Neurocomputing 463 (2021) 596–608

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Automatic topic labeling using graph-based pre-trained neural embedding

Dongbin He [a,b,e], Yanzhao Ren [c,b], Abdul Mateen Khattak [a,d], Xinliang Liu [a,b], Sha Tao [a,b], Wanlin Gao [a,b,*]

[a] College of Information and Electrical Engineering, China Agricultural University, Beijing, China
[b] Key Laboratory of Agricultural Informatization Standardization, Ministry of Agriculture and Rural Affairs, Beijing, China
[c] College of Science, China Agricultural University, Beijing, China
[d] Department of Horticulture, The University of Agriculture, Peshawar 25120, Pakistan
[e] College of Computer Science and Engineering, Shijiazhuang University, Shijiazhuang, China

ARTICLE INFO

Article history:
Received 27 July 2020
Revised 19 April 2021
Accepted 15 August 2021
Available online 19 August 2021
Communicated by Zidong Wang

Keywords:
Automatic labeling
Neural embedding
Graph-based ranking
Topic model
Topic label
Latent dirichlet allocation (LDA)

ABSTRACT

It is necessary to reduce the cognitive overhead of interpreting the native topic term list of the Latent Dirichlet Allocation (LDA) style topic model. In this regard, automatic topic labeling has become an effective approach to generate meaningful alternative representations of topics discovered for end-users. In this study, we introduced a novel two-phase neural embedding framework with the redundancy-aware graph-based ranking process. It demonstrated how pre-trained neural embedding could be usefully applied in topic terms, sentence presentations, and automatic topic labeling tasks. Moreover, reranking the topic terms optimized the discovered topics with fewer yet more representative terms while retaining the topic information integrality and fidelity. It further decreased the burden of computation caused by neural embedding and improved the overall effectiveness of the labeling system. Compared with the prevailing state-of-the-art and classical labeling systems, our efficient model boosted the quality of the topic labels generated and discovered more meaningful topic labels.

1. Introduction

Topic modeling is a significant technique in natural language processing (NLP) tasks, such as information retrieval and text mining [1–3]. The topics discovered are usually represented by a list of topic terms with a marginal probability [4,5]. If the users want to understand the meaning of a topic discovered, they must carefully check the topic term list ranked in descending order of marginal probability. Unfortunately, it is difficult to fully understand a topic discovered and distinguish it from others by merely relying on the topic terms, especially when the users lack the background knowledge in the field of topics. Although manually labeled topic labels are more interpretive and understandable, doing so requires considerable human labor to review the massive data of the corpus. Moreover, manual labels tend to be subconsciously influenced by subjective opinions [6].

It is well known that the existing automatic topic labeling methods that generate phrases [3,6–13], textual summaries [1,14–16], or images [17–19] for the discovered topics to assist in their interpretation have received more attention and are very challenging.

However, the phrases are usually insufficient to interpret the discovered topics owing to their finite length, and images do not fit into most NLP scenarios [16]. To make a clearer and more comprehensive description of the discovered topics, researchers began to study automatic summarization methods [1,14,16]. However, the mentioned methods relied only on surface features and faced two common pitfalls during topic labeling tasks. First, exact word matching leads to performance underestimation in case of non-similar semantically correct words. Second, they fail to capture the contextual meaning of the words or sentences when computing semantic distances.

For a text generation task such as textual summarizing, methods that utilize contextualized embedding semantics perform better than those using surface forms [20]. Over the years, several different methods have been proposed to build neural embeddings, such as Word2vec [21], Doc2vec [22], and BERT [23]. Word2vec trains words into a multidimensional vector based on the hypothesis that the words neighboring similar words tend to have similar meanings. Doc2vec can learn neural embedding for word sequences, for example, sentences, paragraphs, and documents.

[*] Corresponding author.
E-mail address: wanlin_cau@163.com (W. Gao).

https://doi.org/10.1016/j.neucom.2021.08.078
0925-2312/© 2021 Elsevier B.V. All rights reserved.

Obviously, BERT is the first fine-tuning-based representation model that achieved state-of-the-art performance in multiple downstream NLP tasks [23]. In contrast to previous pre-training models, it captures true bidirectional context information. BERT has the advantage of polysemy over Word2vec because it generates word embeddings that are dynamically determined by the words around them; however, the neural embedding generated by Word2vec or Doc2vec has fixed representation.

Unfortunately, there is currently no dynamic neural embedding scheme to represent topic terms. Therefore, we use Doc2vec and Word2vec to generate static neural embedding for sentences, words, and topic terms. To the best of our knowledge, we are the first to introduce a redundancy-aware graph-based topic label ranking model with pre-trained neural embedding (TLRE) for topic labeling tasks. The method relies on continuous neural vector representation that captures the semantic and syntactic information of the topic terms and sentences.

However, using continuous neural vector to represent sentences and topics introduces an additional burden of increased computation in the training process. To address this problem, we need to optimize topics using the topic ranking method. Topic ranking can optimize the discovered topics, which can then be represented by fewer yet more representative terms. We find that using the top20 topic terms instead of 500 [1] to represent the topic reduces the number of topic terms by 96%, which greatly decreases the computational burden of word embedding in our model. Besides, topic ranking provides us more important topic terms and their optimal marginal probability, which results in the selection of candidate sentences with stronger correlation and coherence, further improving the quality of subsequent topic label generation.

The experimental results show that the TLRE can extract salient sentences and generate meaningful topic labels with minimum redundancy. Moreover, the textual summaries (topics labels) for the discovered topics are more accurate and explicitly satisfy the three critical criteria proposed by Mei et al. [6] and Wan et al. [1], namely, high Relevance, Coverage, and Discrimination. In short, this study makes three key contributions. (1) Topic ranking can optimize the discovered topics, so that they can be represented by fewer yet more representative terms. It also results in the selection of strongly correlated and coherent candidate sentences, and greatly decreases the computational burden of word embedding in the model and improves the overall effectiveness of the labeling system. (2) We leverage neural embedding computation to address the problem of semantically correct words and sentences being penalized in the extraction process owing to unmatched surface features. Thus, a novel graph-based ranking method with neural embedding is proposed. It extracts the salient sentence to form topic labels and simultaneously reduces redundancy. (3) To evaluate the generated topic labels more objectively and impartially, we propose a novel approach adapted from BERTScore [24], which works well using neural embedding.

2. Related work

The topic model is a popular approach to find hot spots and track the event development trends in a corpus [12]. To help users understand discovered topics clearly, Blei et al. [4] proposed a frame to visualize topics discovered by the Latent Dirichlet Allocation (LDA) model to generate multi-word distributions of topics. Most works have employed phrases to interpret the topics of the LDA model; for example, Wang et al. proposed an n-gram topic model to obtain more interpretable topics and meaningful phrases [2]; Mei et al. [6] introduced an unsupervised method to label the discovered topic and treated the label task as an optimization problem. Using surface feature computation, it ranked candidate labels based on minimized Kullback-Leibler divergence (KLD) between phrases and topic models.

Magatti et al. [7] proposed an algorithm derived from the hierarchical topic model for the automatic labeling of discovered topics. The labeling method aims to find the fittest labels among given topics and the hierarchy obtained from gDir (Google Directory) service. Lau et al. [8] introduced a supervised learning approach to rank the candidate topic labels that involved noun chunks extracted from external sources, that is, Wikipedia and the top5 topic terms. Kou et al. [11] proposed a framework to generate topic labels with word vectors and letter trigram vectors. A chunk parser generates the majority of candidate labels and at least one additional top10 term is involved. Alokaili et al. were the first to propose a seq2seq topic labeling model to produce appropriate textual labels for discovered topics and use BERTScore to measure the generated topic labels [3]. Although the experimental results were efficient and concise, they might still be insufficient to express rich topics.

Recently, researchers have been paying more attention to the use of automatic summarization technology to label topic models for obtaining more informative and meaningful topic labels. Identifying dominant words in the topic models of Twitter, Basave et al. [14] first introduced an independent summarization framework, independent of external sources (e.g., Wikipedia, WordNet), to automatically label the topics of social media. It shows that summarization algorithms can achieve better performance in labeling topics than other methods (using terms or phrases).

Barawi et al. [15] studied the problem of labeling sentiment-bearing topics and introduced a method to label them with descriptive sentences; the method not only outperformed others, but also promoted the explanation and interpretation of the discovered topics.

To label discovered topics with higher Relevance, Coverage, and Discrimination, Wan et al. [1] proposed a novel two-stage textual summary framework (candidate sentence selection and summary generation). The framework was based on submodular optimization to generate a fixed-length summary involving the top-score sentences for each discovered topic.

Baralis et al. [25] introduced a graph-based automatic summarization method (GRAPHSUM), in which the nodes represent combinations of two or more terms. It learned and exploited a specific strategy and used a PageRank algorithm to choose significant sentences for topic labeling tasks. He et al. [16] introduced a novel graph-based ranking model, which employed two specific strategies based on surface features to restrain or enhance the voting rate for each graph node. Their method also generated less redundant text summaries with high Relevance, Coverage, and Discrimination.

Bhatia et al. [12] proposed a topic labeling method to select the most relevant labels (semantically correct phrases) for discovered topics by computing neural embedding of documents and words. They trained a Doc2vec model on the English Wikipedia corpus to generate sentence and word embedding (the latter was generated by Word2vec during the internal training process). Finally, compared with the competitor methods, their model achieved the best results across a wide range of domains.

3. Problem definition

A topic discovered using the LDA [4,5] modeling approach is a soft cluster of weighted terms based on their co-occurrence in the documentation set [26]. In this study, we set K as the number of topics discovered using the LDA model (it has to be fixed a-priori [16]) and used the Gibbs algorithm method to fit the model. A discovered topic T is a probability distribution list of term $\{p_T(w)\}_{w\in V}$, where V is a vocabulary set of the corpus. Besides, for each topic T, we have $\sum_{w\in V} p_T(w) = 1$[10].

Unlike previous studies [22,27], contextual neural embedding such as BERT [23] outputs very different vector representations of the same word in different sentences, according to the surrounding words [24]. To date, a reasonable way to use BERT embedding to represent topic terms has not been discovered. Therefore, we applied a Doc2vec[1] model trained with English Wikipedia literature[2] to generate static neural embedding for sentences, words, and topic terms.

It is especially important that the embedding of words in sentences is learned during the training process as the Doc2vec model runs Word2vec internally [12]. Thus, we can suggest that the Doc2vec trained model can infer the embedding representation of the sentences, words in the sentences, words in the topic labels, and terms of topics in the same vector space.

3.1. Topic labeling

To reduce the cognitive overhead of interpreting the topic discovered by the LDA model, automatic topic labeling has become an effective approach to generate meaningful alternative representations of topics for end-users. In general, there are three forms of topic labels generated by topic labeling methods: phrases, textual summaries, and images. In this study, to focus on generating summarization topic labels (textual summaries), we introduced a novel two-phase neural embedding framework with a redundancy-aware graph-based ranking process.

The existing topic labeling methods [1,16,25] have two distinct processes in common: "sentence scoring" and "sentence selection". The main task of the first process is to calculate the relevance scores between candidate sentences and a given topic, and to find the sentences most relevant to the given topic. If we directly select a few sentences and assemble them into a summarization topic label according to the ranking result (relevance score between candidate sentence and given topic), there may be redundancy in the generated topic label owing to an overlap between the sentences. The working of this process will be described in detail in Section 4.3.

Therefore, assuming a limited length of the generated topic label, the task of the second process is usually to select sentences with higher relevance from those ranked by the relevance score and then assemble them into a topic label with the least redundancy. In other words, the task is to find sentences with the least overlap and higher diversity to generate the summarization topic label. Therefore, this process is frequently called "sentence selection". We present a new method of sentence selection, a stochastic graph-based method with neural embedding, in Section 4.4. According to the ranking results output in the previous process, a transition matrix (TM) is initialized. Based on the similarity between candidate sentences, the control redundancy strategy is integrated into the new graph-based reranking process. Therefore, as the final goal, the graph-based ranking method not only selects the sentences that are most relevant to the topic, but also considers the redundancy of the final generated topic label in the ranking process.

3.2. Topic embedding

We know that the topics discovered by the LDA model are typically represented as a list of top-N terms. Thus, to improve the performance of the topic labeling task, following prior studies [12,28,29], we usually use two types of term embeddings to represent the topics discovered, described as follows:

$$E_{Mean}(T) \;=\; \sum_{w\in T} E_{w2v}(w)P_T(w) \tag{1}$$

Here, T is a discovered topic; $E_{Mean}(T)$ represents the average words embedding of all top-N terms in the topic T; $E_{w2v}(w)$ is the Word2vec embedding of term w; and $P_T(w)$ is the marginal probability of a term w of the topic T. According to Kenter et al. [29], averaging the embeddings of the sentence's constituent words is an efficient way of computing a sentence embedding.

The terms in a discovered topic have no positional relationship among themselves. Therefore, it is a good choice to represent a discovered topic by averaging the embeddings of top-N terms.

$$E_{List}(T) \;=\; \left[E_{w2v}(w_1),\;\cdots\cdots\;,E_{w2v}(w_{|T|})\right] \tag{2}$$

Here, E_{List} (T) represents the vector of the word embeddings of all $|T|$ terms in topic T. Notably, the order of the term embeddings in the list is the same as that of the top-N terms in the given topic.

3.3. Sentence embedding

To represent the sequential text using neural embedding, three different equations are used in different scenes. These equations are described as follows:

$$E_{Mean}(S) \;=\; \frac{1}{|S|}\sum_{w\in S} E_{w2v}(w) \tag{3}$$

Here, E_{Mean} (S) denotes an averaged "word embedding" vector of all words in sentence S [28]. In this study, it was used to compute the Relevance (similarity) between the candidate sentences and discovered topics.

$$E_{List}(TL) \;=\; \left[E_{w2v}(w_1),\;\cdots\cdots\;,E_{w2v}(w_n)\right] \tag{4}$$

Here, TL represents a generated topic label, and E_{List} (TL) is a list of "word embedding" of all words in the TL. In this study, it was frequently used to compute the Relevance between a generated topic label and discovered topic. The TL has n words, i.e., n = $|S|$.

$$E_{Sen}(S) \;=E_{d2v}(S) \tag{5}$$

Here, E_{Sen} (S) represents the "sentence embedding" of a candidate sentence S, which is deduced by equation E_{d2v} (S) of a fine-tuned Doc2vec model.

In this case, we can view the sentence S as a short document, and $E_{d2v}(S)$ outputs the "sentence embedding" of sentence S based on Doc2vec. This equation is usually applied in the graph-based ranking process. It suppresses or enhances the matrix transition probability according to the textual similarity between any vertex (sentence) pair. Refer to Eqs. (18) and (19) for details.

3.4. Relevance between topic and topic label

According to BERTScore [24] and ROUGE-WE [30], the contextualized embedding is trained to effectively capture distant dependencies, ordering, and meanings, etc. The Relevance (similarity) metrics for neural embedding have been shown to correlate very well with human judgments. Therefore, we provide an adapted equation $F_{doc2vec}$ to estimate the Relevance between a discovered topic and a generated topic label. First, the *tfIdf* and the similarity of word embedding are described as follows.

$$tfIdf(w_j) = \frac{[\![w_j \text{ in } TL]\!]}{|TL|}\log\left(\frac{K}{\sum_{i=1}^{K}[\![w_j \in TL_i]\!]}\right) \tag{6}$$

[1] We used Gensim's implementation for doc2vec and word2vec modeling https://radimrehurek.com/gensim/.

[2] We used a pre-trained doc2vec model based on the English Wikipedia articles. https://github.com/sb1992/NETL-Automatic-Topic-Labelling-.

Here, $|TL|$ and w_j stand for the number of words and a word in TL, respectively. As mentioned in Section 3, K represents the number of not only the topics discovered but also the generated topic labels. In this study, $[\![\cdot]\!]$ is an indicator function.

$$Sim(w_i, w_j) = \begin{cases} 0 & \text{if } w_i \text{ or } w_j \text{ are OOV} \\ cosine(E_{w2v}(w_i), E_{w2v}(w_j)) & \text{otherwise} \end{cases} \tag{7}$$

Here, $Sim(w_i,w_j)$ represents the cosine similarity score between two words (w_i and w_j represent a term in the topic T and a word in the generated topic label, respectively), and OOV means a situation where the word w_i or w_j is not involved in the pre-trained neural model [30].

Then, for a discovered topic T and corresponding generated topic label TL, the Recall, Precision, and F1 scores are as under.

$$R_{doc2vec}(T, TL) = \frac{\sum_{w_i \in T} P(w_i) \sum_{w_j \in TL} Sim(w_i, w_j)}{K + \ell} \tag{8}$$

$$P_{doc2vec}(T, TL) = \frac{\sum_{w_j \in TL} tfIdf(w_j) \sum_{w_i \in T} Sim(w_i, w_j)}{K + \ell} \tag{9}$$

$$F_{doc2vec} = 2\frac{P_{doc2vec} \cdot R_{doc2vec}}{P_{doc2vec} + R_{doc2vec}} \tag{10}$$

Here, $R_{doc2vec}(T,TL)$ denotes the Recall measure with Doc2vec embedding, ℓ is a denominator smoothing parameter (refer to Section 5.1), $P_{doc2vec}(T,TL)$ and $F_{doc2vec}$ denote the Precision measure and F1 measure with Doc2vec embedding, respectively, where the latter is used to compute the Relevance between the discovered topic T and generated topic label TL.

4. Method

4.1. Topic reranking

At present, several topic-related studies employ technology to rank the terms of the topics discovered, surface the more important and discriminative terms with the ranking method, and filter the background or unimportant ones. Their research results show that topic reranking can significantly improve coherence and greatly improve the quality of the topics [13,31–33].

To find more representative top topic terms, we reranked the topic terms for each discovered topic by further considering their marginal probabilities over every other discovered topic [33], as shown in Algorithm 1. The experimental results show that this can lead to more representative top topic terms, and improve the quality of "sentence scoring" and the performance of "sentence selection", that is, subsequent topic label generation. For details, see Tables 3 and 4.

Algorithm 1 Topics Terms and Probability Reranking

```
Input: K, TTCount, allTTs, allProbs
Output: allTTsRanked, allProbsRanked
1: allTTsRanked = Matrix[K, TTCount],
   allProbsRanked = zeros_like(allProbs)
2: for i in range(K):
3:     bow = allProbs[i] - allProbs
4:     weight = sqrt(sum(square(bow),axis = 0)) * allProbs[i]
5: weight = weight /sum(weight)
6: order = argsort(weight*(-1))
7: for i in range(K):
8:     allTTsRanked[i] = allTTs[i][order[i]]
9:     allProbsRanked[i] = allProbs[i][order[i]]
10:return allTTsRanked, allProbsRanked
```

In the algorithm, among the input parameters, K is the number of topics discovered on the AP/SIGMOD; *TTCount* is the number of all the topic terms on the AP/SIGMOD; *allTTs* is a matrix [*K*, *TTCount*] of all the topics terms, where K rows represent K topics, and each row *allTTs* [*K*, :] represents a topic term set that includes *TTCount* terms and belongs to this topic; *allProbs* [*K*, *TTCount*] stores the conditional probabilities of all the topic terms of topic K, its total number is *TTCount*. The output parameters *allTTsRanked* and *allProbsRanked* have the same shape as the previously mentioned matrices and store the ranked data. In the fourth line of the algorithm, in the *ith* loop, the bow is a matrix [*K*, *TTCount*] storing the difference of conditional probabilities of each topic term between the *ith* topic and the other *K*-1 topics. The sum (square (*bow*), axis = 0) is a scalar, and can be seen as a matrix [1, *TTCount*], which sums the difference values of *allProbs* [*K*, *TTCount*] in the column direction. The value of its elements indicates the importance of the topic term to this topic; the higher the value, the more important it is to the current *ith* topic. In the sixth line of the algorithm, argsort (*weight**(-1)) returns a descending order.

4.2. Candidate sentences

There are approximately 20,000 and 40,000 sentences in SIGMOD and APNews, respectively. Computing all the sentences one by one will consume huge amounts of memory and computing resources; however, most of the sentences have little relevance with the topic. Wan [1] suggested that most sentences in the corpus are less correlated with the given topic, and the sentences with low relevance with the given topic are not suitable for generating summarization topic labels.

Bigi [34] argues that the Kullback-Leibler divergence (KLD) method outperforms the conventional methods involving the tfidf technique. To find candidate sentences, Wan et al. [1] used KLD to measure the similarity between the discovered topics and all the sentences in the corpus. The equation is presented as follows.

$$KLD(T, S) = \sum_{w \in S \cup T} P_T(w) * \log \frac{P_T(w)}{tf(w, S)/|S|} \tag{11}$$

Here, $KLD(T, S)$ denotes the KLD measured between topic T and sentence S, w is a word belonging to S or T, $tf(w, S)$represents the frequency of w in S, and $|S|$ denotes the word count of S. According to the strategy introduced by Wan et al. [1], if a word w is not in S, the $tf(w, S)/|S|$ would be replaced with 0.00001.

However, the KLD method relies on surface-form similarity only, and fails to capture the semantic similarity correctly between the sentences and discovered topics. Thus, we propose a simple and effective method to select candidate sentences by computing the cosine similarity of the average word neural embedding between the topic and the sentence. It is described by the following equation.

$$Sim_{TSME}(T, S) = cosine(E_{Mean}(T), E_{Mean}(S)) \tag{12}$$

To improve the efficiency of the topic labeling task, we exploited the method proposed by Wan et al. [1] to rank all the sentences in descending order of relevance scores (see Eq. 12). Then we extracted the top 500 sentences that are the most strongly correlated with each discovered topic, and assigned them as candidate sentences sets (*CSSets*).

4.3. Sentence scoring

In this paper, we present an improved version of TLRank [16], that is, a novel two-phase neural embedding framework with the redundancy-aware graph-based ranking process. The first phase called TLRE-C ranks candidate sentences in descending order of the "overall centrality" values, and directly fetches the top sentences into the topic label right before the length limit of the topic label is exceeded. The second phase, called TLRE-G, generates a topic label based on the graph-based ranking scores for each topic discovered in a single ranking and selection process. In other words, the "sentence selection" process integrates into the "sentence scoring" process via a certain suppressed and enhanced strategy.

In the TLRE-C phase, we first identify the three central features of candidate sentences and then amalgamate them into an overall centrality. It is a scalar value and the basis for natural "sentence scoring".

4.3.1. Relevance centrality

The Relevance centrality (*RelCen*) [16] is a measurement based on the textual similarity between the candidate sentence and discovered topic. It is computed using embedding cosine and KLD. *RelCen* can be defined as:

$$RelCen(S,T)=\begin{cases} exp(Sim_{TSME}(T,S)) & \text{embedding cosine} \\ exp(KLD(T,S)^{-1}) & \text{KLD} \end{cases} \tag{13}$$

Here, *T* and *S* represent the topic and candidate sentence, respectively. Choosing the equation to compute the *RelCen* depends on the specific similarity measure adopted by *T* and *S*.

4.3.2. Coverage centrality

It is quite intuitive if a generated topic label has optimal Coverage, then each containing sentence should also cover the topic to a great extent. Besides, for a topic, the sum of the marginal probabilities of different non-repetitive words in the sentence can sufficiently reflect the sentence coverage [35]. Thus, the coverage centrality (*CovCen*) [16] equation for the sentence is defined as follows.

$$CovCen(S,T)=\sum_{w\in S}P_T(w)\ tf(w,S)/|S|^a \tag{14}$$

Here, *a* is an exponential smoothing parameter that can be used to optimize the result. The effect of the length (the number of topic terms) of the discovered topics is ignored because it is fixed at 20 or 500 (refer to Section 5.1). According to the definition in Eq. (14), the *CovCen* of any candidate sentence would be certainly greater than 0.

4.3.3. Discrimination centrality

For a candidate sentence *S*, if it is more important to a discovered topic *T*, the overall distribution ratio of *T* will be greater than that of all other topics. Hence, the discrimination centrality (*DisCen*) [16] equation can be written as follows.

$$DisCen(S,T)=\frac{\sum_{w\in S}P_T(w)\ tf(w,S)}{\sum_{T^*\in U}\sum_{w\in S}P_{T^*}(w)\ tf(w,S)} \tag{15}$$

Here, *DisCen*(*S*, *T*) denotes the difference in degrees of a sentence *S* belonging to a given topic *T* from that belonging to other topics; *U* represents the entire collection of topics discovered, and *w* is a word in the sentence *S*.

4.3.4. Overall centrality

The three centrality features of the candidate sentences, *RelCen*, *CovCen*, and *DisCen*, correspond to the three criteria, that is, Relevance, Coverage, and Discrimination, respectively, for measuring the quality of generated topic labels. The purpose of identifying the three centrality features is to score the candidate sentences accurately. For convenience, we introduced a scalar value overall centrality (*OC*), which combined the three existing centrality features. It helped us in finding more suitable candidate sentences and generated better topic labels. The *OC* equation is defined as follows.

$$OC(S,T)=\alpha RelCen(S,T)+\beta CovCen(S,T)+(1-\alpha-\beta)DisCen(S,T) \tag{16}$$

Here, α and β are proportion parameters to be set empirically, where $\alpha>0,\beta>0,\alpha+\beta<1$.

4.4. Sentence selection

In the TLRE-G phase, we first introduced a stochastic graph-based method with neural embedding and then combined "sentence scoring" and "sentence selection" into a single process to generate a topic label for each discovered topic.

This process is described as follows. First, we built a directed complete graph, where the vertices had a one-to-one correspondence with the sentences in *CSSets*. Second, according to the approach in [16], the most important part of the graph-based algorithm was the establishment of the TM. Finally, we used the graph-based ranking method to rank candidate sentences in *CSSets* for generating topic labels.

4.4.1. Create directed complete graph

Consider $G=(Vertices,Edges)$ to be a directed complete graph. Let $x,y\in Vertices$ represent sentences S_x and S_y, respectively, and $edge_{xy}\in Edges$ is an edge from point × to *y*. The weights of the *edge* are a critical factor to the creation of the TM, for example, $edge_{xy}$ determines the voting rate from vertex × to y in the ranking process. Therefore, it is feasible to adjust the ranking order of sentences by modifying the weight of each edge in the graph. For $edge^*{}_y$ which represents the edges from any point × to *y*, the initial voting rates come from the overall centrality of vertex *y* (sentence S_y), that is, $edge_{\cdot y}$ = $OC(S_y,T)$, which means that the initial values of each column of matrix TM are the same. Therefore, the graph needs to fine-tune each TM_{xy} using suppressed or enhanced strategies and can generate topic labels with the least redundancy and the highest diversity.

4.4.2. Suppress strategy

To restrict the redundancy of the generated topic label, TM is built according to the overall centrality value and then tuned with a specific strategy. The strategy is simple and effective for a sentence pair. Their similarity determines how far apart they are in the ranked queue; that is, more similar sentences should be far apart from each other.

In a directed complete graph, $edge_{xy}$ and $edge_{yx}$ are opposite sides, where $edge_{xy}$ represents the voting rate from vertex *x* to *y*. The suppression of the direction edge ($edge_{xy}$ or $edge_{yx}$) is determined by the vertex with the smaller overall centrality, that is, if the vertex *y* has a lower overall centrality than *x* , it needs to suppress $edge_{xy}$ according to the similarity degree between the vertices, and vice versa. This is described in detail in Eq. (18).

Here, we use the cosine similarity of sentence neural embedding to measure the similarity of any vertex pair (sentence pair) in the graph. In general, for any two sentences S_x and S_y, the equation is defined as:

$$Sim_{VOG}(S_x,S_y)=cosine(E_{sen}(S_x),E_{sen}(S_y)) \tag{17}$$

where $Sim_{VOG}(S_x,S_y)$ represents the similarity between the vertex pairs (sentence pairs) of the graph and $E_{sen}(S_x)$ and $E_{sen}(S_y)$ denote the "sentence embedding" of the candidate sentences S_x and S_y, respectively. Refer to Eq. (5) for details.

If any vertex × (S_x) and y (S_y) in the graph are similar, and $OC(S_x, T) > OC(S_{y,}T)$, then the $edge_{xy}$ (not $edge_{yx}$) should be suppressed, and the equation can be described as follows.

$$edge_{xy} = edge_{xy}/e^{Sim_{VOG}(S_x,S_y)} \tag{18}$$

4.4.3. Enhance strategy

If the directed $edge_{xy}$ pointing to the weak vertex y is suppressed, should the opposite side $edge_{yx}$ be enhanced? It is not necessary, because only the highly weighted vertices with broad representation have the privilege of receiving more votes.

A threshold t is set if the similarity between two vertices surpasses t and the degree of both vertices is increased by 1 [36]. If a vertex y has a higher degree, it has stronger representation, such that the $edge_{\cdot y}$ that represents a set of edges pointing to vertex y should be enhanced. In this case, TLRE-G will choose the corresponding sentence S_y preferentially to generate the topic label. The equation is shown as follows.

$$edge_{\cdot y} = edge_{\cdot y}\, r^{\left(\mathrm{degree}_y/\sum_{x\in CSSets}\mathrm{degree}_x\right)^a} \tag{19}$$

Here, $\mathrm{degree}_\mathrm{y}$ denotes the degree of vertex y, $\mathrm{edge}_{\cdot\ \mathrm{y}}$ represents a set of edges pointing to vertex y, $\sum_{x\in CSSets}\mathrm{degree}_x$ is the sum of the degrees of all vertices in graph G, and γ represents the cardinality parameter set according to experience. Besides, if $degree_y$is zero, the value of $edge_{\cdot y}$ remains the same.

4.4.4. Graph-based ranking progressing

According to Eqs. (13)–(19), we know that $edge_{xy} > 0$ and $edge_{xy} \neq edge_{yx}$ in any condition. Thus, we can create a weight-directed complete graph and obtain global optimal results every time. We, hereby, use the graph-based ranking method [16] to regularize the sentence scores.

Moreover, we provide a context-sensitive TM-based sentence similarity of neural embedding. According to Erkan et al. [36], the TM must be a positive matrix, that is, an irreducible and aperiodic Markov chain. Consequently, our graph-based algorithm converges to a stable state within a finite iteration. Thus, it further improves the performance of graph-based ranking processing and acquires better topic labels with high Relevance, Coverage, and Discrimination.

5. Experiment

5.1. Experiment setup

Based on previous research [1,6,16], we used two different public document collections: SIGMOD[3] and APNews[4]. Several pretreatments were applied, including removing punctuations and a few unnecessary stop words, and filtering out elements (other than nouns, verbs, adjectives, adverbs, and pronouns). Consequently, we achieved a total of 3016 documents, 22,195 sentences, and 10,378 vocabularies in the SIGMOD and 2246 documents, 40,766 sentences, and 26,777 vocabularies in APNews.

In this study, all methods were implemented using Python. We used Gensim's implementation[5] to discover topics with the LDA model. The parameters were either directly borrowed from prior studies or empirically set; for example, to discover the topics, it is necessary to fix the topic number $K = 25$ [1], and set the parameters num_topics = 25, random_state = 10, update_every = 1, chunksize = 100, passes = 20, alpha = 'auto', iterations = 50, gamma_thresh old = 0.001, decay = 0.5, offset = 1.0, eval_every = 10, and per_word_ topics = True. The other parameters in our experiment were set as follows: a = 0.85, α = 0.3, β = 0.4, γ = 1.75, ℓ = 3, t = 0.2, and m = 3.

To get a concise and meaningful summarization topic label, following the recommended length of the summary in DUC conferences, we set its length to 100 words [37,38].

5.2. Comparison labeling methods

For convenience, we considered *CSSets* as the input to all labeling methods and all the compared methods are listed below.

5.2.1. Baseline

To generate the topic label, we directly fetched the top sentences in *CSSets* without considering redundancy. This is recognized as the baseline method.

5.2.2. LexRank

The LexRank[6] creates a graph based on the candidate sentences and votes equally to related vertices. The method is based on a graph-based automatic summarization method, which was proposed by Erkan et al. [36] for the first time.

5.2.3. TextRank

TextRank[7] is regarded as an improved version of LexRank. The vertex no longer votes equally, instead it votes according to the ratio of their similarities [39]. For LexRank and TextRank, if *CSSets* is viewed as a single document, the topic labeling task can be transformed into a single document summary task.

5.2.4. Submodular

Wan et al. [1] proposed a text summary method, named Submodular, based on submodular optimization for topic labeling tasks. The method scores and selects sentences based on a greedy algorithm in the topic labeling process.

5.2.5. TLRank and TLRE

Based on surface features computing, He et al. [16] proposed a method for automatic topic labeling using graph-based ranking. The two phases of the topic labeling process in this method are called TLRank-C and TLRank-G. Base on neural embedding, we propose an improved version, called TLRE, which offers a significant improvement.

6. Results and discussion

In this section, inspired by the BERTScore method, we propose a new automatic metric that displays a high correlation with human judgments and does not rely on perfect matching, which can improve the quality of graph ranking [1,6,16]. Compared with TLRank, in the first phase, our method does not just rely on handcrafted features when computing the relevance between the given topic and sentences. Instead, we use the neural embeddings of topic T and sentence S by word2vec to compute $RelCen(S, T)$ (see Eq. 13) and $OC(S, T)$ (see Eq. 16) to obtain higher quality *CSSets*. In the second phase, we utilized the neural embedding combined with handcrafted features to optimize the definition of *RelCen* and *OC* for candidate sentences, which made TLRE-C surpass TLRank-C. In addition, according to *OC* and the similarity between

[3] A total of 3016 SIGMOD summaries (1975–2018) downloaded from ACM DL, https://dl.acm.org/citation.cfm?id=500080&picked=prox.

[4] A total of 2246 APNews articles downloaded from GitHub, https://github.com/Blei-Lab/lda-c/blob/master/example/ap.tgz.

[5] https://radimrehurek.com/gensim/.

[6] LexRank method implemented by lexrank (Python tools package) https://github.com/crabcamp/lexrank.

[7] TextRank method implemented by summa-textrank (Python tools package) https://github.com/summanlp/textrank.

sentences computed by the neural embeddings obtained by the doc2vec, the graph-based ranking method determined whether to enhance or suppress the voting rate to a node in the graph, and that was the reason why TLRE-G surpassed TLRank-G in most aspects. Besides, the topics reranking further improved the quality of the generated topic label which we will demonstrate and discuss in detail below. Moreover, unless noted otherwise, the discovered topic was by default represented by the top20 reranked topic terms (see Section 6.2), and the length of topic label was limited to 100 words.

6.1. Evaluation of labeling methods

According to the evaluation metrics framework proposed by Mei et al. [6] and Wan et al. [1], a good topic label should satisfy three critical criteria: Relevance, Coverage, and Discrimination. At present, it is still a comprehensive and advantageous choice to evaluate the performance of topic labeling tasks. In this section, we present the evaluation of the different labeling methods with automatic measures in case of SIGMOD and APNews.

6.1.1. Relevance

It is well known that a topic label, with a higher semantic Relevance to its topic, conveys more accurate meaning to the user. Instead of exact matches of surface features, we computed textual similarity using neural embedding in our work. This way, the generated topic label had higher Relevance with the corresponding topic. Zhang et al. [24] have proved that BERTScore (an automatic evaluation metric for neural embedding text generation) correlates better with human judgment than the existing metrics. To adapt to the changes brought by neural embedding and realize a more objective and impartial evaluation, we applied the $F_{doc2vec}$ (refer to Eq. 10) to estimate the Relevance between the discovered topics and generated topic labels.

6.1.2. Coverage

According to the study of Wan et al. [1], Coverage is defined as the ratio of words that appeared in the top20 topic terms, and the equation is as follows. It is important to note that $UNI(S_i)$ returns a set of unique characters of sentence S_i.

$$Coverage = \frac{\sum_{i=1}^{K}(\sum_{w \in UNI(S_i)}[\![w \in Top20(T_i)]\!]/|Top20(T_i)|)}{K} \tag{20}$$

As both the topic terms and sentence words are represented by neural embedding, the same meaning can be expressed by different words and even the same word has a different meaning in a different sentence. As a result, it seems that Coverage is no longer needed to evaluate the generated topic labels. However, in this study, encouraging generated topic labels to incorporate as many top20 terms as possible is an important reason why it was still used to evaluate labeling methods.

6.1.3. Discrimination

It is meaningless if a generated topic label has high Relevance with several topics simultaneously, because we cannot use one topic label to simultaneously interpret multiple topics. Thus, we used a measure, Discrimination [1], to evaluate the topic labeling models by calculating the degree of differentiation between generated topic labels. First, for the given K topics, we generated corresponding K topic labels; next, we computed the similarity value between any two topic labels; finally, we averaged all the similarity values as Discrimination. The equation of Discrimination is described as follows.

$$\begin{aligned} TLsims &= \sum_{i=1}^{K}\sum_{j=i+1}^{K-1} cosine(E_{sen}(TL_i), E_{sen}(TL_j))_{i \neq j} \\ Discrimination &= TLsims/|TLsims| \end{aligned} \tag{21}$$

We compared the performance of TLRE with other labeling methods in Relevance, Coverage, and Discrimination in SIGMOD and APNews. The detailed results are as follows.

For each labeling method, the averaged Relevance, Coverage, and Discrimination scores for all discovered topics in the two different corpora are shown in Tables 1 and 2.

Our approach outperforms others in terms of Relevance and Discrimination, but it ranks third in Coverage. The main reason is that TLRE, represented by neural embedding, pays more attention to the semantic features of sentences and words. It selects other more explanatory words with the same semantic meaning to replace the top20 words, which may ultimately promote instead of weaken the meaning and interpretation of the topics discovered.

Table 1
Relevance, Coverage, and Discrimination of generated topic labels in SIGMOD.

Approach	Relevance	Coverage	Discrimination
Baseline	0.6808	0.1880	0.3159
LexRank	0.6871	0.1820	0.3119
TextRank	0.5250	**0.2780**	0.3474
Submodular	0.8899	0.2300	0.3713
TLRE	**0.9050**	0.2100	**0.3783**

Table 2
Relevance, Coverage, and Discrimination of generated topic labels in APNews.

Approach	Relevance	Coverage	Discrimination
Baseline	0.6804	0.2220	0.2931
LexRank	0.6813	0.1960	0.2573
TextRank	0.4030	**0.2840**	0.2971
Submodular	0.8754	0.2680	0.3278
TLRE	**0.8886**	0.2220	**0.3305**

6.2. Reranking topics

It is known that the topic discovered is usually in a top-N terms list with the highest marginal probabilities. According to Chi et al. [33], we reranked the topic terms by further considering marginal probabilities on terms of every other topic. It filtered out background words from topic discovered and further widened the gap of the marginal probability of topic terms. The experimental results are presented below.

Figs. 1 and 2 reveal that after ranking the topics, the average marginal probability of top1 term increased sharply, for example, from 0.276 to 0.626 in case of SIGMOD. The average marginal probability of top5 terms was close to 90%, and the average and minimum marginal probability of top20 terms both exceeded over 0.9995.

Therefore, it can be concluded that after topic reranking, the marginal probabilities increase for a few top topic terms, while they decrease for others, and the gaps between them expand rapidly. The top20 terms can almost cover the topic, and their representational capacity is not different from top500.

Further, we investigated the impact on the topic labeling task to know whether or not the topics were reranked and the discovered topics were represented by the top20 or top500 terms.

Table 3 reveals the evaluation results of automatic topic labeling methods using surface features under different conditions. For convenience, according to whether the discovered topics are

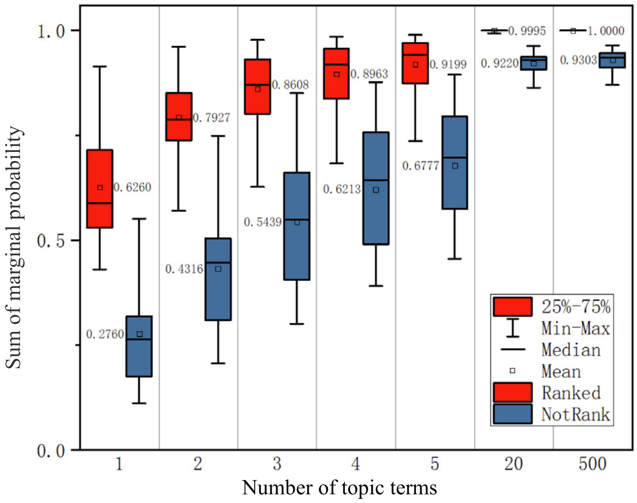

Fig. 1. For all discovered topics in the SIGMOD, the lowest, highest, average, second-quartile, median, and third-quartile of the marginal probability sum of the top 1, 2, 3, 4, 5, 20, and 500 terms. The red and blue boxes denote the ranked and non-ranked topics, respectively.

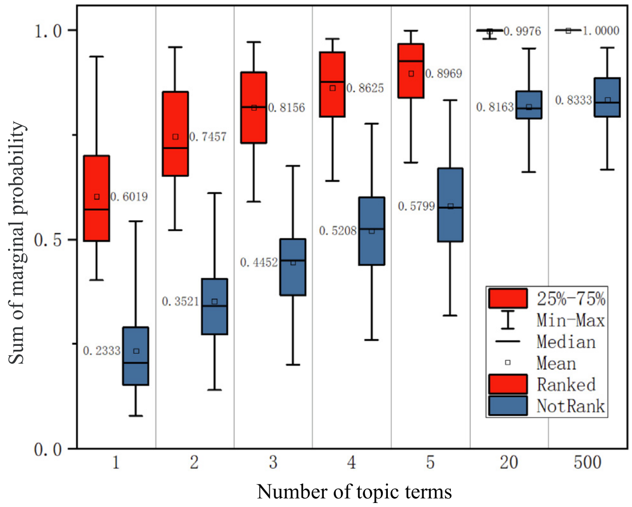

Fig. 2. For all discovered topics in APNews, the lowest, highest, average, second-quartile, median, and third-quartile of the marginal probability sum of the top 1, 2, 3, 4, 5, 20, and 500 terms. The red and blue boxes denote the ranked and non-ranked topics, respectively.

represented by top20 or top500 terms, we added a digital suffix to the method name to distinguish it in different conditions. For example, TLRank-G-20 denotes the method that will work with a discovered topic represented by top20 topic terms.

In this study, we propose a novel neural embedding framework with a graph-based ranking. Our method is in contrast with TLRank. Table 4 lists the evaluation results of our automatic topic labeling method (TLRE) under different conditions.

As depicted in Tables 3 and 4, whichever the method used, "ranked" topics always generated topic labels with higher Relevance and Discrimination scores compared with "non-ranked" topics.

However, this rule seemed to be inapplicable to Coverage. All "ranked" generated topic labels were much lesser than the "non-ranked" ones. The reason may be that a ranked topic always has fewer topic terms with higher marginal probability; nevertheless, the probabilities of most terms are very small and the gap between the two types grows even larger. As shown in Figs. 1 and 2, the sum of marginal probabilities of the top5 terms of a ranked topic increases from 50 to 60% to 90%. It shows that the top5 topic terms closely represent the entire topic. Therefore, the model prefers to choose sentences that contain top5 terms more than those that contain top6-20. Obviously, the Coverage of topic labels generated by ranked topic declines significantly.

As evident from Figs. 1 and 2, fewer (20 instead of 500) terms can be used to represent the discovered topic without decreasing its quality. The experimental results in Tables 3 and 4 show that the difference is almost negligible between the top20 and top500 term topics. Especially, after introducing high-dimensional neural embedding to represent topic terms and sentences, reranking and representing the topics with top20 terms provide us another apparent advantage. It fundamentally reduces the workload of computing neural embedding.

Besides, comparing the Coverage between Tables 3 and 4, we find evidence that the Coverage of topic labels generated by the neural embedding framework significantly declines. This indicates that the neural embedding method has a higher Relevance and Discrimination, but lower Coverage. This is consistent with the previous conclusions according to Tables 1 and 2 (refer to Section 6.1).

6.3. TLRE-C vs. TLRE-G

As mentioned in the Section 4, TLRE has two phases: TLRE-C and TLRE-G. In the first phase, TLRE-C identifies the overall centrality score for each candidate sentence to directly generate a topic label for each topic. Then in the next phase, TLRE-G first creates a TM according to the TLRE-C score (overall centrality value) to rule the voting process. Then, it applies a specified strategy to suppress or enhance a certain graph voting rate. Finally, it runs a graph-based ranking process to generate a topic label.

6.3.1. Improvement of TLRE-G

According to Table 5, in the case of SIGMOD and APNews, the suppression and enhancement strategies of TLRE-G are reasonable and effective, and the improvement in the results of TLRE-C is sufficient and significant.

Interestingly, in Table 5, TLRE-G displays an unexpected Coverage value (0.21) in the case of APNews, which is lower than that of TLRE-C (0.22). The specific reasons are further explored below.

Table 3
Evaluation of topic labeling methods in APNews based on surface features for a discovered topic in different conditions (whether or not a topic was ranked and was represented by top20 or top500 terms).

	Topic ranked			Topic not ranked		
Surface feature method	Relevance	Coverage	Discrimina-tion	Relevance	Coverage	Discrimina-tion
Baseline-20	0.3826	0.1160	0.2519	0.3564	0.1160	0.2519
TLRank-C-20	0.7199	0.0780	0.3756	0.6487	0.0760	0.3798
TLRank-G-20	0.7229	0.0780	0.3889	0.6486	0.0740	0.3882
Baseline-500	0.4633	0.1600	0.2862	0.3996	0.2420	0.2725
TLRank-C-500	0.8006	0.1280	0.3926	0.7032	0.1380	0.3886
TLRank-G-500	0.8017	0.1260	0.3988	0.7026	0.1560	0.3998

Table 4
Evaluation of topic labeling methods in APNews based on neural embedding for a discovered topic in different conditions (whether or not a topic was ranked and was represented by top20 or top500 terms).

	Topic ranked			Topic not ranked		
Neural embedding method	Relevance	Coverage	Discrimina-tion	Relevance	Coverage	Discrimina-tion
Baseline-20	0.6808	0.1880	0.3159	0.6163	0.1860	0.3159
TLRE-C-20	0.9049	0.2200	0.3694	0.7829	0.2620	0.3704
TLRE-G-20	0.9050	0.2100	0.3783	0.7856	0.2560	0.3763
Baseline-500	0.6817	0.1880	0.3152	0.6172	0.1860	0.3152
TLRE-C-500	0.9048	0.2200	0.3688	0.7825	0.2620	0.3705
TLRE-G-500	0.9048	0.2100	0.3775	0.7848	0.2720	0.3765

Table 5
TLRE-G vs. TLRE-C in terms of Relevance, Coverage, and Discrimination.

		Relevance	Coverage	Discrimination
SIGMOD	TLRE-C	0.8720	0.2180	0.3205
	TLRE-G	**0.8886**	**0.2220**	**0.3305**
APNews	TLRE-C	0.9049	**0.2200**	0.3694
	TLRE-G	**0.9050**	0.2100	**0.3783**

We denoted the topic labels generated by TLRE-C and TLRE-G as label-c and label-g, respectively. Table 6 shows the statistical details of the top20 topic terms contained in both the generated topic labels. In case of SIGMOD, even though the ATCT20 value of label-g is lower than that of label-c, the former has a higher ACAT20 value (4.44) than the latter (4.36). It is also evident that the Coverage of label-g is higher than that of label-c (refer to Table 5). In case of APNews, both the ATCT20 and ACAT20 values of label-g are lower than those of label-c. Obviously, this is the reason why the former's Coverage is lower than the latter.

All the evidence above proves that TLRE-G prefers topic terms with higher marginal probability. The same trend is also depicted in Tables 3 and 4, as the gap in the marginal probabilities of the ranked top20 terms increases sharply. Consequently, label-g will contain more leadership terms, as described below.

6.3.2. Coverage measurement

To further illustrate the Coverage details of the topic label, we gathered certain correlated data in APNews. Here, the total top20 terms contained in the label-g and label-c of each topic are shown in Fig. 3, and the number of categories of top20 terms in the label-g and label-c are shown in Fig. 4. For brevity, we have retained only five topics with the different number of categories of top20 terms in label-g or label-c, for example, topics 11,12,18,19, and 24 (see Fig. 4).

We numbered the 25 discovered topics as topic 1 to topic 25. Topic 12 was selected as an example to investigate the effects of the top20 topic terms on Coverage. The topic labels generated by TLRE-G and TLRE-C denote label-g12 and label-c12. As shown in Fig. 3, the label-g12 and label-c12 have 19 and 20 top20 terms, respectively. According to Fig. 4, the numbers of categories of the top20 terms in label-g12 and label-c12 are 4 and 5, respectively. The detailed data are listed in Table 7 below.

Table 6
Average number of top20 terms (ATCT20) and average number of categories of top20 terms (ACAT20) contained in the topic label

	SIGMOD		APNews	
	label-g	label-c	label-g	label-c
ATCT20	21.32	22.32	21.6	22.6
ACAT20	4.44	4.36	4.2	4.4

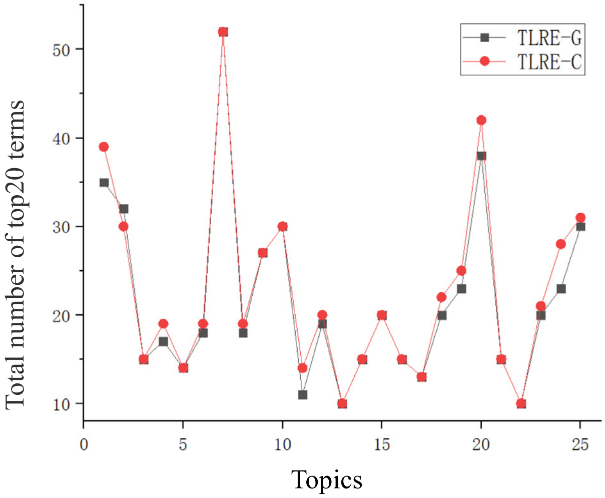

Fig. 3. Total number of top20 terms contained in label-g and label-c for each topic.

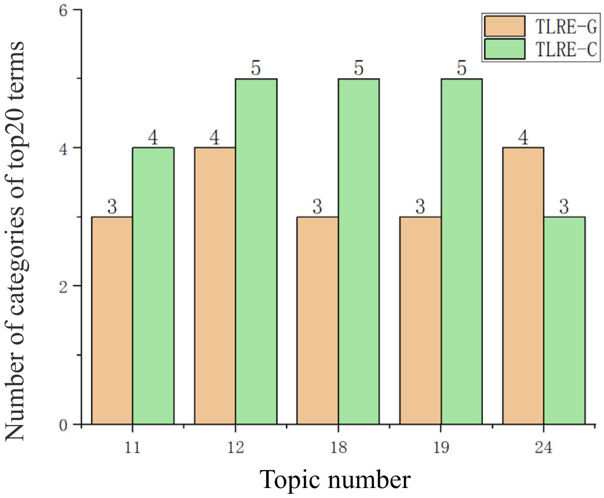

Fig. 4. Uneven number of categories of top20 terms in label-g and label-c for five topics.

Table 7
Number of top20 terms in label-g12 and label-c12.

	Work	Think	Monday	Never	Early
Marginal probability	0.4393	0.1549	0.1489	0.0138	9.02E-07
Label-g12	**12**	5	0	1	1
Label-c12	11	**6**	**1**	1	1

As can be seen from Table 7, compared with label-c12, label-g12 has lesser top20 topic terms and their categories. However, representative words (higher marginal probability) appear more frequently in label-g12 than in label-c12. In a more intuitive form, for label-g12 or label-c12, we summed the results by multiplying each top20 term probability by its amount in the label, and obtained 6.061 and 5.926 as the values of label-g12 and label-c12, respectively, as shown in Fig. 5. It shows that TLRE-G prefers topic terms with stronger representativeness, especially when the top20 topic terms have higher probability values after the reranking of the topics, and this is the reason why TLRE-G performs better.

6.3.3. nDCG measurement

To further illustrate the improvement effect of TLRE-G on TLRE-C, we used normalized discounted cumulative gain (nDCG) [12,40] to evaluate the effectiveness of the graph-based ranking process.

For each topic, TLRE-C and TLRE-G first create the corresponding sentence lists based on the ranking score and then choose sentences sequentially to generate the topic label according to limited length (the default value is 100). We employed three human annotators to rate the sentence lists for each topic, and the average score of manual evaluation for each sentence became the final gold standard. To show the improvement of TLRE-G relative to TLRE-C, based on calibration relative to the manual gold standard, we computed the nDCG for the top-2 (nDCG-2), top-4 (nDCG-4), and top-6 (nDCG-6) ranked topic labels. The results are presented in Table 8.

It is imperative to mention that the boldface figures in Table 8 represent TLRE-G, which demonstrates better results than TLRE-C, and their absolute difference is greater than 0.05. It means that the ranking sequence of TLRE-G is closer to the manual sequence than that of TLRE-C. This is also consistent with the perspective of Table 5. As the APNews corpus is relatively divergent and human ratings are conservative, the manual scores of the sentences involved in label-g and label-c are not ideal.

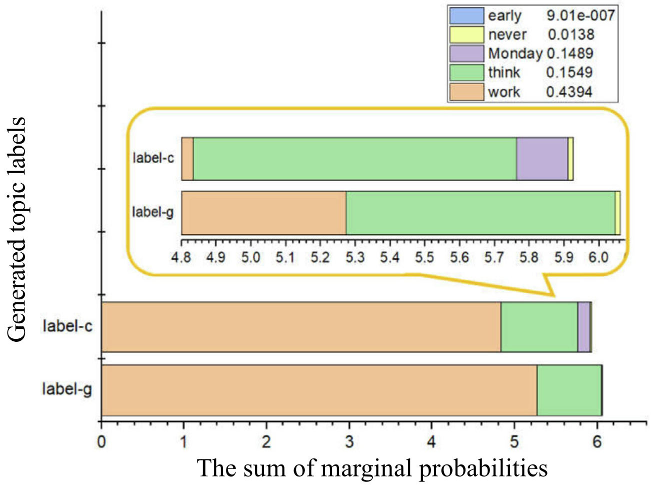

Fig. 5. For topic 12, the sum of marginal probabilities for all top20 terms in label-g or label-c in the case of APNews.

Table 8
Comparison of ranking performance with TLRE-G and TLRE-C

		nDCG-2	nDCG-4	nDCG-6
SIGMOD	TLRE-C	0.727020	0.813647	0.854963
	TLRE-G	**0.839653**	**0.884084**	**0.918552**
APNews	TLRE-C	0.674477	0.738645	0.768860
	TLRE-G	**0.807168**	**0.849254**	**0.883487**

However, generally, the nDCG scores (nDCG-2, nDCG-4, nDCG-6) of the TLRE-C and TLRE-G in the two corpora SIGMOD and APNews are relatively lower, indicating that there is still a huge gap in the abstractive ability between our proposed method and humans.

6.4. Example analysis

According to Section 6.2, it is known that the top20 terms can represent the topic in high fidelity. We demonstrate the top20 terms of a topic discovered in the case of SIGMOD (Table 9). However, it is still confusing to read only these topic terms, as in such a case, we cannot understand the exact content of the topic, only the most probable. Fortunately, by reading the summarization topic label, we can really understand the meaning of the topics. Therefore, two generated topic label examples, generated by Baseline and TLRE, are shown below. It should be noted that after preprocessing the corpus, all generated topic labels and the candidate sentences in the *CSSets* do not contain any stop words and numbers. Therefore, for the sake of smooth and coherent reading, we used the original sentences that contained the stop words and numbers to form the two examples of topic labels, which are longer than 100 characters (178 words).

Topic label example 1 (label-TLRE): VerdictDB exploits a **novel** technique for error estimation called variational subsampling, which is amenable to **efficient** computation via SQL. We introduce a **novel** moving object indexing technique that together with a **novel** road network partitioning **scheme** restricts computations within the partial road network. It employs a **novel** action-aware indexing **scheme** that exploits users interaction characteristics with visual interfaces to support **efficient** retrieval. To further enable instant exploration, we devise a **novel** index **structure** and develop **effective** pruning and materialization techniques. Our **novel search** strategy is coupled with a number of optimization techniques to further prune the **search** space and **efficient**ly maintain the lattice. In this context, we introduce a **novel** cost framework that allows for the application of techniques from record-linkage to the **search** for good repairs. We then study a **novel** indexing **structure** called SI-**tree**, which combines both signature and length filtering strategies, for **efficient** string similarity

Table 9
Top20 terms of a given topic in SIGMOD.

Index	Topic terms
1–10	**Novel**, **structure**, **efficient**, find, execution, **tree**, node, **search**, constraint, **scheme**
11–20	Control, possible, processor, **effect**, actually, impose, **graph**, demonstrate, **memory**, dataset

Table 10
Comparison of label-TLRE and label-Baseline.

		Label-TLRE		Label-Baseline	
Topic term	Marginal probability	Number of topic terms	Sum of marginal probabilities	Number of topic terms	Sum of marginal probabilities
Novel	0.4301967	9	3.87177	7	3.01138
Structure	0.1392892	2	0.27858	2	0.27858
Efficient	0.0574821	5	0.28741	7	0.40237
Tree	0.0519811	1	0.05198	3	0.15594
Search	0.0426962	4	0.17078	3	0.12809
Scheme	0.0263727	2	0.05275	0	0
Effect	0.0019657	1	0.00197	0	0
Graph	7.78E-07	1	7.8E-07	0	0
Memory	4.87E-07	0	0	1	4.9E-07
		25 (8 categories)	4.71524	23 (6 categories)	3.97636

joins with synonyms. In this paper, we present an **efficient** and robust sub-graph **search** solution, called TurboISO, which is turbo-charged with two **novel** concepts, candidate region exploration.

Topic label example 2 (label-Baseline): Particularly, we propose **efficient search** algorithms together with **novel** indexes to speed up the processing of TPMFP. This paper introduces a **novel** problem called the best region **search** (BRS) problem and provides **efficient** solutions to it. In this paper, we propose an **efficient** data **structure** called QC-**tree** and an **efficient** algorithm for directly constructing it from a base table, solving the first problem. We propose a **novel** approach to performing **efficient** similarity **search** and classification in high dimensional data. The present paper proposes a **novel**, R*-**tree** based indexing technique that supports the **efficient** querying of the current and projected future positions of such moving objects. We identify key performance bottlenecks and tradeoffs, and propose **novel** techniques to make these holistic UDAFs fast and space-**efficient** for use in high-speed data stream applications. In this paper, we present a **novel** index **structure** to speed up processing of high-dimensional K-nearest neighbor (KNN) queries in main **memory** environment. In this paper, we introduce Tributary-Delta, a **novel** approach that combines the advantages of the **tree** and multi-path approaches by running…

After observing the label-TLRE (generated by our model, TLRE) and label-Baseline (generated by Baseline), we found that top1-10 topic terms are more preferable than top11-20 ones (6 vs. 3, bold terms in Table 9). Compared with label-Baseline, label-TLRE had more top20 topic terms (25 vs. 23), more top1-10 topic terms (23 vs. 22), more top11-20 topic terms (2 vs. 1), more categories of topic terms (8 vs. 6), and a higher sum of marginal probabilities of words belonging to top20 topic terms (4.71524 vs. 3.97636). Besides, in the top11-20 topic terms, the marginal probability of "memory" in label-Baseline is much lower than "effect" and "graph" in label-TLRE. The details are shown in Table 10. It is obvious after observing the statistical results in Table 10 or reading the summary text of the topic that label-TLRE performs better. It not only contains higher total number of top20 topic terms, but also has more categories of different top20 topic terms. Therefore, the content of label-TLRE is more comprehensive and rich, and has better relevance and diversity.

7. Conclusion and future work

To improve the performance of topic labeling tasks, we proposed a novel topic labeling model based on pre-trained neural embedding with a redundancy-aware graph-based ranking process. The model is named TLRE, which aims to generate a topic label for each topic modeled by LDA to help the user understand it clearly. The experimental evaluation results demonstrate that TLRE significantly and consistently outperforms the prevailing state-of-the-art and classic models of topic labeling tasks. In future research, we will try to use BERT (or GPT-3) to obtain the contextual features to represent sentences, and exploit multi-head self-attention to jointly model that relationship between candidate sentences, the correlation between candidate sentences and given topic, and the redundancy between candidate sentences and generated topic labels. It can provide a more accurate semantic understanding of sentences and better redundancy control over the topic label, further improving the performance of topic labeling tasks.

CRediT authorship contribution statement

Dongbin He: Conceptualization, Methodology, Software, Writing – original draft. **Yanzhao Ren:** Data curation, Writing - review & editing, Validation. **Abdul Mateen Khattak:** Investigation, Writing - review & editing, Validation. **Xinliang Liu:** Methodology, Investigation, Supervision. **Sha Tao:** Formal analysis, Resources, Writing - review & editing. **Wanlin Gao:** Conceptualization, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the Project of Scientific Operating Expenses, Ministry of Education of China, under Grant 2017PT19. The authors highly acknowledge and appreciate this support.

References

[1] X. Wan, T. Wang, Automatic labeling of topic models using text summaries, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016, pp. 2297–2305.
[2] X. Wang, A. McCallum, X. Wei, Topical n-grams: Phrase and topic discovery, with an application to information retrieval, Seventh IEEE international conference on data mining (ICDM), IEEE 2007 (2007) 697–702.
[3] A. Alokaili, N. Aletras, M. Stevenson, Automatic Generation of Topic Labels, in: 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1965–1968.
[4] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Machine Learn. Res. 3 (2003) 993–1022.
[5] D.M. Blei, J.D. Lafferty, Topic models, Text mining: classification, clustering, and applications, 10 (2009) 34.
[6] Q. Mei, X. Shen, C. Zhai, Automatic labeling of multinomial topic models, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 490–499.
[7] D. Magatti, S. Calegari, D. Ciucci, F. Stella, Automatic labeling of topics, in: 2009 Ninth International Conference on Intelligent Systems Design and Applications, 2009, pp. 1227–1232.

[8] J.H. Lau, K. Grieser, D. Newman, T. Baldwin, Automatic labelling of topic models, ACL (2011) 1536–1545.

[9] I. Hulpus, C. Hayes, M. Karnstedt, D. Greene, Unsupervised graph-based topic labelling using dbpedia, in: Proceedings of the sixth ACM international conference on Web search and data mining, 2013, pp. 465–474.

[10] N. Aletras, M. Stevenson, Labelling topics using unsupervised graph-based methods, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 631–636.

[11] W. Kou, F. Li, T. Baldwin, Automatic Labelling of Topic Models Using Word Vectors and Letter Trigram Vectors, Springer, AIRS, 2015, pp. 253–264.

[12] S. Bhatia, J.H. Lau, T. Baldwin, Automatic labelling of topics with neural embeddings, The 26th International Conference on Computational Linguistics, 2016, pp. 953–963.

[13] A. Alokaili, N. Aletras, M. Stevenson, Re-ranking words to improve interpretability of automatically generated topics, in: Proceedings of the 13th International Conference on Computational Semantics, 2019, pp. 43–54.

[14] A.E.C. Basave, Y. He, R. Xu, Automatic labelling of topic models learned from twitter by summarization, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 618–624.

[15] M.H. Barawi, C. Lin, A. Siddharthan, Automatically labelling sentiment-bearing topics with descriptive sentence labels, International Conference on Applications of Natural Language to Information Systems, Springer, 2017, pp. 299-312.

[16] D. He, M. Wang, A.M. Khattak, L. Zhang, W. Gao, Automatic labeling of topic models using graph-based ranking, IEEE Access 7 (2019) 131593–131608.

[17] N. Aletras, A. Mittal, Labeling topics with images using a neural network, Eur. Conf. Inform. Retrieval Springer (2017) 500–505.

[18] N. Aletras, M. Stevenson, Representing topics using images, Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 158-167.

[19] I. Sorodoc, J.H. Lau, N. Aletras, T. Baldwin, Multimodal topic labelling, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2017, pp. 701–706.

[20] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C.M. Meyer, S. Eger, MoverScore text generation evaluating with contextualized embeddings and earth mover distance, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 563–578.

[21] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, International Conference on Learning Representations, 2013.

[22] Q. Le, T. Mikolov, Distributed representations of sentences and documents, Int. Conf. Machine Learn. (2014) 1188–1196.

[23] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT, Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 4171–4186.

[24] T. Zhang, V. Kishore, F. Wu, K.Q. Weinberger, Y. Artzi, BERTScore: Evaluating Text Generation with BERT, Eighth International Conference on Learning Representations, 2019.

[25] E. Baralis, L. Cagliero, N. Mahoto, A. Fiori, GRAPHSUM: discovering correlations among multiple terms for graph-based summarization, INFORM Sciences 249 (2013) 96–109.

[26] N.A. Sanjaya, M.L. Ba, T. Abdessalem, S. Bressan, Harnessing truth discovery algorithms on the topic labelling problem, in: Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services, 2018, pp. 8–14.

[27] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Adv. Neural Inform. Process. Syst. (2013) 3111–3119.

[28] M. Peyrard, T. Botschen, I. Gurevych, Learning to Score System Summaries for Better Content Selection Evaluation, in: Proceedings of the Workshop on New Frontiers in Summarization, 2017, pp. 74–84.

[29] T. Kenter, A. Borisov, M. De Rijke, Siamese cbow: Optimizing word embeddings for sentence representations, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016, pp. 941–951.

[30] J. Ng, V. Abrecht, Better summarization evaluation with word embeddings for rouge, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1925–1930.

[31] M. Taddy, On estimation and selection for topic models, Artificial Intell. Statist. Conf. PMLR (2012) 1184–1193.

[32] J. Chuang, C.D. Manning, J. Heer, ermite: visualization techniques for assessing textual topic models, in: Proceedings of the international working conference on advanced visual interfaces, 2012, pp. 74–77.

[33] J. Chi, J. Ouyang, C. Li, X. Dong, X. Li, X. Wang, Topic representation: Finding more representative words in topic models, PATTERN RECOGN LETT 123 (2019) 53–60.

[34] B. Bigi, Using Kullback-Leibler distance for text categorization, Eur. Conf. Inform. Retrieval Springer (2003) 305–319.

[35] R. Arora, B. Ravindran, Latent dirichlet allocation based multi-document summarization, Proceedings of the second workshop on Analytics for noisy unstructured text data, ACM, 2008, pp. 91-97.

[36] G. Erkan, D.R. Radev, Lexrank: Graph-based lexical centrality as salience in text summarization, J. Artificial Intell. Res. 22 (2004) 457–479.

[37] P. Ren, F. Wei, C. Zhumin, M.A. Jun, M. Zhou, A redundancy-aware sentence regression framework for extractive summarization, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, 2016, pp. 33–43.

[38] P. Ren, Z. Chen, Z. Ren, F. Wei, L. Nie, J. Ma, M. De Rijke, Sentence relations for extractive summarization with deep neural networks, ACM Trans. Inform. Syst. (TOIS) 36 (2018) 39.

[39] R. Mihalcea, P. Tarau, Textrank: Bringing order into text, Proceedings of the 2004 conference on empirical methods in natural language processing, ACL, 2004, pp. 404-411.

[40] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inform. Syst. (TOIS) 20 (2002) 422–446.

Dongbin He received the B.S. degree in Computer and its Application from Hebei University of Technology, Tianjin, China, in 1996, and the M.S. degree in Computer Software and Theory from Inner Mongolia University, Hohhot, China, in 2006. He is currently pursuing the Ph.D. degree at the College of Information and Electrical Engineering, China Agricultural University, Beijing, China, and he is an associate professor at the College of Computer Science and Engineering, Shijiazhuang University. His research interests include Internet of Things technologies, Data Mining and Natural Language Processing.

Yanzhao Ren, Post Doctorate. He obtained his master of Engineering in Agricultural Machine Engineering from Shandong University of Technology, Zibo, China in 2012. He is currently pursuing his PhD in Agricultural Electrification and Automation at China Agricultural University, Beijing. His research direction is computer application technology and intelligent information processing. His research subjects includes agricultural IoT, automatic operating equipment, intelligent video projection equipment.

Abdul Mateen Khattak received the Ph.D. degree in horticulture and landscape from the University of Reading, U.K., in 1999. He was a Research Scientist in different agriculture research organizations before joining Agricultural University Peshawar, Pakistan, where he is currently a Professor with considerable experience in teaching and research at the Department of Horticulture. He has conducted academic and applied research on different aspects of tropical fruits, vegetables, and ornamental plants. He was also with Alberta Agriculture and Forestry, Canada, as a Research Associate, and with the Organic Agriculture Centre of Canada as a Research and Extension Coordinator (for Alberta province). There he helped in developing organic standards for greenhouse production and energy saving technologies for Alberta greenhouses. He is also a Visiting Professor with the College of Information and Electrical Engineering, China Agricultural University, Beijing. He has published 55 research articles in scientific journals of international repute. His research interests include greenhouse production, medicinal, aromatic and ornamental plants, light quality, supplemental lighting and temperature effects on greenhouse crops, aquaponics, and organic production.

Xinliang Liu, Ph.D. candidate, in computer science, College of Information and Electrical Engineering, China Agricultural University. He also is an associate professor in the School of E-Commerce and Logistics, Beijing Technology and Business University. His current research interests include Knowledge graph, cross-media retrieval, and Web information processing.

Sha Tao received the Ph.D. degree from College of Food Science & Nutritional Engineering, China Agricultural University, in 2013. From 2014 to 2016, she was a Post-Doctoral Research Fellow with the Key Laboratory of Agricultural Informatization Standardization, Ministry of Agriculture (Prof. Wanlin Gao Laboratory), College of Information and Electrical Engineering, China Agricultural University, where she is currently a lecturer in this college. Her research mainly focuses on processing and storage of agriculture products technologies.

Wanlin Gao received the B.S., M.S., and Ph.D. degrees from China Agricultural University, in 1990, 2000, and 2010, respectively. He is a Principal Investigator of over 20 national plans and projects, a member of the Science and Technology Committee of the Ministry of Agriculture, a member of the Agriculture and Forestry Committee of Computer Basic Education in Colleges and Universities, a Senior Member of the Society of Chinese Agricultural Engineering, and so on. He has published 90 academic papers in domestic and foreign journals, and among them, over 40 are cited by SCI/EI/ISTP. He has written two teaching materials, which are supported by the National Key Technology R&D Program of China during the 11th Five-Year Plan Period, and has written five monographs. Moreover, he holds 101 software copyrights, 11 patents for inventions, and 8 patents for new practical inventions. His research interests include the informationization of new rural areas, intelligence agriculture, and the service for rural comprehensive information.