymbol{C}_t) = \lambda_c tr(\boldsymbol{C}_t^T \boldsymbol{H}_{t-1} \boldsymbol{C}_t), \tag{7}$$

where λ_c is a hyperparameter.

[2] https://en.wikipedia.org/wiki/Root_(linguistics)

3.2.5 Constraint on $\boldsymbol{B}_t$

The first part of $\Omega(\boldsymbol{B}_t)$ is similar to the word-word relationship regularization for $\Psi(\boldsymbol{U}_t)$ and $\Phi(\boldsymbol{C}_t)$. It holds that embeddings of words that are closely related in $\mathcal{SG}$ would be more similar than those have less connection in $\mathcal{SG}$. We also use the Graph Laplacian to depict that under each dimension in $\boldsymbol{B}_t$, the closer two words are connected in $\mathcal{SG}$, the closer their representations are. A word-word relationship matrix $\boldsymbol{S}_t \in R^{M \times M}$ is constructed from $\mathcal{SG}_t$, as follows:

$$s_{ij} = \begin{cases} 1, & i = j \\ \dfrac{(W_s)_{ij}}{max(W_s)}, & (W_s)_{ij} \geq R_s max(W_s) \\ 0, & otherwise, \end{cases} \quad (8)$$

where $(W_s)_{ij}$ represents the weight of pair (w_i, w_j) in SG_t, and R_s denotes the threshold ratio of SG. R_s helps exclude and ignore wrong connections in $\mathcal{SG}$. In addition, the cooperation of $\mathcal{KG}$ and $\mathcal{SG}$ can further reduce the influence of unimportant edges. For example, a connection that is only selected by R_k may not be as important as the connection that is selected by both of R_k and R_s simultaneously.

The second part is a sparsity constraint on $\boldsymbol{B}_t$, which depicts that each word only has features of a limited number, because we aim to learn sparse representations so that the generated domain-specific word embeddings are more interpretable.

Finally, $\Omega(\boldsymbol{B}_t)$ is defined as follows:

$$\Omega(\boldsymbol{B}_t) = \lambda_{b1} tr(\boldsymbol{B}_t^T \boldsymbol{N}_t \boldsymbol{B}_t) + \lambda_{b2} \left\| \boldsymbol{B}_t \right\|_1, \quad (9)$$

where λ_{b1} and λ_{b2} are hyperparameters. $\boldsymbol{N}_t = diag(\boldsymbol{S}_t \cdot \mathbf{1}) - \boldsymbol{S}_t$ is the Graph Laplacian of $\boldsymbol{S}_t$.

3.3 Alternately Iterative Algorithm

We develop an alternately iterative algorithm to achieve a good compromise between ease of implementation and speed. Take $\boldsymbol{B}_t$ as an example, we first calculate the derivative of the objective function L on $\boldsymbol{B}_t$ as follows:

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{B}_t} = & -2\boldsymbol{X}_t\boldsymbol{C}_t + 2\boldsymbol{B}_t\boldsymbol{C}_t^T\boldsymbol{C}_t \\ & + 2\lambda_{b1} diag(\boldsymbol{S}_t \cdot \mathbf{1})\boldsymbol{B}_t - 2\lambda_{b1}\boldsymbol{S}_t\boldsymbol{B}_t \\ & + \lambda_{b2} \cdot \mathbf{1} \cdot \mathbf{1}^T. \end{aligned} \quad (10)$$

Based on the derivative of L on $\boldsymbol{B}_t$, the updating rule for $\boldsymbol{B}_t$ is given below:

$$\boldsymbol{B}_t \leftarrow \boldsymbol{B}_t \circ \frac{\boldsymbol{X}_t\boldsymbol{C}_t + \lambda_{b1}\boldsymbol{S}_t\boldsymbol{B}_t}{\boldsymbol{B}_t\boldsymbol{C}_t^T\boldsymbol{C}_t + J(\boldsymbol{B}_t)}, \quad (11)$$

where $J(\boldsymbol{B}_t) = \lambda_{b1} diag(\boldsymbol{S}_t \cdot \mathbf{1})\boldsymbol{B}_t + \frac{\lambda_{b2}}{2} \cdot \mathbf{1} \cdot \mathbf{1}^T$. Note that $\boldsymbol{U}_t$, $\boldsymbol{V}_t$, $\boldsymbol{C}_t$, and $\boldsymbol{B}_t$ always satisfy the non-negativity because they are updated in this multiplication form. Due to the limited space, we provide the updating rules for matrices $\boldsymbol{U}_t$, $\boldsymbol{V}_t$, and $\boldsymbol{C}_t$, the parameter inference process, the theoretical proof of the algorithmic convergence, and the time complexity analysis in Appendices A-D.

3.4 Model Scalability

Our model has a good scalability due to the "divide-and-conquer" strategy. First, we partition the large corpus into several small document chunks that belong to different domains, and we only decompose matrices of one chunk at any time. Second, we use sparse matrices to store $\mathcal{KG}$ and $\mathcal{SG}$, thus they are scalable and can be processed fast. Third, when facing a large single domain corpus in text streams, we can partition it into small sub-domain corpora and process one corpus at each time.

4 Experiments

4.1 Dataset

We evaluate our LCM on the real-world Amazon Review dataset[3] (McAuley et al., 2015; He and McAuley, 2016) from 28 departments (i.e., the first-level category). Following (Xu et al., 2018), we consider all the reviews under each second-level category as a domain. Each domain has several third-level categories, which will be used for all down-stream tasks and model evaluation. We randomly select 9 in-domain corpora to carry out experiments. Table 2 summarizes the characteristics of the selected 9 corpora, i.e., domain names, numbers of reviews and labels, the average text lengths, and vocabulary sizes. The selected corpora are pre-processed by eliminating stop-words and words with frequency (in the total reviews from 9 domains) lower than 15. Also, reviews with less than 20 words are removed.

Note that we choose the above dataset instead of other datasets for lifelong topic modeling (Gupta et al., 2020) since we focus on in-domain corpora with rare overlaps, in which, documents share few common information. For completeness, we also shuffle these 7 training domains randomly and show the results under different permutations in Appendix E.

[3] http://snap.stanford.edu/data/web-Amazon.html

Domain Name	#Docs	#Labels	Length	Vocab
Safety Signs & Signals	11,298	8	32.12	16,090
Pet Behavior Center	7,450	5	43.34	14,301
SIM Cards & Prepaid Minutes	13,872	4	42.29	16,726
Horses	13,571	11	35.98	19,608
Video	10,088	5	76.35	23,196
Keyboards & MIDI	15,585	3	59.09	24,703
Characters & Series	16,663	6	67.77	35,393
Science Education	7,975	13	38.05	18,107
Cult Movies	15,252	13	83.25	47,668

Table 2: Dataset statistics.

4.2 Experimental Setting

We simulate an endless lifelong learning process using the 9 corpora. Word-word relationships in the knowledge graph are gradually accumulated on the first 7 domains. With the help of this knowledge graph, we conduct experiments on the last 2 domains and compare the model performance on both lifelong topic modeling and domain-specific word embedding learning.

We randomly select 5% reviews from "Cult Movies (CM)" for validation. To build the knowledge graph for the validation set, 5% reviews from the first 7 corpora are sampled and reviews from these 8 small-scale in-domain corpora also construct a stream of document chunks. The idea of grid search (Fayed and Atiya, 2019) is used to select the best parameters for each metric. We provide our hyperparameters search space for grid search in Appendix F. The remaining 95% reviews of the first 7 domains construct the training set. All reviews from "Science Education (SE)" and the remaining 95% documents from CM are used as the testing set.

4.3 Baselines

For lifelong topic modeling, we compare our method with the following baselines: LDA-LTM[4] (Chen and Liu, 2014b), NMF-LTM (Chen et al., 2020b), and LNTM[5] (Gupta et al., 2020). For word embedding learning, we adopt FastText[6] (Bojanowski et al., 2017), L-DEM (Xu et al., 2018), SPINE[7] (Subramanian et al., 2018), and Word2Sense[8] (Panigrahi et al., 2019). To the best of our knowledge, L-DEM is the only work focuses on lifelong learning of domain-specific word embedding. Following (Xu et al., 2018), we train other baselines in two ways for evaluation, i.e., only on the new in-domain corpus, and on the total document set by fusing the new corpus and all corpora from the previous domains. We implement NMF-LTM and L-DEM by Python according to the original papers. For the sake of fairness, the parameters of all baselines[9] are selected on the validation set in the same experimental setting as LCM.

4.4 Evaluation Metrics

Suggested by (Lau et al., 2014; Chen and Liu, 2014a,b; Wang et al., 2016; Isonuma et al., 2020), we use the normalized pointwise mutual information (NPMI) (Aletras and Stevenson, 2013) score, which closely matches human judgments, to measure the coherence of representative words of topics generated by lifelong topic models. Following (Chen et al., 2020b), top 20 words of each topic are used for calculation. Considering that it is important to discover discriminative topics, we also adopt the topic uniqueness (TU) score (Nan et al., 2019) to measure the diversity of topics. In addition, the sparsity score of the document-topic distribution (TS-U) and the topic-word distribution (TS-V) proposed by Lin et al. (2019) is further used to measure the topic sparsity quantitatively. Particularly, we use 1e-20 as the threshold to count the number of zero values in document-topic and topic-word distributions. Only values that are smaller than 1e-20 can be set to zero. Although several studies (Chang et al., 2009; Newman et al., 2010) stated that the perplexity is unable to reflect the real semantic coherence of topics and even negatively correlated with human judgements, we show this metric of each model for completeness.

As domain-specific dictionaries are relatively small and meanwhile may contain some uncommon words, it is inappropriate to evaluate domain-specific word embeddings in traditional ways, e.g., calculating the word similarity. Following (Xu et al., 2018), we build a down-stream text classification task to evaluate domain-specific word embeddings generated by different models. The two testing sets, i.e., SE and CM, are used for text classification with their third-level categories as classification labels. For each review, we use the average embedding of all of the words as its feature vector to train a SVM classifier (Bayot and Gonçalves, 2016; Qin and Wang, 2009). We use

[4] https://github.com/czyuan/LTM

[5] https://github.com/pgcool/Lifelong-Neural-Topic-Modeling

[6] https://github.com/bamtercelboo/cw2vec

[7] https://github.com/harsh19/SPINE

[8] https://github.com/abhishekpanigrahi1996/Word2Sense

[9] The search scope of each parameter that is different with our method is obtained from the original papers.

Method	Science Education (SE)					Cult Movies (CM)				
	NPMI↑	TU↑	TS-U↑	TS-V↑	Perplexity↓	NPMI↑	TU↑	TS-U↑	TS-V↑	Perplexity↓
LDA-LTM	-0.0201	**0.6620**	0.0000	0.0000	9081.5853	0.0509	0.6060	0.0000	0.0000	14191.7899
LNTM	-0.3855	0.3610	0.0000	0.0000	9031.3161	-0.2072	0.4160	0.0000	0.0000	14057.3928
NMF-LTM	-0.0453	0.3940	0.2012	0.1725	**8564.4900**	0.0386	0.2660	0.1125	0.0099	**10870.6600**
LCM	**0.0012**	0.5940	**0.7782**	**0.5526**	8706.1200	**0.0633**	**0.6340**	**0.6781**	**0.4466**	13040.8700

Table 3: Performance comparison of lifelong topic models. For all metrics, "↓" after the metric indicates smaller is better while "↑" indicates larger is better. The best performance on each measure is highlighted by boldface.

accuracy to evaluate the effectiveness of word embeddings on the down-stream text classification task as in (Xu et al., 2018).

4.5 Result Comparison

4.5.1 Lifelong Topic Modeling

Table 3 shows the performance of different models on the lifelong topic discovery task, from which we can observe that LCM performs the best or the second best on each measure.

For the baselines, NMF-LTM achieves the best perplexity while almost the poorest TU. As mentioned in (Burkhardt and Kramer, 2019), there is a tradeoff between perplexity and TU in some cases, which means that models generating a lot of redundant topics may have a meaningless low perplexity. The reason of obtaining a low TU for NMF-LTM may be that it enforces documents within the same class would have more similar topic distributions, which is unsuitable to handle in-domain text streams since all documents in the in-domain corpus come from one class. This also influences its sparsity. For example, a document only has non-zero values for 10 topics while another document from the same class has non-zero values for another 10 topics. To get similar, they may both become non-zero for 20 topics. LNTM faces the same problem because it does not constrain the diversity among topics explicitly. The coherence score of LNTM is also unsatisfactory. A possible reason is that it treats word embeddings as topic distributions of words, which deteriorates the local semantic information captured by context word embeddings. LNTM entirely neglected the sparsity of document-topic and topic-word distributions, thus its TS-U and TS-V are zero.

LDA-LTM has a main difference with the other models, i.e., it does not construct the in-domain text stream based on time series, but fuses all of the previous domains together and accumulates the knowledge from this large corpus to help with the current in-domain corpus. Even though it is difficult to compare LDA-LTM with other lifelong topic models fairly, our LCM performs better than LDA-LTM in most cases, because we can exploit the information from local context and subwords.

The qualitative analysis of topics generated by these models is provided in Appendix G.

4.5.2 Word Embedding Learning

Figure 2 reports the accuracy of text classification using word embeddings with different dimensions. Note that sparse interpretable word embeddings always perform better when the embedding dimension is relatively large (Murphy et al., 2012). LCM performs the best in each testing set under each dimension, although some baselines (i.e., FastText, SPINE, and Word2Sense) are trained on the total 9 corpora and access more information without considering time series. L-DEM cannot learn high-quality embeddings because it needs a large amount of in-domain corpora to train its meta-learner, which is not accessible in most applications. Compared to FastText that incorporates subword information, our word embeddings perform better on classification in all cases. One possible reason is that the global context provides LCM with extra information. Two sparse word embedding models, i.e., SPINE and Word2Sense, overemphasize the sparsity while ignore the quality for downstream tasks. LCM balances well between sparsity and quality with the help of global context and subwords. We evaluate sparsity and interpretability of word embeddings in Appendix H.

For completeness, we also replace the SVM classifier with a neural network classifier (Chen et al., 2020a) consisting of 3 fully-connected layers. The results are shown in Appendix I.

4.5.3 Ablation Experiments

Take SE as an example, we report results of ablation experiments on NPMI, TU, and Accuracy in Table 4. "LCM-$\mathcal{KG}_\mathcal{G}$", "LCM-$\mathcal{KG}_\mathcal{L}$", and "LCM-$\mathcal{SG}$" represent LCM without the participation of global context in $\mathcal{KG}$, local context in $\mathcal{KG}$, and subwords in $\mathcal{SG}$, respectively. Deleting each part leads to performance degradation, which validates the

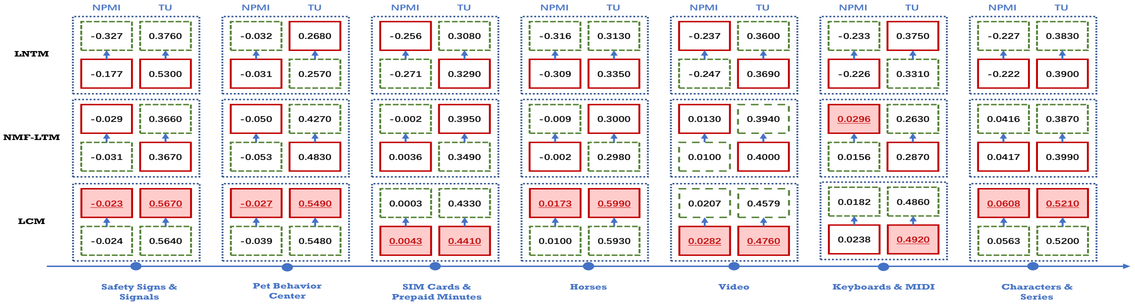

Figure 3: Analysis on catastrophic forgetting. The x-axis shows the names of domains. For each model, the bottom line indicates metrics in the original lifelong learning process, while the top line shows the model performance with the help of $\mathcal{KG}$ generated by 7 training sets and CM. The boxes with red and solid edges mark the better metric for each model in each domain, while boxes filled in red with underlined values mark the best metric in each domain among all models.

effectiveness of global context, local context, and subwords. Compared to the subword information, $\mathcal{KG}$ contributes more to topic discovery and the local context information plays the most important role in word embedding learning. We replace BPE of $\mathcal{SG}$ with character n-gram features (Bojanowski et al., 2017) in "LCM-$\mathcal{SG}_{\mathcal{BPE}}$", which indicates the effectiveness of BPE on capturing subwords.

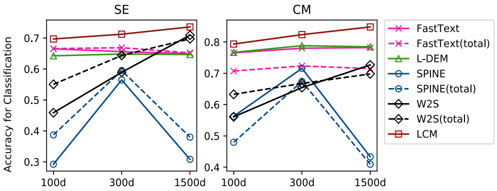

Figure 2: Classification performance comparison of SVM classifiers with word embeddings generated by different methods. Models marked with "total" are conducted on the combination of 7 training sets and the current testing set. W2S represents Word2Sense.

Method	NPMI	TU	Accuracy
LCM-$\mathcal{KG}_{\mathcal{G}}$	-0.0099	0.5790	0.7283
LCM-$\mathcal{KG}_{\mathcal{L}}$	-0.0109	0.5740	0.7244
LCM-$\mathcal{SG}$	-0.0076	0.5810	0.7288
LCM-$\mathcal{SG}_{\mathcal{BPE}}$	-0.0123	0.5800	0.7314
LCM	**0.0012**	**0.5940**	**0.7360**

Table 4: Results of ablation experiments.

4.5.4 Analysis on Catastrophic Forgetting

Catastrophic Forgetting (Robins, 1995; Kirkpatrick et al., 2017), which is a big challenge for lifelong topic models, will not be a serious problem for LCM. The learning process of LCM only accumulates knowledge in $\mathcal{KG}$, and the model is trained on each in-domain corpus independently by following (Chen et al., 2020b). To further investigate the ability of LCM in avoiding catastrophic forgetting, we use the final updated $\mathcal{KG}$ after CM to "go back" to help train the model on the 7 training set one by one. As LDA-LTM does not construct the in-domain text stream based on time series, we only take NMF-LTM and LNTM for comparison. In terms of NPMI and TU, Figure 3 shows that LCM has the best ability to alleviate catastrophic forgetting. For all domains, the latest $\mathcal{KG}$ will not have a significant negative impact on LCM (i.e., the catastrophic forgetting is limited), and sometimes it even helps with the new task. For example, both NMF-LTM and LCM achieve better NPMI scores with the latest $\mathcal{KG}$ in "SIM Cards & Prepaid Minutes". One possible reason is that later domains provide valuable information through $\mathcal{KG}$.

5 Conclusions

In this work, we propose a lifelong collaborative model (LCM) for learning topics and domain-specific word embeddings. LCM deals with the new in-domain corpus by coordinating global and local context information from previous domains, and subword information from the current corpus. A knowledge graph based on word-word relationships is leveraged during the learning process. Experiments on real-world in-domain text streams demonstrated the superior performances of LCM. In the future, we plan to incorporate contextualized word representations into topic models (Bianchi et al., 2020, 2021) for alleviating collapsing of word senses and learning more coherent topics.

Acknowledgment

We are grateful to the reviewers for their constructive comments and suggestions on this study. This work has been supported by the National Natural Science Foundation of China (61972426) and Guangdong Basic and Applied Basic Research Foundation (2020A1515010536).

References

Nikolaos Aletras and Mark Stevenson. 2013. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS)*, pages 13–22.

Roy Khristopher Bayot and Teresa Gonçalves. 2016. Multilingual author profiling using word embedding averages and SVMs. In *Proceedings of the 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, pages 382–386.

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2020. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. *CoRR*, abs/2004.03974.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021. Cross-lingual contextualized topic models with zero-shot learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1676–1683.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Sophie Burkhardt and Stefan Kramer. 2019. Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model. *Journal of Machine Learning Research*, 20:131:1–131:27.

Dallas Card, Chenhao Tan, and Noah A. Smith. 2018. Neural models for documents with metadata. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2031–2040.

Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS)*, pages 288–296.

Hong-You Chen, Sz-Han Yu, and Shou-de Lin. 2020a. Glyph2vec: Learning chinese out-of-vocabulary word embedding from glyphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2865–2871.

Yong Chen, Junjie Wu, Jianying Lin, Rui Liu, Hui Zhang, and Zhiwen Ye. 2020b. Affinity regularized non-negative matrix factorization for lifelong topic modeling. *IEEE Transactions on Knowledge and Data Engineering*, 32(7):1249–1262.

Yong Chen, Hui Zhang, Junjie Wu, Xingguang Wang, Rui Liu, and Mengxiang Lin. 2015. Modeling emerging, evolving and fading topics using dynamic soft orthogonal NMF with sparse representation. In *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM)*, pages 61–70.

Zhiyuan Chen. 2015. Lifelong machine learning for topic modeling and beyond. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 133–139.

Zhiyuan Chen and Bing Liu. 2014a. Mining topics in documents: Standing on the shoulders of big data. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1116–1125.

Zhiyuan Chen and Bing Liu. 2014b. Topic modeling using topics from many domains, lifelong learning and big data. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 703–711.

Ling-Yun Dai, Rong Zhu, and Juan Wang. 2020. Joint nonnegative matrix factorization based on sparse and graph laplacian regularization for clustering and co-differential expression genes analysis. *Complexity*, 2020:3917812:1–3917812:10.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 795–804.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Adji Bousso Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Hatem A. Fayed and Amir F. Atiya. 2019. Speed up grid-search for parameter selection of support vector machines. *Applied Soft Computing*, 80:202–210.

Pankaj Gupta, Yatin Chaudhary, Thomas A. Runkler, and Hinrich Schütze. 2020. Neural topic modeling with continual lifelong learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 3907–3917.

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

Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 507–517.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 50–57.

Patrik O. Hoyer. 2004. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469.

Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2020. Tree-structured neural topic model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 800–806.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 530–539.

Daniel D. Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2177–2185.

Tianyi Lin, Zhiyue Hu, and Xin Guo. 2019. Sparsemax and relaxed wasserstein for topic sparsity. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 141–149.

Hongyin Luo, Zhiyuan Liu, Huan-Bo Luan, and Maosong Sun. 2015. Online learning of interpretable word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1687–1692.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 43–52.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR)*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil A. Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2302–2310.

Brian Murphy, Partha Pratim Talukdar, and Tom M. Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 1933–1950.

Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic modeling with wasserstein autoencoders. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 6345–6381.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Proceedings of the Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 100–108.

Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. 2019. Word2sense: Sparse interpretable word embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 5692–5705.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword rnns. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 102–112.

Yu-ping Qin and Xiu-kun Wang. 2009. Study on multi-label text classification based on SVM. In *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 300–304.

Anthony V. Robins. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725.

Daniel L. Silver. 2011. Machine lifelong learning: Challenges and benefits for artificial general intelligence. In *Proceedings of the 4th International Conference on Artificial General Intelligence (AGI)*, pages 370–375.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard H. Hovy. 2018. SPINE: Sparse interpretable neural embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 4921–4928.

Shuai Wang, Zhiyuan Chen, and Bing Liu. 2016. Mining aspect-specific opinion using a holistic lifelong topic model. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 167–176.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Lifelong domain word embedding via meta-learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4510–4516.

Guangxu Xun, Vishrawas Gopalakrishnan, Fenglong Ma, Yaliang Li, Jing Gao, and Aidong Zhang. 2016. Topic discovery for short texts using word embeddings. In *Proceedings of the 16th IEEE International Conference on Data Mining (ICDM)*, pages 1299–1304.

Guangxu Xun, Yaliang Li, Jing Gao, and Aidong Zhang. 2017a. Collaboratively improving topic discovery and word embeddings by coordinating global and local contexts. In *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 535–543.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017b. A correlated topic model using word embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4207–4213.

Appendices

A Updating Rules for $\boldsymbol{U}_t$, $\boldsymbol{V}_t$, and $\boldsymbol{C}_t$

First, we transform the objective function L as:

$$
\begin{aligned}
L = & \|\boldsymbol{D}_t - \boldsymbol{U}_t\boldsymbol{V}_t\|_F^2 + \left\|\boldsymbol{X}_t - \boldsymbol{B}_t\boldsymbol{C}_t^T\right\|_F^2 \\
& + \lambda_v \|\boldsymbol{V}_t\|_1 \\
& + \lambda_{u1} tr(\boldsymbol{U}_t^T\boldsymbol{H}_{t-1}\boldsymbol{U}_t) + \lambda_{u2}\left\|\boldsymbol{U}_t^T\boldsymbol{U}_t - \boldsymbol{I}_K\right\|_F^2 \\
& + \lambda_c tr(\boldsymbol{C}_t^T\boldsymbol{H}_{t-1}\boldsymbol{C}_t) \\
& + \lambda_{b1} tr(\boldsymbol{B}_t^T\boldsymbol{N}_t\boldsymbol{B}_t) + \lambda_{b2}\|\boldsymbol{B}_t\|_1 \\
= & tr(\boldsymbol{D}_t^T\boldsymbol{D}_t - 2\boldsymbol{D}_t^T\boldsymbol{U}_t\boldsymbol{V}_t + \boldsymbol{V}_t^T\boldsymbol{U}_t^T\boldsymbol{U}_t\boldsymbol{V}_t) \\
& + tr(\boldsymbol{X}_t^T\boldsymbol{X}_t - 2\boldsymbol{X}_t^T\boldsymbol{B}_t\boldsymbol{C}_t^T + \boldsymbol{C}_t\boldsymbol{B}_t^T\boldsymbol{B}_t\boldsymbol{C}_t^T) \\
& + \lambda_v tr(\boldsymbol{1}^T\boldsymbol{V}_t\boldsymbol{1}) \\
& + \lambda_{u1} tr\left(\boldsymbol{U}_t^T diag(\boldsymbol{W}_{t-1}\cdot\boldsymbol{1})\boldsymbol{U}_t\right) \\
& - \lambda_{u1} tr(\boldsymbol{U}_t^T\boldsymbol{W}_{t-1}\boldsymbol{U}_t) \\
& + \lambda_{u2} tr(\boldsymbol{U}_t^T\boldsymbol{U}_t\boldsymbol{U}_t^T\boldsymbol{U}_t - 2\boldsymbol{U}_t^T\boldsymbol{U}_t) \\
& + \lambda_c tr\left(\boldsymbol{C}_t^T diag(\boldsymbol{W}_{t-1}\cdot\boldsymbol{1})\boldsymbol{C}_t\right) \\
& - \lambda_c tr(\boldsymbol{C}_t^T\boldsymbol{W}_{t-1}\boldsymbol{C}_t) \\
& + \lambda_{b1} tr\left(\boldsymbol{B}_t^T diag(\boldsymbol{S}_t\cdot\boldsymbol{1})\boldsymbol{B}_t\right) \\
& - \lambda_{b1} tr(\boldsymbol{B}_t^T\boldsymbol{S}_t\boldsymbol{B}_t) \\
& + \lambda_{b2} tr(\boldsymbol{1}^T\boldsymbol{B}_t\boldsymbol{1}) + const.
\end{aligned}
$$

Then, the updating rules for matrices $\boldsymbol{U}_t$, $\boldsymbol{V}_t$, and $\boldsymbol{C}_t$ can be derived as follows:

$$
\begin{aligned}
\boldsymbol{U}_t &\leftarrow \boldsymbol{U}_t \circ \frac{\boldsymbol{D}_t\boldsymbol{V}_t^T + \lambda_{u1}\boldsymbol{W}_{t-1}\boldsymbol{U}_t + 2\lambda_{u2}\boldsymbol{U}_t}{\boldsymbol{U}_t\boldsymbol{V}_t\boldsymbol{V}_t^T + Z(\boldsymbol{U}_t)}, \\
\boldsymbol{V}_t &\leftarrow \boldsymbol{V}_t \circ \frac{\boldsymbol{U}_t^T\boldsymbol{D}_t}{\boldsymbol{U}_t^T\boldsymbol{U}_t\boldsymbol{V}_t + \frac{\lambda_v}{2}\cdot\boldsymbol{1}\cdot\boldsymbol{1}^T}, \\
\boldsymbol{C}_t &\leftarrow \boldsymbol{C}_t \circ \frac{\boldsymbol{X}_t^T\boldsymbol{B}_t + \lambda_c\boldsymbol{W}_{t-1}\boldsymbol{C}_t}{\boldsymbol{C}_t\boldsymbol{B}_t^T\boldsymbol{B}_t + \lambda_c diag(\boldsymbol{W}_{t-1}\cdot\boldsymbol{1})\boldsymbol{C}_t}.
\end{aligned}
$$

In the above, $Z(\boldsymbol{U}_t) = \lambda_{u1} diag(\boldsymbol{W}_{t-1}\cdot\boldsymbol{1})\boldsymbol{U}_t + 2\lambda_{u2}\boldsymbol{U}_t\boldsymbol{U}_t^T\boldsymbol{U}_t$.

B Alternately Iterative Algorithm

The inference method of our LCM is shown in Algorithm 1.

C Convergence Analysis

In this section, we analyze the convergence of Algorithm 1.

Theorem C.1. *Algorithm 1 is guaranteed to converge to a locally-optimal solution.*

Algorithm 1 Alternately Iterative Algorithm

```
Input: $\{\mathcal{DOC}_t\}_{t=1}^T$ ($T$ can be infinite);
Output: $\{\boldsymbol{U}_t\}_{t=1}^T$, $\{\boldsymbol{V}_t\}_{t=1}^T$, $\{\boldsymbol{C}_t\}_{t=1}^T$, $\{\boldsymbol{B}_t\}_{t=1}^T$.
$\mathcal{KG}_0 = \emptyset$;
for $t$ = 1, 2, 3, ..., $T$ do
    Randomly initialize non-negative matrices $\boldsymbol{U}_t^{(0)}$, $\boldsymbol{V}_t^{(0)}$, $\boldsymbol{C}_t^{(0)}$, $\boldsymbol{B}_t^{(0)}$;
    Construct $\boldsymbol{S}_t$ from $\mathcal{SG}_t$;
    Construct $\boldsymbol{K}_{t-1}$ from $\mathcal{KG}_{t-1}$;
    Set the iteration number $i$ to 0;
    repeat
        $i = i + 1$;
        Compute $\boldsymbol{B}_t^{(i)}$, $\boldsymbol{U}_t^{(i)}$, $\boldsymbol{V}_t^{(i)}$, and $\boldsymbol{C}_t^{(i)}$;
    until convergence;
    Compute $\mathcal{KG}_t$ using Eq. (3).
end
```

First, we prove the convergence of the update rule of $\boldsymbol{B}_t$ in Eq. (11).

Definition C.1. *$G(x, z)$ is an auxiliary function for $F(x)$ if the following conditions are satisfied.*

$$G(x, z) \geq F(x), G(x, x) = F(x).$$

Lemma C.1. *If G is an auxiliary function, then F is non-increasing under the following update rule:*

$$x^{t+1} = arg\min_x G(x, x^t). \tag{12}$$

Proof.

$$F(x^{t+1}) \leq G(x^{t+1}, x^t) \leq G(x^t, x^t) = F(x^t).$$

□

If we could prove that the updating rule of $\boldsymbol{B}_t$ confirms to Eq. (12) for an appropriate auxiliary function, we would conclude that $\boldsymbol{B}_t$ converges to a local minimum.

Lemma C.2. *Let $z = (\boldsymbol{B}_t)_{ij} > 0$, $G(x, z)$ is an auxiliary function for $F(z) = L\left((\boldsymbol{B}_t)_{ij} = z\right)$.*

$$
\begin{aligned}
G(x, z) = & F(z) + \frac{\partial F(z)}{\partial z}(x - z) + \frac{(x - z)^2}{z} \\
& (\boldsymbol{B}_t\boldsymbol{C}_t^T\boldsymbol{C}_t + \lambda_{b1} diag(\boldsymbol{S}_t\cdot\boldsymbol{1})\boldsymbol{B}_t \\
& + \frac{\lambda_{b2}}{2}\cdot\boldsymbol{1}\cdot\boldsymbol{1}^T)_{ij}.
\end{aligned}
$$

Proof. Clearly, $G(x, x) = F(x)$. Taylor expansion of $F(x)$ is:

$$F(x) = F(z) + \frac{\partial F(z)}{\partial z}(x - z) + \frac{1}{2}\frac{\partial^2 F(z)}{\partial z^2}(x - z)^2.$$

In order to show G is an auxiliary function, we have to show $G(x,z) \geq F(x)$:

$$\frac{\left(\boldsymbol{B}_t\boldsymbol{C}_t^T\boldsymbol{C}_t + \lambda_{b1} diag(\boldsymbol{S}_t \cdot \mathbf{1})\boldsymbol{B}_t + \frac{\lambda_{b2}}{2} \cdot \mathbf{1} \cdot \mathbf{1}^T\right)_{ij}}{z}$$

$$\geq \frac{1}{2}\frac{\partial^2 F(z)}{\partial z^2},$$

where $\frac{\partial^2 F(z)}{\partial^2 z} = 2(\boldsymbol{C}_t^T\boldsymbol{C}_t)_{jj} + 2\lambda_{b1}(diag(\boldsymbol{S}_t \cdot \mathbf{1}))_{ii} - 2\lambda_{b1}(\boldsymbol{S}_t)_{ii}$.

To prove the inequation, we first verify how the inequality holds on the first term:

$$\begin{aligned}\frac{(\boldsymbol{B}_t\boldsymbol{C}_t^T\boldsymbol{C}_t)_{ij}}{(\boldsymbol{B}_t)_{ij}} =& \frac{1}{(\boldsymbol{B}_t)_{ij}}\sum_k (\boldsymbol{B}_t)_{ik}(\boldsymbol{C}_t^T\boldsymbol{C}_t)_{kj} \\ =& \frac{1}{(\boldsymbol{B}_t)_{ij}}\sum_{k=j} (\boldsymbol{B}_t)_{ik}(\boldsymbol{C}_t^T\boldsymbol{C}_t)_{kj} \\ &+ \frac{1}{(\boldsymbol{B}_t)_{ij}}\sum_{k\neq j} (\boldsymbol{B}_t)_{ik}(\boldsymbol{C}_t^T\boldsymbol{C}_t)_{kj} \\ =& \frac{1}{(\boldsymbol{B}_t)_{ij}}(\boldsymbol{B}_t)_{ij}(\boldsymbol{C}_t^T\boldsymbol{C}_t)_{jj} \\ &+ \frac{1}{(\boldsymbol{B}_t)_{ij}}\sum_{k\neq j} (\boldsymbol{B}_t)_{ik}(\boldsymbol{C}_t^T\boldsymbol{C}_t)_{kj} \\ \geq& (\boldsymbol{C}_t^T\boldsymbol{C}_t)_{jj}.\end{aligned}$$

Similarly, we can get:

$$\lambda_{b1}\left(diag(\boldsymbol{S}_t \cdot \mathbf{1})\boldsymbol{B}_t\right)_{ij} \geq \lambda_{b1}\left(diag(\boldsymbol{S}_t \cdot \mathbf{1})\right)_{ii}.$$

Since λ_{b1}, λ_{b2}, and each element in $\boldsymbol{S}_t$ are nonnegative, we have the above inequation. This establishes that G is an auxiliary function for F. □

Proof. To show that Algorithm 1 converges (i.e., Theorem C.1), we need to show that update rule for $\boldsymbol{B}_t$ follows Eq. (12). $\frac{\partial G(x,z)}{\partial x}$ is listed as follows:

$$\begin{aligned}\frac{\partial G(x,z)}{\partial x} =& (-2\boldsymbol{X}_t\boldsymbol{C}_t + 2\boldsymbol{B}_t\boldsymbol{C}_t^T\boldsymbol{C}_t \\ &+ 2\lambda_{b1} diag(\boldsymbol{S}_t \cdot \mathbf{1})\boldsymbol{B}_t - 2\lambda_{b1}\boldsymbol{S}_t\boldsymbol{B}_t \\ &+ \lambda_{b2} \cdot \mathbf{1} \cdot \mathbf{1}^T)_{ij} \\ &+ \frac{x-z}{z} \cdot 2(\lambda_{b1} diag(\boldsymbol{S}_t \cdot \mathbf{1})\boldsymbol{B}_t \\ &+ \boldsymbol{B}_t\boldsymbol{C}_t^T\boldsymbol{C}_t + \frac{\lambda_{b2}}{2} \cdot \mathbf{1} \cdot \mathbf{1}^T)_{ij}.\end{aligned}$$

Solving $\frac{\partial G(x,z)}{\partial x} = 0$ for x, we get the update rule as mentioned in Eq. (11). Since G is the auxiliary function for F, the value of F is non-increasing. We can prove the convergence of update rules for $\boldsymbol{U}_t$, $\boldsymbol{V}_t$, and $\boldsymbol{C}_t$ similarly. Thus, Algorithm 1 is guaranteed to converge to a local minimum. □

D Time Complexity Analysis

In this section, we analyze the time complexity of Algorithm 1. For updating matrices $\boldsymbol{B}_t$, $\boldsymbol{U}_t$, $\boldsymbol{V}_t$, and $\boldsymbol{C}_t$, the time complexity of one iteration is $O((3E+1)M^2+(2E^2+3E)M)$, $O((3E+1)M^2+(2E^2+2E)M)$, $O(MNK+2M^2K+3MK^2+NK^2+M^2+2MK)$, and $O(3KMN+3KN)$, respectively. Thus, the time complexity of each iteration is $O(4MNK+(6E+3)M^2+(4E^2+5E)M+2M^2K+3MK^2+NK^2+2MK+3KN)$ for our method, which spends an extra time cost of $O((6E+2)M^2+(4E^2+5E)M-(2K+1)N^2)$ to learn word embeddings when compared with the previous NMF-based lifelong topic model, i.e., NMF-LTM (Chen et al., 2020b). Although the time complexity is proportional to M, we can alleviate the scalability issue simply. For example, if M (i.e., the vocabulary) of a single domain is too large, we can partition this domain into several small subdomain corpora. At each time, we only process matrices of one sub-domain, and M of each small sub-domain will not be too large.

As an illustration, it costs about 30 seconds per iteration for training LCM based on a workstation equipped with Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50 GHz, 8 cores and 128G memory. To achieve convergence, LCM costs about 1 hour to update $\boldsymbol{B}_t$, $\boldsymbol{U}_t$, $\boldsymbol{V}_t$, and $\boldsymbol{C}_t$ for all domains in order. NMF-LTM costs about half an hour accordingly.

E Different Permutations of Training Domains

We shuffle training domains randomly for 5 times, and show the results under these different permutations in Table 5, which indicates that LCM is robust to domain permutations.

F Search Space of Hyperparameters

Table 6 shows the search space of hyperparameters for grid search in our LCM. To ensure a fair comparison with NMF-LTM (Chen et al., 2020b), *topT* in LCM is set to 5. For each word, our knowledge graph selects 5 most similar words from global and local context information, respectively. In other words, each word is connected to 10 similar words, which is equivalent to NMF-LTM. Note that for the topic number, we select 50 as topic numbers for all evaluations of LCM after validation. For fair comparison, we directly choose 50 as topic numbers for baselines and this is not a hyperparameter

Permutation	Science Education (SE)				Cult Movies (CM)			
	NPMI↑	TU↑	Perplexity↓	Accuracy↑	NPMI↑	TU↑	Perplexity↓	Accuracy↑
1	-0.0072	0.5870	8708.0200	0.7207	0.0639	0.6340	13350.4800	0.8455
2	-0.0064	0.5840	8704.2400	0.7196	0.0570	0.6090	13420.5600	0.8461
3	-0.0106	0.5960	8701.5600	0.7196	0.0580	0.6170	13373.0800	0.8443
4	-0.0053	0.5880	8705.9500	0.7171	0.0524	0.6090	13369.4400	0.8467
5	-0.0013	0.5700	8744.0600	0.7135	0.0562	0.6030	13554.5300	0.8480

Table 5: Performance comparison of LCM within different permutations of the dataset. For all metrics, "↓" after the metric indicates smaller is better while "↑" indicates larger is better.

Hyperparameter	Search Space
λ_v	[0.001]
λ_{u2}	[0.5]
λ_{b2}	[0.001, 0.1]
λ_{u1}, λ_c, λ_{b1}	[0.1, 1, 10]
$topT$	[5]
η	[0.1, 0.5, 1, 2, 10]
R_k, R_s	[0.5, 0.6]
topic number	[50, 200]
word embedding dimension	[100, 300, 1500]
iterations	[100, 200]

Table 6: Hyperparameters' search space.

Hyperparameter	NPMI	TU	Perplexity	Accuracy
λ_v	0.001	0.001	0.001	0.001
λ_{u2}	0.5	0.5	0.5	0.5
λ_{b2}	0.001	0.1	0.1	0.1
λ_{u1}	10	10	10	0.1
λ_c	10	0.1	10	10
λ_{b1}	0.1	1	10	1
$topT$	5	5	5	5
η	1	1	2	1
R_k	0.5	0.5	0.6	0.5
R_s	0.6	0.6	0.6	0.5
topic number	50	50	50	50
word embedding dimension	1500	1500	1500	300
iterations	200	200	200	200

Table 7: Best hyperparameters on the validation set.

Hyperparameter	Range	Var-NPMI	Var-Accuracy
λ_{u1}	[0.01, 0.1, 1, 10, 100]	1.28E-05	1.79E-05
λ_{b1}	[0.01, 0.1, 1, 10, 100]	2.71E-05	3.50E-03
λ_c	[0.01, 0.1, 1, 10, 100]	9.32E-07	5.15E-06
R_k	[0.3, 0.5, 0.6, 0.8]	6.57E-05	3.08E-05
iterations	[50, 100, 200, 300]	1.28E-05	3.92E-05

Table 8: Variances of results.

of baselines. We list the best hyperparameters on the validation set for different metrics in Table 7. We provide the search space of hyperparameters for grid search in all of our baselines and their corresponding best hyperparameters in our codes.

For hyperparameters, we first varied search spaces for sensitive analysis and observed that LCM was robust to most hyperparameters. Thus, we used final search spaces in Table 6. For completeness, we also show the variances of results under different hyperparameter values in Table 8. Take some hyperparameters in CM as examples, we vary each parameter when others are fixed, and compute the variances of NPMI and Accuracy, which indicates that LCM is robust to most hyperparameters.

G Qualitative Analysis of Topics

Following (Chen et al., 2020b), we map the topics learned by LCM with ones by LDA-LTM (Chen and Liu, 2014b), LNTM (Gupta et al., 2020), and NMF-LTM (Chen et al., 2020b), respectively. Particularly, we represent each topic by its top 20 words, and compute the cosine similarity between every two topics. Take SE as an example, we randomly show 3 topics, as listed in Table 9. Irrelevant words are marked by italics.

LDA-LTM, NMF-LTM, and LCM learn topics well in most cases. However, NMF-LTM captures some high-frequency words (i.e., amazon, anatomy, anatomical) in the corpus, which are not related to the topic "Design and production of handicrafts". LDA-LTM also assigns an irrelevant word "print" to the topic "Machinery and industrial manufacturing technology", and an irrelevant word "spring" to the topic "Body structure and anatomy".

The ability of LNTM in generating cohesive topics is poor. It is noteworthy that the topics generated by LNTM seem not related to other models, because we use cosine similarity to map topics. If the cosine similarities are the same (for LNTM, it is 0 sometimes), the topics with smaller IDs will be chosen. For the sake of fairness, we also show a relatively coherent topic generated by LNTM separately in Table 10. The result of LNTM is still worse than other models, which contains 7 irrelevant words in the top 10 word list.

H Evaluating Interpretability

To evaluate the interpretability of our domain-specific word embeddings, we follow (Murphy et al., 2012) to show top 5 words for 5 randomly chosen dimensions in word embeddings generated from CM. Although there exists noisy words, the dimensions are generally semantically coherent and interpretable. We also choose one polysemous word "cell" in SE to measure the ability of our method in capturing the polysemous nature of words. We select the two highest values of the word vector learned by our LCM for "cell", and find top 5 words in these two dimensions. From Table 11, we can observe that the two dimensions focus on cells in biology and cell phones, which reflect the two different meanings of "cell".

ID	LDA-LTM	LNTM	NMF-LTM	LCM
		Machinery and industrial manufacturing technology		
1	bed	*deserved*	mm	mm
2	mm	clamped	bed	bed
3	axis	*mothers*	frame	frame
4	screws	*crowds*	screws	screws
5	screw	*players*	axis	axis
6	belt	*suspiciously*	screw	set
7	motor	*resolutions*	stepper	screw
8	nuts	*wasps*	power	stepper
9	*print*	*military*	wires	lead
10	set	*collapses*	tape	tape
		Design and production of handicrafts		
1	model	*deserved*	model	brain
2	brain	*crowded*	brain	paint
3	pieces	*mothers*	models	models
4	models	*rider*	structures	pieces
5	paint	*mutual*	*anatomy*	structures
6	heart	*trends*	paint	lines
7	motor	*wasp*	*amazon*	*budget*
8	quality	*bloom*	detail	detail
9	painted	*fumbling*	*anatomical*	white
10	job	*habitat*	stand	detailed
		Body structure and anatomy		
1	skull	*deserved*	skull	skull
2	teeth	*crowded*	teeth	teeth
3	jaw	*mothers*	anatomy	anatomy
4	quality	*rider*	jaw	jaw
5	size	*mutual*	bone	mandible
6	top	*trends*	mandible	foramen
7	*spring*	*wasp*	foramen	bone
8	removable	*bloom*	detail	detail
9	life	*fumbling*	skulls	study
10	skulls	*habitat*	removable	removable

Table 9: Qualitative analysis of topics. Top 10 words are listed with irrelevant ones marked in italics.

ID	LNTM
	Music and instruments
1	*tens*
2	*deserved*
3	*habitat*
4	*circuitry*
5	guitars
6	*mothers*
7	headphones
8	piano
9	*rider*
10	*foray*

Table 10: Qualitative analysis of LNTM. Top 10 words are listed with irrelevant ones marked in italics.

Top Words	depleted	constant	drained	dissolve	exhausted
	painted	spray	paint	grease	di
	icing	cake	riffing	esp	wrapping
	weed	pot	smoking	smoke	smoked
	angora	sweater	dressing	sweaters	skirt
Close Words for "cell"	cell	sewn	animal	cells	believes
	phones	cell	games	fascinating	computers

Table 11: Examples of evaluating interpretability.

Table 12 shows the classification accuracy and the sparsity (i.e., the proportion of zeros) of word embeddings generated by different methods, where the dimension is 1500. Compared with other models, our LCM generates domain-specific word embeddings with a good balance between sparsity and classification accuracy.

Method	Science Education		Cult Movies	
	Accuracy	Sparsity	Accuracy	Sparsity
SPINE	0.3080	0.9969	0.4332	0.9979
Word2Sense	0.7043	0.9920	0.7267	0.9920
LCM	0.7360	0.9438	0.8481	0.9099

Table 12: Classification accuracy and sparsity of word embeddings generated by different methods.

I Evaluating Word Embedding Learning Models by Neural Networks

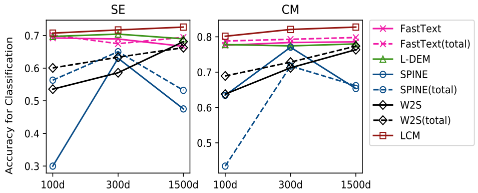

Figure 4: Classification performance comparison of neural networks with word embeddings generated by different methods.

To compare different word embedding learning models comprehensively, we replace the SVM classifier with a neural network classifier consisting of 3 fully-connected layers. As shown in Figure 4, LCM also performs the best in each testing set under each dimension.