he_2021_automatic_topic_labeling_using_graph_based_pre_trained_neural_embedding

Year

2021

Author(s)

Dongbin He and Yanzhao Ren and Abdul Mateen Khattak and Xinliang Liu and Sha Tao and Wanlin Gao

Title

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

Venue

Neurocomputing

Topic labeling

Fully automated

Focus

Primary

Type of contribution

Novel

Underlying technique

graph-based pre-trained neural embedding

Topic labeling parameters

Summary length: 100 words

Label generation

We applied a Doc2vec model trained with English Wikipedia literature to generate static neural embedding for sentences, words, and topic terms.
In this study, to focus on generating sum- marization 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 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.
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.

We present a new method of sentence selection, a stochastic graph-based method with neural embedding.
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.

Topic embedding

We usually use two types of term embeddings to represent the topics discovered, described as follows:

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

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 PT (w) is the marginal probability of a term w of the topic T.

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.

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

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.

Sentence embedding

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

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

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

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

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|.

$$E_{Sen}(S) = E_{d2v}(S)$$

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.

Relevance between topic and topic label

We provide an adapted equation Fdoc2vec to estimate the Relevance between a discovered topic and a generated topic label.

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

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

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

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

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

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

F1 measure is used to compute the Relevance between the discovered topic T and generated topic label TL.

Method

Topic reranking

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, as shown in Algorithm 1

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*

Candidate sentences

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.

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

Sentence scoring

In this paper, we present an improved version of TLRank, 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".

Relevance centrality

The Relevance centrality (RelCen) 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:

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

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. Thus, the coverage centrality (CovCen) equation for the sentence is defined as follows.

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

Here, a is an exponential smoothing parameter that can be used to optimize the result.

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 (Dis-Cen) equation can be written as follows.

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

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.

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

Here, a and b are proportion parameters to be set empirically, where a > 0; b > 0; a + b < 1.

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.

1. First, we built a directed complete graph, where the vertices had a one-to-one correspondence with the sentences in CSSets.
2. Second, the most important part of the graph-based algorithm was the establishment of the TM.
3. Finally, we used the graph-based ranking method to rank candidate sentences in CSSets for generating topic labels.

[TO FINISH...]

Motivation

It is necessary to reduce the cognitive overhead of interpreting the native topic term list of the Latent Dirichlet Allocation (LDA) style topic model.

Topic modeling

LDA

Topic modeling parameters

Nr of topics (k): 25

random_state = 10,

update_every = 1,

chunk- size = 100,

passes = 20,

alpha = ’auto’,

iterations = 50,

gamma_thresh old = 0.001,

decay = 0.5,

offset = 1.0,
eval_every = 10,
per_word_ topics = True.

a = 0.85,
alpha = 0.3,
beta = 0.4,
gamme= 1.75,
l = 3,
t = 0.2
m = 3

Nr. of topics

25

Label

Summary of up to 100 words

Label selection

Label quality evaluation

Assessors

Domain

Paper:

Dataset:

Problem statement

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 use- fully 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.

Corpus

SIGMOD

Origin: ACM DL

Nr. of documents: 3016

Details:

Origin: APNews

Nr. of documents: 2246

Details:

Document

Pre-processing

- removing punctuations and a stop words
- filtering out elements (other than nouns, verbs, adjectives, adverbs, and pronouns)

```
@article{he_2021_automatic_topic_labeling_using_graph_based_pre_trained_neural_e
mbedding,
    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.},
    author = {Dongbin He and Yanzhao Ren and Abdul {Mateen Khattak} and Xinliang Liu and Sha Tao and Wanlin Gao},
    date-added = {2023-04-27 19:52:25 +0200},
    date-modified = {2023-04-27 19:52:25 +0200},
    doi = {https://doi.org/10.1016/j.neucom.2021.08.078},
    issn = {0925-2312},
    journal = {Neurocomputing},
    keywords = {Automatic labeling, Neural embedding, Graph-based ranking, Topic model, Topic label, Latent dirichlet allocation (LDA)},
    pages = {596-608},
    title = {Automatic topic labeling using graph-based pre-trained neural embedding},
    url = {https://www.sciencedirect.com/science/article/pii/S0925231221012686},
    volume = {463},
    year = {2021}}
```

#Thesis/Papers/FS