WETM: A Word Embedding-based Topic Model with Modified Collapsed Gibbs Sampling for Short Text

Junaid Rashid[a], Jungeun Kim[b], Amir Hussain[c], Usman Naseem[d]

[a]Department of Data Science, Sejong University, Seoul, 05006, Republic of Korea, Email:junaid.rashid@sejong.ac.kr
[b]Department of Software, Kongju National University, Cheonan, 31080, Republic of Korea, Email:jekim@kongju.ac.kr
[c]Data Science and Cyber Analytics Research Group, Edinburgh Napier University, Edinburgh, EH11 4DY, UK, Email:a.hussain@napier.ac.uk
[d]School of Computer Science, University of Sydney, Sydney, Australia, Email:usman.naseem@sydney.edu.au

Abstract

Short texts are a common source of knowledge, and the extraction of such valuable information is beneficial for several purposes. Traditional topic models are incapable of analyzing the internal structural information of topics. They are mostly based on the co-occurrence of words at the document level and are often unable to extract semantically relevant topics from short text datasets due to their limited length. Although some traditional topic models are sensitive to word order due to the strong sparsity of data, they do not perform well on short texts. In this paper, we propose a novel word embedding-based topic model (WETM) for short text documents to discover the structural information of topics and words and eliminate the sparsity problem. Moreover, a modified collapsed Gibbs sampling algorithm is proposed to strengthen the semantic coherence of topics in short texts. WETM extracts semantically coherent topics from short texts and finds relationships between words. Extensive experimental results on two real-world datasets show that WETM achieves better topic quality, topic coherence, classification, and clustering results. WETM also requires less execution time compared to traditional topic models.

Keywords: Topi Modeling, Short Text, Classification, Topic Coherence

1. Introduction

With the proliferation of mobile devices, short text has become a common way of information delivery. Numerous natural language processing tasks, such as emerging topic detection, content analysis, question answering, sentiment analysis, automatic summarization, and recommendation systems, need to discover potential topics from a short text [1]. Nevertheless, the sparse information in short texts leads to insufficient contextual information, and the differences in language phrases ensure that it is difficult to analyze them using standard methods.

A topic model defines topics as multinomial word distributions. A document is composed of multinomial topics, and words are based on topics. However, the topics are distributed across the entire collection. Therefore, each topic is considered the center of the group of words grouped based on patterns of co-occurrence. Words that appear in several documents should be assigned to a similar topic. Therefore, limitations in the length of the document, the pattern of word co-occurrence is rarely found in the small text corpus, the main conventionally imperfect topic [2]. In another method, semantic relationships between terms are not adequately described in a short text.

Probabilistic topic models consider documents a combination of probabilistic topics, such as a probability distribution over words, and are used to identify topics from large sets of documents automatically. The conventional topic models like Probabilistic Latent Semantic Analysis (PLSA) [3] and Latent Dirichlet Allocation (LDA) [4] infer the topic by extracting word co-occurrence information, and terms with a high co-occurrence belong to some topic [5]. In a long text, the conventional topic model performance is good, but in a short text, they perform poorly due to a lack of word co-occurrence patterns [6, 7]. There are some topic models which utilize word embedding. However, word embedding is a method of embedding that focuses on the relationship across words with contexts. Specific words are no longer treated as distinct symbols but instead reflect similarities and correlations between words. In particular, word embedding represents a continuous vector of words for low-dimensional Euclidean space [8, 9, 10]. The Latent feature topic model (LFTM) does not explain the difference between topics and words [11].

A neural topic model is presented in [12], where topics are not associated with vectors. Some other topic models use word embedding that prohibits topics representing global word correlation. Short texts are compared with long papers using this random technique that ensures a power-law distribution. As a result, only words with a significant geometric similarity between their embedded vectors are grouped, leading to poor topic quality [9, 12, 13]. Also, instead of assigning an embedding to each topic in the document, some approaches build document or topic embedding at the corpus level [14, 15, 16]. The topic model and embedding technique only reflect part of the semantics of the text content because they focus on two different types of text patterns in the text. The combination of topics and word-

*Corresponding Authors: Junaid Rashid and Jungeun Kim

Table 1: Notations

Variable	Description
D	Number of documents
w	Number of words
$\|g\|$	Different words
y	Word embedding
$\|V\|$	Vocabulary size
G	List of related terms
Φ_Z	Word distribution of topics
θ	Global topic distribution
$\|G\|$	Numbers of multiterm
$m_{w\|z}$	Assignment of words to topics

ing has recently become a realistic way of creating a complete text representation. However, the previous methods based on word embedding forbid topics that reflect global word correlation. Only words with a strong geometric resemblance between their embedded vectors are grouped into the same topics, resulting in poor quality. Therefore, we present a novel topic model for short text documents that uses word embedding to solve the sparsity problem and find the structural information of topics and words to produce high-quality topics. In the proposed topic model, semantically or syntactically similar words are derived more accurately to extract suitable topics. The proposed topic model preserves the semantic meaning of each word in a short document without losing contextual information.

The main contribution of this research work is as follows:

- We proposed a novel word embedding-based topic model (WETM) using word2vec, which addresses the sparsity problem in short texts and discovers structural information about topics and words using a word embedding to find semantically related words and extract the most appropriate topics.
- We also proposed a modified collapsed Gibbs sampling algorithm to find the parameters of the proposed topic model.
- We performed qualitative and quantitative analysis. Experiments on two real-world datasets showed that the proposed topic model achieves better classification, topic coherence, topic quality, and clustering results than other topic models. The proposed topic model execution time is also lower than baseline topic models.

2. Related Work

The probabilistic topic modeling methods LDA and PLSA extract meaningful information from text collections by the co-occurrence of document-level words. It is difficult for conventional topic models to identify coherent topics in collections of short texts because semantically related terms rarely occur in short text documents. Some heuristic strategies, such as [6, 17, 18], have already been presented. There are several approaches to topic models that compensate for short text features and increase the pattern of occurrence of words without additional information. The BTM [19] is possibly a short text topic model developed specifically for short text data. It means such words inside a biterm are generated with the same topic, which provides explicit information about the co-occurrence of the words. The DMM technique [20] implies that each document has a single topic suitable for large but short documents. Clustering and LDA are combined in the self-aggregation topic model (SATM)[21]. The experiments show that the suggested SATM model can extract topics and enhance accuracy in short texts. The Pseudo-document topic model (PTM) [22] used pseudo documents to allow for the implied aggregation for short texts. The Sparse-PTM SPTM [23] implies that every short text is simply an excerpt from a lengthy pseudo document. The long pseudo documents are associated with topics and solve the sparsity problem. In [24], the SeaNMF model is presented, which uses semantic word-text relationships during the training phase to increase semantic coherence in topics. There are several other topic modeling approaches that are used for word embeddings, one of the most common methods of representing word vectors, has been widely used in many NLP tasks [25, 26, 27], including relationship extraction [28, 29, 30], question answering [31, 32, 33], and topic modeling [34]. Word embedding is a method that focuses on the relationship between the word level and its context. Each word is no longer treated as a separate symbol in word embeddings but instead reflects the similarities and correlations between words. In particular, the word embedding method expresses the word as a continuous vector in low-dimensional Euclidian space. Word embeddings learned from massive corpora are preprogrammed with generalized semantic and structural data about the words considered prior information. A Gaussian mixture model for grouping the word embeddings are used [35]. Gaussian [15] replaces textural words with word embeddings for the LDA generative process, and every topic is a Gaussian multivariate distribution on word embeddings. The Gaussian LDA in the DMM model development process includes word embeddings and a base topic to extract topics from a short collection [36]. In [37] proposed a TSSE-DM topic model based on the topic subdivision for improvement of interpretability. The DMM-based topic model GPU-DMM [38] employs word embeddings to capture semantic relevance information between words, which is further used by the GPU model in the inference process to improve topic coherence. Word embeddings are a prerequisite for grouping short corpus before using the random-field Markov LDA [39].

3. Methodology

We present our topic model in this section. The problem is defined first, and then we briefly explain our proposed topic model.

3.1. Problem Description

In topic modeling, topics are prohibited from revealing the global correlation of words due to the sparsity problem in short texts. Words with strong geometric similarities between embedded vectors are classified in the same topics and are of low quality. We have a collection of D text documents $G = (d_1, d_2, \ldots\ldots, d_n)$ and each document contains a number of words $(w_1, w_2, w_3, \ldots..w_{|g|})$. The d_n, w_g represent the numbers of documents and words. Each word $w_{|g|}$ accepts a value from vocabulary V. A topic is a distribution of vocabulary words.

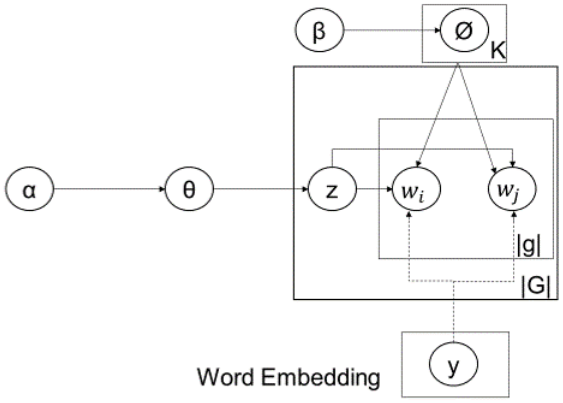

Figure 1: WETM graphical representation

3.2. Description of Proposed Topic Model

Table 1 shows the notations used in the paper. A graphic representation of the proposed topic model WETM is shown in Figure 1. The proposed topic model retains the semantic meaning of each word without losing contextual information.

- The $|g|$ represents different words in the documents. The terms list of text documents d is $G_d = (g1, g2, g3, \ldots\ldots, g_{G_d})$.
- The G_d is represented by several terms in documents d. The G is the global list of the related terms.
- The generative process after the extractions of the related term is $\Phi_z \sim Dir(\beta)$, $\Phi \sim Dir(\alpha)$, $w_i \sim terms(\Phi_Z)$, $w_j \sim terms(\Phi_Z)$ and $z \sim terms(\theta)$.
- The word embedding is utilized for the y. Term relationships with high semantic relevance may not occur regularly in short texts due to their small size.

WETM solves the sparsity problem and finds the structural information of topics and words in short texts to discover more appropriate and high-quality topics. Word2vec, a model based on neural networks, represents words in a corpus as a vector with contextual understanding [40]. The words in the short text documents are converted into vectors. The smaller the distance between two vectors in vector space, the greater the similarity between the two words. Learned word embeddings capture general patterns of word co-occurrence, as they are learned to retain the general contextual information of words. In latent space, semantic or syntactically similar words are inferred more closely. Word2Vec, like word co-occurrence, uses contextual information about words, and word2vec concentrates on the likelihood that specific terms will come around a word after it has been in the text. The significant advantage of the word2vector is that it preserves the semantic significance of distinct words for short text document with no loss of contextual information. The embedding vector is generated, and its size is very small. Every dimension of an embedding vector represents one aspect of a word. The $w_i = w_j = p(w|z, \Phi, y)$ is assigned. The hyperparameters of Dirichlet are α and β. The overall topic distribution of the entire corpora is θ, z is a related term of topic assignment, and Φ_Z means the word distribution of topic z. Each word w_i is generated using the Multinomial distribution. Equation 1 and 2 shows the probability of related term g and whole terms G.

$$p(g) = \sum_z p(z) \prod_{w_i \in m} p(w_i|z) \tag{1}$$

$$p(G) = \prod_g p(z) \prod_{w_i \in m} p(w_i|z) \tag{2}$$

We assume that a document topic distribution equals the topic distribution anticipation for each term generated from the document $p(z|d)$. After that, we find the value of $p(z|g)$ through the Bayes formula in equation 3.

$$p(z|d) = \frac{p(z) \coprod_{i,j \epsilon g} p(w_{i,j}|z)}{\sum_z p(z) \coprod_{i,j \epsilon g} p(w_{i,j}|z)} \tag{3}$$

The $p(g|d)$ for estimation [19] is found using equation 4. The $m_{G_d}(g)$ shows the terms frequency sequence in G_d.

$$p(g|d) = \frac{m_{G_d}(g)}{\sum_g m_{G_d}(g)} \tag{4}$$

As a result, topic distribution $p(z|g)$ is calculated in equation 5.

$$p(z|d) = \sum_g p(z|g)\, p(g|d) \tag{5}$$

The values of $p(g|d)\, and p(z|d)$ are used to find the new equation is 6. The topic distribution for every term is calculated.

$$p(z|d) = \sum_g \frac{p(z) \coprod_{i,j \epsilon g} p(w_{i,j}|z)\, m_{G_d}(g)}{\sum_z p(z) \coprod_{i,j \epsilon g} p(w_{i,j}|z) \sum_g m_{G_d}(g)} \tag{6}$$

The parameters are inferred with the most widely used Gibbs sampling method. Equation 7 shows the conditional distribution. Where, z_{-g} represents the topic assignments for all terms except the e. The m_z shows the occurrence of terms e, which are given to z. The $m_{w|z}$ shows the assignment of word w_i to the topic z. The $|V|$ is vocabulary size.

$$p\left(z|z_{-g},\ G, \alpha, \beta\right) \infty (m_z + \alpha) \prod_{w_{i \epsilon j}} \frac{m_{w_i}|z + \beta}{\sum_{w_j} m_{w_j}|z + |V|\beta} \tag{7}$$

3.3. Collapsed Gibbs Sampling Algorithm

Algorithm 1 represents a simplified inference process for the proposed topic model. In this modified Collapsed Gibbs Sampling Algorithm, the inputs are topics, terms, and hyperparameters α and β. Each term is assigned a topic. The modified Collapsed Gibbs Sampling Algorithm computes the θ and Ø.

4. Experiments and Results

4.1. Datasets

We used two real-world datasets, Web Snippet [41] and Amazon Review [42], for experiments. There are 12340 search snippets in this dataset, and each is considered a small document. Snippets are divided into eight groups, and Amazon reviews contain product reviews and metadata from May 1994 to June 2014. As a training dataset, we randomly selected 20000 brief reviews, each of which falls into one of seven categories.

Algorithm 1 Proposed Topic Model Modified Collapsed Gibbs Sampling Algorithm

The process will compute the θ and $\emptyset$

Input:

Number of topics K

Related terms G

Hyperparameters α, β

Process:

For every term, assign topic assignments

for $m_{iter} = 1$ to M_{iter} **do**

  for each g in the G **do**

    Draw z_g from the $p\left(z|z_{-g}, G, \alpha, \beta\right)$

    Update $m_z,\ w_i|z$

  end for

end for

Compute the θ and $\emptyset$ with the following equations:

$\emptyset_{w_i|z} = \frac{m_{w_i}|z+\beta}{\sum_{w_j} m_{w_j}|z+|V|\beta}$

$\theta_z = \frac{m_z+\alpha}{|G|+K\alpha}$

4.2. Preprocessing

Preprocessing steps like tokenization, stop word removal, normalization and lemmatization are applied to text documents, and the term list is extracted. Tokenization is a technique for separating the text into tokens. Blank spaces, periods, commas, semicolons, and quotation marks are used to separate words. A punctuation mark or stop word is the most familiar unfavorable term. As a result, they are removed from words because they don't make sense in context. The process of converting input text into canonical form is known as text normalization. It is required for noisy text, such as social media comments and popular text messages with abbreviations and misspellings. It also focuses on removing inconsistencies in language variations, necessary for the preprocessing phase, and an imperative component for natural language processing applications. It is a procedure for determining a word-based form.

4.3. Baseline Topic Models

The WETM is compared to the state-of-the-art topic models BTM, PTM, SPTM, PYSTM, DMM, GPU-DMM, and GLTM. BTM [19] is a topic model for the short text that has been proposed recently. BTM simulates the creation of biterms, which are made up of two words that are not in any specific order. Every short document is based on a lengthy pseudo document, according to PTM [22], and lengthy pseudo documents create topics. SPTM [23] obtains a concentrated topic for each long pseudo-document, replacing the PTM symmetric Dirichlet prior with a point prior and a slab preface. DMM [20] assumes that each short document has only a single topic, which is unrealistic for most texts but appropriate for short documents. GPU-DMM [38] is a short text topic model used to obtain additional information using word2vec. Global word embeddings are derived from a huge external corpus in GLTM [2]. Pitman-yor Process Self-aggregated Topic Model (PYSTM) [43] uses the pitman-yor process to generate short texts. For these BTM, PTM, SPTM, PYSTM, DMM, GPU-DMM, GLTM, and WETM, we set common parameters such as $\alpha = 50/K, \beta = 0.01$, and maximum iteration =1500.

Table 2: Classification accuracy in percent for the web snippet dataset

	Method	K=20	K=40	K=60	K=80
Web Snippet Dataset	BTM	0.86	0.83	0.77	0.76
	PTM	0.67	0.67	0.67	0.65
	SPTM	0.65	0.61	0.53	0.53
	PYSTM	0.69	0.68	0.67	0.66
	DMM	0.85	0.84	0.83	0.83
	GPU-DMM	0.87	0.85	0.84	0.82
	GLTM	0.86	0.87	0.86	0.84
	WETM	**0.89**	**0.90**	**0.91**	**0.88**

Table 3: Classification accuracy in percent for the amazon review dataset

	Method	K=20	K=40	K=60	K=80
Amazon Review Dataset	BTM	0.81	0.81	0.81	0.79
	PTM	0.79	0.79	0.81	0.81
	SPTM	0.80	0.75	0.71	0.72
	PYSTM	0.78	0.76	0.73	0.71
	DMM	0.80	0.79	0.78	0.77
	GPU-DMM	0.81	0.81	0.81	0.81
	GLTM	0.81	0.83	0.82	0.83
	WETM	**0.84**	**0.86**	**0.85**	**0.86**

4.4. Quantitative Analysis

4.4.1. Classification

We conduct classification experiments with topic distributions on $p(z|d)$. A text classification task is performed externally as part of the evaluation of topic models. It is considered that the topics in each short document are features, and the probabilities derived from the topic-document distribution are the features' values. We use Random Forest as the classification algorithm for both web snippet and Amazon review datasets and compute classification accuracy using five-fold cross-validation. In each short text, we create 20, 40, 60, and 80 topics, with the distribution of topics per document as the value of the features. Tables 2 and 3 show the accuracy of the WETM for web snippet and Amazon review datasets. The WETM achieved an accuracy of 0.89 percent with 20 topics, 0.90 percent with 40 topics, 0.91 percent with 60 topics, and 0.88 percent with 80 topics on the web snippet dataset. The performance of PTM and SPTM on the online snippet dataset is poor when compared with BTM. Perhaps it is due to the indirect manner in which PTM and SPTM topic-document distributions are derived. On the website snippet dataset with different topics, GLTM, GUP-DMM, DMM, and PYSTM topic models perform worse than the proposed model. The accuracy on the Amazon review dataset is 0.84 percent with 20 topics, 0.86 percent with 40 topics, 0.85 percent with 60 topics, and 0.86 percent with 80 topics. With different topics in the Amazon dataset, the proposed model outperformed BTM, PTM, SPTM, PYSTM, DMM, GPU-DMM, and GLTM topic models. These

Table 4: Topic Coherence of web snippet dataset

Method	K=20	K=40	K=60	K=80
BTM	-900.72	-879.23	-859.49	-833.57
PTM	-976.44	-967.30	-918.62	-907.55
SPTM	-1026.03	-1035.15	-1088.98	-1096.38
PYSTM	-1012.01	-1021.10	-1074.88	-1080.26
DMM	-946.27	-909.18	-880.34	-870.76
GPU-DMM	-914.61	-883.21	-859.59	-851.32
GLTM	-886.48	-872.20	-842.20	-830.38
WETM	**-882.13**	**-867.84**	**-837.85**	**-826.03**

Table 5: Topic Coherence of amazon review dataset

Method	K=20	K=40	K=60	K=80
BTM	-851.76	-848.21	-878.32	-888.90
PTM	-941.02	-956.42	-939.71	-917.86
SPTM	-923.95	-991.35	-987.40	-907.86
PYSTM	-913.84	-987.24	-977.28	-905.67
DMM	-855.51	-913.34	-912.74	-916.86
GPU-DMM	-821.45	-865.03	-896.44	-907.82
GLTM	-848.40	-843.84	-866.47	-859.87
WETM	**-817.10**	**-839.49**	**-862.12**	**-855.52**

classification results indicate that our method WETM performs best across all datasets that contain a wide range of topic counts.

4.4.2. Topic Coherence

We utilize the topic coherence [23] to quantify the semantic coherence of topics extracted by topic models. Equation 8 is used to calculate the topic coherence of a topic k.

$$C\left(k, v^k\right) = \sum_{x=2}^{X} \sum_{l=1}^{x-1} log \frac{|D(v_x^k, v_l^k|}{|Dv_l^k|} \tag{8}$$

The v^k shows the top X words for topics k. The $|D(v_x^k, v_l^k|$ total number of documents for a collection that contains both the words v_x^k and v_l^k.The average topic coherence for the topics extracted by the topic model is used to evaluate the performance of a topic model. When the topic coherence value is high, the topic model performs better. Tables 4 and 5 show the topic coherence scores for web snippet and amazon review datasets. In the web snippet dataset with the number of topics 20, 40, 60, and 80, the WETM topic coherence scores are -8802.13, -867.84, -837.85, and -826.03. The score for topic coherence on the Amazon reviews dataset is -817.10, -838.49, -862.12, and -855.52 for topics 20, 40, 60, and 80, respectively. GPU-DMM, PTM, BTM, and GLTM produce inferior outcomes compared to other models. Generally, the topic coherence results of WETM are better than BTM, PTM, SPTM, PYSTM, DMM, GPU-DMM, and GLTM. Therefore, WETM outperforms other baseline topic models in terms of topic coherence.

4.4.3. Clustering

We also evaluated the clustering performance of WETM. We consider that every topic is a label, and every document is assigned to topic k. The maximum value of $p(z|d)$ value constructs various clusters. The clustering results are evaluated with Purity and Entropy.

4.4.4. Purity

Purity ranges between 0 and 1, with higher purity representing better clustering results. In equation 9, purity is represented by a formula.

$$Purity = \frac{1}{D} \sum_{k=1}^{K} \max(k) \tag{9}$$

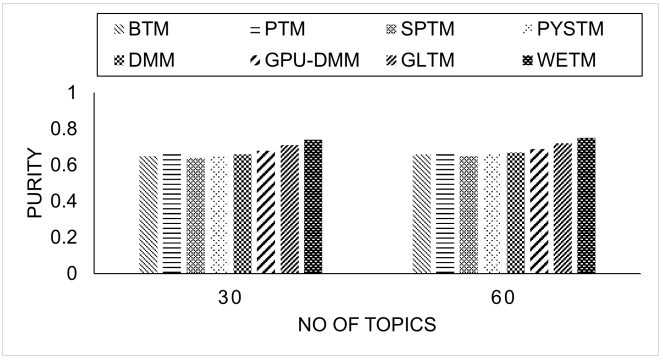

Figure 2: Clustering results using purity on web snippet dataset

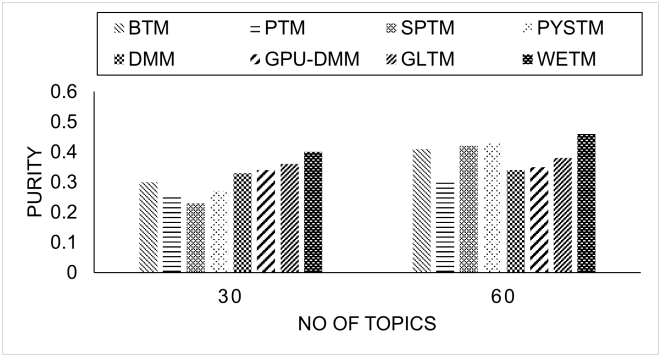

Figure 3: Clustering results using purity on amazon reviews dataset

Where, $max(k)$ signifies the dominating category document number in cluster k. Figures 2 and 3 demonstrate the clustering outcomes measured by purity for K = 30 and 60 topics on web snippet and amazon reviews datasets. The purity results indicate that WETM purity results are greater than other state-of-the-art topic models on web snippet and amazon reviews datasets. It means that WETM clustering performance is better against other baseline topic models.

4.4.5. Entropy

Equation 10 finds the entropy. In entropy, the homogenous of the cluster is measured. The lowest score of entropy means clustering performance is better.

$$Entropy = -\sum_{k=1}^{K} \frac{C_k}{D} \sum_{l=1}^{L} \frac{C_{kl}}{C_k} \log(\frac{C_{kl}}{C_k}) \tag{10}$$

The category numbers are L. The number of categorized documents by category l in cluster k and the total number of doc-

Table 6: Qualitative analysis for topic “health” for web snippet dataset

Topic Model	Top 5 words
BTM	gov, news, information, research, nutrition
PTM	nutrition, hiv, medical, gov, health
SPTM	information, party, diseases, gov, food
PYSTM	food, news, hiv, research, party
DMM	care, gov, diet, party, home
GPU-DMM	care, party, research, medical, gov
GLTM	nutrition, drug, information, research, gov
WETM	cancer, disease, treatment, medical, health

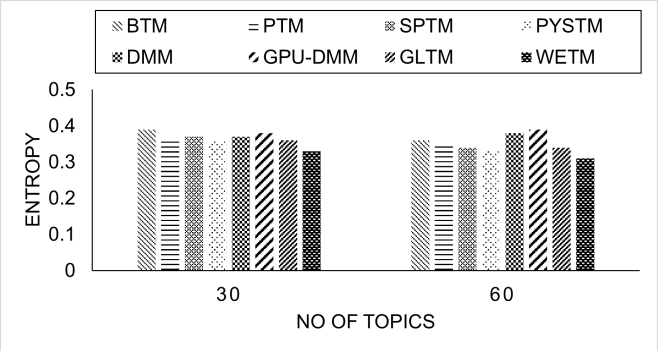

Figure 4: Clustering results using entropy on web snippet dataset

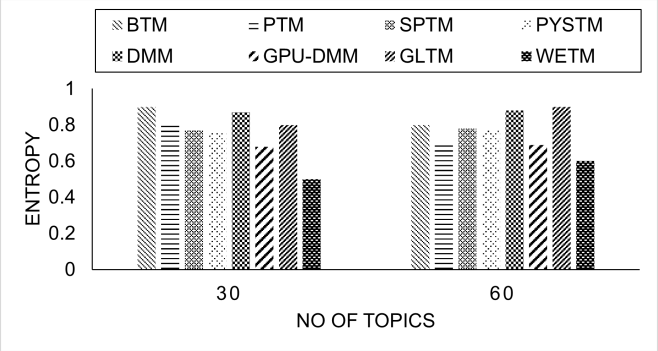

Figure 5: Clustering results using entropy on amazon reviews dataset

uments in cluster k is represented by C_{kl} and C_k. Figures 4 and 5 show the clustering outcomes based on entropy for web snippets and Amazon reviews for K = 30 and 60 topics, respectively. GPU-DMM entropy for the web snippet dataset is higher than GLTM and DMM. WETM entropy is lower than BTM, PTM, SPTM, PYSTM, DMM, GPU-DMM, and GLTM for web snippet datasets. The GPU-DMM entropy for the Amazon dataset is higher than that of GLTM and DMM. On the web snippet dataset, WETM entropy is lower than BTM, PTM, SPTM, PYSTM, DMM, GPU-DMM, and GLTM. We find from the results that WETM has lower entropy than other state-of-the-art topic models for datasets comprised of web snippets and Amazon reviews. It indicates that WETM clustering performance is better than baseline topic models.

4.4.6. Execution time

In this section, we compare the time efficiency of several topic models with 20, 40, 60, and 80 topics. For all topic models, we utilize Gibbs sampling for parameter estimation. Figures 6 and 7 show the time cost for one iteration for web snippets and Amazon reviews datasets, respectively. The BTM has demonstrated robustness for some datasets without external data, but the time complexity of the model prevents its wide application. The graphical structure of the PTM is complex and there are multiple latent variables for each sample, which makes processing time-consuming. The execution time of PTM for both datasets is high, and BTM takes less time than PTM. SPTM takes less time than BTM, PTM and PYSTM and GLTM also takes less time than BTM, PTM and SPTM. Overall, the proposed topic model takes less time than other baseline topic models for web snippet and Amazon reviews datasets.

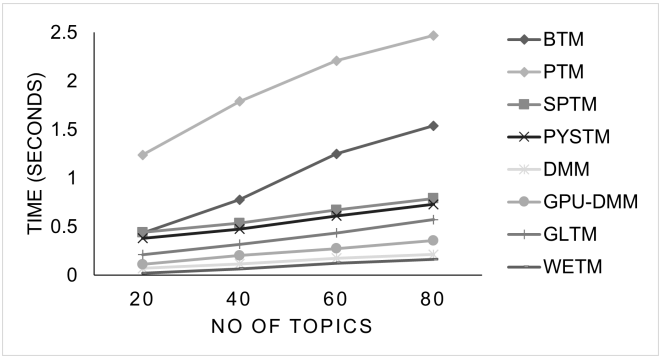

Figure 6: Execution time on web snippet dataset

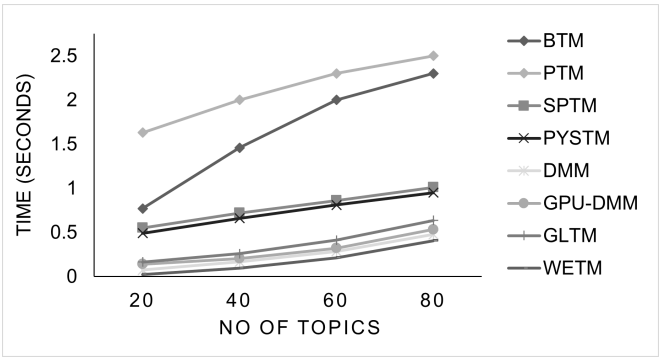

Figure 7: Execution time on amazon reviews dataset

4.5. Qualitative Analysis

Table 6 shows the top 5 words of the extracted topic ”health” for all topic models for the web snippet dataset. WETM discovered more related words such as cancer, disease, treatment, medical, health. However, in GPU-DMM, DMM and GLTM, other terms such as ”government”, ”party” and ”home” are irrelevant to the topic of health. Some words in BTM, SPTM, PYSTM and PTM are also irrelevant to the topic of health, such as ”news” and ”party”. The result shows that the topics discovered by our topic model are more appropriate and suitable than other baseline topic models.

5. Conclusion

Topic modeling for short texts is a useful task due to the prominence of short texts on the Internet. The sparsity in short texts is a major challenge for traditional topic models. In this

paper, we presented a novel topic model WETM based on word embedding. Semantically related words are discovered for the extraction of suitable topics. A modified collapsed Gibbs sampling inference algorithm is also proposed for short texts. The proposed topic model solves the sparsity problem in short text documents and finds topics and words structural information. We performed both qualitative and quantitative analysis for experiments. The result of the proposed topic model experiments is effective for topic quality, classification, topic coherence, clustering, and execution time. Compared to baseline topic models, the proposed topic model discovers more appropriate topics. The classification results of the proposed topic model are better than the basic topic models. The proposed topic model performed better in topic coherence and generated more diverse topics for real-world datasets. The performance of the proposed topic model is better in clustering than other baseline topic models. The proposed topic model takes less time to run than conventional topic models.

Acknowledgements: This research was partly supported by the Technology Development Program of MSS [No. S3033853] and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A4A1031509).

References

[1] L. Li, Y. Sun, C. Wang, Semantic augmented topic model over short text, in: 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), IEEE, 2018, pp. 652–656.

[2] W. Liang, R. Feng, X. Liu, Y. Li, X. Zhang, Gltm: A global and local word embedding-based topic model for short texts, IEEE access 6 (2018) 43612–43621.

[3] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pp. 50–57.

[4] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of machine Learning research 3 (Jan) (2003) 993–1022.

[5] M. Divya, K. Thendral, S. Chitrakala, A survey on topic modeling, International Journal of Recent Advances in Engineering & Technology (IJRAET) 1 (2013) 57–61.

[6] L. Hong, B. D. Davison, Empirical study of topic modeling in twitter, in: Proceedings of the first workshop on social media analytics, 2010, pp. 80–88.

[7] A. F. Ibrahim, M. Hassaballah, A. A. Ali, I. A. Ibrahim, A study of sentiment analysis approaches in short text, in: Digital Transformation Technology, Springer, 2022, pp. 143–151.

[8] S. Li, J. Zhu, C. Miao, A generative word embedding model and its low rank positive semidefinite solution, arXiv preprint arXiv:1508.03826 (2015).

[9] O. Levy, Y. Goldberg, I. Dagan, Improving distributional similarity with lessons learned from word embeddings, Transactions of the association for computational linguistics 3 (2015) 211–225.

[10] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[11] D. Q. Nguyen, R. Billingsley, L. Du, M. Johnson, Improving topic models with latent feature word representations, Transactions of the Association for Computational Linguistics 3 (2015) 299–313.

[12] Z. Cao, S. Li, Y. Liu, W. Li, H. Ji, A novel neural topic model and its supervised extension, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29, 2015.

[13] S. Li, T.-S. Chua, J. Zhu, C. Miao, Generative topic embedding: a continuous representation of documents (extended version with proofs), arXiv preprint arXiv:1606.02979 (2016).

[14] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Twenty-ninth AAAI conference on artificial intelligence, 2015.

[15] J. Law, H. H. Zhuo, J. He, E. Rong, Ltsg: Latent topical skip-gram for mutually improving topic model and vector representations, in: Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Springer, 2018, pp. 375–387.

[16] R. Das, M. Zaheer, C. Dyer, 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), 2015, pp. 795–804.

[17] R. Mehrotra, S. Sanner, W. Buntine, L. Xie, Improving lda topic models for microblogs via tweet pooling and automatic labeling, in: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, 2013, pp. 889–892.

[18] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, X. Li, Comparing twitter and traditional media using topic models, in: European conference on information retrieval, Springer, 2011, pp. 338–349.

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

[20] J. Yin, J. Wang, A dirichlet multinomial mixture model-based approach for short text clustering, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 233–242.

[21] X. Quan, C. Kit, Y. Ge, S. J. Pan, Short and sparse text topic modeling via self-aggregation, in: Twenty-fourth international joint conference on artificial intelligence, 2015.

[22] Y. Zuo, C. Li, H. Lin, J. Wu, Topic modeling of short texts: A pseudo-document view with word embedding enhancement, IEEE Transactions on Knowledge and Data Engineering (2021).

[23] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, H. Xiong, Topic modeling of short texts: A pseudo-document view, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 2105–2114.

[24] T. Shi, K. Kang, J. Choo, C. K. Reddy, Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1105–1114.

[25] M. Habib, M. Faris, A. Alomari, H. Faris, Altibbivec: A word embedding model for medical and health applications in the arabic language, IEEE Access 9 (2021) 133875–133888.

[26] A. Fesseha, S. Xiong, E. D. Emiru, M. Diallo, A. Dahou, Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya, Information 12 (2) (2021) 52.

[27] A. Roy, S. Pan, Incorporating extra knowledge to enhance word embedding, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4929–4935.

[28] D. Sorokin, I. Gurevych, Context-aware representations for knowledge base relation extraction, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1784–1789.

[29] M. K. Najafabadi, M. B. Nair, A. Mohamed, Tag recommendation model using feature learning via word embedding, in: 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI), IEEE, 2021, pp. 000305–000310.

[30] M. Sänger, U. Leser, Large-scale entity representation learning for biomedical relationship extraction, Bioinformatics 37 (2) (2021) 236–242.

[31] J. Wen, H. Tu, X. Cheng, R. Xie, W. Yin, Joint modeling of users, questions and answers for answer selection in cqa, Expert Systems with Applications 118 (2019) 563–572.

[32] S. Gao, X. Chen, Z. Ren, D. Zhao, R. Yan, Meaningful answer generation of e-commerce question-answering, ACM Transactions on Information Systems (TOIS) 39 (2) (2021) 1–26.

[33] A. Ali, I. Schwartz, T. Hazan, L. Wolf, Video and text matching with conditioned embeddings, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 1565–1574.

[34] M. Peng, Q. Xie, Y. Zhang, H. Wang, X. J. Zhang, J. Huang, G. Tian, Neural sparse topical coding, in: Proceedings of the 56th Annual Meet-

ing of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 2332–2340.
[35] V. K. R. Sridhar, Unsupervised topic modeling for short texts using distributed representations of words, in: Proceedings of the 1st workshop on vector space modeling for natural language processing, 2015, pp. 192–200.
[36] G. Xun, V. Gopalakrishnan, F. Ma, Y. Li, J. Gao, A. Zhang, Topic discovery for short texts using word embeddings, in: 2016 IEEE 16th international conference on data mining (ICDM), IEEE, 2016, pp. 1299–1304.
[37] C. Mai, X. Qiu, K. Luo, M. Chen, B. Zhao, Y. Huang, Tsse-dmm: Topic modeling for short texts based on topic subdivision and semantic enhancement, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2021, pp. 640–651.
[38] C. Li, Y. Duan, H. Wang, Z. Zhang, A. Sun, Z. Ma, Enhancing topic modeling for short texts with auxiliary word embeddings, ACM Transactions on Information Systems (TOIS) 36 (2) (2017) 1–30.
[39] P. Xie, D. Yang, E. Xing, Incorporating word correlation knowledge into topic modeling, in: Proceedings of the 2015 conference of the north American chapter of the association for computational linguistics: human language technologies, 2015, pp. 725–734.
[40] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).
[41] J. Rashid, S. M. A. Shah, A. Irtaza, Fuzzy topic modeling approach for text mining over short text, Information Processing & Management 56 (6) (2019) 102060.
[42] J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Proceedings of the 7th ACM conference on Recommender systems, 2013, pp. 165–172.
[43] Y. Niu, H. Zhang, J. Li, A pitman-yor process self-aggregated topic model for short texts of social media, IEEE Access 9 (2021) 129011–129021.