Research Paper

JIS

Journal of Information Science
1–17

DOI: 10.1177/0165551520977453
journals.sagepub.com/home/jis
SAGE

Topic attention encoder: A self-supervised approach for short text clustering

Jian Jin and Haiyuan Zhao
Department of Information Management, School of Government, Beijing Normal University, China

Ping Ji
Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China

Abstract

Short text clustering is a challenging and important task in many practical applications. However, many Bag-of-Word–based methods for short text clustering are often limited by the sparsity of text representation, while many sentence embedding–based methods fail to capture the document structure dependencies within a text corpus. In considerations of the shortcomings of many existing studies, a topic attention encoder (TAE) is proposed in this study. Given topics derived from corpus by the techniques of topic modelling, the cross-document information is introduced. This encoder assumes the document-topic vector to be the learning target and the concatenating vectors of the word embedding and corresponding topic-word vector to be the input. Also, a self-attention mechanism is employed in the encoder, which aims to extract weights of hidden states adaptively and encode the semantics of each short text document. With captured global dependencies and local semantics, TAE integrates the superiority of Bag-of-Word methods and sentence embedding methods. Finally, categories of benchmarking experiments were conducted by analysing three public data sets. It demonstrates that the proposed TAE outperforms many document representation benchmark methods for short text clustering.

Keywords

Global dependency; local semantic; self-attention; short text clustering; text representation; topic attention encoder

1. Introduction

Short text clustering is of great importance due to its various applications, such as user profiling [1] and recommendation [2], for nowadays social media data sets emerge day by day. Some classical clustering algorithms such as *k*-means, density-based spatial clustering of applications with noise (DBSCAN), Gaussian mixture models (GMMs) and their variants are often applied in these applications to get the cluster result. However, the premise of a good clustering algorithm is appropriate text representation of documents. Also, learning good representations of texts plays an important role in many natural language processing (NLP) tasks, such as document classification, ranking and sentimental analysis. Different representations may capture and disentangle different degrees of explanatory hidden ingredients in the corpus. It attracts interests of many researchers, and various types of models have been proposed for text representation. In this study, the text representation learning for short text clustering is focused. In comparison to the text representation learning for other NLP tasks, such as sentiment analysis, named-entity recognition and machine translation [3], the representation of short text clustering expects that learned document vectors from different categories should be well-separated.

Commonly used techniques about text representation can be divided into two main categories, that is, Bag-of-Words (BoW)-based approaches [4] and sentence embedding–based approaches. Within BoW-based approaches, a document is fundamentally represented by the count of word occurrences within a document. For decades, this approach has been

Corresponding author:
Jian Jin, Beijing Normal University, B503, Main Building, No. 19, XinJieKouWai St., HaiDian District, Beijing 100875, China.
Email: jinjian.jay@bnu.edu.cn

shown to be effective in solving various text mining tasks [5–7]. The main advantage is that BoW-based approaches produce an interpretable and effective vector representation of a document and each feature in this vector denotes the word occurrence in a document. However, when the size of vocabulary is large, the generated document vector can be high-dimensional extremely sparse. As the dimension and the sparsity of the document vectors increase, conventional distance metrics become ineffective in representing the proper proximity between documents, which often reduce the overall clustering performance. Meanwhile, clustering high-dimensional data also consume a lot of resources and computation time. In considerations of these limitations, some research efforts concentrated on how to improve traditional methods for normal text clustering or designing new models to analyse short texts. Some of these famous methods include probabilistic topic models, such as latent semantic indexing (LSI) [8], probabilistic latent semantic analysis (pLSA) [9], latent Dirichlet allocation (LDA) [10], biterm topic model (BTM) [11] and collapsed Gibbs sampling algorithm for the Dirichlet multinomial mixture model (GSDMM) [12], and heuristic optimisation methods, such as nonnegative matrix factorisation methods [13,14]. The main idea of these methods is to analyse relations between pairs of terms in order to compensate for the sparsity of short texts and reduce the dimension of document vectors. However, these BoW-based models ignore the word order and syntactic features which make them potentially fail to capture semantic information in the scenario of short text clustering.

Recently, neural sentence embedding–based methods provide new solutions for text representation and, compared with BoW-based methods, they achieve remarkable improvements in many applications. Due to the redundancy of natural language, models extract features from variable-length texts, that is, phrases, sentences and documents. In order to capture these features, neural network models are widely used as the feature extractor, such as recurrent neural networks (RNNs) [15,16] and convolutional neural network (CNN) [17]. Based on neural network structures, previous studies leveraged variable training approaches to generate the distributed representation of a sentence/document within unsupervised and semi-supervised scenarios. One mainstream training approach is AutoEncoder [18] which aims to learn the hidden representation of input vectors in the latent space and try to reconstruct the original input. This training approach helps to capture the latent semantics of sentence through a non-linear transform and shows its effectiveness in many previous studies [19]. Another mainstream training approach is Skip-Thought [20,21], which is to learn the representation of a sentence by predicting the next sentence given the current one. In the scenario of short text clustering, many sentence embedding–based methods utilise the pre-trained word embedding to generate the representation of a document. In this scheme, however, sentence representations trained by those abovementioned training methods only encode semantic information within itself or its local context. Therefore, these representations are unable to form high correlations within the same cluster category and may even form high similarities among different categories, thus misleading the cluster algorithms.

To further illustrate the inferiors of BoW-based approaches and sentence embedding–based approaches in short text clustering, two categories of examples which might cause mistakes are as follows:

- Sentence A: US stocks dip on weak Asian data.
- Sentence B: Wall Street edges lower after record.

For instance, Sentences A and B are sampled from the same category but have no common words. Therefore, neither traditional BoW-based methods nor improved BoW-based methods based on matrix decomposition cannot generate a reasonable representation of these two sentences. However, sentence embedding–based models may relieve such problems. Representations of Sentences A and B encoded by sentence embedding approaches are expected to have high correlations with each other since pre-trained word embedding are utilised as input and word similar local semantics are captured.

However, documents with similar local semantics might not be the same category. For instance, the following two sentences are sampled from different categories, but they are assumed to have similar semantics if they are encoded via sentence embedding:

- Sentence C: How do I sort my code in Visual Studio?
- Sentence D: How do I sort my data in Excel?

With the pre-trained word embedding, such as, Word2Vec or Glove, the words *code, Visual Studio* in Sentence C and the words *data, Excel* in Sentence D have high similarity. However, this similarity is counterproductive to the clustering task because it gives too much similarity between documents from different categories. Hence, besides the local semantic information of each document, the representation of short text also needs to capture the global dependencies between documents within a corpus. The global dependency uncovers the correlations of each document's vector

representation in the corpus vector space. However, these pre-trained word embedding-based approaches neglect the dependencies between documents within a corpus due to the fact that the training procedure in a neural network is to optimise the representation of a single document or its local context. In contrast to sentence embedding methods, some improved BoW-based methods, such as LDA, are efficient in capturing the global dependencies due to the assumption about the word occurrence in different documents under different distributions. More specifically, the co-occurrence of words provides a strong linkage between different documents, thus to introduce the cross-document information for encoding sentences. Short text document sharing more words can be naturally clustered into a category, but it does not always hold as Sentence C and Sentence D present.

In view of the visual research gap, in this study, neural sentence embedding methods are integrated with BoW-based models for better short text representations and, specifically, a self-supervised approach is proposed, referred to as topic attention encoder (TAE). As a BoW-based model is built on the basis of matrix factorisation, LDA may generate dimensionally reduced document representations according to the document-topic matrix and word representations considering the reference to the global dependency of a corpus. Given topics derived from a corpus by LDA, the cross-document information is then introduced. It assumes the document-topic vector to be the learning target and the concatenation of word embedding and corresponding topic-word vector to be the input. In this way, the sequential model encodes the global dependency of each document. Also, a self-attention mechanism is utilised, which aims to extract the weights of hidden states adaptively and contribute to encode the final semantics of each short document. In this scheme, TAE encodes the global dependencies and local semantics simultaneously so as to derive a better clustering result.

To summarise, this study presents a text representation approach to improve short text clustering. The contributions of this study are at least twofold:

- A novel self-supervised learning approach is provided in the text representation learning for short text clustering via a topic model, which captures the local semantics and global dependencies of a short text document. Meanwhile, the attention mechanism is utilised to improve the encoding of local semantics.
- Categories of experiments were conducted in three real-world data sets, which demonstrated the ability of proposed approach in short text clustering. In comparison with different approaches, the proposed approach obtains state-of-the-art performance with different evaluation metrics.

The rest of the article is organised as follows. Section 2 presents some related studies. In Section 3, the proposed approach is explained. Categories of experiments are presented in Section 4, and relevant discussions are given in Section 5. This study is concluded in Section 6.

2. Related work

2.1. BoW methods on short text clustering

Various BoW-based studies for short text clustering are reported, which can be briefly divided into three categories.

The first category is to expand and enrich the context of short text document via external textual information such as dictionary, encyclopaedia and knowledge graph. In Banerjee et al. [22], to improve the accuracy of short text clustering, additional features from Wikipedia are utilised to enrich the representation. Similarly, Gabrilovich and Markovitch [23] represent the meaning of texts in a high-dimensional space of concepts derived from Wikipedia. Fodeh et al. [24] incorporate the semantic knowledge from an ontology into text clustering. However, these studies need solid NLP knowledge and employ high-dimensional representation which may result in a waste of both memory and computation time. Also, it introduces new challenges regarding how to properly use those external resources [25]. Practically, solving this new problem itself is time-consuming and complicated.

The second category is to utilise the global dimensionality reduction method to reduce the BoW representation to a low-dimensional latent space. For example, some famous models include Latent Semantic Analysis (LSA) [8], pLSA [9] and LDA [10]. These approaches allow documents of the same category to have closer distances in a latent semantic space with dimensionality reduction. On this basis, some researchers explored sophisticated models to cluster short texts. For example, a Dirichlet multinomial mixture model-based approach was proposed for short text clustering [12]. Although they generally provide better representation, compared with the traditional BoWs, these matrix factorisation methods reduce the matrix in a linear space, which neglect non-linear semantic relations between words. Moreover, some studies focus the above two aspects. For example, with matrix factorisation, a novel framework that enriches text features by the techniques of machine translation and reduces original features simultaneously is proposed [26].

The third category is to seek word communities in the vocabulary and to get conception representation. In Kim et al. [27], bag-of-concepts is introduced, in which the concept vector of vocabulary is obtained by clustering word vectors and represent each document as combination of concepts. In Jia et al. [28], concept vectors are created by identifying semantic word communities from a weighted word co-occurrence network, which is extracted from a short text corpus or a subset thereof. The cluster membership of short texts is then estimated by mapping the original short texts to the learned semantic concept vectors.

Despite some methods improving the performance of short text clustering, most of them ignore the word order and syntactic features, which make them cannot encode semantic information between words into the representation of document.

2.2. Neural sentence embedding methods on text representation

To overcome the defect of BoW-based approach in the representation of short text document, some neural network structures were employed to learn the compact representation vectors of sentences. On the basis of word embedding such as Word2Vec [29], Doc2Vec was proposed to represent a document by treating a document ID as a word in the document [30]. Doc2Vec is still a shallow neural network model and needs a larger corpus to generate better performance. Some sequential models utilise the pre-trained word embedding to generate the sentence representation, such as long short-term memory (LSTM) [15] and gated recurrent unit (GRU) [16], which have presented their advantages in many NLP problems, such as machine translation [31], speech recognition [32] and text conversation [33]. Meanwhile, CNN has also been applied to NLP tasks and achieves good results. For instance, TextCNN was proposed and applied in a text classification problem [17], which uses a number of different size kernels to extract key information in a sentence to better capture local dependencies.

Besides various neural network structures, some studies attempted to find better sentence embedding in a self-supervised scenario. Denoising AutoEncoder [18] was proposed to eliminate noises that are added into the inputs intentionally and reconstruct the original data. Variational AutoEncoder (VAE) [34] was proposed to learn the latent distribution of original data, thus making the representations more robust. Skip-Thought [20,21] was reported on the hypothesis of distribution semantics, which is to learn the representation of a sentence by predicting the next sentence given the current one, thus making the learnt representation has a higher similarity. However, the decoder of Skip-Thought is a sequential structure. Accordingly, QuickThought [35] was proposed, which leverages a sequential decoder into a classifier to predict the context of current one. Meanwhile, [36] trains the sentence embedding with the help of supervised data in The Stanford Natural Language Inference Datasets and shows a good generalisation performance on a variety of natural language tasks.

As presented, pre-trained word embedding vectors are often utilised to represent a document in many embedding-based methods. However, a single document or its local context is reckoned only to encode the semantic information. It induces that the learned representations potentially lead to form low correlations within the same cluster and high similarities among different clusters, which put much difficulty on a clustering algorithm.

2.3. The integration of BoW methods and neural sentence embedding methods

To integrate the advantages of BoW-based methods and sentence embedding–based methods, recently, some studies are reported to enhance the neural network model with topic models or topical information. Cao et al. [37] analysed topic model in a neural network perspective and proposed a novel neural topic model where the representation of words and documents are combined into a uniform framework. Liu et al. [38] employed latent topic models to assign a topic for each word in the text corpus and learned topical word embedding vectors based on both words and their topics. In this way, document representations were built, which are claimed to be more expressive than some widely used document models such as latent topic models. Dieng et al. [39] proposed the TopicRNN model, which integrates the merits of RNN and latent topic models. It captures local dependencies using an RNN and global dependencies using latent topics. Li et al. [40] proposed a generative topic embedding model to combine the superiority of word embedding and topic model. Lau et al. [41] presented a neural language model that incorporates document context in the form of a topic model-like architecture, providing a representation of the broader document context outside of the current sentence. Peng et al. [42] proposed a novel sparsity enhanced topic model, which focuses on replacing the complex inference process with the back propagation to make the model easy to be extended.

Compared with these models, the proposed model in this article has two main merits. First, different from previous combined models, the word embedding vector is concatenated with topic-word vector as the input of TAE, which helps

Table 1. Notation description.[a]

Name	Description
D, d	A document D and its embedding d
$\boldsymbol{R_d}$	Document embedding matrix $\{d_1, d_2, \ldots, d_V\}$
$\boldsymbol{R_w}$, w_i	Word embedding matrix $\{w_1, w_2, \ldots, w_M\}$
V	The size of vocabulary
M	The number of document in a corpus
K	Topic number
X	The input sequence of GRU $\{x_1, x_2, \ldots, x_T\}$
h_t	The tth hidden state of GRU
$\boldsymbol{H}$	The output matrix of GRU $\{h_1, h_2, \ldots, h_t\}$
R	Attention aspect number
M	The final document embedding

GRU: gated recurrent unit.a

[a]Only lists the special notations used in this article and notations of LDA, and GRU parameters are not presented.

to capture local semantics and global dependencies respectively. Second, in order to learn a document representation which has global semantics, a novel self-supervised training approach is designed in TAE.

3. TAE for short text clustering

3.1. Problem statement

Given a short text with n words in a corpus, that is, $D = (w^1, w^2, ..., w^n) \in \mathbb{R}^{n \times T}$, where w^i is the ith word in a short text and T is the embedding size of a word. The encoder aims to learn a mapping from D to a vector d which is the representation of this document. Notice that, in many scenarios of short text, a document D is primarily made up by one single sentence

$$d = F(D) \tag{1}$$

$F(\cdot)$ is an unknown mapping function to be learned. Then, accordingly, a cluster algorithm is utilised on the corpus, which helps to assign a cluster label for each document. For clarity, some notations are listed in Table 1.

3.2. Model

In this study, a self-supervised approach is proposed, referred to as TAE, which integrates the superiority of BoW-based methods and neural sentence embedding–based methods. The framework of TAE is shown in Figure 1.

As explained, a BoW-based model is built on the basis of matrix factorisation, and LDA is able to generate dimensionally reduced document representations and word representations after references to the global dependency of a corpus. To introduce the cross-document information into the encoding, as presented in Figure 1, two improvements are conducted. First, the word embedding vector is concatenated with topic-word vector as the input of model. Second, the corresponding document-topic vector is assumed to as the learning target. In this way, the sequential model can encode the global dependency of each document. Then, a self-attention mechanism is applied to extract weights of hidden states adaptively and help to encode the final semantics of each short text document. In this scheme, TAE encodes the global dependencies and local semantics simultaneously, which aims to derive a better result for short text clustering.

3.2.1. LDA. The LDA is a type of Bayesian-based topic model which is used to detect latent topics within a text corpus. The structure of LDA is illustrated in Figure 2.

LDA assumes that the prior of document-topic α and topic-word β are Dirichlet distributed. When a document is generated, a topic distribution θ is selected from α. Then, a topic z is selected from topic distribution θ on the basis of prior distribution β. With N times repeated, a document with N words is generated. The joint probability of generating a document by LDA is

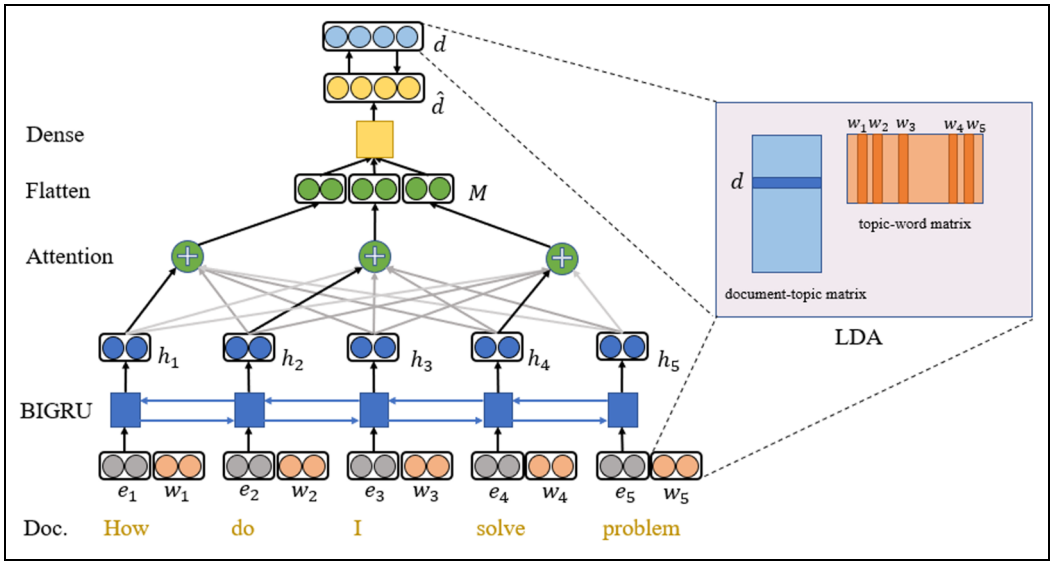

Figure 1. Graphical illustration of the topic attention encoder.

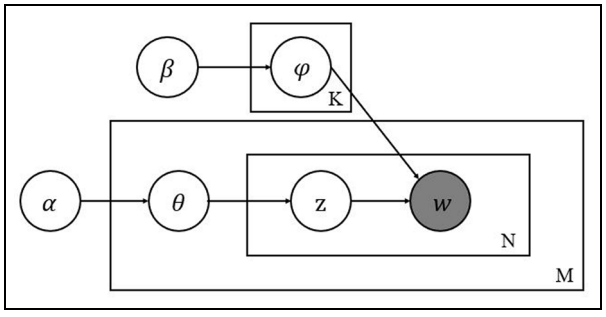

Figure 2. The probability graph structure of latent Dirichlet allocation (LDA).

$$p(\theta_m, z_m, w_m, \Phi|\alpha, \beta) = p(\Phi|\beta)p(\theta_m|\alpha)\prod_{n=1}^{N} p(z_{m,n}|\theta)p\left(w_{m,n}|\Phi_{z_{m,n}}\right) \tag{2}$$

In this study, LDA is presented as a matrix decomposition form

$$\boldsymbol{C} = \boldsymbol{R}_{\boldsymbol{d}}^{T}\boldsymbol{R}_{\boldsymbol{w}} \tag{3}$$

$\boldsymbol{C} \in \mathbb{R}^{V \times M}$ is the corpus inversed index matrix and each element in C is a Boolean value which denotes whether a word is presented. V is the vocabulary size and M is the number of documents is the corpus. C can be decomposed into a topic-word matrix $\boldsymbol{R}_{\boldsymbol{d}}^{T} \in \mathbb{R}^{V \times K}$ and a document-topic matrix $\boldsymbol{R}_{\boldsymbol{w}} \in \mathbb{R}^{K \times M}$, where K is the topic number. Then, a word representation and a document representation can be derived according to these two matrixes

$$\boldsymbol{R}_{\boldsymbol{d}}^{T} = (d_1, d_2, \ldots, d_V)^T \tag{4}$$

$$\boldsymbol{R}_{\boldsymbol{w}} = (w_1; w_2; \ldots; w_M) \tag{5}$$

$w_i \in \mathbb{R}^{1 \times K}$ is the topic distribution of word i and $d_j^T \in \mathbb{R}^{1 \times K}$ is the topic distribution of document j. In this study, they are illustrated as the vector representation.

3.2.2. Bidirectional gated units. The encoder is essentially an RNN that encodes the input sequences into a feature representation. For time series prediction, given the input sequence $\mathrm{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_T)$ with $\mathrm{x}_t \in \mathbb{R}^{(E+K)}$, where T is the number

of terms in an input short text, E is the embedding size of pre-training word embedding, K is the topic number of LDA model. In this stage, the word embedding e_t and topic representation w_t are concatenated

$$\mathrm{x}_t = [\mathrm{e}_{\boldsymbol{t}}; \mathrm{w}_{\boldsymbol{t}}] \tag{6}$$

An encoder f_1 is then applied to learn a mapping from x_t to h_t at time step t with

$$h_t = f_1(h_{t-1}, \mathrm{x}_t) \tag{7}$$

$h_t \in \mathbb{R}^m$ is the hidden state of the encoder at time t, m is the size of the hidden state and f_1 is a non-linear function that could be an LSTM [15] or GRU [16]. In this study, a bidirectional GRU is utilised as f_1 to capture long-term dependencies. Also, the reason for using GRU unit is that the cell state sums activities over time, which helps to overcome the problem of gradients vanishing and better capture long-term dependencies in time series. There are two sigmoid gates in each GRU unit, that is, a reset gate r_t and an update gate z_t. The update of a GRU unit can be summarised as follows

$$r_t = \sigma(W_r[h_{t-1}; x_t]) \tag{8}$$

$$z_t = \sigma(W_z[h_{t-1}; \mathrm{x}_t]) \tag{9}$$

$$\tilde{h}_t = \tanh\left(W_{\tilde{h}}[r_t h_{t-1}; \mathrm{x}_t]\right) \tag{10}$$

$$h_t = (1 - z_t)h_t + z_t h_{t-1} \tag{11}$$

$$y_t = \sigma(W_o h_t) \tag{12}$$

$[h_{t-1}; \mathrm{x}_{\boldsymbol{t}}] \in \mathbb{R}^{m+E+K}$ is a concatenation of the previous hidden state h_{t-1} and the current input $\mathrm{x}_{\boldsymbol{t}}$. W_r, W_z, $W_{\tilde{h}} \in \mathbb{R}^{m \times (m+E+K)}$ are parameters to be learned. σ is the sigmoid function. To gain the dependencies between adjacent words within a single sentence, a bidirectional structure is utilised

$$\overrightarrow{h_t} = \overrightarrow{GRU}\left(W_t, \overrightarrow{h_{t-1}}\right) \tag{13}$$

$$\overleftarrow{h_t} = \overleftarrow{GRU}\left(W_t, \overleftarrow{h_{t-1}}\right) \tag{14}$$

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right] \tag{15}$$

Finally, each $\overrightarrow{h_t}$ is concatenated with $\overleftarrow{h_t}$ to obtain a hidden state h_t. Let the hidden unit number for each unidirectional GRU be u. For simplicity, note all the h_t as $\boldsymbol{H} \in \mathbb{R}^{V \times 2u}$, where V is the number of terms in an input short text.

3.2.3. Self-attention mechanism. A common approach in many previous studies creates a simple vector representation using the final hidden states of RNN or the max (or average) pooling from either RNNs hidden states or convolved n-grammes. However, carrying the semantics along all time steps of a recurrent model is relatively hard and not necessary. Different from previous approaches, a self-attention mechanism was proposed [43], which allows to extract different aspects of a sentence into multiple vector representations. Specifically, in this study, the self-attention mechanism is introduced on top of a GRU in the proposed sentence embedding model. It enables the attention to be used in those cases when there are no extra inputs. In addition, due to the direct access to hidden representations from all time steps, it relieves some long-term memorisation burden from an encoder. The multi-aspect self-attention mechanism can be summarised as follows

$$\boldsymbol{S} = W_{a2} \tanh\left(W_{a1}\boldsymbol{H}^T\right) \tag{16}$$

$$\boldsymbol{A} = \frac{\exp(S_{ri})}{\sum_{j=1}^{T} \exp\left(S_{rj}\right)} \tag{17}$$

$W_{\boldsymbol{a}1} \in \mathbb{R}^{d_1 \times 2u}$ and $W_{\boldsymbol{a}2} \in \mathbb{R}^{r \times d_1}$ are the parameters to be learned. d_1 is a hyper parameter and r is the aspect number expected to be focused on. Equation (17) is a row-wise *softmax* function which ensures all the computed weights in the aspect dimension sum up to 1. $\boldsymbol{A} \in \mathbb{R}^{r \times T}$ is the annotation matrix. Each row of $\boldsymbol{A}$ focuses on a different specific

Table 2. Statistics for the text data sets.

Data set	|C|	Num.	Max Len.	Average Len.	|V|
StackOverflow	20	20,000	31	6.41	5584
Biomedical	20	20,000	39	8.87	9588
News Categories	20	20,000	27	7.70	11,177

|C| means the number of classes; Num. means the data set size; Max Len. means the max length of texts, Average Len. means the mean length of texts and |V| means the vocabulary size.

component of a sentence, like a special set of related words. Then, r-weighted sums are estimated by multiplying the annotation matrix $\boldsymbol{A}$ and GRU hidden states $\boldsymbol{H}$. Finally, the flattened vector, $\boldsymbol{M} \in \mathbb{R}^{r*2u}$, is assumed to be the sentence embedding

$$\boldsymbol{M} = Flatten(\boldsymbol{AH}) \tag{18}$$

Here, the function *Flatten* denotes to flatten a matrix to a vector.

3.2.4. Dense layer decoder and loss function. To derive the same dimension prediction vector with corresponding document-topic vector, a two-layer fully connected dense layer is leveraged to be the decoder

$$\hat{d} = Softmax(W_{s2}Relu(W_{s1}\boldsymbol{M})) \tag{19}$$

$W_{s1} \in \mathbb{R}^{d_2 \times (r*2u)}$ and $W_{s2} \in \mathbb{R}^{K \times d_2}$ are the parameters to be learned and d_2 is a hyper parameter. Here, the function *Relu* denotes the rectified linear unit activation function, which is to filter out the negative output of the first layer. Due to that the corresponding document-topic vector is the topic probabilistic distribution of the document, the second layer of decoder utilises the *softmax* activation function to calculate the posterior probability distribution $\hat{d} \in \mathbb{R}^K$.

The final loss function can be defined as follows,

$$\mathcal{L} = KL(d||\hat{d}) = \sum_{k=1}^{K} d^{(k)} \log\left(\frac{d^{(k)}}{\hat{d}^{(k)}}\right) \tag{20}$$

$d \in \mathbb{R}^K$ is the sentence representation calculated by LDA, and it can be also seen as the probability distribution of each sentence on the topic. Then, the Kullback–Leibler divergence is utilised to measure the entire loss over the whole corpus involving N sentences.

4. Experiments

4.1. Data sets description

In the experiment, three public short text data sets are evaluated. In Table 2, some summary statistics of these data sets are described, and, in Table 3, class labels and exemplary short texts of these data sets are illustrated. All of these data sets are used in a previous study about short text clustering [44]. The average length of instances in three data sets are less than 9, which allow to evaluate the performance with a large volume of short texts. Besides, three data sets come from different fields, which help to eliminate potential influences from domain characteristics. Accordingly, different approaches were benchmarked and the general performance of TAE is presented.

StackOverflow.[1] This data set was published in Kaggle.com. The raw data set consists of 3,370,528 questions, which were posted through 31 July 2012 to 14 August 2012. In the experiment, 20,000 question titles were randomly selected from 20 different tags.

Biomedical.[2] This data set was published in BioASQ's official website. In the experiments, 20,000 paper titles were randomly selected from 20 different MeSH major topics.

News Categories.[3] This public data set was published in Kaggle.com. The raw data set consists of 126,466 samples with 31 categories. 20,000 news articles' titles were randomly selected from 20 different categories.

Table 3. Class labels and example short text.

Data set	Class labels	Example short text
StackOverflow	svn, oracle, bash, apache, excel, matlab, cocoa, visual-studio, osx, wordpress, spring, hibernate, scala, sharepoint, ajax, drupal, qt, Haskell, linq, magento	*How to tab between buttons on an OSX dialog box?* (class label: **osx**)
Biomedical	aging, chemistry, cats, erythrocytes, glucose, potassium, lung, lymphocytes, spleen, mutation, skin, norepinephrine, insulin, prognosis, risk, myocardium, sodium, mathematics, swine, temperature	*The relationship to age and cerebral vascular accidents of fibrin and fibrinolytic activity.* (class label: **aging**)
News Categories	crime, entertainment, world news, impact, politics, weird news, black voices, women, comedy, queer voices, sports, business, travel, media, tech, religion, science, latino voices, education, college	*There Were 2 Mass Shootings in Texas Last Week, But Only 1 On TV.* (class label: **crime**)

4.2. Benchmark approaches

The proposed TAE was compared with nine methods on all three public data sets. Compared approach is widely used and has been proved to achieve good results in text representation. The first four are BoW-based methods, while the other five are sentence embedding–based methods:

Term frequency - inverse document frequency (TF-IDF). This baseline is simple but efficient in many natural language tasks and has been proved to achieve good results in text representation.
Concept frequency - inverse document frequency (CF-IDF) [27]. This baseline uses bag-of-concepts to get the concept vector of vocabulary by clustering word vector and represent each document as combination of different concepts' frequency and inverse document frequency.
LSI [8]. This baseline derives reduced subspace vectors generated by singular value decomposition (SVD) method.
LDA. This baseline is a typical Bayesian-based topic model which is used to detect the latent topics of a document. The topic distribution of each word and document can be derived with this approach.
Doc2Vec [30]. This baseline generates a fixed size vector for a document which is an unsupervised method to learn distributed representation of words and documents.
VAE [34]. This baseline aims to learn the latent distribution of original data, thus making the representations learned by the model more robust.
Convolutional AutoEncoder (*Conv-AE*). This baseline utilises CNN as the main structure of encoder and decoder, which captures the n-gramme feature of input sentences.
LSTM [15]. This baseline has a memory cell with the state at each time step. Access to the memory cell is controlled by three sigmoid gates forget gate, input gate and output gate. In this study, the bidirectional structure of LSTM is utilised to strengthen this baseline and detect the result of max-pooling and average-pooling of hidden states.
GRU [16]. This baseline is an improvement on LSTM which simplified the structure of networks. There are two sigmoid gates in GRU, that is, reset gate and update gate. In this study, the bidirectional structure of LSTM is utilised to strengthen this baseline and detect the result of max-pooling and average-pooling of hidden states.

Each short text document was encoded into a representation vector via those approaches and was then evaluated by the metrics of clustering.

4.3. Evaluation metrics

The clustering performance is evaluated by comparing the clustering results of texts with the tags/labels provided by the text corpus. Two metrics, the normalised mutual information (NMI) metric and the adjusted Rand index (ARI) are used to measure the clustering performance.

NMI between tag/label set T and cluster set C is a popular metric used for evaluating clustering tasks. It is defined as follows

$$NMI(T, C) = \frac{MI(T, C)}{\sqrt{H(T)H(C)}} \tag{21}$$

$MI(T, C)$ is the mutual information between T and C. $H(\cdot)$ is the entropy function and the denominator $\sqrt{H(T)H(C)}$ is used for normalising the mutual information to be in a range of [0, 1].

The ARI is another evaluation metric of clustering. If C is the ground truth class assignment and K is the result from clustering, then a is defined as the number of pairs of elements that are in the same set in C and in the same set in K, and b as the number of pairs of elements that are in different sets in C and in different sets in K. The raw Rand index is then given by

$$RI = \frac{a + b}{C_2^{nsamples}} \tag{22}$$

$C_2^{nsamples}$ is the total number of possible pairs in the data set. However, RI score does not guarantee that random label assignments will get a value close to zero, especially if the number of clusters is in the same order of magnitude as the number of samples. To counter this effect, the expected RI of random labelling, $E[RI]$, can be discounted by defining the adjusted Rand index as follows

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{23}$$

4.4. Parameter settings

For TAE and different sentence embedding approaches, the Adam optimizer was utilised [45] to train the model. The size of the minibatch was 64, and the amount of epoch was 10. The learning rate started from 0.001 and was reduced by 10% after each 10,000 iterations. TAE utilised the equation (20) as the loss function and the hyper parameter r in the attention layer was set as 5. For the latent dimension of GRU, hyper parameters d_1 and d_2 were 50. For TAE and sentence embedding approaches, the input sequence was truncated and padded into the average document length of the corpus. The word embedding leveraged in this study was pre-trained Glove with 300 dimensions [46].[4]

For CF-IDF, word embedding representations were clustered in Glove by k-means and the concept number was 500. For LDA and LSI, trading off the efficiency and performance, the topic number was set as 100. The iteration of LDA was 1000, and the pass of each iteration was set as 100. For Doc2Vec, the training approach was distributed memory model of paragraph vectors (PV-DM). The vector size was 300 with the window size of 5 and the minimum count number was set as 2. The Gensim toolkit was utilised to training Doc2Vec, LDA and LSI models. For VAE, the batch size was set as 64 and the latent dimension was set as 100. The input word embedding matrix of a document was flattened as a vector. Then, input word embedding matrix was reduced to a low-dimensional vector and the intermediate dimension of reduced dimensional vector was set as 256. The optimisation method was Adam with default parameter. For convolutional auto encoder, the word embedding matrix was used as input. Then, one-dimensional convolution and max-pooling were adopted in the encoder component, which aim to combine the n-gramme features of a sentence. The optimisation method was Adam with default parameter and L2 loss was adopted as optimal target. For GRU and LSTM, the latent dimension was set as 50. The optimise method was Adam with default parameter, and L2 loss was adopted as optimal target.

For all approaches in the experiment, the k-means was utilised as the cluster approach and the method of initialising the cluster centres was K-means$++$. Due to the fact that the concentration in this study is on the representation of short text clustering, the cluster number was set as the ground truth of each corpus.

4.5. Result

In Table 4, the NMI and ARI are reported by comparing the proposed approaches with the other baseline methods. Some findings are summarised into five aspects:

1. The performance of sentence embedding–based approach is more stable. In all of three data sets, all different sentence embedding–based approaches do not perform the worst, although they do not always get the highest scores. One potential reason can be that the pre-trained word embedding encodes the uniform concept and knowledge,

Table 4. Cluster result in three data sets.

Models	StackOverflow		Biomedical		News Categories	
	NMI (%)	ARI (%)	NMI (%)	ARI (%)	NMI (%)	ARI (%)
Bag-of-Word-based methods						
CF-IDF	12.81	05.67	10.97	03.93	08.32	03.11
TF-IDF	15.64	07.31	25.43	05.43	07.06	00.60
LSI	05.26	00.81	08.82	01.47	02.86	00.84
LDA	41.78	23.76	19.46	06.30	03.63	00.48
Neural sentence embedding methods						
Doc2Vec	01.93	00.50	01.11	00.23	00.68	00.08
VAE	02.23	00.65	11.97	04.70	02.06	04.70
Conv-AE	04.37	01.48	12.67	05.62	04.37	01.48
Bi-LSTM (max-pooling)	18.80	08.56	26.02	14.17	08.62	03.37
Bi-LSTM (average-pooling)	16.02	06.93	23.99	12.66	08.62	03.40
Bi-GRU (max-pooling)	20.47	09.39	26.06	14.72	08.86	03.40
Bi-GRU (average-pooling)	20.63	09.68	25.68	13.79	09.25	03.43
Proposed method						
TAE (K = 100, r = 5)	**62.80**	**45.38**	30.98	19.51	**11.73**	**05.69**
TAE (K = 100, r = 10)	61.82	45.10	**32.51**	**19.76**	09.94	04.92
TAE (K = 100, r = 15)	60.02	44.10	31.08	19.04	10.33	04.88
TAE (K = 200, r = 5)	11.15	03.67	22.00	11.59	09.67	03.70
TAE (K = 300, r = 5)	16.98	08.03	24.09	12.98	08.86	03.50

NMI: normalised mutual information; ARI: adjusted Rand index; LSI: latent semantic indexing; LDA: latent Dirichlet allocation; VAE: Variational AutoEncoder; LSTM: long short-term memory; GRU: gated recurrent unit; TAE: topic attention encoder.
Best performance in boldface, another top 3 of benchmark model are underscored.

thus making the performance of sentence embedding–based approach does not change much more with corpus types.

2. BoW-based methods are still of strong baseline in the task of short text cluster task. As illustrated before, BoW-based methods directly model the linkage of documents, which help to introduce the cross-document information. As global information, cross-document information makes the representations of documents which share common words more relevant. Therefore, in those corpora which have a smaller vocabulary size, which indicates that, in same document number, there will have more co-occurrence between each document, BoW-based methods can get roughly the same or better performance. See the results in the StackOverflow and Biomedical, which indicate the argument discussed above.
3. It is known that max-pooling can improve the performance of sentence embedding in some NLP tasks in previous studies [43,47]. In this study, similarly, it is found that max-pooling is still slightly better than average pooling in both LSTM and GRU in the scenario of short text clustering.
4. Compared with other approaches, TAE obtained state-of-the-art results and demonstrated the superiority in all three data sets. It is worth noting that, as an integration of BoW-based method and sentence embedding–based method, TAE relies on the performance of LDA and GRU. As presented, in the experiment of News Categories data set, the performance of TAE is not much higher than LDA and GRU, when these two methods do not perform well.
5. Different data sets have different optimal parameters of the proposed model. The best parameter combination is K = 100, r = 5 among all these data sets. The detailed parameter sensitivity analysis will be presented at Section 5.1.

5. Discussion

5.1. Parameter sensitivity analysis

5.1.1. Effect of attention aspect number. Leveraging multiple aspect attention in TAE is expected to provide more abundant information from the text corpus [43]. It makes sense to evaluate how significant the improvement can be brought by the attention aspect number r. Taking all three data sets in Section 4.2, r was evaluated by ranging from 1 to 30. Other training parameter remained the settings in Section 4.4. The sensitivity analysis result is illustrated in Figure 3. Obviously, an

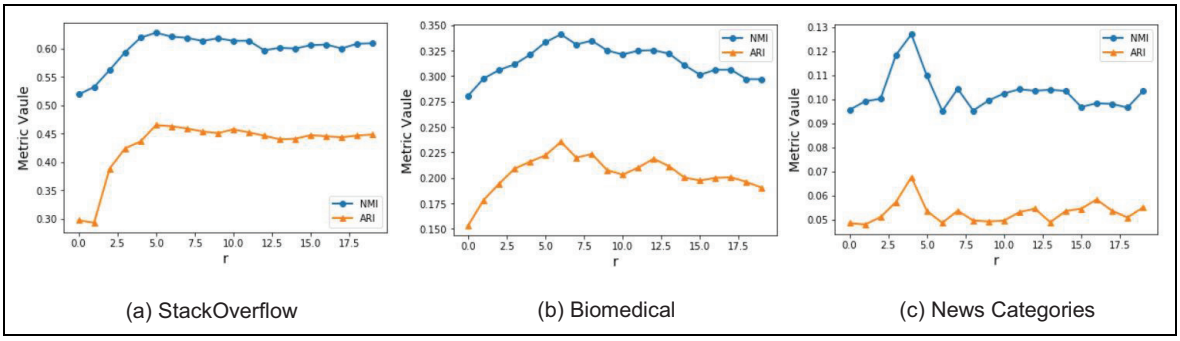

Figure 3. Effect of the number of attention aspects *r* in TAE.

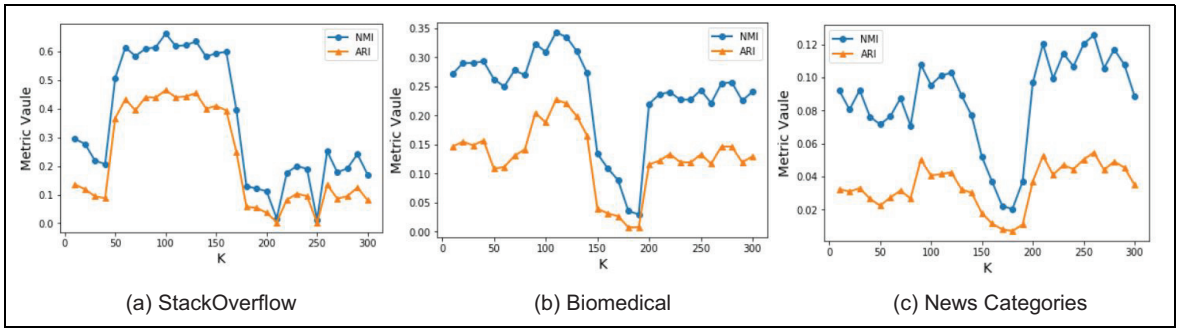

Figure 4. Effect of the number of topics *K* in TAE.

appropriate number of attention aspects is expected to improve TAE on clustering. However, with the increase in number of attention aspects, the performance will reach a plateau and the increase in embedding size also leads to more time consumption and less efficiency.

5.1.2. Effect of topic number. Notably, the topic number *K* in LDA affects both document-topic vectors, which are utilised as the learning target, and word-topic vectors, which are utilised as a proportion of input that decides the proportion of local and global information. To this end, it makes sense to evaluate the effect to the clustering result brought by the topic number *K*. Taking all three data sets in Section 4.2, *K* was evaluated by ranging from 10 to 300 with 10-size interval. Other training parameter retained the settings in Section 4.4. The sensitivity analysis result is illustrated in Figure 4. It is observed that the result of clustering can be worse when the topic number is too small or too big. In brief, the best topic number range is 80–120.

5.2. Ablation study

Pervious categories of experiments show that TAE framework has a large margin of improvement than benchmark methods. However, results presented so far do not show specific contributions from each aspect of TAE. Note that one of the most important differences between TAE and benchmarked approaches is the input, which concatenates the pre-trained word embedding vector and the word-topic vector. Moreover, there are two main components in TAE, which includes self-attention mechanism and bidirectional GRU. These two components together form the encoder module in the model, thus capturing both global and local information. Accordingly, the ablation study over facets of inputs and model structure are presented in this section to better understand their relative importance. In this study, four variants of two facets for TAE and its original model were examined on three data sets and all results of the ablation study are listed in Table 5.

In terms of the input ablation study, for the first variant, the input is the Glove vector only, which is a kind of widely used word embedding. In the second variant, the input has word-topic vector only. As presented in Table 5, it indicates that original TAE performs the best among all three data sets. With the Glove vector only as the input, critical local information is focussed and global information is neglected. Comparatively, with the topic-word vector only as the input,

Table 5. Ablation study of TAE.

Models	StackOverflow		Biomedical		News Categories	
	NMI (%)	ARI (%)	NMI (%)	ARI (%)	NMI (%)	ARI (%)
TAE	62.80	45.38	30.98	19.51	11.73	05.69
Input ablation study						
TAE (Only Glove vector input)	49.61	27.64	26.46	13.70	10.46	04.35
TAE (Only topic-word vector input)	24.91	11.32	20.07	08.47	03.74	01.20
Structure ablation study						
W/O attention	60.72	42.67	30.06	18.09	09.15	03.44
W/O bidirectional	61.73	44.71	30.38	18.79	11.02	05.46

NMI: normalised mutual information; ARI: adjusted Rand index; TAE: topic attention encoder.

global information is paid attention and there is lack of local information that encodes the semantics of short text, which leads to the worst performance. In terms of the structure ablation study, for the first variant, the self-attention mechanism is removed from TAE. As a consequence, it is replaced by an average pooling operation. In contrast to self-attention mechanism, average pooling gives each output of bidirectional GRU an equal weight, which means the information at each time step is equally important. For the second variant, the bidirectional GRU is replaced by a unidirectional GRU, which only captures the global and local information from one side. As presented in Table 5, the original TAE is still the best in comparison with above two variants. However, compare to the change of inputs, when the attention mechanism and the bidirectional GRU are removed from TAE, the performance of these variants just slightly declines. It indicates that most performance improve of TAE come from the combined inputs. Coupled with the use of novel training targets, TAE can better learn global and local information. It is worth noting that comparing with the removal of the bidirectional GRU, the model presented an obvious declined performance with the removal of the attention mechanism. It indicates that attention mechanism can bring more benefit for TAE.

5.3. Efficiency comparison

In this section, an efficiency comparison was conducted between TAE and other benchmark models. Categories of experiments were conducted to evaluate the time consumption per epoch during the training procedure of neural sentence embedding methods. All results were obtained on a GeForce GTX 1070 Max-Q GPU, and they are presented in Table 6. Note that all the BoW-based methods do not apply gradient-based optimisation method, and it is inaccessible and meaningless to give the training time per epoch for these models. As seen from Table 6, TAE achieves state-of-the-art results in approximate time consumption compared with other neural sentence embedding–based methods.

5.4. Case study

In Section 1, two categories of sentences are taken as examples to demonstrate problems that some BoW-based methods or word embedding–based methods fail to deal with. In this section, a case study of these two categories of sentences' cosine similarity was conducted. The reason for choosing cosine similarity is that the encoded embedding vectors are normalised before clustering, since the k-means algorithm often invites the Euclidean distance as a metric and it is obvious that the Euclidean distance of normalised vectors is proportional to their cosine similarity. Also, since that the maximum cosine similarity is one, choosing cosine similarity helps to understand the metric used in clustering. The results of two categories of sentences are listed in Table 7.

Sentences A and B are sampled from the same category of the News Categories data set. The higher the similarity, the better the result. According to Table 7, BoW-based methods cannot estimate the similarity between Sentence A and Sentence B reasonably since they do not share common words. However, in this circumstance, local semantics help indicate the similarity of the two sentences. Such kind of information indicates the semantics of each document, and semantics are embedded in document itself. Accordingly, sentence embedding models relieve such problems, which introduce the semantic information into the sentence representations. Even though Sentences A and B do not share common words, all of the neural sentence embedding methods give a high similarity between Sentence A and Sentence B since they use pre-trained word embedding vectors as the input, except Doc2Vec. On the contrary, TAE assigns a relatively high similarity between Sentence A and Sentence B. Although not higher than that are given by most of neural sentence

Table 6. Efficiency comparison of each model in StackOverflow data set.

Models	Time (s/epoch)	ARI (%)	NMI (%)
Bag-of-Word-based methods			
CF-IDF	N/A	12.81	05.67
TF-IDF	N/A	15.64	07.31
LSI	N/A	05.26	00.81
LDA	N/A	41.78	23.76
Neural sentence embedding methods			
Doc2Vec	**0.814**	01.93	00.50
VAE	1.487	02.23	00.65
Conv-AE	1.985	04.37	01.48
Bi-LSTM (max-pooling)	10.584	18.80	08.56
Bi-LSTM (average-pooling)	9.637	16.02	06.93
Bi-GRU (max-pooling)	8.852	20.47	09.39
Bi-GRU (average-pooling)	8.954	20.63	09.68
Proposed method			
TAE (K = 100, r = 5)	10.089	**62.80**	**45.38**

NMI: normalised mutual information; ARI: adjusted Rand index; LSI: latent semantic indexing; LDA: latent Dirichlet allocation; VAE: Variational AutoEncoder; LSTM: long short-term memory; GRU: gated recurrent unit; TAE: topic attention encoder.

Best performance in boldface, another top 3 of benchmark model in underscore.

Table 7. Case study of sentences' cosine similarity.

Models	Sentences A and B	Sentences C and D
Bag-of-Word-based methods		
CF-IDF	0.000	0.000
TF-IDF	0.000	0.000
LSI	0.000	0.044
LDA	0.000	0.044
Neural sentence embedding methods		
Doc2Vec	0.814	0.442
VAE	0.712	0.689
Conv-AE	0.888	0.872
Bi-LSTM (max-pooling)	0.964	0.953
Bi-LSTM (average-pooling)	0.959	0.982
Bi-GRU (max-pooling)	0.960	0.983
Bi-GRU (average-pooling)	0.959	0.985
Proposed method		
TAE (K = 100, r = 5)	0.737	0.380

NMI: normalised mutual information; ARI: adjusted Rand index; LSI: latent semantic indexing; LDA: latent Dirichlet allocation; VAE: Variational AutoEncoder; LSTM: long short-term memory; GRU: gated recurrent unit; TAE: topic attention encoder.

Sentences A and B are sampled from the same category of News Categories data set. The higher the similarity, the better the result. Sentences C and D are sampled from different categories of StackOverflow data set. The lower the similarity, the better the result.

embedding methods, TAE can still build a high similarity between the two sentences, avoiding the mistake induced by BoW-based methods.

Sentences C and D are sampled from two different categories of the StackOverflow data set. The lower the similarity, the better the result. According to Table 7, neural sentence embedding–based methods cannot identify the differences between Sentences C and D, and the generated sentence embedding vectors are similar. One important reason might be that they do not encode global dependency information and thus give a high similarity unreasonably from the introduced word embedding. All BoW-based methods give a lower similarity between Sentence C and Sentence D since they estimate the similarity with shared common words, which is a natural global dependency. Such information mirrors the cross-document relationship of each document. One type of natural cross-document relationship is common words, which appear in different documents and connect them, which can be inferred by considering the entire corpus. Comparatively,

TAE gives a relatively low similarity between Sentence C and Sentence D. Although not lower than those given by all BoW-based methods, TAE can still identify the differences between the two sentences, avoiding the mistake induced by neural sentence embedding methods.

6. Conclusion

In this article, a novel TAE is proposed, which shows the effectiveness in short text clustering by comparing different approaches on three large data sets. In order to integrate the superiority of BoW-based method and sentence embedding–based method, the cross-document information is introduced via the document-topic vector as learning target and the concatenation of the word embedding and corresponding topic-word vector as input. Then, a self-attention mechanism is utilised, which aims to extract the weights of hidden states adaptively and encode the final semantics of each short text document. Thorough empirical studies-based upon public data sets demonstrate that TAE outperforms popular document representation benchmark methods for short text clustering.

For the future, there exist several promising tasks to be investigated. For instance, TAE is limited by the performance of representation derived through topic model. In the future, better short text clustering methods with an end-to-end structure are to be explored. In addition, the training of TAE is time-consuming to gain topics from massive text data. With the selected three data sets, topic estimation accounted for a large proportion of overall training time. In the future, a model with higher scalability should be developed. Finally, one innovation of TAE is to introduce cross-document information into the representation learning of document, which implicitly introduces such information via the integration of bidirectional GRU and topic model. Since the relationship of documents can be regarded as graph, in the future, graph neural network [48] can be considered to model the relevance of different documents explicitly.

Acknowledgements

The authors thank the anonymous reviewers for their comments that have contributed to important improvements of the paper.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by grants from the National Nature Science Foundation of China (project no. NSFC 71701019/G0114) and National Social Science Foundation of China (Grant.19ATQ005).

ORCID iD

Jian Jin https://orcid.org/0000-0002-3239-2294

Notes

1. https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow/download/train.zip
2. http://participants-area.bioasq.org
3. https://www.kaggle.com/rmisra/news-category-dataset
4. http://nlp.stanford.edu/data/glove.840B.300d.zip

References

[1] Li J, Ritter A and Hovy E. Weakly supervised user profile extraction from Twitter. In: *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers)* (ed K Toutanova and H Wu), Baltimore, MD, 23–25 June 2014, pp. 165–174. Stroudsburg, PA: ACL.

[2] Wang J, Li Q, Chen YP et al. Recommendation in internet forums and blogs. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*, Uppsala 11–16 July 2010, pp. 257–265. Stroudsburg, PA: ACL.

[3] Zhou J, Cao Y, Wang X et al. Deep recurrent models with fast-forward connections for neural machine translation. *Trans Assoc Comput Linguist* 2010; 4: 371–383.

[4] Harris ZS. Distributional structure. In: ZS Harris and H Hiż (eds) *Papers on syntax*. Dordrecht; London: Reidel, 1981, pp. 3–22.

[5] Huang A. Similarity measures for text document clustering. In: *Proceedings of the sixth New Zealand computer science research student conference*. Christchurch, New Zealand, 2008, pp. 49–56, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.332.4480&rep=rep1&type=pdf
[6] Kolari P, Java A, Finin T et al. Detecting spam blogs: a machine learning approach. In: *The twenty-first national conference on artificial intelligence (AAAI-06)*, Boston, MA, 16–20 July 2006, pp.1351. Menlo Park, CA: AAAI.
[7] Wu L, Hoi SCH and Yu N. Semantics-preserving bag-of-words models and applications. *IEEE Trans Image Process* 2010; 19: 1908–1920.
[8] Deerwester S. Improving information retrieval with latent semantic indexing. In: *Proceedings of the 51st annual meeting of the American society for information science*, 1988, pp. 36–40, https://www.bibsonomy.org/bibtex/18dc6270c4038a38c8b53a97e1f737a54/sb1989
[9] Hofmann T. Learning the similarity of documents: an information-geometric approach to document retrieval and categorization. In: *Advances in neural information processing systems* (NIPS 2000), 2000, pp. 914–920, https://papers.nips.cc/paper/1999/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html
[10] Blei DM, Ng AY and Jordan MI. Latent Dirichlet allocation. *J Mach Learn Res* 2003; 3: 993–1022.
[11] Yan X, Guo J, Lan Y et al. A biterm topic model for short texts. In: *Proceedings of the 22nd international conference on World Wide Web* (ed D Schwabe), Rio de Janeiro, Brazil, 13–17 May 2013, pp. 1445–1456. Geneva: International World Wide Web Conferences Steering Committee.
[12] Yin J and Wang J. 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* (eds S Macskassy, C Perlich, J Leskovec, W Wang and R Ghani), New York, 24–27 August 2014, pp. 233–242. New York: ACM.
[13] Yan X, Guo J, Liu S et al. Clustering short text using Ncut-weighted non-negative matrix factorization. In: *Proceedings of the 21st ACM international conference on information and knowledge management* (ed X Chen, G Lebanon, H Wang et al.), Maui, HI, 29 October–2 November 2012, p. 2259. New York: ACM.
[14] Yan X, Guo J, Liu S et al. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: *Proceedings of the 2013 SIAM international conference on data mining* (ed J Gosh), 2013, pp. 749–757. Philadelphia, PA: SIAM, https://pdfs.semanticscholar.org/b5d0/36429877568a648389531e323ea0983a5148.pdf
[15] Hochreiter S and Schmidhuber J. Long short-term memory. *Neur Comput* 1997; 9: 1735–1780.
[16] Cho K, van Merrienboer B, Bahdanau D et al. On the properties of neural machine translation: encoder-decoder approaches, https://arxiv.org/abs/1409.1259
[17] Kim Y. Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (ed A Moschitti, G Bo Pang and W Daelemans), Doha, Qatar, 25–29 October 2014, pp. 1746–1751. Stroudsburg, PA: ACL.
[18] Vincent P, Larochelle H, Bengio Y et al. Extracting and composing robust features with denoising autoencoders. In: *Proceedings of twenty-fifth international conference on machine learning* (ed A McCallum and S Roweis) Helsinki, 5–9 July 2008, pp. 1096–1103. Helsinki: University of Helsinki.
[19] Socher R, Huang EH, Pennin J et al. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: *Advances in neural information processing systems 24 (NIPS 2011)*, Granada, 12–17 December 2011, pp. 801–809. Cambridge, MA: MIT Press.
[20] Kiros R, Zhu Y, Salakhutdinov RR et al. Skip-thought vectors. In: *Advances in neural information processing systems 28 (NIPS 2015)*, Montréal, QC, Canada, 7–12 December 2015, pp. 3294–3302. Cambridge, MA: MIT Press.
[21] Tang S, Jin H, Fang C et al. Trimming and improving skip-thought vectors, https://arxiv.org/abs/1706.03148
[22] Banerjee S, Ramanathan K and Gupta A. Clustering short texts using Wikipedia. In: *30th annual international ACM SIGIR conference on research and development in information retrieval* (ed. CLA Clarke), Amsterdam, The Netherlands, 23–27 July 2017, p. 787. New York: ACM.
[23] Gabrilovich E and Markovitch S. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: *Proceedings of the twentieth international joint conference on artificial intelligence (IJCAI-2007)*. Hyderabad, India, January 2007, pp. 1606–1611. Burlington, MA: Morgan Kaufmann.
[24] Fodeh S, Punch B and Tan P-N. On ontology-driven document clustering using core semantic features. *Knowl Inf Syst* 2011; 28: 395–421.
[25] Milne D, Medelyan O and Witten I. Mining domain-specific thesauri from Wikipedia: a case study. In: *2006 IEEE WICACM international conference on web intelligence (WI 2006 main conference proceedings) (WI '06)*, Hong Kong, China, 18–22, December 2006, pp. 442–448. Los Alamitos, CA: IEEE.
[26] Tang J, Wang X, Gao H et al. Enriching short text representation in microblog for clustering. *Front Comput Sci* 2012; 6: 88–101.
[27] Kim HK, Kim H and Cho S. Bag-of-concepts: comprehending document representation through clustering words in distributed representation. *Neurocomputing* 2017; 266: 336–352.
[28] Jia C, Carson MB, Wang X et al. Concept decompositions for short text clustering by identifying word communities. *Patt Recog* 2018; 76: 691–703.

[29] Le Q and Mikolov T. Distributed representations of sentences and documents. In: *International conference on machine learning*, 2014, pp. 1188–1196, https://cs.stanford.edu/~quocle/paragraph_vector.pdf
[30] Mikolov T, Chen K, Corrado G et al. Efficient estimation of word representations in vector space, https://arxiv.org/abs/1301.3781
[31] Sutskever I, Vinyals O and Le QV. Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems 27 (NIPS 2014)*. Montréal, QC, Canada, 8–13 December 2014, pp. 801–809. Cambridge, MA: MIT Press.
[32] Graves A and Mohamed A-r Hinton G. Speech recognition with deep recurrent neural networks. In: *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Vancouver, BC, Canada, 26–31 May 2013, pp. 6645–6649. Piscataway, NJ: IEEE.
[33] Shang L, Lu Z and Li H. Neural responding machine for short-text conversation, https://arxiv.org/abs/1503.02364
[34] Kingma DP and Welling M. Auto-encoding variational Bayes, https://arxiv.org/abs/1312.6114
[35] Logeswaran L and Lee H. An efficient framework for learning sentence representations, https://arxiv.org/abs/1803.02893
[36] Conneau A, Kiela D, Schwenk H et al. Supervised learning of universal sentence representations from natural language inference data. In: *Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP)*, Copenhagen, 7–11 September 2017, pp. 670–680. Stroudsburg, PA: ACL.
[37] Cao Z, Li S, Liu Y et al. A novel neural topic model and its supervised extension. In: *Proceedings of the twenty-ninth AAAI conference on artificial intelligence*, Austin, TX, 25–29 January 2015, pp. 2210–2216. Menlo Park, CA: AAAI.
[38] Liu Y, Liu Z, Chua TS et al. Topical word embeddings. In: *Proceedings of the twenty-ninth AAAI conference on artificial intelligence*, Austin, TX, 25–29 January 2015, pp. 2418–2424. Menlo Park, CA: AAAI.
[39] Dieng AB, Wang C, Gao J et al. TopicRNN: a recurrent neural network with long-range semantic dependency, https://arxiv.org/abs/1611.01702
[40] Li S, Chua TS, Zhu J et al. Generative topic embedding: a continuous representation of documents. In: *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, Berlin, 7–12 August 2016, pp. 666–675. Stroudsburg, PA: ACL.
[41] Lau JH, Baldwin T and Cohn T. Topically driven neural language model. In: *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, Vancouver, BC, Canada, 30 July–4 August 2017, pp. 355–365. Stroudsburg, PA: ACL.
[42] Peng M, Xie Q, Zhang Y et al. Neural sparse topical coding. In: *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers)*, Melbourne, VIC, Australia, 15–20 July 2018, pp. 2332–2340. Stroudsburg, PA: ACL.
[43] Lin Z, Feng M, Santos CN et al. A structured self-attentive sentence embedding, https://arxiv.org/abs/1703.03130
[44] Xu J, Xu B, Wang P et al. Self-taught convolutional neural networks for short text clustering. *Neur Netw* 2017; 88: 22–31.
[45] Kingma DP and Ba J. Adam: a method for stochastic optimization, https://arxiv.org/abs/1412.6980
[46] Pennington J, Socher R and Manning C. Glove: global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, 25–29 October, 2014, pp. 1532–1543. Stroudsburg, PA: ACL.
[47] Conneau A, Kiela D, Schwenk H et al. Supervised learning of universal sentence representations from natural language inference data. In: *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, Vancouver, BC, Canada, 30 July–4 August 2017, pp. 670–680. Stroudsburg, PA: ACL.
[48] Kipf TN and Welling M. Semi-supervised classification with graph convolutional networks, https://arxiv.org/abs/1609.02907