IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 26, 2020, accepted May 9, 2020, date of publication May 19, 2020, date of current version June 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995776

Topic Detection and Tracking Based on Event Ontology

WEI LIU, LEI JIANG, YUSEN WU, TINGTING TANG, AND WEIMIN LI
School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

Corresponding author: Wei Liu (liuw@shu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0117500, and in part by the National Social Science Fund Major Project of China under Grant 19ZDA301.

ABSTRACT In recent years, Topic Detection and Tracking (TDT) has served as a core technology for searching, organizing and structuring news oriented textual materials from a variety of internet news and social media. The biggest challenges of TDT are the sparsity and complexity of data, organization of topic granularity, unexpectedness of emergency topics, and the unpredictability of topics evolution. This paper proposes a new TDT method based on event ontology for hierarchical topic detection and tracking topic evolution, named TDTEO. As domain-oriented event knowledge base, the event ontology provides event classes hierarchy based on domain common sense, as well a set of scenario models that describe the occurrence and evolution of different types of emergency events. The proposed method solves the problem that new emerging events are easy to be missed in the process of topic detection, and solves the problem that the topic model is easy to result in semantics drift due to the dynamic evolution of topic, and it can effectively improve the accuracy of topic detection and tracking. Experiments show our models achieve satisfactory performance of topic detection with a maximum macro-F1 value of 85.25%, and the $(C_{Det})_{Norm}$ of topic tracking in the datasets is as low as 0.1028. Experimental results show that hierachical topic model can effectively detect the topics and tracking model based on event scenario can reflect the trend of topic evolution.

INDEX TERMS Event ontology, topic detection, topic tracking, hierarchical topic model, event scenario.

I. INTRODUCTION

The Topic Detection and Tracking program originated from a pilot study of technologies for automatically organizing news texts sponsored by DARPA in 1996 [1]. The core task of topic detection is to discover new topics from the data stream and collect subsequent related reports (data or information). On the Internet, a sequence of web media content on a specific subject forms a topic. A topic can be defined fromally as "a seminal event or activity, along with all directly related events and activities" [2]. Most research on new topic detection mainly focuses on the method based on topic models (also include its variants) and text clustering [3]–[7]. In approaches based on topic model, topics are typically detected through co-occurrence analysis of document-words. The construction of the topic model usually utilizes the combination of one or more calculation methods such as word feature vector, NER (Named Entity Recognition), TF-IDF [8] weight calculation. In recent years, researchers proposed online topic models and temporal topic models to realize real-time detection of topics, but there still exist deficiencies, including time-consuming and high rate of missed detection [9]. The reasons for these deficiencies include the sparsity of data and the lack of effective organization of topics granularity. The topic tracking task is fundamentally similar to the standard routing and filtering tasks of Information Retrieval. Given a detected topic which described with a few sample instances of news reports, the task is to identify any and all subsequent news reports describing the same topic [1]. The biggest challenge of topic tracking is the drift of topic core content while reduction of the number of reports or evolution of topic over time. Traditional topic tracking methods based on topic models usually include vector retrieval and probability retrieval, neural networks, KNN, dynamic clustering, decision trees and so on [3], [10]–[12]. In recent years, in response to online topic tracking, some adaptive topic tracking technologies have emerged, such as GE, Dragon, UMass [13]. Compared with traditional topic

The associate editor coordinating the review of this manuscript and approving it for publication was Jiankang Zhang.

tracking technology, adaptive topic tracking technologies embed a self-learning mechanism, which enables topic model to be automatically updated along with the development and evolution of topic in real time, thus to track topic evolution trend effectively. At present, adaptive topic tracking usually automatically learns the evolution trend of the topic and the drift trigger point based on system feedback. However, this kind of adaptive tracking usually lacks supervision. When there is too much feedback (which is easy to cause information clutter) and too little feedback, it is easy to result in more serious drift of topic model, which is called pseudo-feedback.

Aiming at the shortcomings of existing online topic detection and tracking methods, this paper proposes a topic detection and tracking method based on event ontology, named TDTEO (Topic Detection and Tracking based on Event Ontology). Event ontology is a prior knowledge base constructed according to the law of occurrence and evolution of events in different domain [14]. The event class is the basic knowledge unit in event ontology. Event ontology contains formal descriptions (including action, objects, place and time of event) of main event classes and event instances in a specific domain. Event ontology also provides event classes hierarchy model based on domain commonsense, as well a set of event scenario models that describe the occurrence and evolution of different types of events. The knowledge structure of the event ontology is very suitable for topic detection and tracking. The event hierarchy model (taxonomy relationship model) can be transformed into a hierarchy of topics in specific domain, in which an event class corresponds to a topic, and topic model can be constructed by using event information. The hierarchical topic model can overcome the shortcomings of scattered, lacking of organization and uneven granularity of topics in traditional topic detection. The event scenario model (non-taxonomy relationship model) in the event ontology describes the semantic relationships of a series of subsequent events triggered by a seed event, which reflects the pattern of event occurrence and evolution. By using event ontology, while a certain topic is detected, the event scenario model in the event ontology can be used to predict the occurrence of subsequent events in advance, thereby updating the topic model by accumulating features of the subsequent events, and eventually to improve the effectiveness of topic tracking.

In this paper, we take topic detection and tracking in the domain of *Science and Technology* as an example to study TDT based on event ontology. An event ontology about *Science and Technology* is constructed in advance. The process of topic detection and tracking includes three steps. First, according to the event class hierarchy model in the event ontology, visit each event class from top to bottom in turn, obtain linguistic expression (a set of keywords to represent the event class, an example will be seen in TABLE 1) of the event class. Second, create a corresponding topic model by using the word vector of these keywords respectively. The topic models can be calculated by using text vectorization models (Word2Vec [15]–[17], GloVe [18] or FastText [19]). Third, update the topic model by using feature vectors (such as specific event participants, entities, places, etc.) created from the detected event instances to enhance the adaptability of the topic model. While tracking a specific topic, we regard each topic detection model as an initial topic tracking model. First, find the corresponding event class and its scenario model in event ontology, and then query its subsequent event classes (there exist *sequence* or *causal* relationship with it) and obtain the feature data from subsequent event classes to update the topic model. If there are multiple choices for subsequent events (*choice* relationship between the subsequent events), we can obtain features data of subsequent events according to different branches, and then update these feature data to topic model respectively. Each branch will generate a new topic model, so that the evolution process of the topic can be accurately tracked.

The paper is structured as follows. Section 2 discusses the related works about topic detection and tracking. Section 3 introduces the event ontology model structure. Section 4 and 5 discuss the topic detection and tracking method based on event ontology. In Section 6, we illustrate some experimental results in real labeled datasets. Section 7 concludes the paper.

II. RELATED WORK

The early works of The Topic Detection and Tracking (TDT) began in 1990s [1]. Over the past decades, many methods have been proposed for TDT. Since 1998, the National Institute of Standards and Technology(NIST) has held an international conference on topic detection and tracking every year. The meeting specified the criteria for evaluation of TDT, and divided the TDT tasks into five sub-tasks: report division, topic tracking, topic detection, first report detection and association detection [20].

A. TOPIC DETECTION

Topic detection technology can accurately detect topics in the news media stream, and is used to track the dynamic evolution process of the topic. Therefore, the most critical issue lies in topic detection. TDT includes methods based on clustering models [1]. When the text is described by a vector space model, the report of the similar topic is closer to the distance in the vector space, so many clustering algorithms can be applied to topic detection. Yang *et al.* [4] proposed a historical event detection method based on average grouping hierarchical clustering, which made use of the characteristics of events clustering over a period of time, so that the aggregated result had higher average similarity. Kumaran and Allan [5] used VSM to express news topics and reports, and endowed higher weights to named entities for entity detection. Li *et al.* [6] proposed a news event detection method based on probability generation model which integrated the content and time information of news events into a unified framework and combined the content and time information to detect historical events. Wei-Hua and Yuman-Quan [7] proposed a topic discovery algorithm based on multi-layer clustering, which divides all data into related groups, and optimizes the groups clustering with multiple strategies to improve the effect of topic detection. These cluster-based models have some drawbacks, such as the selection of appropriate number of clusters, and the number of topics obtained by clustering may be significantly different from the actual topics.

The theme-based model is a generation model of probability maps by establishing an association between information and topics. Zeng and Zhang [21] used the hidden Markov model to represent the topics in the text. Based on the model, the TDT algorithm combined with the theme transformation improves the accuracy of subject detection. Yuan *et al.* [10] proposed a self-aggregating text topic model that uses automatic aggregation in the modeling process of the topic. Based on Twitter's huge amount of information and short text, Ding *et al.* [22] proposed a Dirichlet processing model to realize the topic detection of short text stream. Due to the fact that social network text is short and sparse, the traditional topic detection methods can't solve the problem of text sparsity. Shi *et al.* [3] proposed a topic discovery method based on RNN and topic model, which use RNN to learn the relationship between words as the a priori knowledge of topic model, and simultaneously constructs word pairs to solve the sparsity problem of text topic modeling. Chen *et al.* [23] proposed a heterogeneous topic model for large amount of heterogeneous information in the media, and realized the correspondence of text themes by iteratively updating the themes and words distribution. These theme-based models are more affected by high frequency words. As a result, the model inclines to high-frequency feature words, resulting in insufficient text differentiation.

In recent years, with the rapid development of deep learning technology, deep learning has also been applied to topic detection. Zhou *et al.* [24] combined CNN and RNN to capture the local features of phrases and semantic information of sentences to achieve the classification of the text. Xiang *et al.* [25] took character-level text as the original information and used an one-dimensional convolutional neural network to classify the text. Pappas and Popescu-Belis [26] used hierarchical attention mechanism to indicate the importance of words or sentences combined with contextual information to achieve classification of text. Zhou *et al.* [27] suggested a hierarchical neural network with automatic semantic feature selection to improve the whole performance of Chinese conversation topic classification tasks. Zhang *et al.* [28] proposed a coordinated CNN-LSTM-Attention model for emotional classification of text. These deep learning methods can achieve higher accuracy and reduce the redundancy of artificial design features, but the training cost of the model is higher and it takes a lot of time and computing resources, the trained model cannot modify learned parameters unless retrain.

B. TOPIC TRACKING

The research of topic tracking can be divided into two types: non-adaptive topic tracking and adaptive topic tracking. Knowledge-based and statistics-based methods are the main methods of non-adaptive topic tracking. Allan *et al.* [1] proposed a topic track method based on decision trees, it must rely on multi-layer tree structure to obtain the correct tracking strategy, which would lead to missing detection. Papka [29] adopt KNN classification algorithm to extract K reports similar to current reports for tracking. Zhang *et al.* [11] introduced entity words in topic tracking, which improved the tracking effect. Chen *et al.* [30] proposed a method of topic tracking based on semantic relevance, which solved the problems of sparse feature and inaccurate topic tracking. Because non-adaptive topic tracking is to build a topic model based on a small number of topic reports, and users usually have very little knowledge of sudden topics, the topic model is usually insufficiently practical in application.

Adaptive topic tracking can achieve continuous tracking through self-learning mechanism, which can not only embeds new features for topic, but also dynamically adjust the weight of features. Rao *et al.* suggested a multi-relational term scheme for first story detection [31]. In adaptive topic tracking, Franz *et al.* [32] proposed a topic tracking method based on supervised and unsupervised. The topic model of initial training in practical is not sufficient and accurate, Ren *et al.* [12] proposed an adaptive topic tracking method based on K-Modes clustering. Yeh *et al.* [33] proposed a dynamic concept implicit Dirichlet distribution model for topic tracking in conversational. Compared with the traditional Latent Dirichlet Allocation(LDA) model, this model takes time characteristics into account by introducing dynamic concepts. Cai *et al.* proposed a novel framework for constructing temporal event map [34]. Xu *et al.* [35] and Vargas-Calderón *et al.* [36] used LDA model to extract topic information from news texts, then improved single-channel algorithm for topic tracking and introduced time decay function to improve the similarity between topics, Syed and Spruit [37] testified that use full-text data of documents can increase quality of extracted topic information.

III. PREREQUISITES

A. EVENT ONTOLOGY DEFINITIONS AND STRUCTRUE

In recent years, ontology has been used to provide semantic information for topic detection and tracking [38]. However, most of these ontologies are traditional conceptual ontologies, which describe the static relationship between concepts. The semantics of conceptual relationships are weak, especially it is difficult to represent the evolution of topics, so the effectiveness of improving topic detection and tracking is limited. Event ontology is a shared, formal and explicit specification of an event class system model that exists in real world objectively [39]. Compared with traditional conceptual ontology, event ontology pays more attention to the dynamic features of event. It is a dynamic knowledge base for representation of events, and can describe the occurrence and evolution of events in news reports more effectively. Event ontology can be represented as a knowledge base composed of a set of event classes, a set of relationships between event classes and a rule sets. It can be formally defined as a 3-tuple:

$$EO ::= < ECs, Rs, Rules >$$

where *ECs* denotes a set of all event classes involved in event ontology and *Rules* denotes a set of rules for event knowledge inference. R_s represents a set of f relations between event classes, including taxonomy relation (*is_a* relation) and 5 non-taxonomy relations:

$$Rs ::= \{R_{is\_a}, R_{follow}, R_{choice}, R_{cause}, R_{compose}, R_{concur}\}$$

Fig.1 shows the structure of a domain event ontology. It consists of an overall event hierarchy model and a set of event scenario models. Event hierarchy model is usually a directed acyclic graph that is composed of event classes and taxonomy relations between them. Event scenario model is usually a directed cycling graph that consists of event classes and non-taxonomy relations, which is used to describe the pattern about occurrence of a seed event and a set of subsequent events triggered by it. Event scenario models usually express the rules and patterns of event occurrence or evolution. Elements of event can also be described as a conceptual hierarchy model, such as object hierarchy, organization hierarchy and place hierarchy.

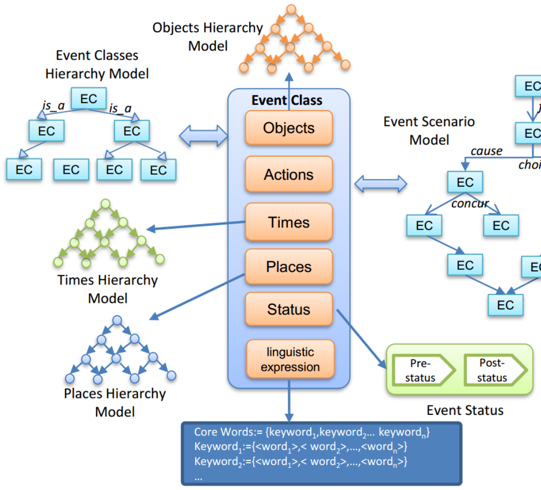


FIGURE 1. The structure of event ontology.

We use the definition of event in [39] to describe the event in event ontology. The event in the news media can be described as a knowledge unit consisting of actions (trigger the event), objects involved in the event, place, time, status and linguistic expression. The linguistic expression of event is usually described in the form of keywords dictionary, which can be used to calculate the features vector of the event. For example, an event "*Exposure of Paper Plagiarism*" can be described with framework-based specification as TABLE 1.

TABLE 1. Specification of Event Class "*Exposure of Paper Plagiarism*".

Frame <*Exposure of Paper Plagiarism*>	
Inheritance	*Exposure of Academic Misconduct*

Objects	*Role 1: Author; Role 2: Plagiarist;*
	Role 3: Paper; Role 4: Author Affiliation;
	Role 5: Journal; Role 6: Journal Editorial Department

Action	*Plagiarize paper content; Falsify the data; Expose; Criticize*
Place	*Social Media Class*
Time	*Time Class*
Pre-status	*not infringed (author);* *not reputation-damaged (plagiarist)*

Post-status	*infringed (plagiarist);*
	reputation-damaged (plagiarist);
	called-in-question (plagiarist);
	criticized (plagiarist)

Linguistic Expression	*Action: Exposure,Criticize, Disclosure, Plagiarism, Duplication, Forgery, Falsify, Violation* *Object: Paper, Periodical, Title, Author, Plagiarist Similarity , Editor, Experiment* *Place: Social Media, Weibo, WeChat*

IV. TOPIC DETECTION

A. TOPIC DETECTION MODEL

Traditional topic detection is usually an unsupervised learning task (with no labeled samples and prior knowledge). Topic detection includes discovering unrecognized topics in cumulative news texts or real-time news streams. Due to the sparsity and complexity of massive news data, topic detection methods are difficult to create effective topic models and obtain high-accuracy detection. In practical applications of online topic detection, the topic to be detected is usually highly focused (the semantics of topic is identified in advance). For the reason that event classes hierarchy model in event ontology contains the semantic information of different levels of event types in a specific domain, it can be used as a priori knowledge to construct effective and adaptive topic detection models. The semantic information of event could be obtained from textual linguistic expression of event and event elements, which are usually expressed in the form of keywords or phrases like TABLE 1. As a result, we propose an algorithm of topic detection model construction, see algorithm 1.

In algorithm 1, the vectorization process in step 4 is to embed topics into the vector space, and we use the pre-trained model(Word2Vec, GloVe or FastText) to construct the topic model, which enhances the semantics of topic model.

By using algorithm 1, a hierachical topic model can be derived based on event class hierarchy, which could cover semantics of news texts to the greatest extent for a specific domain. Fig. 2 is a partial event class hierarchy model in the domain of *Science and Technology Events*, which shows part of event classes with taxonomy relationships between them. Therefore, we can construct topic models for all event classes in the event class hierarchy by using the process above. For example, according to specification of event class "*Expose Paper Plagiarism*" in TABLE 1, by using step 2 and step 3 of the algorithm 1, a set of feature words could be extracted and extended as:

Algorithm 1 Topic Detection Model Construction Algorithm

```
Require:
    1. Event classes hierarchy model from a domain event ontology.
    2. training dataset of the domain topic.
Ensure:
1: Select the root node of event ontology as root event class $EC$.
2: Start from $EC$ of the event hierarchy model, create a topic $TP$ with the same name as $EC$, and obtain the linguistic expression of the actions, objects, places and status of $EC$ in turn, name them as $LE = \{LE_j | 1 \leq j \leq 4\}$ respectively.
3: Expand $LE$ of $TP$ by reusing the hypernym and hyponym in WordNet. The extended $LE$ is called $LE' = \{LE_j' | 1 \leq j \leq 4\}$.
4: Extract $u$ keywords from training dataset by TF-IDF, and we use $\overrightarrow{V_{TP}}$ describe $TP$ in the vector space by using formula 1:
       $$\overrightarrow{V_{TP}} = 1/2 \sum_{j=1}^{4} \sum_{k=1}^{l_j} \frac{\omega_j \overrightarrow{V_{jk}}}{l_j} + 1/2 \sum_{i=1}^{u} \overrightarrow{V_i} \quad (1)$$
   $l_j$ is the keywords number of $LE_j'$, we use the vetorization model like Word2Vec, GloVe or FastText to describe the keyword in the vector space, $\overrightarrow{V_{jk}}$ represents the vector of the $k^{th}$ keyword vector of $LE_j'$. $\omega_j (1 \leq j \leq 4)$ denotes weights of different elements in $EC$, $\overrightarrow{V_i}$ represents the vector of the $i^{th}$ keyword vector of the keywords from training dataset.
5: Continue to read the next $n$ sub-event class of $EC$ named $EC_{lower}$, $EC_{lower_i}$ represents the $i^{th}$ of the $EC$ children. Repeat step 2 to step 4 and create topic $TP_{lower_i}$ for each $EC_{lower_i}$, then calculate feature vector of $TP_{lower_i}$ as $\overrightarrow{V_{TP_{lower_i}}}$ to construct topic models. After that, update $\overrightarrow{V_{TP_{lower_i}}}$ by using formula 2:
       $$\overrightarrow{V_{TP_{lower_i}}} = \lambda \overrightarrow{V_{TP_{lower_i}}} + (1 - \lambda) \overrightarrow{V_{TP}} (0 < \lambda < 1) \quad (2)$$
   where $\lambda$ denotes the offset ratio, it gives $\overrightarrow{V_{TP_{lower_i}}}$ an effective offset.
6: Iterate over the children of $EC_{lower_i}$, and set $EC_{lower_i}$ as the new root $EC$, then repeat step 5 until all event classes in event class hierarchy model are visited and their corresponding topic models are generated.
7: return  A hierarchical topic detection model.
```

Feature words = {*Entity: Paper, Article, Journal, Periodical, Title, Author, Plagiarist, Similarity, Editor, Experiment, Experiment Data, Experiment Result;*
Actions: Disclosure, Expose, Plagiarism, Duplication, Forgery, Falsify, Violation, Deny, Admit, Apologize;
Place: University, Institute, State, Province, Country, Weibo, WeChat;
Time: Year, Jan, Feb, …;
Status: Infringed, Criticized, Reputation Damaged }.

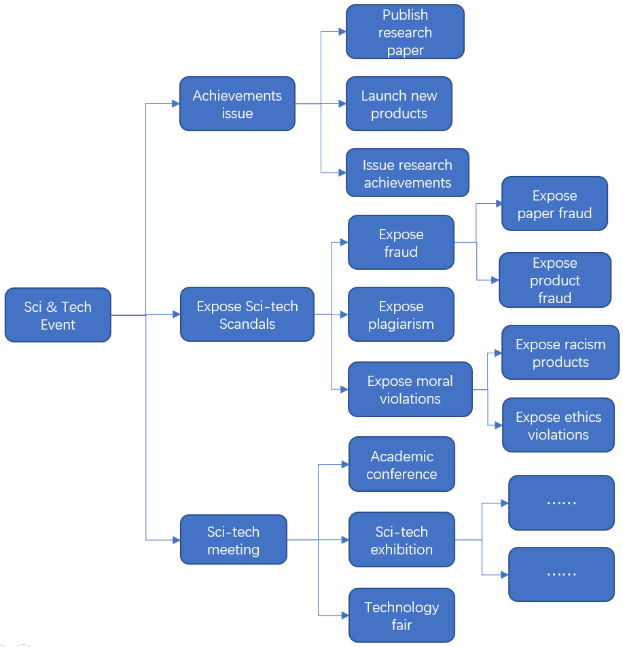


FIGURE 2. **Partial Event Class Hierarchy of *Science and Technology Events*.**

The words are expanded by HowNet [40] and TF-IDF [8]. And then, we can construct a topic model "*Expose Paper Plagiarism*" with these feature words. By using formula in step 3 above, the vector space of the topic model can be generated as $\{0.0352, -0.2352, \ldots, 0.1632, -0.0725\}$. The size of vector is the dimension size of the model.

B. DETECTION ALGORITHM

The topic detection model is constructed from event ontology of specific domain, which has good hierarchical structure and rich semantics. Based on topic models above, we can identify topics from a sequence of news events, which uses the vector space model to construct the cluster centers by using the linguistic expression of the nodes in the topic detection model, then calculates the similarities between each topic center and vectorized news events. The algorithm does not need to repeatedly search cluster centers like K-means [41], and the computational complexity is close to *O(n)*. It dynamically updates the topic centers by accumulating newly-detected events vectors, and then updates the detection model [8]. Fig. 3 shows the process of topic detection, which is described in algorithm 2. We use cosine similarity to calculate the similarity of two vectors by formula 3 in detection algorithm.

$$SIM_{\overrightarrow{v}\overrightarrow{u}} = \frac{\overrightarrow{v} \cdot \overrightarrow{u}}{|\overrightarrow{v}||\overrightarrow{u}|} = \frac{\sum_{i=1}^{n} v_i \times u_i}{\sqrt{\sum_{i=1}^{n} (v_i)^2} \times \sqrt{\sum_{i=1}^{n} (u_i)^2}} \quad (3)$$

where $\overrightarrow{u}$ and $\overrightarrow{v}$ are vectors with the same dimension n.

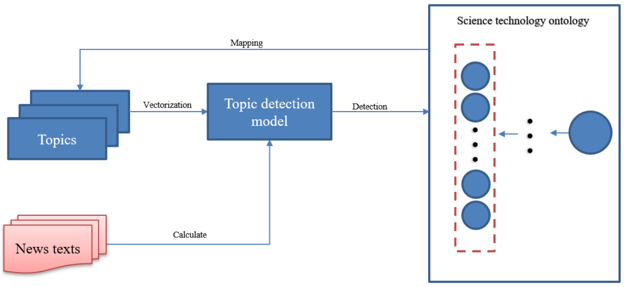


FIGURE 3. **The process of topic detection.**

Algorithm 2 Topic Detection Algorithm of TDTEO

```
Require:
    1. Topic detection model $TM$.
    2. A news text(event) set $NS$ with implicit topics.
Ensure:
1: Select the root node of $TM$ as intial root $TP$ and news text set $NS$ as initial input texts.
2: Get $n$ children topic nodes of root $TP$ as $TP_{lower}$, and $\overrightarrow{V_{TP_i}}$ describes $TP_{lower_i}$ which is the $i^{th}$ child of $TP$ in the vector space.
3: $NS_j$ represents the $j^{th}$ event of $NS$, we divide each $NS_j$ text into $w$ words after removing stop words, then use formula 4 to calculate the vector for $NS_j$ as $\overrightarrow{V_{E_j}}$:
       $$\overrightarrow{V_{E_j}} = \frac{1}{w}\sum_{k=1}^{w}\overrightarrow{V_k} \qquad (4)$$
   where $\overrightarrow{V_k}$ represents the $k^{th}$ word in the vector space.
4: For each $TP_{lower_i}$ in $TP_{lower}$ and each $NS_j$ in $m$ events, calculate the similarity of $\overrightarrow{V_{TP_i}}$ and $\overrightarrow{V_{E_j}}$ as $SIM_{\overrightarrow{V_{TP_u}}\overrightarrow{V_{E_v}}}$. If the relationship $i, j = \arg\max_{u,v} SIM_{\overrightarrow{V_{TP_u}}\overrightarrow{V_{E_v}}}$ is satisfied, it means that the similarity between $NS_j$ and $TP_{lower_i}$ is the highest, $NS_j$ belongs to the topic node $TP_{lower_i}$. Repeat step 4 until all events in the node $TP$ are detected to $TP_{lower}$.
5: Set each $TP_{lower_i}$ in $TP_{lower}$ as the new root $TP$, and set the events detected to the new root $TP$ becomes the input events $NS$. Repeat step 2 to step 4 until the model has no children nodes.
6: return A set of topics with news texts.
```

For the reason that it is difficult to identify which words are the most prominent and most effective in the models, we averages the semantics of the text based on text vectorization methods in step 3, which can improve the robustness of the model and reduce the interference caused by randomness [42]. The average vectorization is similar to average-pooling in deep learning, which can reduce vector offsets, so it will not result in "overfitting" that often happens in neural networks. For example, for the topic of "*Expose Paper Plagiarism*", most of the high-frequency words in news texts are descriptive words with strong sentiments like "shock" and "shame". On the contrary, the frequency of verbs words like "plagiarize", "publish" and "duplicate" is relatively small, which are directly linked to the topic of "*Expose Paper Plagiarism*". Obviously, increasing the weight of these verbs while describing the topic of "*Expose Paper Plagiarism*" will make the topic model more precise. However, when describing sports topics, more professional sport nouns are more likely to discriminate different sport topics. When describing a topic, we usually need to use prior knowledge to determine which types of keywords is more important (need to increase weight). Therefore, when describing an unknown new topic, it is a wise choice to set average weights for different types of keywords. Although there are many words in a news text that can not accurately describe the topic, or even deviate from the topic, other texts will also have similar problems. If there are similar offsets in vector space, no matter whether these offsets are eliminated or not, there is no effect on accuracy of topic model. Therefore, we use the average vectorization to model topics, thus to reduce the random influence which may be caused by different artificial prior knowledge, and enhances the robustness of topic models.

The proposed hierarchical topic model consists of topics with different granularity. In general, the hypernym topic contains the semantic features of all its hyponym topics, which enable fine-grained topic detection, but also avoid missing detection of topic. For example, while identifying the topic of *Zhaitianlin's plagiarism* with the topic model in Fig.2, if an article can be matched with the feature vector of the *Science and Technology* topic, it will be classified as the topic of "*Expose Paper Plagiarism*". If it can not be matched, the article further matches its hypernym topic *Expopse Sci-tech Scandals*. Since the topic Expopse Sci-tech Scandals topic contains more semantic information, including feature vector of the topic *Expose Fraud* and *Expose Moral Violations*. As a result, the article has a high probability of being matched with the topic of *Expopse Sci-tech Scandals*. Therefore, the hierarchical topic model can effectively improve the efficiency of topic detection.

V. TOPIC TRACKING

A. TOPIC TRACKING MODEL

The event scenario model (non-taxonomy relationship model) in the event ontology describes the semantic relationships of a series of subsequent events triggered by a seed event, which reflects the pattern of event occurrence and evolution. On the basis of topic detection mode, the topic tracking model integrates the semantic information of topic evolution by accumulating word features of the subsequent events in the event scenario model, and eventually achieves topic tracking and evolution. The construction of topic tracking models is described as algorithm 3.

Fig. 4 describes a scenario model of *'Expose Paper Plagiarism"* in event ontology, which can be transformed into a topic model vector group **TMV**. If we use a vectorization model Word2Vec, and the dimension of it is n, each node in

Algorithm 3 Topic Tracking Model Construction Algorithm

```
Require:
    1. An domain event ontology.
    2. One topic node $TP$ in detection model.
Ensure:
1: For the topic $TP$, the corresponding event scenario model is queried from the event ontology according to the topic name.
2: Extract the seed event class of the node corresponding event scenario model as $event_{seed}$, obtain the linguistic expression of the actions, objects, places and status of it in turn, name them as $LE$, and $LE = \{LE_i | 1 \le i \le 4\}$ respectively. Then expand $LE$ of $TP$ by reusing the hypernym and hyponym in WordNet to enhance vocabulary richness. The extended $LE$ is called $LE'$, and $LE' = \{LE'_i | 1 \le i \le 4\}$.
3: Calculate the vector of $LE'$ with a text vectorization method to describe $event_{seed}$ in a vector space. For example, $LE'_j$ contains $m$ words, and $\overrightarrow{V_k}$ represents the $k^{th}$ word in the vector space of the model, $\overrightarrow{LE'_j}$ represent the vector of $LE'_j$ which can be calculated by using formula 5:
       $$\overrightarrow{V_{LE'_j}} = \frac{1}{m} \sum_{k=1}^{m} \overrightarrow{V_k} \qquad (5)$$
4: We use $\overrightarrow{V_{seed}}$ to represent the average vector generated by the elements and extended elements of $event_{seed}$. $\overrightarrow{V_{seed}}$ can be calculated by using formula 6:
       $$\overrightarrow{V_{seed}} = \sum_{j=1}^{4} \omega_j \overrightarrow{V_{LE'_j}} \qquad (6)$$
   where $\omega_j (1 \le j \le 4)$ denotes weights of different elements. Then add it to the topic model vector group TMV, and TMV = $\{\overrightarrow{V_{seed}}\}$.
5: According to event scenario model obtained in step 1, extract the subsequent event classes (including causality, concurrency, etc.) of $event_{seed}$ by breadth-first traversal, then calculate the vector of each subsequent event classes by step 2 to step 4, and add the vectors to TMV.
6: Repeat step 5 until all nodes of the scenario model are calculated and added the vectors of them to TMV. If the scenario model has $k$ event class nodes, the vector group of the topic template will be expanded as TMV $= \{\overrightarrow{V_c} | c \in \{1, \ldots, k\}\}$. TMV represents the vector group as the topic tracking model.
7: return A topic tracking model with scenarios based on event ontology.
```

the scenario can be transformed into a vector $\{v_1, v_2, \ldots, v_n\}$ by the above construction method, and the **TMV** will be represented as:

$$\{\{v_1^{(1)}, v_2^{(1)}, \ldots, v_{n-1}^{(1)}, v_n^{(1)}\}, \{v_1^{(2)}, v_2^{(2)}, \ldots, v_{n-1}^{(2)}, v_n^{(2)}\}, \ldots, \{v_1^{(k)}, v_2^{(k)}, \ldots, v_{n-1}^{(k)}, v_n^{(k)}\}\}$$

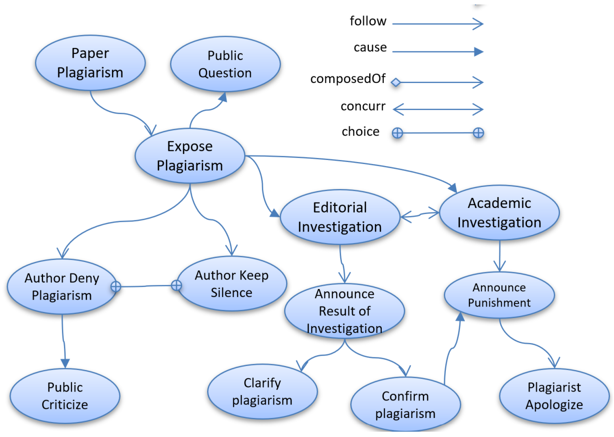


FIGURE 4. A scenario model of ***"Expose Paper Plagiarism"*** in event ontology.

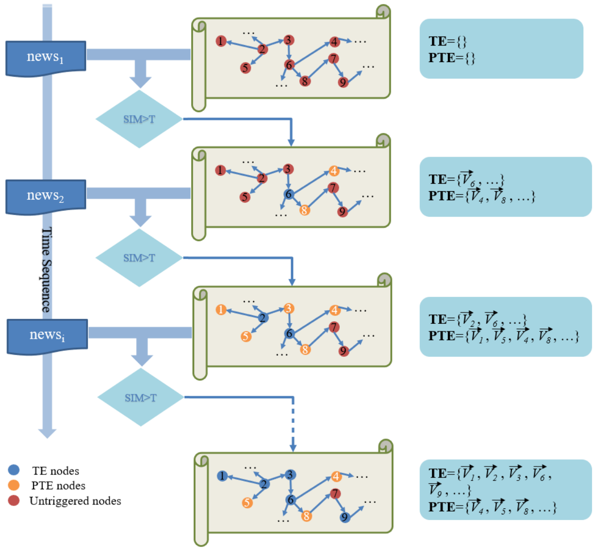

news$_1$
news$_2$
news$_i$
SIM>T
SIM>T
SIM>T
Time Sequence
TE={}
PTE={}
TE=$\{\overrightarrow{V_6}, \ldots\}$
PTE=$\{\overrightarrow{V_4}, \overrightarrow{V_8}, \ldots\}$
TE=$\{\overrightarrow{V_2}, \overrightarrow{V_6}, \ldots\}$
PTE=$\{\overrightarrow{V_1}, \overrightarrow{V_5}, \overrightarrow{V_4}, \overrightarrow{V_8}, \ldots\}$
TE=$\{\overrightarrow{V_1}, \overrightarrow{V_2}, \overrightarrow{V_3}, \overrightarrow{V_6}, \overrightarrow{V_9}, \ldots\}$
PTE=$\{\overrightarrow{V_4}, \overrightarrow{V_5}, \overrightarrow{V_8}, \ldots\}$
TE nodes
PTE nodes
Untriggered nodes

FIGURE 5. The process of topic tracking (with tracking distance = 1).

B. TRACKING ALGORITHM

The core task of topic tracking is to determine which newly-coming news events are semantically related to a identified topic, which we use a threshold value to measure. When a sequence of news events are input to the tracking system, the news events that are similar to the original tracking model with a similarity higher than a given threshold value will be identified (be tracked), and the topic tracking model will be updated dynamically by accumulating features of the newly tracked events. The topic tracking algorithm is described in algorithm 4, Fig.5 illustrates the process of topic tracking according to algorithm 4.

As we all know, the keywords in a text usually represent the core semantics of the text to a large extent. In algorithm 4, we use keywords to calculate the vectors of each news text and assign a certain weight λ to the keywords. In the process of tracking, the model will change weight distribution by triggering event scenarios that integrated in the topic tracking model, which gives the model self-updating ability.

Algorithm 4 Topic Tracking Algorithm of TDTEO

```
Require:
    1. A topic tracking model for specified topic.
    2. A news text (event) set $NS$ entered in time sequence.
    3. A threshold as $T$ that triggers tracking, it is also the similarity threshold between tracking model and news text.
    4. Distance $d$, it represents the distance of topic evolution.
Ensure:
1: By using algorithm 3, construct model vector group $\mathbf{TMV} = \{\overrightarrow{V_c} | c \in \{1, \ldots, k\}\}$ of specified topic tracking model. Create a list TL for tracked news, an empty vector group TE for triggered event node and an empty vector group PTE for event nodes that follow the triggered nodes within distance $d$ (called tracking distance), then calculate the base vector of the topic tracking model as $V_{TP}$ by formula 7:
       $\overrightarrow{V_{TP}} = \frac{1}{k} \sum_{c=1}^{k} \overrightarrow{V_c}$    (7)
2: For each news in $NS$, we use $NS_i$ to represent the $i^{th}$ event of $NS$, and we divide each $NS_i$ text into $u$ words after removing stop words, extract $v$ keywords of $NS_i$ by TF-IDF, and then calculate the weight vector $\overrightarrow{V_{E_i}}$ by formula 8:
       $\overrightarrow{V_{E_i}} = \frac{1-\lambda}{u} \sum_{t=1}^{u} \overrightarrow{V_{w_t}} + \frac{\lambda}{v} \sum_{t=1}^{v} \overrightarrow{V_{k_t}}$    (8)
   where $\overrightarrow{V_{w_t}}$ presents the vector of the $t^{th}$ of $u$ words,$\overrightarrow{V_{k_t}}$ presents the vector of the $t^{th}$ of $v$ keywords, $\lambda$ denotes the offset ratio of keywords.
3: Calculate the similarity between $V_{TP}$ and $V_{E_i}$,if $SIM_{\overrightarrow{V_{TP}}\overrightarrow{V_{E_i}}} < T$, jump to step 2 and calculate the next news text. If $SIM_{\overrightarrow{V_{TP}}\overrightarrow{V_{E_i}}} >= T$, the $NS_i$ belongs to the topic, add $NS_i$ to TL.
4: If the relationship $j = \arg\max_{c} SIM_{\overrightarrow{V_c}\overrightarrow{V_{E_i}}}$ is satisfied, add $\overrightarrow{V_j}$ to TE. Find the event node $node_j$ corresponding to $\overrightarrow{V_j}$ in the topic tracking model, and check all subsequent nodes with distance less than $d+1$, then add the vectors of these nodes to PTE. If PTE contains $\overrightarrow{V_j}$, pop it out, then update the current topic scenario vector by formula 9 as new $\overrightarrow{V_{TP}}$:
       $\overrightarrow{V_{TP}} = \frac{\sum_{t=1}^{l_1} \overrightarrow{V_{TE_t}} + \sum_{t=1}^{l_2} \overrightarrow{V_{PTE_t}}}{l_1 + l_2}$    (9)
   where $l_1$ and $l_2$ represent the size of TE and PTE, $\overrightarrow{V_{TE_t}}$ is the $t^{th}$ vector of TE, $\overrightarrow{V_{PTE_t}}$ is the $t^{th}$ vector of PTE.
5: Repeat step 2 to step 4 until all news in $NS$ are tracked.
6: return TL
```

VI. EXPERIMENTS AND ANALYSIS

A. MODELS AND DATASETS

We construct an event hierarchy model including five major types of event classes (*Sports*, *Military*, *Entertainment*, *Society*, *Science and Technology*) and 42 fine-grained event classes according to the structure of event ontology. Fig.2 shows a part of the event hierarchy model of *"Science and Technology Events"*, which be used to construct a topic detection model by using algorithm 1. We select *"Expose Paper Plagiarism"* as a specific domain for topic tracking, and construct an event scenerio model of *"Expose Paper Plagiarism"* as Fig.4, which is used to construct the topic tracking model by using algorithm 3.

TABLE 2. Hyperparameters of pre-trained models.

Name	Window size	Negative sampling	CBOW	Size
w2v1	5	5	0	100
w2v2	5	5	1	100
GloVe	15	–	–	100
FastText	5	5	0	100

Window size means context window size. *Negative size* represents the use of negative sampling to optimize computational speed. *CBOW*=0 means that the model is trained using the continuous skip-gram. *Size* represents the dimension of the output word vector. "*w2v1*" represents a Word2Vec model trained using skip-gram, "*w2v2* represents a Word2Vec model trained using CBOW.

We use Word2Vec (CBOW and skip-gram), GloVe and FastText to train Sogou news corpus [43] after word segmentation and stop words removal. The hyperparameters we selected are shown in TABLE 2. We collect 21,778 Chinese news texts from *Global News*[1] from January to July 2019. They are divided into five main topics for topic detection experiment: *Sports (1,124), Military (5,412), Entertainment (4,714), Society (6,683), Science and Technology (3,845)*. We labeled 2,048 news texts of *Science and Technology* with 42 fine-grained topics to verify our method is also effective to detect fine-grained topics. We labeled 138 events of *Zhaitianlin's plagiarism* in 3,845 news for topic tracking experiment.

For topic detection, we used the overall accuracy, F1 value of various topics, the macro-precision, the macro-recall and the macro-F1 to evaluate effectiveness of topic detection. For topic tracking, the evaluation method in TDT2004 [44] is usually adopted. The proposed method is measured and evaluated according to the loss rate and false positive rate of the relevant news in the topic detection and tracking results. They can be calculated as formula 10:

$$C_{track} = C_{miss} P_{miss} P_{target} + C_{FA} P_{FA} P_{non-target} \quad (10)$$

where C_{miss} and C_{FA} are the costs of misses and false alarms, P_{target} and $P_{non-target}$ are a priori probabilities of whether a news is related to a tracking topic. In addition, P_{miss} is the loss rate of the tracking news and P_{FA} is the false positive rate of the tracking news. We use $(C_{Det})_{Norm}$ to evaluate the result of

[1] https://huanqiu.com/

TABLE 3. The experiment results for detection of 5 major topics.

Method	Overall Accuracy(%)	Sports(%)	Military(%)	Entertainment(%)	Society(%)	Technology and Science(%)	Time-consumption(s)
TF-IDF+K-means	41.58	14.68	96.27	50.00	4.02	11.94	2304
w2v1+K-means	89.12	94.58	93.08	91.79	87.72	81.74	353
w2v2+K-means	88.25	93.53	92.67	90.38	85.45	82.75	346
GloVe+K-means	89.35	94.67	93.55	91.79	88.51	80.46	357
LDA+30% dataset	70.51	67.54	81.00	82.59	51.93	74.09	279
LDA+70% dataset	84.78	89.32	92.56	85.23	80.23	79.88	599
w2v1+1% for 5	86.52	89.27	86.45	92.63	89.01	72.87	**94**
w2v1+1% for 10	86.05	91.37	91.65	91.74	82.99	75.00	98
w2v1+10% for 20	88.48	93.81	91.95	93.02	87.02	79.20	100
w2v1+10% for 100	90.67	94.06	93.50	93.77	89.74	83.65	101
w2v2+1% for 5	84.17	88.26	84.32	90.26	86.84	70.68	96
w2v2+1% for 10	85.90	88.73	85.57	91.32	88.35	74.66	95
w2v2+10% for 20	87.09	90.54	88.43	92.79	87.54	76.32	99
w2v2+10% for 100	89.31	94.06	93.50	93.77	89.74	83.65	103
GloVe+1% for 5	85.31	90.97	89.80	90.58	85.77	65.14	125
GloVe+1% for 10	90.07	93.08	92.44	93.64	89.30	83.45	129
GloVe+10% for 20	91.16	92.02	93.84	90.88	90.66	85.30	133
GloVe+10% for 100	**91.93**	94.30	94.76	93.48	91.29	86.91	136
FastText+1% for 5	82.36	86.23	83.58	85.56	87.98	65.83	253
FastText+1% for 10	83.66	87.58	85.57	87.66	86.84	69.42	275
FastText+10% for 20	86.68	89.59	86.62	89.56	91.54	74.84	279
FastText+10% for 100	89.30	91.58	90.44	91.76	89.96	82.88	283

"TF-IDF" means using TF-IDF vectorization, "30% dataset" means using 30% dataset as training data to train the LDA model [37], then use the model to calculate indicators. "1% for 5" represents using 1% of the dataset as training data and expand 5 keywords to create the topic model, the others are the same.

topic tracking methods by formula 11:

$$(C_{Det})_{Norm} = \frac{C_{track}}{min(C_{miss}P_{target}, P_{FA}P_{non-target})} \tag{11}$$

$(C_{Det})_{Norm}$ represents the error recognition cost of the tracking system. The smaller value of $(C_{Det})_{Norm}$, the better system tracking performance. In general, P_{target}, C_{miss}, C_{FA} and $P_{non-target}$ are set to the following values: 0.02, 1, 0.1 and 0.98.

B. TOPIC DETECTION EXPERIMENT

In order to verify the effectiveness of the TDTEO methods, we compare TDTEO method with K-means and LDA [33], [45] by detection experiments of five major types topics (*Sports*, *Military*,*Entertainment*, *Society*, *Science and Technology*) and 42 fine-grained topics in *Science and Technology Events*.

In the detection experiment based on K-means [46], we assume that the number of clustering centers is the number of topics to be detected. We use different vectorization models with TDTEO method. In TDTEO, topic models are constructed from linguistic expression keywords of event and event elements from event ontology. We use TF-IDF to extend 5,10,20 and 100 keywords based on datasets with a ratio which are selected randomly and constructing the topic centers by algorithm 1. We compare the overall accuracy, F1 value and running cost of topic detection in this experiment.

According to the experiment results in TABLE 3, the time-consumption of the method based on TF-IDF vectorization is much higher than that of the method based on pre-trained models. The vectorization of TF-IDF needs to be initialized, which costs more time. It also can be seen that TF-IDF method has a good effect in detecting topics about the military, due to the features of topics related to the military having a high differentiation from other topics.

TABLE 3 also shows that LDA is obviously effective, but it takes more time to train the data for calculating the topics distribution. When the ratio of training dataset increases, the effect is significantly improved, but time-consumption increases obviously. The overall accuracy of LDA reaches 84.78% when the ratio of training dataset achieves 70%, but it is worse than TDTEO method based on vectorization model like Word2Vec. The vectorization model is generally a pre-trained model contained rich semantic information, and its time-consumption is lower than TF-IDF and LDA, so it has better performance in TDT tasks of real-time data flow.

The TDTEO method proposed in this paper has an overall accuracy of 86.52% when using *w2v1* with 1% training dataset to expand 5 keywords. With the increase of the proportion of used data and the expansion of keywords, there is a small increase of the time-consumption, and the overall accuracy is close to K-means. When using 10% of training dataset to expand 100 keywords in the topic detection model, the overall accuracy achieves 90.67%, and the time-consumption is 101 seconds, which is only 28.58% of the time-consumption of K-means method. The TDTEO method based on *w2v1* and *w2v2* reaches a competitive overall accuracy, and the accuracy result of TDTEO method based on *GloVe* reaches 91.93%. In the TDTEO topic detection method, topic centers could be constructed quickly while the topic model maps to vector space, which are similar to the clustering centers in K-means method. In addition, the computational complexity of topic centers found in TDTEO is only $O(n)$, which is better than K-means method with the computational complexity $O(n^2)$ for finding cluster center.

We select 2,048 news texts from topic *Science and Technology* for fine-grained topic detection experiment. The dataset is manually labeled with 42 topics according to the topic model that we constructed from event class hierarchy of *Science and Technology Events* shown as Fig. 2. We use macro-precision, macro-recall and macro-F1 as indicators to evaluate the results of the experiment, as shown in TABLE 4. The macro-precision of the K-means algorithm based on *FastText* is only 33.65%, and the macro-recall is 72.38%. So, the macro-F1 value is generally lower, it means K-means method based on the vectorization performs poorly in the fine-grained topic detection experiments. LDA also has a low macro-precision and high macro-recall, and its macro-F1 value is 45.56%, which is better than K-means method. The TDTEO method with *w2v1* has the best result with 65.03% macro-precision, 75.23% macro-recall and 69.76% macro-F1 value. As well, *w2v2*-based model performs well. Though all indicators decreased significantly when using TDTEO with *GloVe*, the macro-F1 achieves 48.93%, which is better than LDA. This shows that the TDTEO method based on pre-trained models performs better in fine-grained topic detection tasks, and Word2Vec models are more competitive than *GloVe* and *FastText*.

TABLE 4. The experiment results comparison of fine-grained topics detection.

Method	macro-P	macro-R	macro-F1
w2v1+K-means	0.3388	0.7238	0.4616
w2v2+K-means	0.3674	0.7023	0.4824
GloVe+K-means	0.3413	0.7456	0.4683
FastText+K-means	0.3365	0.7199	0.4425
LDA+70%dataset	0.4556	0.6432	0.5334
w2v1+10% for 100	**0.6503**	**0.7523**	**0.6976**
w2v2+10% for 100	0.6453	0.7301	0.6851
GloVe+10% for 100	0.4893	0.6532	0.5595
FastText+10% for 100	0.4651	0.6189	0.5358

"*w2v1*+K-mean" means K-means based on *w2v1*, "10% for 100" represents using 10% of detected topics and extracting 100 keywords to update topic models.

In order to study the influence of the training dataset ratio and the number of extended keywords on the detection effect, we choose 6 different training dataset ratios and 7 different keyword expansion numbers to compare the accuracy of topic detection. Fig. 6 shows the effect on the detection accuracy with different models while adjusting the training

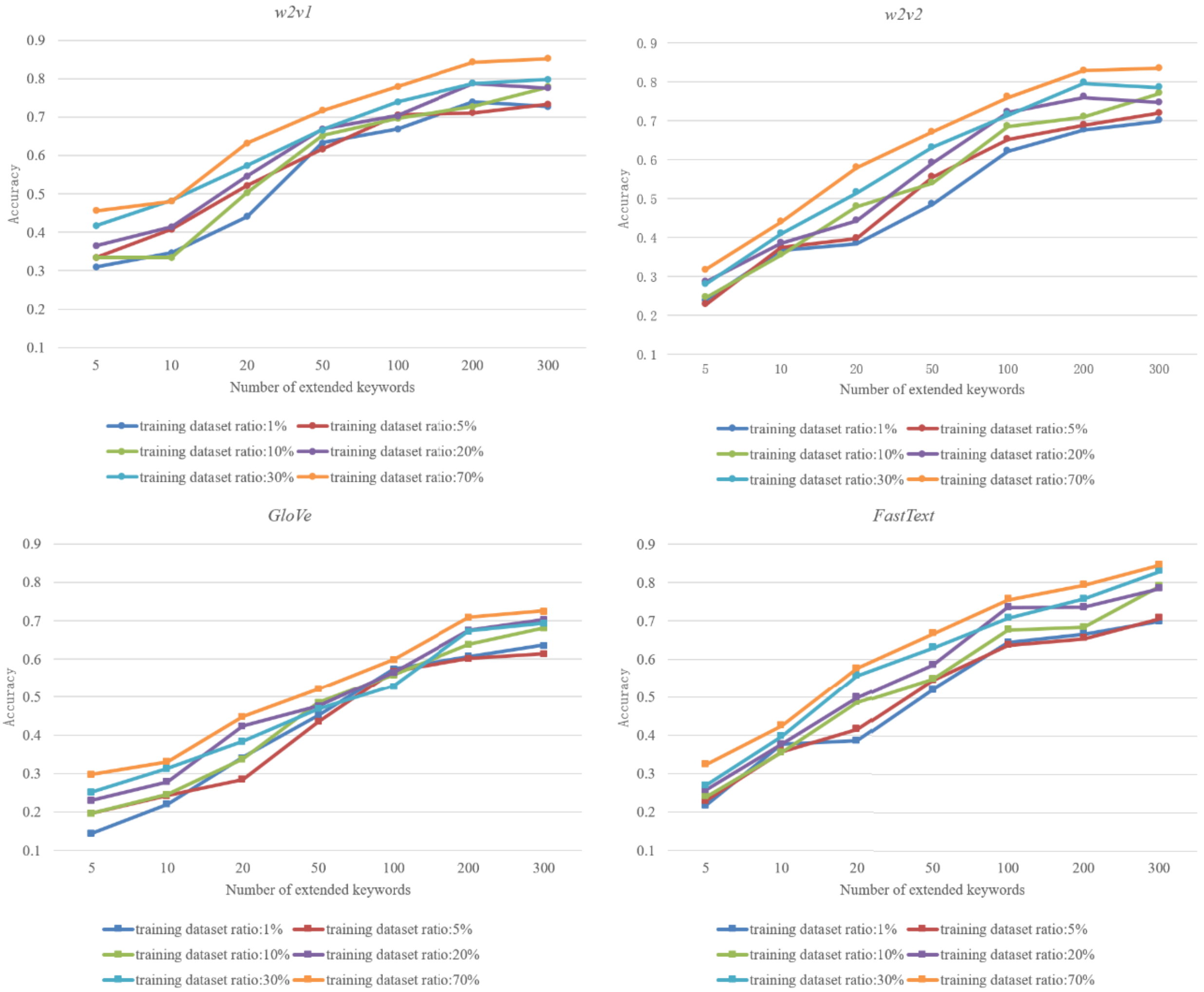


FIGURE 6. Accuracy comparison of topic tracking based on four models with different training dataset ratios and different number of extended keywords.

dataset ratio and the number of expanded keywords. When the training dataset ratio increases, the accuracy also increases slightly. This effect can be explicitly observed by comparing the training dataset ratios with 1%,5%,... and 70%. When the number of extended keywords increases slightly, the accuracy also increases. When the number of extended keywords reaches 300 and the ratio of training dataset ratios is 70%, the accuracy achieves 85.25% by using *w2v1* model. From the trends in Fig. 6, we can see that *GloVe* and *FastText* work effectively when the number of extended keywords and training percent increase. However, they can not achieve the effect of *w2v1* and *w2v2* models.

C. TOPIC TRACKING EXPERIMENT

According to the topic detection experiments, we find that Word2Vec models are always better than other models, and the *w2v1* model trained by skip-gram method is better in topic detection, so we only choose *w2v1* for topic tracking experiment.

In the topic tracking experiment, we use the indicator $(C_{Det})_{Norm}$ to evaluate the effect of topic tracking. We choose a specific topic model as the target topic (about *Zhai tianlin, a Chinese actor, who was revealed of plagiarizing in his doctoral dissertation*) to be tracked, we choose 138 events of *Zhaitianlin's plagiarism* in 3845 news for topic tracking experiments. The topic started on January 31th and lasted until June 30th in 2019.

TABLE 5. The experiment results comparison of topic tracking with different parameters.

Methods	P_{miss}	P_{FA}	$(C_{Det})_{Norm}$
d=1, T=0.5	0.0623	0.1224	**0.1352**
d=2, T=0.5	0.0652	0.1556	0.1689
d=3, T=0.5	0.0434	0.2113	0.2202
d=1, T=0.7	0.1159	0.1309	**0.1546**
d=2, T=0.7	0.1012	0.1531	0.1737
d=3, T=0.7	0.0809	0.1851	0.2016
d=1, T=0.9	0.6812	0.0520	**0.1910**
d=2, T=0.9	0.5354	0.0910	0.2003
d=3, T=0.9	0.4790	0.1434	0.2411

"d=1" means updating tracking model by combining subsequent events with distance 1, "T=0.5" means that the trigger threshold is 0.5 .

We calculate P_{miss}, P_{FA} and $(C_{Det})_{Norm}$ while using different distances of subsequent events information and different trigger threshold to update topic tracking model. The experimental results are shown in TABLE 5. When the trigger threshold is lower as 0.5, the value of P_{miss} is usually lower, which means the input news text is easy to be matched to the target topic. With the trigger threshold increases, some news texts can't match the corresponding topic, resulting in the increase of P_{miss} and the decrease of P_{FA}. In topic tracking experiment, we update the tracking model of *Zhaitianlin's plagiarism* with subsequent events with distance of 1, and set the threshold as 0.5, 0.7 and 0.9, which result in P_{miss} increases obviously, and P_{FA} decreases. However, due to the high weight of P_{FA} in $(C_{Det})_{Norm}$, $(C_{Det})_{Norm}$ does not increase significantly. It come to a conclusion that increasing T can effectively decrease P_{FA}, but at the cost of increasing of P_{miss}. According to the experimental results, we also find that in the process of tracking, using different distances of subsequent events information to update the topic model will result in different topic tracking effect. When all thresholds are set to 0.5, the P_{FA} is 0.2113 while updating the topic model by using the subsequent events information with distance of 3. In the scenario model, the farther the subsequent events are from the trigger event, the greater the semantic distances are from the latest topic tracking model that has not been updated, so it will increase the false rate. According to the experimental results, when updating the topic model by using the subsequent events information of the trigger event with distance of 1, its $(C_{Det})_{Norm}$ is as low as 0.1352. This shows that in the process of topic tracking, the topic evolution can be predicted according to the directly subsequent events information (distance of 1) in the knowledge base, which usually result in higher tracking accuracy and higher topic aggregation.

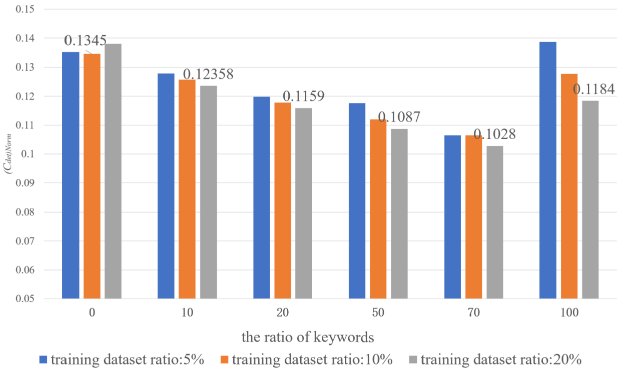


FIGURE 7. $(C_{Det})_{Norm}$ **based on different keywords ratios and training dataset ratios.**

According to TABLE 5, $(C_{Det})_{Norm}$ achieves the lowest value (means best tracking effect) when $T = 0.5$ and $d = 1$. Also, we find that different training set ratio and the ratio of keywords extracted from training set have different effects on $(C_{Det})_{Norm}$. Fig.7 shows $(C_{Det})_{Norm}$ values for different ratios of keywords when ratio of training set is 5%, 10% and 20% respectively. It can be seen that the effect of topic tracking achieve best when the training set ratio is around 20% and the ratio of keywords is around 70%, and the minimum of $(C_{Det})_{Norm}$ is around 0.1028. The experimental results show that in a certain range, the higher the ratio of keywords is, the better the tracking effect is. However, when ratio of keywords reaches 100%, the effect of topic tracking decreases, because the high ratio of keywords is prone to semantics drift, and it is difficult to trigger subsequent events to update the tracking model.

VII. CONCLUSION

In this paper, we propose a new TDT method based on event ontology for hierarchical topic detection and tracking topic evolution. Event class hierarchy model and event scenario models in event ontology are used for topic detection and topic tracking, respectively. By using event information in event class hierarchy models, we can construct a hierarchical topic model for topic detection. Hierarchical topic models can overcome the shortcomings of scattered, lacking organization and uneven granularity of topics in traditional topic detection. The event scenario model in the event ontology can be used to predict the occurrence of subsequent events in advance, thereby updating the topic model by accumulating features of the subsequent events, and eventually to improve the effectiveness of topic tracking. Experimental results show that the proposed method can effectively improve the accuracy of topic detection and tracking. In the proposed method, the construction of domain event ontology will inevitably bring extra labor costs, even high costs for the detection and tracking of topics. Fortunately, with more attention paid to the importance of event-based knowledge base, event ontologies in different domains will be constructed and reused to the greatest extent, such as domain of natural disaster, public security, prevention and control of infectious diseases, etc. Meanwhile, with the increasing attention paid to the construction of common event knowledge bases from the semantic web community, more and more issued open event knowledge bases will gradually reduce the cost.

REFERENCES

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study final report," in *Proc. Darpa Broadcast News Transcription Understand. Workshop*, 1998, pp. 194–218. [Online]. Available: http://repository.cmu.edu/compsci/341

[2] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," in *Topic Detection and Tracking: Event-Based Information Organization*. 2002.

[3] L. Shi, J. Du, and M. Liang, "Social network bursty topic discovery based on RNN and topic model," *J. Commun.*, vol. 39, no. 4, pp. 189–198, 2018.

[4] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1998, pp. 28–36.

[5] G. Kumaran and J. Allan, "Using names and topics for new event detection," in *Proc. Conf. Human Lang. Technol. Empirical Methods Natural Lang. Process. (HLT)*. Stroudsburg, PA, USA: Association Computational Linguistics, 2005, pp. 121–128.

[6] Z. Li, B. Wang, M. Li, and W.-Y. Ma, "A probabilistic model for retrospective news event detection," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*. Stroudsburg, PA, USA: Association Computational Linguistics, 2005, pp. 106–113.

[7] L. Wei-Hua and X. Yuman-Quan, "The study of topic detection based on algorithm of division and multi-level clustering with multi-strategy optimization," *J. Chin. Inf. Process.*, vol. 20, no. 1, pp. 29–36, 2006.

[8] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *Proc. 1st Instructional Conf. Mach. Learn.*, Piscataway, NJ, USA, vol. 242, Dec. 2003, pp. 133–142.

[9] X. Gui *et al.*, "Survey on temporal topic model methods and application," *Comput. Sci.*, vol. 2, Jun. 2017.

[10] Y. Wang, J. Liu, Y. Huang, and X. Feng, "Using hashtag graph-based topic model to connect semantically-related words without co-occurrence in microblogs," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1919–1933, Jul. 2016.

[11] K. Zhang, J. Zi, and L. G. Wu, "New event detection based on indexing-tree and named entity," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*. New York, NY, USA: ACM, 2007, pp. 215–222.

[12] X. D. Ren, Y. K. Zhang, and X. F. Xue, "Adaptive topic tracking technique based on K-modes clustering," *Comput. Eng.*, vol. 35, no. 9, pp. 222–224, May 2009.

[13] L. Jiang, H. Zhang, and X. Yang, "Research on semantic text mining based on domain ontology," in *Proc. Int. Conf. Comput. Comput. Technol. Agricult.* Berlin, Germany: Springer, 2012, pp. 336–343.

[14] Z. T. Liu, M. Huang, W. Zhou, Z. Zhong, J. Fu, J. Shan, and H. Zhi, "Research on event-oriented ontology model," *Comput. Sci.*, vol. 36, no. 11, pp. 189–192, 2009.

[15] X. Rong, "Word2vec parameter learning explained," *Comput. Sci.*, vol. 39, no. 3, pp. 359–378, 2014.

[16] V. Vargas-Calderón and J. E. Camargo, "Characterization of citizens using word2vec and latent topic analysis in a large set of tweets," *Cities*, vol. 92, pp. 187–196, Sep. 2019.

[17] Z. Sheng, Z. Xin, C. Jia-Jun, and W. Hui, "Chinese sentiment classification using extended word2vec," *J. Donghua Univ.*, vol. 28, pp. 142–145, 2016.

[18] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[19] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.Zip: Compressing text classification models," 2016, *arXiv:1612.03651*. [Online]. Available: http://arxiv.org/abs/1612.03651

[20] Y. Chen and L. Liu, "Development and research of topic detection and tracking," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Aug. 2016, pp. 170–173.

[21] J. Zeng and S. Zhang, "Incorporating topic transition in topic detection and tracking algorithms," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 227–232, Jan. 2009.

[22] W. Ding, Y. Zhang, C. Chen, and X. Hu, "Semi-supervised Dirichlet-Hawkes process with applications of topic detection and tracking in Twitter," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 869–874.

[23] L. Chen, H. Zhang, J. M. Jose, H. Yu, Y. Moshfeghi, and P. Triantafillou, "Topic detection and tracking on heterogeneous information," *J. Intell. Inf. Syst.*, vol. 51, no. 1, pp. 115–137, Aug. 2018.

[24] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," *Comput. Sci.*, vol. 1, no. 4, pp. 39–44, 2015.

[25] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2015, pp. 649–657.

[26] N. Pappas and A. Popescu-Belis, "Multilingual hierarchical attention networks for document classification," in *Proc. IJCNLP*, 2017, pp. 1–11.

[27] Y. Zhou *et al.*, "Hierarchical hybrid attention networks for Chinese conversation topic classification," in *Proc. Int. Conf. Neural Inf. Process.*, Nov. 2017, pp. 540–550.

[28] Y. Zhang, J. Zheng, Y. Jiang, G. Huang, and R. Chen, "A text sentiment classification modeling method based on coordinated CNN-LSTM-Attention model," *Chin. J. Electron.*, vol. 28, no. 1, pp. 120–126, Jan. 2019.

[29] R. Papka, "On-line new event detection, clustering, and tracking," Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, 1999.

[30] H. Chen, J. Lu, F. Wang, Y. Zhang, and S. Zhao, "A new method of topic tracking for micro-blog texts based on semantic relevance," in *Proc. 9th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, vol. 2, Aug. 2017, pp. 349–353.

[31] Y. Rao, Q. Li, Q. Wu, H. Xie, F. L. Wang, and T. Wang, "A multi-relational term scheme for first story detection," *Neurocomputing*, vol. 254, pp. 42–52, Sep. 2017.

[32] M. Franz, T. Ward, J. S. McCarley, and W.-J. Zhu, "Unsupervised and supervised clustering for topic tracking," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2001, pp. 310–317.

[33] J.-F. Yeh, Y.-S. Tan, and C.-H. Lee, "Topic detection and tracking for conversational content by using conceptual dynamic latent Dirichlet allocation," *Neurocomputing*, vol. 216, pp. 310–318, Dec. 2016.

[34] Y. Cai, H. Xie, R. Y. K. Lau, Q. Li, T.-L. Wong, and F. L. Wang, "Temporal event searches based on event maps and relationships," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105750.

[35] G. Xu, Y. Meng, Z. Chen, X. Qiu, C. Wang, and H. Yao, "Research on topic detection and tracking for online news texts," *IEEE Access*, vol. 7, pp. 58407–58418, 2019.

[36] V. Vargas-Calderón, M. S. Dominguez, H. Vinck-Posada, and J. E. Camargo, "Using machine learning and information visualisation for discovering latent topics in Twitter news," 2019, *arXiv:1910.09114*. [Online]. Available: http://arxiv.org/abs/1910.09114

[37] S. Syed and M. Spruit, "Full-text or abstract? Examining topic coherence scores using latent Dirichlet allocation," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2017, pp. 165–174.

[38] J. Makkonen, "Investigations on event evolution in TDT," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol. Proc. HLT-NAACL Student Res. Workshop (NAACL)*, vol. 3. Stroudsburg, PA, USA: Association Computational Linguistics, 2003, pp. 43–48.

[39] W. Liu, Z. Liu, J. Fu, R. Hu, and Z. Zhong, "Extending owl for modeling event-oriented ontology," in *Proc. Int. Conf. Complex, Intell. Softw. Intensive Syst.*, Feb. 2010, pp. 581–586.

[40] Z. Dong, Q. Dong, and C. Hao, *HowNet and the Computation of Meaning*. Princeton, NJ, USA: Citeseer, 2006.

[41] S. Li, X. Lv, T. Wang, and S. Shi, "The key technology of topic detection based on K-means," in *Proc. Int. Conf. Future Inf. Technol. Manage. Eng.*, vol. 2, Oct. 2010, pp. 387–390.

[42] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. London, U.K.: Chapman & Hall/CRC, 2012.

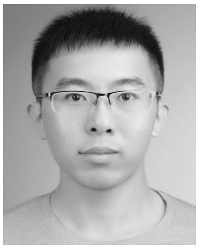
[43] C. Wang, M. Zhang, S. Ma, and L. Ru, "Automatic online news issue construction in Web environment," in *Proc. 17th Int. Conf. World Wide Web (WWW)*. ACM, 2008, pp. 457–466.

[44] V. Giedraitis, H. Modin, M. Callander, A. M. Landtblom, R. Fossdal, K. Stefansson, J. Hillert, and J. Gulcher, "Genome-wide TDT analysis in a localized population with a high prevalence of multiple sclerosis indicates the importance of a region on chromosome 14q," *Genes Immunity*, vol. 4, no. 8, pp. 559–563, Dec. 2003.

[45] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[46] T. W. S. Chow, H. Zhang, and M. K. M. Rahman, "A new document representation using term frequency and vectorized graph connectionists with application to document retrieval," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12023–12035, 2009.

WEI LIU was born in 1978. He received the Ph.D. degree from Shanghai University, in 2005. He was a Senior Visiting Scholar with the Department of Computer Science, Wright State University, USA, from 2013 to 2014. He is currently an Associate Professor with Shanghai University. His current research interests include knowledge representation and reasoning, semantic web and ontology technologies, knowledge graph, and so on. He is a member of CCF.



LEI JIANG was born in 1996. He graduated from Shanghai University, where he is currently pursuing the master's degree. His research interests include natural language processing, machine learning, and deep learning.

YUSEN WU was born in 1997. He is currently pursuing the M.S. degree with Shanghai University. His research interests include knowledge representation, deep learning, and so on.

TINGTING TANG was born in 1996. She is currently pursuing the bachelor's degree with Shanghai University. Her research interests include natural language processing, machine learning, and deep learning.

WEIMIN LI was a JSPS Research Fellow with the Department of Human Informatics and Cognitive Sciences, Waseda University, Japan, from 2012 to 2013. He was a Visiting Scholar with the Department of Computer Science, University of California at Santa Barbara, supported by the China Scholarship Council, from 2015 to 2016. He is currently an Associate Professor with the School of Computer Engineering and Science, Shanghai University, China. He has been involved with the extensively research works in the fields of computer science, service computing, business process management, and database technology. His current research interests include social computing, data mining and analytics, group behavior modeling and simulating, and service recommendation.