Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

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

Topic-informed neural approach for biomedical event extraction

Junchi Zhang[a], Mengchi Liu[a,*], Yue Zhang[b,c]

[a] *Computer School, Wuhan University, Wuhan, Hubei, China*
[b] *School of Engineering, Westlake University, Hangzhou, Zhejiang, China*
[c] *Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou, Zhejiang, China*

ARTICLE INFO

Keywords:
Neural topic model
Variational inference
Biomedical event extraction
Neural network

ABSTRACT

As a crucial step of biological event extraction, event trigger identification has attracted much attention in recent years. Deep representation methods, which have the superiorities of less feature engineering and end-to-end training, show better performance than statistical methods. While most deep learning methods have been done on sentence-level event extraction, there are few works taking document context into account, losing potentially informative knowledge that is beneficial for trigger detection. In this paper, we propose a variational neural approach for biomedical event extraction, which can take advantage of latent topics underlying documents. By adopting a joint modeling manner of topics and events, our model is able to produce more meaningful and event-indicative words compare to prior topic models. In addition, we introduce a language model embeddings to capture context-dependent features. Experimental results show that our approach outperforms various baselines in a commonly used multi-level event extraction corpus.

1. Introduction

With the rapid growth of biomedical text, there have been surge of interests in the development of biomedical information extraction techniques. Biomedical event detection (BEE) is one crucial task in the construction of biomedical knowledge base and ontology, which facilitates following researches of biomedical science [1]. The goal of BEE is to identify event triggers of specified types and their arguments in text. Event triggers are generally nominalizations or verbs as the key words evoking the corresponding events and arguments are entities connecting triggers with particular relation types. For example, in Table 1, there is a target sentence drawn from multi-level event extraction (MLEE) corpus [2], which includes a component entity “Reactive oxygen species” and a *Synthesis* event mention, both triggered by the word “produced”. Identifying this event is challenging for a BEE system, because the same trigger word could present different event types in a different context.

Biomedical event corpus are typically annotated in document context. However, existing studies mainly focus on developing sentence-level event extraction system. Both statistical methods and neural representation-based methods have been used. The former mainly rely on kernel classification methods such as support vector machines [2–4] with hand-crafted features, which require domain-specific knowledge and feature engineering effort. In contrast, deep neural network methods explore distributed representation to capture meaningful semantic information [5–7].

Intuitively, the broader document-level context potentially contains a more informative description of the main topics that a document talk about. For humans, if we cannot figure out the meaning of an expression or make sure the idea conveyed in a limited context, we may try to read more description in a wider document context to understand the meaning. Such cases are indeed prominent in the biomedical domain, in the sense that proper words and compound words occur more often than in the News domains. Under this observation, we argue that machines can also take advantage of document-level context. For instance, if we only examine the target sentence in Table 1 alone, it is hard to determine whether “produced” triggers a *Positive_regulation* event, which is defined as a process that increases the frequency, rate or extent of gene expression, or whether it refers to a process of decomposition. On the other hand, if we read the surrounding sentences or the whole article and find it to be a production story of component ROS, it is more confident to tag “produced” as a *Synthesis* event.

Upon such observation, there have been attempts [8–10] that construct heuristic rules to capture cross-sentence information. These approaches often require off-the-shelf NLP tools to connect multiple sentences (e.g., coreference resolution, dependency tree), which are prone to involving propagated errors. Zhou and Zhong [11] alternatively exploit hidden topics of a sentence as distance features to improve BEE

[*] Corresponding author at: Computer School, Wuhan University, Wuhan, Hubei, China.
E-mail addresses: zjc.whu@gmail.com (J. Zhang), mengchiliu1@gmail.com (M. Liu).

https://doi.org/10.1016/j.artmed.2019.101783
Received 16 September 2019; Received in revised form 25 November 2019; Accepted 26 December 2019
0933-3657/

Table 1
A sample target sentence (T) and its surrounding sentences (S_1-S_3) drawn from a biomedical document. Trigger words and their event types are marked in underlined. Words in black bold font are named entities. This table shows that surrounding sentences provide more informative description of compound entity ROS.

Target sentence

T: **Reactive oxygen species** (**ROS**) [**produced**]$_{Synthesis}$ in the course of cellular oxidative phosphorylation...

Surrounding sentences

S_1: The [**excessive**]$_{Positive\_regulation}$ [**production**]$_{Synthesis}$ of **ROS** can [**damage**]$_{Catabolism}$ protein, **lipids**, **nucleic acids**, and **matrix components**.

S_2: **Oxygen** [**metabolism**]$_{Metabolism}$ has an important role in the pathogenesis of rheumatoid arthritis.

S_3: They also serve as important intracellular signaling molecules...

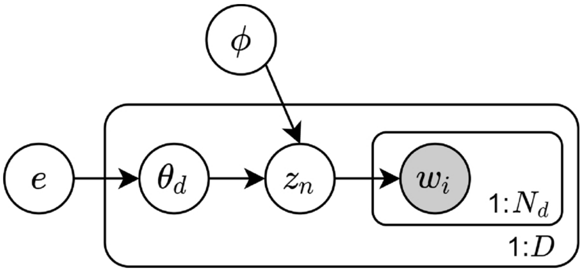

Fig. 1. Probabilistic graph representation of LDA in a plate diagram. We assume that there are D documents in a biological corpus, and each document contains N_d words. e is the prior of the Dirichlet distribution. For latent topic proportion θ_d, LDA samples a word-level topic z_n conditioned on a topic-word mixture distribution ϕ. Finally, each word w_i is generated based on the topic vector z_n.

performance. However, sentence-level topics are limited and suffer from word sparsity. In addition, topics infered by Latent Dirichlet Allocation (LDA) [12] model are fixed when training downstream tasks. Being not specific in event extraction, this two-stage procedure is hard to take advantage of joint training of topic models and BEE.

To tackle the issues mentioned above, we propose a novel neural framework, named *topic informed neural model* (TINM) for biomedical event trigger detection. TINM is capable of identifying topic words, practically indicative words for BEE, e.g., “produced” in T, via jointly exploiting the document-level word co-occurrence patterns, such as “damage” and “production” in S_1. In addition, considering that many triggers have only a few training instances that probably influence the classification performance of neural methods, topic vectors can sever as knowledge inferred from documents which are beneficial for alleviating the data sparsity issue [3].

Previous work have shown that document-level latent topics are helpful for other tasks [13,14]. The usefulness of neural topic models, nevertheless, have not been explored in existing event extraction research, particularly in the biomedical domain. Our model is built upon the success of variational neural topic models [15,16], enabling end-to-end training of latent topic modeling and trigger detection. Experimental results show that our model is superior to various baselines. The quantitative and qualitative analysis reveals the capability of our model in inferring coherence topic words that are meaningful and specific for BEE.

2. Related work

Our work mainly follows the line of two prior work: event trigger detection and topic modeling.

2.1. Event trigger detection

In general domain, existing work could broadly be categorized into two areas. The first is statistical methods, which explore various features with SVM model [2–4]. In contrast to sentence-level event detection, [8] applies heuristic rules to model document information flow. Huang and Rilo [17] proposes a bottom-up architecture integrating textual cohesion properties into event extraction. They also observe that a view of a larger context is beneficial for modeling the relevances of entities and events. Yang and Mitchell [10] further improves statistical method by considering cross-sentence event interactions with the graph model. The second area includes representation methods. Most of these work are based on CNN [18], RNN [6], enhanced with attention mechanisms [5,7], or extend them with hierarchical attention over document context [19]. Distinguished from them, we propose to use document-level latent topic representations where topics are informed jointly with trigger detection, resulting in an end-to-end training.

2.2. Topic model

Traditional LDA [12] is a hierarchical probabilistic model and it is widely used in downstream applications, such as information retrieval [20]. For event trigger detection, latent topics have been introduced as a distance metric between sentences, aiming at including more training instances [11]. However, they perform the topic model on sentence-level, and thus hard to capture domain-general topics. The closest work to ours is [21] that attempts to use topic features to improve event extraction performance on standard News corpus. Despite the simple formalism of their approach, it is difficult to integrate their sparse topic features into a neural model. Recently, with the emergence of variational autoencoding (VAE) [22], posterior distribution can be approximated directly in neural networks. Therefore, neural topic models (NTM) [15,16] have showed superior in perplexity and normalized pointwise mutual information (NPMI) score over LDA. Different from existing work, we study the effectiveness of VAE-style topic models that learned together with trigger detection, whose effect and interpretability in the biomedical domain is the focus of this work.

3. Backgound: Latent Dirichlet allocation (LDA)

Here, we briefly introduce the basic knowledge of LDA [12]. Formally, given a collection $\mathcal{D}$ with $|D|$ biological documents $\{d_1, d_2, ..., d_{|D|}\}$, LDA assume each document d_i is represented as a mixture of topics $\phi = (\phi_1 ... \phi_K)$, where each topic ϕ_k is a probability distribution over the vocabulary $\mathcal{V}$. Accordingly, the generative process of LDA can be described in Algorithm 1. Latent variables θ_d and z_n represent the topic proportion of d, and the topic assignment for the observed word w_n, respectively. e is the hyper-parameter of the Dirichlet prior. By representing LDA in a probabilistic graph view (Fig. 1), we can write the marginal likelihood of document d as:

$$p(d|e, \phi) = \int_{\theta} p(\theta|e) \prod_{n} \sum_{z_n} p(w_n|\phi_{z_n}) p(z_n|\theta) d\theta \tag{1}$$

However, direct optimization of Eq. (1) is intractable due to the coupling between the θ and ϕ under the multinomial assumption [23].

Algorithm 1. Generative process of LDA.

```
Input: collection $\mathcal{D}$, prior $e$, randomly initialized topic-word matrix $\phi$.
for each document $d$ in $\mathcal{D}$ do
    Draw topic distribution $\theta_d \sim \text{Dirichlet}(e)$;
    for each word $w_n$ in $d$ do
        Sample topic $z_n \sim \text{Multinomial}(1, \theta_d)$;
        Sample word $w_n \sim \text{Multinomial}(1, \phi_{z_n})$;
    end
end
```

3.1. Variational inference (VI)

As alternative to Gibbs sampling methods in traditional LDA [12]. Variational inference [24] approximate an intractable posterior distribution $p(d, \theta|e, \phi)$ with a tractable variational lower bound (also called evidence lower bound (ELBO)):

$$\log p(d|e, \phi) = \int_{\theta} q(\theta|d) \log p(d|e, \phi) d\theta \tag{2}$$

$$= \mathbb{E}_{q(\theta|d)} \log \frac{p(d, \theta|e, \phi)}{q(\theta|d)} + D_{\mathrm{KL}}(q(\theta|d) || p(\theta|d, e, \phi)) \tag{3}$$

$$\geq \mathbb{E}_{q(\theta|d)} \log \frac{p(d, \theta|e, \phi)}{q(\theta|d)} (\text{ELBO}) \tag{4}$$

where $q(\theta|d)$ is a tractable variational distribution (e.g., Dirichlet distribution), D_{KL} is the KL divergence:

$$D_{\mathrm{KL}}(P||Q) = \sum_{i} P(i) \log \frac{P(i)}{Q(i)} \tag{5}$$

By integrating Eq. (1), ELBO over document d can be expanded as:

$$\text{ELBO} = \mathbb{E}_{q(\theta|d)} \log p(d|\theta, e, \phi) + D_{\mathrm{KL}}(q(\theta|d) || p(\theta|e, \phi)) \tag{6}$$

Intuitively, the first term in ELBO can be thought of as a reconstruction loss, ensuring that generated words are similar to the original document. The second term, the KL divergence, encourages the variational approximation to be close to the assumed prior $p(\theta)$. For LDA, this optimization has closed form coordinate descent equations due to the conjugacy between the Dirichlet and multinomial distributions.

Despite the effectiveness of VI-based LDA [25], it is difficult to adjust latent topics to be specific and suitable for event extraction due to the unsupervised training process. In addition, the high dimensionality of LDA hinders it from being used in deep neural models. To overcome above issues, we then introduce a neural version of LDA that can be trained jointly with the event model (Section 4.1).

4. Methods

In this section, we present our topic informed trigger extraction framework. The overall architecture is shown in Fig. 2. There are two major modules, including (1) a document-level neural topic model (NTM), shown on the left of the figure, which aims to capture long-range latent topics across documents and (2) a trigger detection module, shown on the right of the figure, which produces tagging sequence for each local input sentence with designated topic informed. These two components can be updated simultaneously via a joint learning process, which is introduced in Section 4.3.

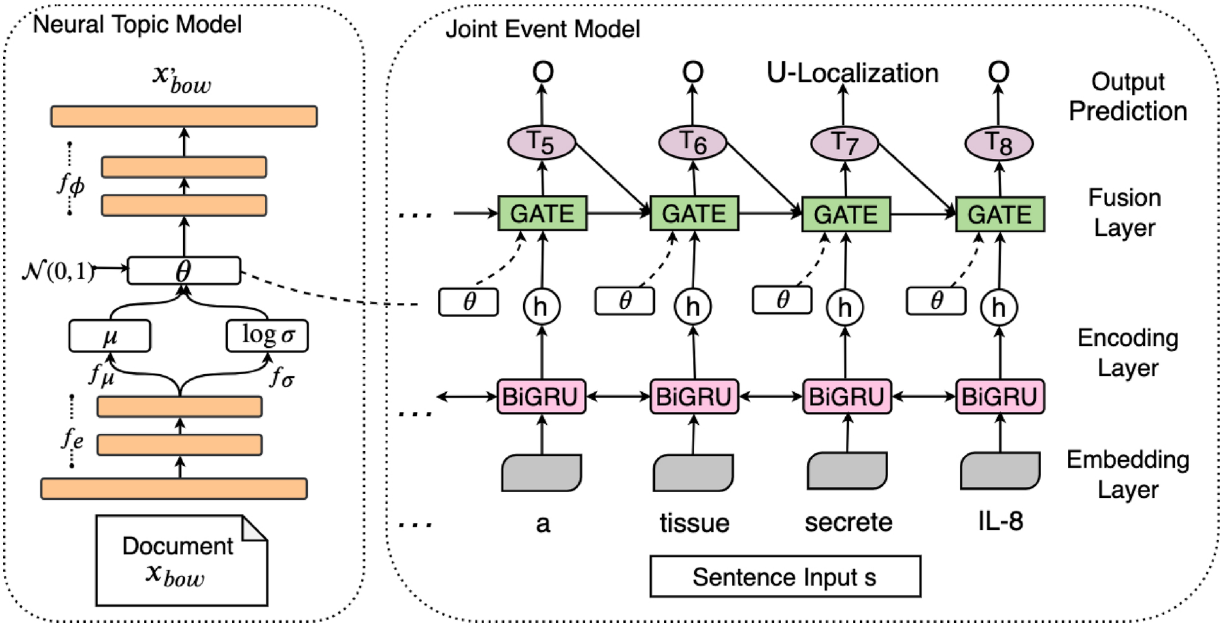

Fig. 2. Network structure of our joint model. Left is the neural topic model and right is the trigger detection model.

4.1. Neural topic model (NTM)

Our neural topic model, based on variational autoencoder (VAE) [22] and neural document modeling [16], consists of an encoder network and a decoder network to resemble the document reconstruction process. In particular, we represent each document $\mathbf{x}$ into a bag-of-words (BoW) term $\mathbf{x}_{\text{bow}}$, which is a one-hot vector over the vocabulary $\mathcal{V}_{\text{bow}}$ and is fed into the encoder network to obtain the continuous latent vector $z \in \mathbb{R}^K$ (where K denotes the number of topics). We take $p(\mathbf{z})$ as a Gaussian distribution $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$,

4.1.1. Bag of words encoder

In LDA, it assumes that the variational posterior $q(\theta|d)$ is a Dirichlet distribution and the potential complex posterior is thus simplified in real applications. To better model $q(\theta|d)$, we borrow the approximation power of neural networks. In NTM, the posterior distribution $q(\mathbf{z}|\mathbf{x}_{\text{BoW}})$ is estimated by adopting an amortized variational inference (AVI) [26]. In particular, the variational parameters of distribution z, namely the mean μ and standard deviation σ, are estimated from the input document with three feed-forward networks (FFN):

$$\mu = f_\mu(f_e(\mathbf{x}_{\text{BoW}})), \quad \log\sigma = f_\sigma(f_e(\mathbf{x}_{\text{BoW}})) \tag{7}$$

where $f_*(x)$ is a multi-layer perceptron that linearly transforms inputs, activated by a non-linear transformation:

$$f_*(x) = \delta(\mathbf{W}x + \mathbf{b}) \tag{8}$$

here $\mathbf{W}$ is the matrix parameter of the MLP and $\mathbf{b}$ is the bias vector, δ is an activation function which we adopt rectified linear units (ReLUs) [27].

4.1.2. Bag of words decoder

Similar to topic models in LDA, the aim of the decoder network is to reconstruct original document $\mathbf{x}_{\text{BoW}}$ with vector $\mathbf{z}$ as input. Therefore, the reconstruction likelihood $p(d|\theta, e, \phi)$ in Eq. (6) is replaced by $p(\mathbf{x}_{\text{BoW}}|\mathbf{z})$ in NTM. As presented in Algorithm 2, we use a feed-forward network f_θ and softmax function to transform a Gaussian random vector $\mathbf{z}$ into a multinomial topic distribution vector θ_d. Then an output network $f_\phi(\theta_d)$ is used to project θ_d to the vocabulary space, and each word w_n can be generated individually with the probabilistic softmax function. Note that weight matrix of $f_\phi(\theta_d)$ is the topic-word distributions $\phi = (\phi_1 \ldots \phi_K)$. In the next section, we adopt the topic mixture θ_d as the topic representations to enhance event trigger detection.

Algorithm 2. Generative process of the neural topic model.

```
Input: collection $\mathcal{D}$
for each document  d in $\mathcal{D}$ do
    Draw latent topic variable $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$;
    Topic distribution $\theta_d = softmax(f_\theta(\mathbf{z}))$;
    for each word $w_n$ in $d$ do
        Draw word $w_n \sim softmax(f_\phi(\theta))$;
    end
end
```

4.1.3. Object function

With above definition and replace the prior distribution of $p(\theta|e, \phi)$ to be $p(\mathbf{z})$, the variational objective function (ELBO) of NTM can be constructed as:

$$\mathcal{L}_{\text{NTM}} = D_{\text{KL}}(p(\mathbf{z})||q(\mathbf{z}|\mathbf{x}_{\text{BoW}})) - \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_{\text{BoW}})}[p(\mathbf{x}_{\text{BoW}}|\mathbf{z})], \tag{9}$$

where the first term is the Kullback-Leibler divergence loss, which encourages the variational approximation to be close to the assumed prior $p(\mathbf{z})$. We take $p(\mathbf{z})$ as a spherical Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, $q(\mathbf{z}|\mathbf{x}_{\text{BoW}})$ and $p(\mathbf{x}_{\text{BoW}}|\mathbf{z})$ are probabilities to describe inference (encode network) and reconstruction process (decode network), respectively. Using the reparameterization trick [22], we replace the expectation with a single-sample approximation, so that:

$$\mathbf{z} = \mu + \sigma \odot \epsilon \tag{10}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is sampled from an independent Gaussian distribution. All parameters derived from the networks $f_\mu, f_e, f_\sigma, f_\theta, f_\phi$ can then be optimized simultaneously by performing stochastic gradient descent on the variational objective function in Eq. (9).

4.2. Neural event extraction model

To predict the trigger words and their event types, we treat event detection as a sequence labeling problem. Specifically, document d is splitted into sentence segments with Stanford CoreNLP [1] and each sentence is tokenized to N words $\{w_1, w_2, \ldots, w_N\}$. We then map each word w_i in a sentence into labels under the BILOU-* scheme, where B/I/L/O indicates the **B**egin, **I**nside, **E**nd and **O**utside of a trigger, respectively, and **U** indicates a single word trigger. * is the event type added along with BILOU tags.

To integrate topic information into the event model, we use θ_d (learned by NTM) as latent topic representations to capture document-level semantics for event extraction.

4.2.1. Input representation

For each word w_i, we use three types of continuous vectors as input features:

Static word embeddings such as word2vec [28] using a word co-occurrence training strategy on a large amount of raw texts, aims to capture semantic relationships between words in a distributed fashion. To reduce domain discrepancy, we employ BioWordVec[2] as input features, which is trained on PubMed text and MIMIC-III Clinical Database.[3] The shortage of static word embeddings is that words are always mapped to the same vector in spite of their context words in practice.

Contextualized word embeddings, in contrast, compute a representation for a target word based on the particular context that the words presented within a sentence. We use Bidirectional Encoder Representations from Transformers (BERT) [29], the base model consists of 12 layers of multi-head self-attention networks. BERT use a predefined wordpiece vocabulary which means that one word may correspond to multiple subword units. As a result, the input sentence s is first tokenized into word pieces by using BERT tokenizer.[4] Then, to match the next sentence prediction strategy of BERT, special tokens [CLS] and

[1] https://stanfordnlp.github.io/CoreNLP/.

[2] https://github.com/ncbi-nlp/BioSentVec.

[3] https://physionet.org/works/MIMICIIIClinicalDatabase/access.shtml.

[4] https://github.com/google-research/bert/blob/master/tokenization.py.

[SEP] are added to the start and end positions of the tokenized input sequence. In order to effectively use multiple hidden states of BERT, we obtain the word embedding $\mathbf{v}_i^{\mathrm{BERT}}$ by applying a mean pooling function to its subword embeddings.

Entity type embeddings are important features for trigger detection, because a candidate event type is often related to several particular types of entities. We obtain the entity type embedding $\mathbf{v}_i^{\mathrm{ENT}}$ by converting gold annotated discrete entity types (under BILOU scheme) into low-dimensional vectors with a random initialized matrix. We use a zero-vector to indicate word w_i is not part of an entity. Note that the embedding matrix is fine-tuned along with model training.

Finally, the overall word representation is the concatenation of:

$$\mathbf{v}_i = [\mathbf{v}_i^{\mathrm{STATIC}}; \mathbf{v}_i^{\mathrm{BERT}}; \mathbf{v}_i^{\mathrm{ENT}}] \tag{11}$$

4.2.2. Sequence encoder

By converting input sentences into dense representation $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$, we employ a bidirectional gated recurrent unit (Bi-GRU) as the encoder. Each word representation $\mathbf{v}_i$ is mapped into forward and backward hidden states (denoted as $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}_i}$, respectively) with the following defined operations:

$$\overrightarrow{\mathbf{h}}_i = f_{\mathrm{GRU}}(\mathbf{v}_i, \mathbf{h}_{i-1}), \tag{12}$$

$$\overleftarrow{\mathbf{h}_i} = f_{\mathrm{GRU}}(\mathbf{v}_i, \mathbf{h}_{i+1}). \tag{13}$$

where $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}_i}$ are the previous step and next step hidden vectors, respectively. The forward and backward representations are concatenated to serves as a bi-directional representation of w_i, $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}_i}]$. We omit the detailed the description of Eqs. (12) and (13) for simplicity.

4.2.3. Topic-aware gated decoder

Our model incorporates topical information by assimilating the document-topic representation ($\theta_{\mathbf{d}}$) with a concatenation operating for each word in the input sentence s. In particular, we adopt a GATE unit similar to a GRU to allow hidden states $\mathbf{h}_i$ to learn the degree of influence of topical information on the trigger detection model:

$$\begin{aligned}
\mathbf{z}_i &= \sigma(\mathbf{W}_z\theta_{\mathbf{d}} + \mathbf{U}_z\mathbf{h}_i + \mathbf{V}_z y_{i-1} + \mathbf{b}_z)\\
\mathbf{r}_i &= \sigma(\mathbf{W}_r\theta_{\mathbf{d}} + \mathbf{U}_r + \mathbf{V}_z y_{i-1} + \mathbf{b}_r)\\
\hat{\mathbf{h}}_i &= \tanh(\mathbf{W}_h\theta_{\mathbf{d}} + \mathbf{U}_h(\mathbf{r}_i \odot \mathbf{h}_i) + \mathbf{b}_h)\\
\mathbf{h}_i' &= (1 - \mathbf{z}_i) \odot \mathbf{h}_i + \mathbf{z}_i \odot \hat{\mathbf{h}}_i
\end{aligned} \tag{14}$$

where $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{V}$ are learnable parameters of the GATE unit. $\mathbf{z}_t$ and $\mathbf{r}_t$ encode the update and reset activations, respectively, at timestep i. y_{i-1} is the predicted tag of the word w_{i-1} ensuring that the output tags sequence obeys the validity of BILOU tagging scheme (such that B-* followed by U is not allowed). The new hidden state $\mathbf{h}_t'$ is connected to a dense layer with linear transformation and softmax output to predict the label of the current word:

$$P(y_i|w_i) = \mathrm{softmax}(\mathrm{Relu}(\mathbf{W}_y\mathbf{h}_i' + \mathbf{b}_y))) \tag{15}$$

where $\mathbf{W}_y$ and $\mathbf{b}_y$ are model parameters.

4.3. Joint learning of topics and events

We treat the trigger detection and topic models as subtasks in a multi-task learning setting, and train both in a joint manner. For the trigger detection model, we minimize the cross-entropy loss over all training instances:

$$\mathcal{L}_{\mathrm{TD}} = -\sum_{m=1}^{M}\sum_{i=1}^{N}\log(P(y_i|w_i)) \tag{16}$$

where M denotes the number of training instances, N is the sentence length. Note that for NTM, we optimize the negative lower bound $\mathcal{L}_{\mathrm{NTM}}$ in Eq. (9). Finally, we define the loss function of the overall framework by combining the trigger detection and neural topic model loss functions:

Table 2
Statistics of event types in MLEE corpus.

Category	Event type	Total count
Anatomical	Cell_proliferation	125
	Development	300
	Blood_Vessel_Development	890
	Death	93
	Breakdown	67
	Remodeling	32
	Growth	163
Molecular	Synthesis	17
	Gene_Expression	342
	Transcription	23
	Catabolism	24
	Phosphorylation	29
	Dephosphorylation	3
General	Localization	415
	Binding	158
	Regulation	540
	Positive_Regulation	966
	Negative_Regulation	683
Planned	Planned_Process	582

$$\mathcal{L} = \mathcal{L}_{\mathrm{TD}} + \lambda\mathcal{L}_{\mathrm{NTM}} \tag{17}$$

where λ is the parameter balancing the effect of topic model and event classification. Our two modules are jointly trained with their parameters updated simultaneously. In decoding, it is not necessary to run the reconstruction process of NTM over the test set, we can use the encoder output θ as topic features for trigger prediction.

5. Experiments

We first examine the performance of our model with comparison to the state-of-the-art methods and ablation study. Then, we study whether our joint learning framework can produce coherent topics. Finally, case study and error analysis are adopted to reveal different aspects of our model.

5.1. Settings

We evaluate our method on the commonly used MLEE [2] corpus. There are 3598 triggers in the training set and 1809 triggers in the test set. The events in this corpus broadly cover 4 categories namely "Planned", "Anatomical", "Molecular", and "General", which can be further divided into 19 sub-categories as shown in Table 2. Following Li et al. [7], we use the standard document splits as training and testing sets, and choose 25% of training set as development set.

5.1.1. Hyperparameters

We tune all the hyper-parameters on the standard development set. Dropout is adopted to mitigate overfitting, with a rate of 0.45 for word embeddings and 0.15 for hidden states. All models are optimized using the Adam optimizer [30], with an initial learning rate of 0.001 and decayed at every 8 epochs with a rate of 0.9. The maximum training epochs is set to 100. The hidden and batch sizes are set to 256 and 32, respectively. For the number of topics, we follow previous settings [31] set topic number K to 50. The trade-off term is set to $\lambda = 0.8$. Finally, to combat unknown words during testing, we replace singleton words with a UNK embedding, with a probability of 0.5.

Table 3
Comparison of our model and baselines in MLEE test set. Models with “*” indicate that results are obtained by reimplementing their models.

Model	P (%)	R (%)	F1 (%)
Pyysalo-SVM	70.79	81.69	75.84
Knowledge-SVM	75.35	81.60	78.32
Topic-Semi	82.26	72.17	76.89
TwoStage-SVM	80.06	78.87	79.75
Dependency-CNN	73.56	83.62	78.27
Attention-GRU	80.65	79.09	79.87
LSTM-CRF	78.08	77.89	78.28
Contextual-GRU	80.74	80.42	80.58
Attention-Document*	79.42	81.33	80.36
Joint-GATE-Sentence	81.58	81.08	81.33
Joint-GATE-Document	82.11	82.53	**82.32**

5.1.2. Metrics

To compare our model with baselines, we use Precision (P), Recall (R) and F-Measure (F1) as the major metric to evaluate model performances, treating a trigger as correct when its offset and type are both correct. We exploit pairwise t-test for measuring significance values.

5.2. Event trigger detection results

We first introduce systems of our implementations for comparison.

Joint-GATE-Document is the proposed model that jointly train document-level latent topics (Section 4.1) and trigger detection (Section 4.2).

Joint-GATE-Sentence is similar to Joint-Document, but latent topics are learned on the sentence-level.

We also compare our method with several state-of-the-art trigger detection methods, which can be divided into statistical methods and representation methods. The statistical methods include:

(1) **Pyysalo-SVM** [2] is a typical feature-based method, which performs trigger prediction using a Support Vector Machine (SVM).
(2) **Knowledge-SVM** [3] is also a feature-based method using SVM but additionally incorporating biomedical domain knowledge by pretraining a neural language model.
(3) **Topic-Semi** [11] is a rule-based semi-supervised method, which uses sentence-level topic features that are automatically learned by LDA on raw texts to include more training instances.
(4) **TwoStage-SVM** [4] is the state-of-the-art statistical method, which first extracts most useful features then classifies event types using SVM.

The neural representation methods are listed as follows: (1) **Dependency-CNN** [18] is the representation-based model, which trains word embeddings by integrating dependency relations into a CNN. Event triggers are decided by a feed-forward network. (2) **Attention-GRU** [5] utilizes argument information by exploiting a supervised attention mechanism. (3) **LSTM-CRF** [6] is a globally normalized neural method, which integrates different types of word embeddings and sentence embeddings. (4) **Contextual-GRU** [7] is the current state-of-the-art method, which takes an encoder-decoder architecture to summarize sentence-level information. They also investigate event label interactions by proposing a GRU-like decoder. All the above methods work on the sentence-level only.

To compare with existing document-level event detection method, we introduce **Attention-Document** [19], which is a recently proposed neural model that hierarchically integrate document-level information with gated multi-level attention. We obtain the result by running their released code on the MLEE corpus.

5.2.1. Comparisons

Table 3 shows the comparisons of the main results of our model with the baselines on the MLEE test set. We can observe that:

Table 4
Ablation results on the MLEE test set.

Model	P (%)	R (%)	F1 (%)
Full model	82.11	82.53	82.32
-previous tag	81.64	82.45	82.04
-entity type	81.31	82.42	81.86
-GATE function	81.26	82.11	81.68
-NTM	81.38	80.94	81.16
-Joint NTM	81.62	81.06	81.34
-BERT	80.45	81.30	80.87

1. Feature-based methods (the first group) perform worse than neural-based methods (the second group) on the average F1-scores (77.70% to 79.25%). Despite the representation strength of the neural networks, it is noticeable that a carefully designed feature-based method **TwoStage-SVM** gives a result of 79.75% which is on par with the second group. It is thus worthy of investigating how to combine two typical models for further improvements.
2. **Joint-GATE-Document** achieves an F1-score of 82.32%, which outperforms the sentence-level topic baselines **Topic-Semi** (76.89)% and **Joint-GATE-Sentence** (81.33)%. These results demonstrate that the topic representations inferred by document context are highly beneficial for trigger detection.
3. Compared with the existing document-level method, our **Joint-GATE-Document** noticeably outperforms **Attention-Document** by 1.96% F1-score ($p < 0.03$). We attribute the advantages of the neural topic model over the hierarchical attention to the ability to find most indicative words (e.g., increase, exhibited) that are helpful for trigger detection. In contrast, softmax-based attention mechanisms are distributions over the entire document context, which might inevitably include noise.
4. Finally, **Joint-GATE-Document** shows better performance than the current best model **Contextual-GRU**, with a 2.11% improvement on recall and 1.74% on F1-score ($p < 0.05$), which sets a new state-of-the-art on the MLEE corpus. We evaluate the contributions and effects of the various components of our framework in the following subsection.

5.3. Ablation study

In this subsection, we perform ablation experiments on the MLEE test set (Table 4). As can be seen, the F1-score is slightly degraded without integrating the tag embedding predicted by the previous step. This is reasonable because the model may predict an invalid BILOU sequence if each softmax layer decides tags individually. On the other hand, contextual labels are important clues to capture interactions between events (e.g., *Negative_regulation* event tends to co-occur with *Positive_regulation event*). We can also observe that entity type embeddings contribute to the model performance, notably for precision (with 0.80% degradation), demonstrating the effectiveness of alleviating false positive event types.

In order to demonstrate the efficiency of the GATE function, we test the case when GATE is removed from our model. We hence use a simple linear transformation layer with tanh activation to compare the difference. From Table 4, we can see that there are 0.85% degradation in precision and 0.42% degradation in recall, respectively, when GATE function is replaced with a linear transformation. This shows that the design of the GATE is effective for balancing the weight of contextual features and topic representations.

We additionally design two baselines to verify the usefulness of latent topics and joint training strategy. In Table 4, “-NTM” indicates our model without topic features, while “-Joint NTM” indicates a pipelined training approach where NTM is pre-trained on MLEE

Table 5
Topic coherence score comparison on MLEE test set. Higher scores indicate better coherence.

Models	NPMI scores
LDA	0.11
BTM	0.14
NTM	0.16
Our model	**0.19**

documents, and then the trigger model uses NTM encoder outputs as fixed topic representations. From Table 4, we can observe that the F1-score is significantly lower (by 1.16%) when topic features are removed ($p < 0.03$), in particular droping 1.60% in recall, demonstrating that the proposed neural topic model can boost the overall results by finding challenging triggers. We also observe that pipelined neural topics only bring small improvement (i.e., 0.17%) over "-NTM". This suggests that the joint training of topics and triggers is crucial to better consume the effectiveness of latent topic representations.

Lastly, without the BERT embeddings, it leads to mostly 1.55% degradation from our full model. This result indicates that by pre-training the masked bidirectional language model on a large amount of corpus [29], the contextualized word embeddings can capture deep generalizable semantic and syntax information across domains, which can be transferred to event extraction.

5.4. Study on learned topics

We have shown that latent topics are useful for event trigger detection. In this subsection, we analyze whether our model can learn meaningful topics.

5.4.1. Topic evaluation

Automatically evaluating topics is a challenging task, and consequently there have been efforts in developing evaluation metrics that attempt to match human judgment of topic quality [32].

Lau et al. [33] showed that among all the competing metrics, normalized pointwise mutual information (NPMI) between all the pairs of words in a set of topics matches human judgment most closely. The measurement of NPMI is:

$$\mathbf{NPMI}(t) = \sum_{i,j \leq M; j \neq i} \frac{\log \frac{P(t_i, t_j)}{P(t_i)P(t_j)}}{-\log P(t_i, t_j)} \tag{18}$$

For each topic t, the top M most probable words are selected to compute NPMI. In this work, we choose M to be 10 and coherence results are reported on the MLEE test set.

For comparison, we consider LDA (implemented with a gensim LdaMulticore package[5]), BTM[6] [34] (a state-of-the-art topic model specifically for short texts), and NTM [16]. For LDA and BTM, we run Gibbs sampling with 500 iterations. From the results in Table 5, we can observe that our model outperforms all the topic models compared by large margins, which implies that jointly exploring trigger detection can in turn help produce coherent topics.

5.4.2. Impact of topic numbers

Fig. 3 shows the event extraction F1-scores of **Joint-GATE-Sentence** (JGS) and **Joint-GATE-Document** (JGD) respect to topic numbers. As we can see, the curves of both the models are not monotonic and JGS peaks at 40 topics while JGD achieved the best 82.32% F1-score at 50 topics, indicating that document-level context can benefit more from larger topic numbers. It can be observed that JGD yields consistently better F1-scores than JGS, which demonstrates the robust performance of JGD over varying number of topics.

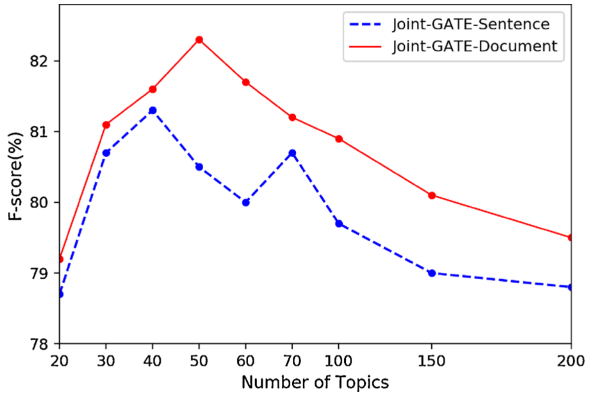

Fig. 3. Learning curve in terms of topic numbers.

5.4.3. Sample topics

To further analyze whether our model can produce coherent topics qualitatively, Table 6 shows the top 10 words of the latent topics reflecting "*breast cancer*" discovered by various models from the corpus. As can be seen, traditional LDA yields non-topic words include "*lung*", "*group*" and "*known*". We define non-topic words to be words that cannot clearly demonstrate the corresponding topic. For the results of BTM and NTM, non-topic words are also involved. Compared with other results, the topic generated by our model appears to be more coherent. For instance, "*mitochondrial*" and "*proliferation*" are indicative words for the topic.

5.5. Case study

In this subsection, we study the effectiveness of our model compare with baselines by selecting two representative cases, from which different aspects of models are reflected.

As shown in Table 7, case one is a situation where the word "therapies" triggers a new *Negative_regulation* event, which has not been observed in training set. This can be viewed as an out-of-label (OOL) problem. Due to the fact that *Planned_process* occurs many times in the training set with "therapies", our baseline, without the GATE mechanism, has challenges in capturing generalizable features between local representations and global topics, resulting in simple memorization of training instances. In contrast, our proposed GATE model dynamically weights the impact of input features, which correctly classifies the trigger.

In case two, "antagonised' has not appeared as a trigger in the training set, which results in an out-of-vocabulary (OOV) issue. Thanks to the use of contextualized word embeddings, our baseline model can identify that word "antagonised" involves an event, but misclassified it to a *Regulation* event. We found that the softmax probabilities of *Regulation* and *Negative_regulation* were 0.35 and 0.31, respectively, indicating that the baseline model was uncertain between these two types. In contrast, with the help of latent topics, the joint model is able to consider silent features from the surrounding sentences (e.g., indicative word "inhibited" in the previous sentence), and thereby correctly infers the event type.

5.6. Error analysis

We take several classification results as examples for error analysis. It is observed that one typical type of incorrect prediction is that the differences among some event types are an underlying factor. For

[5] https://pypi.org/project/gensim/.

[6] https://github.com/xiaohuiyan/BTM.

Table 6
Top 10 terms of the sample latent topics presented by various topic models from MLEE dataset. We interpret the topics as "*breast cancer*" according to the topic-word distributions. Bold and underlined words are **Non-topic words**.

LDA	cancer transfected breast **lung** cervical vector **group** cells activity **known**
BTM	breast tumor cancer containing lymphocyte **dimensional** reduced tissue plasmid angiogenesis
NTM	cancer breast lymphatic normal Matrigel tumor **individuals** model expressed **average**
Our model	breast cancer lymphatic mitochondrial cervical cell lymphocyte tumor Survivin proliferation

Table 7
Trigger prediction of different models. Standard C_i indicates the gold standard annotation. Words in underlined are correct triggers, while the italics are incorrect. Words in black bold font are named entities.

Standard C_1:	Most anti $\underline{\textbf{[angiogenic]}_{Blood\_v essel\_d evelopment}}$ $\underline{\textbf{[therapies]}_{Negative\_r egulation}}$ for malignant **gliomas** are in Phase I/II trials and...
Full model(-GATE):	Most anti $\underline{\textbf{[angiogenic]}_{Blood\_v essel\_d evelopment}}$ [*therapies*]$_{Planned\_p rocess}$ for malignant **gliomas** are in Phase I/II trials and...
Full model:	Most anti $\underline{\textbf{[angiogenic]}_{Blood\_v essel\_d evelopment}}$ $\underline{\textbf{[therapies]}_{Negative\_r egulation}}$ for malignant **gliomas** are in Phase I/II trials and...
Standard C_2:	... derivatives $\underline{\textbf{[inhibited]}_{Negative\_r egulation}}$ the activation of this enzyme. They also $\underline{\textbf{[antagonised]}_{Negative\_r egulation}}$ the $\underline{\textbf{[effects]}_{Regulation}}$ of both **thrombin** and...
Full model(-NTM):	... derivatives $\underline{\textbf{[inhibited]}_{Negative\_r egulation}}$ the activation of this enzyme. They also [*antagonised*]$_{Regulation}$ the $\underline{\textbf{[effects]}_{Regulation}}$ of both **thrombin** and...
Full model:	... derivatives $\underline{\textbf{[inhibited]}_{Negative\_r egulation}}$ the activation of this enzyme. They also $\underline{\textbf{[antagonised]}_{Negative\_r egulation}}$ the $\underline{\textbf{[effects]}_{Regulation}}$ of both **thrombin** and...

example, the trigger "proteolysis" in the sentence "... angiogenesis and extracellular matrix proteolysis." should be tagged as Breakdown. However, failing to understand the biological characteristics of "angiogenesis", our model mistakenly recognizes this candidate as Catabolism. In future work, we would explore external knowledge, such as entity description, to assist informing our model such distinct features.

Other types of error come from failure to capture phrase-level semantics. Taking "Adenovirus-mediated gene transfer of endostatin in vivo results in..." as an example, without instances annotated in training set where the phrase "gene transfer" triggers a *Planned_process* event, our model incorrectly tag these two words with the label *O*. Such error can be potentially alleviated from two perspectives: first, our neural topic model can be enhanced by relaxing the bag-of-words assumption to phrase-discovering topic models [35]; on the other hand, a more resource-intensive direction is to pre-train contextualized embeddings such as BERT by explicitly incorporating phrase-level knowledge [36], which is worthy of investigation in future work.

6. Conclusion

We have presented a novel joint training framework for learning latent topics and event triggers of a biomedical document. Unlike previous methods that focus on sentence-level event extractions, we investigate the usefulness of document-level context by leveraging a neural topic model based on variational autoencoding approaches. In addition, to balance the influences of two sources, we propose a novel gated function to dynamically integrate contextual features and topic representations. Empirical comparisons with state-of-the-art methods on the MLEE corpus demonstrate the validity and effectiveness of our model. Further analysis interprets the superiority to discover topic words that are indicative for biomedical event extraction.

Conflicts of interest

The authors declare no conflicts of interest.

Acknowledgements

We would like to thank the anonymous reviewers for their many valuable comments and suggestions. Work was done when the first author was visiting Westlake University. This work was supported by Natural Science Foundation of China (grant No. 61672389).

References

[1] Ananiadou S, Thompson P, Nawaz R, McNaught J, Kell DB. Event-based text mining for biology and functional genomics. Brief Funct Genomics 2014;14:213–30.
[2] Pyysalo S, Ohta T, Miwa M, Cho H-C, Tsujii J, Ananiadou S. Event extraction across multiple levels of biological organization. Bioinformatics 2012;28:i575–81.
[3] Zhou D, Zhong D, He Y. Event trigger identification for biomedical events extraction using domain knowledge. Bioinformatics 2014;30:1587–94.
[4] He X, Li L, Liu Y, Yu X, Meng J. A two-stage biomedical event trigger detection method integrating feature selection and word embeddings. IEEE/ACM Trans Comput Biol Bioinform 2017;15:1325–32.
[5] Li L, Liu Y. Exploiting argument information to improve biomedical event trigger identification via recurrent neural networks and supervised attention mechanisms. 2017 IEEE international conference on bioinformatics and biomedicine (BIBM). 2017. p. 565–8.
[6] Wang Y, Wang J, Lin H, Zhang S, Li L. Biomedical event trigger detection based on bidirectional lstm and crf. 2017 IEEE international conference on bioinformatics and biomedicine (BIBM). 2017. p. 445–50.
[7] Li L, Huang M, Liu Y, Qian S, He X. Contextual label sensitive gated network for biomedical event trigger extraction. J Biomed Inform 2019:103221.
[8] Ji H, Grishman R. Refining event extraction through cross-document inference. Proc. of ACL-08: HLT 2008:254–62.
[9] Liao S, Grishman R. Using document level cross-event inference to improve event extraction. Proc of ACL 2010.
[10] Yang B, Mitchell T. Joint extraction of events and entities within a document context. Proc of NAACL-HLT 2016:289–99.
[11] Zhou D, Zhong D. A semi-supervised learning framework for biomedical event extraction based on hidden topics. Artif Intell Med 2015;64:51–8.
[12] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. J Mach Learn Res 2003;3:993–1022.
[13] Zeng J, Li J, Song Y, Gao C, Lyu MR, King I. Topic memory networks for short text classification. Proceedings of the 2018 conference on empirical methods in natural language processing 2018:3120–31.
[14] Li J, Song Y, Wei Z, Wong K-F. A joint model of conversational discourse and latent topics on microblogs. Comput Linguist 2018;44:719–54.
[15] Srivastava A, Sutton C. Autoencoding variational inference for topic models. 2017arXiv preprint arXiv:1703.01488.
[16] Miao Y, Grefenstette E, Blunsom P. Discovering discrete latent topics with neural variational inference. Proceedings of the 34th international conference on machine learning – vol. 70. 2017. p. 2410–9.
[17] Huang R, Riloff E. Modeling textual cohesion for event extraction. Twenty-sixth AAAI conference on artificial intelligence 2012.
[18] Wang J, Zhang J, An Y, Lin H, Yang Z, Zhang Y, et al. Biomedical event trigger detection by dependency-based word embedding. BMC Med Genomics 2016;9:45.
[19] Chen Y, Yang H, Liu K, Zhao J, Jia Y. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. Proceedings of the 2018 conference on empirical methods in natural language processing 2018:1267–76.
[20] Wang X, McCallum A, Wei X. Topical n-grams: phrase and topic discovery, with an application to information retrieval. Seventh IEEE international conference on data mining (ICDM 2007). 2007. p. 697–702.
[21] Liao S, Grishman R. Acquiring topic features to improve event extraction: in preselected and balanced collections. Proceedings of the international conference recent advances in natural language processing 2011 2011:9–16.
[22] Kingma DP, Welling M. Auto-encoding variational bayes. stat 2014;1050:10.

[23] Dickey JM. Multiple hypergeometric functions: probabilistic interpretations and statistical uses. J Am Stat Assoc 1983;78:628–37.

[24] Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK. An introduction to variational methods for graphical models. Mach Learn 1999;37:183–233.

[25] Zhai K, Boyd-Graber J, Asadi N, Alkhouja ML. Mr. lda: a flexible large scale topic modeling package using variational inference in mapreduce. Proceedings of the 21st international conference on world wide web. 2012. p. 879–88.

[26] Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: a review for statisticians. J Am Stat Assoc 2017;112:859–77.

[27] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. Proceedings of the 27th international conference on machine learning (ICML-10) 2010:807–14.

[28] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems. 2013. p. 3111–9.

[29] Devlin J, Chang M-W, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies, vol. 1 (Long and Short Papers) 2019:4171–86.

[30] Kingma DP, Ba J. Adam: a method for stochastic optimization. 2014arXiv preprint arXiv:1412.6980.

[31] Dieng AB, Wang C, Gao J, Paisley J. Topicrnn: a recurrent neural network with long-range semantic dependency. 2016arXiv preprint arXiv:1611.01702.

[32] Newman D, Lau JH, Grieser K, Baldwin T. Automatic evaluation of topic coherence. Human language technologies: The 2010 annual conference of the north American chapter of the association for computational linguistics. 2010. p. 100–8.

[33] Lau JH, Newman D, Baldwin T. Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. Proceedings of the 14th conference of the European chapter of the association for computational linguistics 2014:530–9.

[34] Yan X, Guo J, Lan Y, Cheng X. A biterm topic model for short texts. Proceedings of the 22nd international conference on world wide web. 2013. p. 1445–56.

[35] Lindsey RV, Headden III WP, Stipicevic MJ. A phrase-discovering topic model using hierarchical pitman-yor processes. Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. 2012. p. 214–22.

[36] Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q. Ernie: enhanced language representation with informative entities. 2019. https://arxiv.org/abs/arXiv preprint arXiv:1905.07129.