IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ADVANCED DATA MINING METHODS FOR SOCIAL COMPUTING

Received November 30, 2019, accepted December 19, 2019, date of publication December 24, 2019,
date of current version January 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961973

Integrating Hierarchical Attentions for Future Subevent Prediction

LINMEI HU
School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
e-mail: hulinmei@bupt.edu.cn

This work was supported in part by the National Natural Science Foundation of China under Grant 61806020, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT An Event, containing a sequence of subevents, describes a typical thing that happens at a specific time and place. Predicting next probable subevents based on knowledge acquired from large-scale news documents are very important for many real-world applications, such as disaster warning etc. In this paper, we present a novel hierarchical attention based end-to-end model for future (*unknown*) subevent prediction using large-scale historical events. Our model automatically produces a short text which describes a possible future subevent after consuming the texts describing previous subevents. To boost the model's understanding towards subevent sequence, we design a hierarchical LSTM model to compress the knowledge in both the word sequence for a subevent and the subevent sequence for an event. In addition, topic information has been exploited to make context-aware prediction for future subevents. To further consider which subevents and words play a critical role in prediction, we propose a hierarchical attention mechanism to stress on the important previous subevents as well as the the critical words within them. Experimental results on a real-world dataset demonstrate the superiority of our model for future subevent prediction over state-of-the-art methods.

INDEX TERMS Future subevent prediction, hierarchical attentions, subevent sequence.

I. INTRODUCTION

An Event, which describes a specific thing happened at a specific time and place, always consists of a sequence of subevents recording how things developed in detail. Over years, news portals have collected tens of thousands of news articles, and organized them as events. Thus, rich historical data are readily available for us to study various events. The past events do not repeat, but often rhyme. A variety of events typically have similar sequential progressing pattern.

In this paper, we argue that we can learn some common sequential subevent patterns from large-scale historical data. In particular, given a sequence of a few subevents, we aim to automatically predict the future subevent by capturing the sequential transition patterns within an event. The large-scale historical events of various topics help us to learn the sequential transition patterns. As illustrated in Figure 1, taking the event "*Egyptian revolution of 2011*" as an example, given its sequential subevents described by the headlines of news documents,[1] "*Conflict occurred again in Egypt on 22nd, and people plan to hold a million-people march*", and "*Egypt's military will deliver a speech to respond to the conflict between demonstrators and police*", our model generates "*protests*", "*burst*", "*chaos*", "*cause*", "*deaths*", "*injuries*" word by word, which constitutes a possible future subevent. It matches well with a later news report "*The Egyptian protest has caused 32 people dead and more than*

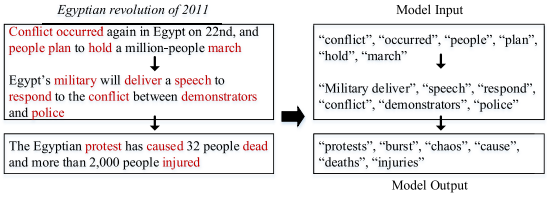

FIGURE 1. An example of future subevent prediction. Our model takes the subevents described by news headlines (after removing stopwords) as input and generates a word sequence which describes the next potential subevent as output.

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

[1]Note that the headlines and model outputs are translated from Chinese into English.

2000 people injured''. Thus, our model is able to predict a future subevent which has not yet happened.

Obviously, it is quite pivotal for governments, companies and news agencies to predict future subevents in advance. Governments can take proactive policy to avoid damages and deaths with the help of predicted subevents. For instance, after an earthquake happens, forecast of chaos could guide the governments to take proactive measures such as arranging more policemen, rather than other reactive actions. In the same way, companies can make strategic decision to better control financial crisis if they know the potential upcoming outcomes beforehand. In addition, news agencies can closely follow those imminent topics of public concern. Thus, future subevent prediction has attracted considerable attention over the past decades.

However, future subevent prediction suffers from the following drawbacks. First, how can we represent an event? Existing works require domain knowledge to extract hand-crafted features for event representation, which is not able to be generalized to other tasks. Events typically consist of dual-level sequential structures: i) word sequence which describes a specific subevent, and ii) subevent sequence illustrating the progress of an event. Therefore, it is a challenging taksk to capture the two-level dependencies in event representation. Second, the latent topic of a future subevent is supposed to be closely related to the topics of previous subevents. Thus topic information is supposed to contribute to next subevent prediction. How can we jointly model both the semantic meanings and topic information i.e. context-aware features within an event? Last, different subevents and words may have different importance for future subevent prediction, how can we pay attention to the previous subevents and words which play a key role in predicting the future subevent?

Although a variety of works have been done on event detection and burst event detection based on social media [27], [34], [36] and search engines [9], only a few works focused on predicting future events. These existing works aimed at target (known) event prediction [25], differently, we work on unknown non-targeted event prediction. Along this line of research, Radinsky et al. [24] extracted the causality relations between two events and generalize them by ontology for predicting events. Recently, with the wide success of neural networks, Granroth-Wilding and Clark [10] extracted event chains [3] from texts and developed a compositional neural network for learning the coherence score of two events. Some works learned a probabilistic language model of event sequences for prediction [18]. Pichotta and Mooney [23] developed a statistical script learning model based on Long Short-Term Memory (LSTM). All the work require hand-crafted features for event representation. Besides, they can only predict future events from given candidate events (choosing from the training set). Hu et al. [15] proposed a generative method to automatically predict the next subevent of an event. In this paper, we propose a hierarchical attention based generative hierarchical LSTM model to predict future subevents.

Given previous subevents represented by news headlines, our model automatically outputs a short text which describes the potential future subevent. The model first reads the words describing a subevent one by one and thus converts a subevent into an *embedding*. Then, it encodes the subevent sequence by taking the vector representations of subevents as input. To enhance the semantic information for prediction, we also incorporate contextual *topic* features of subevents. Additionally, we observe that different previous subevents have quite different influence or impact for future subevent prediction. For example, as the case shown in Fig. 1, the first subevent ''*Conflict occurred again in Egypt on 22nd, and people plan to hold a million-people march*'' is critical in predicting the future subevent ''*The Egyptian protest has caused 32 people dead and more than 2000 people injured*'', while the second subevent ''*Egypt's military will deliver a speech to respond to the conflict between demonstrators and police*'' is less important. If we can pay more attention to the first subevent and the words such as ''conflict'', ''million-people'', and ''march'', we are likely to get more accurate prediction. In this work, to consider which part of the input is most responsible for the current subevent decoding state, we present a hierarchical attention based model CH-LSTM-Att by incorporating a hierarchical attention mechanism. The model links the current decoding state with all the input subevents and words.

Consequently, our main contributions in this paper can be summarized as follows.

1) We propose to predict future potential subevent by automatically generating a short text which describes it.
2) We present a hierarchical attention based neural model CH-LSTM-Att, which not only captures the two-level sequential structures of subevent sequences, but also considers different importance of previous subevents and words within the subevents for future subevent prediction.
3) Experimental results on a real world dataset demonstrate that the model achieves substantial improvements compared with the state-of-the-art methods in the task of future subevent prediction.

The remainder of this paper is organized as follows. In Section II, we formulate the future subevent prediction problem. Section III details our proposed methods. Section IV evaluates the performance of our model. In Section V, we review the related work. At last, conclusion and future work are presented in Section VI.

II. PROBLEM DEFINITION

In this section, we define some concepts and the problem of future subevent prediction.

A. EVENT

An *event* is a particular thing which happened at a specific time and place. It is typically described by a sequence of news articles. Formally, we consider an event $E = \{d_1, \ldots, d_M\}$ as a sequence of M documents, where each document d_m describes a subevent s_m of the event. Therefore,

we can also consider an event as a sequence of subevents $E = \{s_1, \ldots, s_M\}$.

For instance, "2010 Chile earthquake" is an event, which consists of a sequence of news documents describing subevents such as *aftershocks*, *damages and casualties*, *chaos and disorder*, *food scarcity*, and *tsunami*.

B. SUBEVENT

Each subevent s_m is denoted by its description text (e.g., a news document or the title of a news document). The description text contains a sequence of words, i.e. $s_m = (w_{m,1}, \ldots, w_{m,N_m})$, where $w_{m,n} \in \mathcal{V}$ denotes the *n-th* word in the subevent s_m, and $\mathcal{V}$ denotes the vocabulary.

For example, the news document titled "*Tsunami After Major Earthquake Hits Chile*" describes a subevent (*the happening of tsunami*) of the event "2010 Chile earthquake". For simplicity, we can use only the title of the document as the text description of the subevent as it summarizes the key content and main idea of the document.

So far tens of thousands of events have happened and the corresponding reported news documents have been recorded as the events progress. Events of the same (or similar) topic have common sequential transition patterns. For example, in both *earthquake* events and *flood* events, there are sequential topics *rescue effort*, *food scarcity*, *chaos* and so on. With the large scale historical data, we can automatically predict the future subevent given a sequence of observed subevents, which is accomplished by capturing the sequential transition patterns underlying the large-scale historical events. Formally, we define the problem of future subevent prediction as follows.

C. FUTURE SUBEVENT PREDICTION

Given historical events E, where$E = \{d_1, \ldots, d_M\}$, each news document d_m can be considered as a text description of a subevent s_m, we aim to discover the underlying sequential transition pattern among subevents. Specifically, given a sequence of observed subevents $s_1, \ldots, s_{m-1}$ described by texts, we predict the next subevent s_m. To achieve the goal, we learn a probability distribution over all text descriptions describing the next potential subevent s_m. Formally, it can be defined as a language model:

$$P(s_m|s_{1:m-1}) = \Pi_{n=1}^{N_m} P(w_{m,n}|w_{m,1:n-1}, s_{1:m-1}). \quad (1)$$

We propose an end-to-end neural model which takes the previous subevents $s_1, s_2, \ldots s_{m-1}$ described by texts as input and generates a word sequence s_m which describes a possible future subevent. Obviously, a naive method is to concatenate the previous subevents as a whole word sequence, and then apply *n-grams* models to compute conditional probability tables for each word. However, it suffers from the problem of high dimension and thus is intractable for realistic vocabulary size [28]. RNNs (Recurrent Neural Networks) have been proposed to model long *n-gram* contexts [19], while they suffer from the problem of vanishing gradient. To alleviate the problem, LSTM [11] and Gated Recurrent Unit [5] were proposed to improve the RNN models [13]. In the light of this, we build our model based on LSTM.

III. OUR PROPOSED METHODOLOGY

We detail our proposed methodology for the problem of future subevent prediction in this section. The following subsection describes the proposed CH-LSTM-Att model for our future subevent prediction task in turn. Our CH-LSTM-Att captures the content information of a subevent sequence at two levels (i.e. a word sequence for each subevent and a subevent sequence) and the context features by incorporating the topics of the subevents. To improve the prediction performance, our CH-LSTM-Att model adds a hierarchical attention mechanism to consider which part of the words and subevents are most responsible for the current decoding state. Furthermore, the proposed model can learn the representations of subevents automatically, which is generic and can be applied to other tasks.

A. CH-LSTM-Att

As illustrated in Figure 2, the CH-LSTM-Att model is composed of two LSTM encoders for respectively encoding word sequence and subevent sequence, and an LSTM decoder for predicting next subevent. Given a sequence of subevent, $(s_1, s_2, \ldots, s_M)$, the subevent-level LSTM encoder first encodes word sequence $((w_{m,1}, \ldots, w_{m,N_m}))$ into dense vector for each subevent s_m. The event-level LSTM encoder then processes each subevent vector iteratively. The last hidden state of the event-level LSTM represents a summary of the event up to the last subevent. Afterwards, a decoder LSTM is used to generate the text description of the next subevent word by word. We detail the overall process as follows.

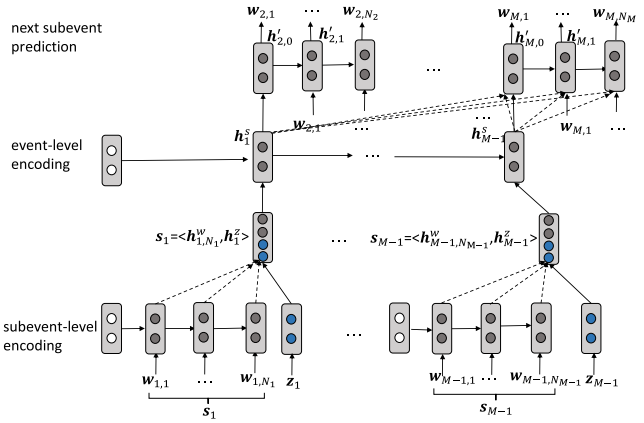

FIGURE 2. Illustration of CH-LSTM-Att model. It consists of three parts, including a subevent-level LSTM encoder, an event-level LSTM encoder and a LSTM decoder. The subevent-level LSTM encoder generates hidden state for each subevent. And, the hidden vector of a specific subevent is concatenated with its corresponding topic embedding to get representation for the subevent. The event-level LSTM encoder takes subevent representations one at a time and update its internal states iteratively to output vectors accounting for subevent sequences. Finally, a LSTM decoder is utilized to decode the text description of the next subevent. During decoding, we consider which subevents and which words within the subevents are most responsible for current-step word generation.

1) SUBEVENT-LEVEL ENCODING

The subevent-level LSTM encoder sequentially consumes the words of a subevent $s_m = (w_{m,1}, \ldots, w_{m,N_m})$, and updates the hidden vector:

$$\mathbf{h}^w_{m,n} = \text{LSTM}^w_{enc}(\mathbf{h}^w_{m,n-1}, \mathbf{w}_{m,n}), \quad n = 1, \ldots, N_m, \quad (2)$$

where $\mathbf{h}^w_{m,n} \in \mathcal{R}^D$ denotes the recurrent state. Initially, we set $\mathbf{h}^w_{m,0} = \{0\}$. The function LSTM^w_{enc} denotes the LSTM function for encoding a word sequence. The last recurrent state $\mathbf{h}^w_{m,N_m}$ can represent the subevent s_m, i.e., $\mathbf{s}_m = \mathbf{h}^w_{m,N_m}$. Overall, the subevent LSTM encoder projects a subevent into an embedding vector without any hand-crafted features.

2) EVENT-LEVEL ENCODING

After learning the representations of subevents, we learn the representations of events which consists of a sequence of subevents. The event-level LSTM encoder takes the subevent embeddings $(\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_m)$ obtained from the subevent-level encoder as input and calculates the event-level recurrent states:

$$\mathbf{h}^s_m = \text{LSTM}^s_{enc}(\mathbf{h}^s_{m-1}, \mathbf{s}_m), \quad m = 1, \ldots, M, \quad (3)$$

where $\mathbf{h}^s_m \in \mathcal{R}^D$ denotes the event-level recurrent state. Initially, we set $\mathbf{h}^s_0 = \{0\}$. The function LSTM^s_{enc} denotes the LSTM function. The event-level recurrent state $\mathbf{h}^s_m$ contains the information of the subevents $(s_1, s_2, \ldots, s_m)$ that have been observed so far.

We further consolidate contextual features, the topics of the subevents by utilizing Latent Dirichlet Allocation (LDA) to learn better representations of subevents. The topics of subevents within a subevent sequence may be particularly related, thus consolidating topic information into subevent representations can leverage the semantic association among the subevents. Via LDA, we can get topic distributions $\{\theta_m\}_{m=1:M}$ of the subevents over K topics, where $\theta_m = \{\theta_{m,k}\}_{k=1:K}$ is a K-dimensional vector. The topic $z_m = \arg\max_k \theta_{m,k}$ with the largest probability is taken as the topic of the subevent s_m. To incorporate the topic information, we represent a subevent as a concatenation of the hidden vectors of words and its topic, namely, $\mathbf{s}_m = \langle \mathbf{h}^w_{m,N_m}, \mathbf{h}^z_m \rangle$.

3) NEXT-SUBEVENT PREDICTION

After getting the representation of a subevent sequence $\mathbf{s}_{1:m-1}$ through the above hierarchical LSTM architecture, we design a LSTM decoder to generate the text description of the next potential subevent $\mathbf{s}_m$. Formally, we aim to estimate the probability $P(\mathbf{s}_m|\mathbf{s}_1, \ldots, \mathbf{s}_{m-1})$ using Eqn.(1).

The desired condition on previous subevents is obtained by using the event-level encoding to initialize the recurrent state of the LSTM decoder, i.e., $\mathbf{h}'_{m,0} = \mathbf{h}^s_{m-1}$, where $\mathbf{h}'_{m,0}$ is the initial recurrent state of the decoder. Formally,

$$\mathbf{h}'_{m,n} = \text{LSTM}_{dec}(\mathbf{h}'_{m,n-1}, \mathbf{w}_{m,n}), \quad n = 1, \ldots, N_m, \quad (4)$$

where $\mathbf{h}'_{m,n-1}$ is the recurrent hidden state of the LSTM decoder. The function LSTM_{dec} is the LSTM function for decoding a word sequence which describes the next potential subevent. In the LSTM decoder, we compute the probability of the next word $\mathbf{w}_{m,n}$ based on the recurrent state $\mathbf{h}'_{m,n-1}$ using a softmax layer. The LSTM decoder will terminate until ⟨end⟩ is predicted. Then the predicted s_m is returned as the final prediction result.

B. HIERARCHICAL ATTENTIONS

We observe in future subevent prediction, the previous subevents and words within the subevents may have different importance. Attention models adopt a look-back strategy by linking the current decoding state with input subevents in an attempt to consider which part of the input is most responsible for the current decoding state. To address the issue, we propose to incorporate a hierarchical attention mechanism.

1) SUBEVENT-LEVEL ATTENTION

In the subevent-level encoding, let $H^w_m = \{\mathbf{h}^w_{m,1}, \ldots, \mathbf{h}^w_{m,N_m}\}$ be the collection of hidden vectors for words in the *m-th* subevent. We take a weighted sum of all the hidden vectors for the words instead of the hidden vector for the last word in the subevent as the representation of the subevent. Formally,

$$\mathbf{s}_m = \sum_{i \in [1,N_m]} \mathbf{a}^w_i \mathbf{h}^w_{m,i} \quad (5)$$

where $\mathbf{a}^w_i$ is the weight of the *i-th* word in the subevent. The weight is determined by both the hidden vector $\mathbf{h}^w_{m,i}$ for the word $w_{m,i}$ and the hidden vector for last-step decoding state. Suppose that $\mathbf{h}'_{n-1}$ denotes the hidden vector for last-step decoding state, the subevent-level attention model would link the current-step decoding information, i.e., $\mathbf{h}'_{n-1}$ with each of the input words in a previous subevent $\mathbf{h}^w_{m,1}, \ldots, \mathbf{h}^w_{m,N_m}\}$, characterized by a strength indicator v^w_i, $i \in [1, N_m]$:

$$v^w_i = \mathbf{U}^T f(\mathbf{W}_1 \mathbf{h}'_{n-1} + \mathbf{W}_2 \mathbf{h}^w_{m,i}) \quad (6)$$

We get the weight $\mathbf{a}^w_i$ by normalizing v^w_i:

$$\mathbf{a}^w_i = \frac{\exp(v^w_i)}{\sum_{i'} \exp(v^w_{i'})} \quad (7)$$

The weights can tell which words in the subevent are more important. The subevent is then represented by averaging weights over all the words within the event.

2) EVENT-LEVEL ATTENTION

Similarly, in the event-level encoding, let $H^s = \{\mathbf{h}^s_1, \mathbf{h}^s_2, \ldots, \mathbf{h}^s_{m-1}\}$ be the collection of hidden vectors for each subevent. Each element in H^s contains information about input subevents with a strong focus on the parts surrounding each specific subevent (time-step). During decoding, suppose that $\mathbf{h}'_{n-1}$ denotes the hidden vector outputted from LSTM_{dec} at previous time step $n-1$, the event-level attention model would link the current-step decoding information, i.e., $\mathbf{h}'_{n-1}$ with each of the input subevents $\mathbf{h}^s_1, \mathbf{h}^s_2, \ldots, \mathbf{h}^s_{m-1}\}$, characterized by a strength indicator v^s_i, $i \in [1, m-1]$:

$$v^s_i = \mathbf{U}'^T f(\mathbf{W}'_1 \mathbf{h}'_{n-1} + \mathbf{W}'_2 \mathbf{h}^s_i) \quad (8)$$

v_i^s is then normalized:

$$\mathbf{a}_i^s = \frac{\exp(v_i^s)}{\sum_{i'} \exp(v_{i'}^s)} \tag{9}$$

The vector $\mathbf{a}_i^s$ stores the weights of all input subevents, which can tell which subevents are more important and which subevents are less important for future subevent prediction. The event-level attention vector is then created by averaging weights over all input subevents:

$$\mathbf{m}_n = \sum_{i \in [1,m-1]} \mathbf{a}_i^s \mathbf{h}_i^s \tag{10}$$

The hidden vectors of CH-LSTM-Att for current step is then achieved by combining $\mathbf{m}_n$, $\mathbf{w}_n$ (the current-step input word) and $\mathbf{h}'(n-1)$:

$$\begin{aligned}
\mathbf{i}_n &= \sigma(\mathbf{W}_{iw}\mathbf{w}_n + \mathbf{W}_{ih}\mathbf{h}'_{n-1} + \mathbf{W}_{im}\mathbf{m}_n + \mathbf{b}_i),\\
\mathbf{f}_n &= \sigma(\mathbf{W}_{fw}\mathbf{w}_n + \mathbf{W}_{fh}\mathbf{h}'_{n-1} + \mathbf{W}_{fm}\mathbf{m}_n + \mathbf{b}_f),\\
\mathbf{o}_n &= \sigma(\mathbf{W}_{ow}\mathbf{w}_n + \mathbf{W}_{oh}\mathbf{h}'_{n-1} + \mathbf{W}_{om}\mathbf{m}_n + \mathbf{b}_o),\\
\mathbf{g}_n &= tanh(\mathbf{W}_{gw}\mathbf{w}_n + \mathbf{W}_{gh}\mathbf{h}'_{n-1} + \mathbf{W}_{gm}\mathbf{m}_n + \mathbf{b}_g),\\
\mathbf{c}_n &= \mathbf{f}_n \odot \mathbf{c}_{n-1} + \mathbf{i}_n \odot \mathbf{g}_n,\\
\mathbf{h}_n &= \mathbf{o}_n \odot tanh(\mathbf{c}_n),
\end{aligned} \tag{11}$$

where $\mathbf{W}_{*w}$ and $\mathbf{W}_{*h}$ denote the transformation matrix from the input to hidden states and the recurrent transformation matrix between the recurrent states $\mathbf{h}_n$ respectively. $\mathbf{b}_*$ represents the bias vector.

Once ⟨end⟩ is predicted, the decoder terminates and the predicted s_m is returned as the final prediction result.

C. TRAINING AND TESTING

By maximizing the log-likelihood of subevents $\{s_m\}_{m\in\{2:M\}}$ given previous ones $s_{1:m-1}$, we learn the model parameters W:

$$\begin{aligned}
\mathcal{L} &= \sum_{e=1}^{E_{train}} \sum_{m=2}^{M} \log P(s_m^e | s_{1:m-1}^e)\\
&= \sum_{e=1}^{E_{train}} \sum_{m=2}^{M} \sum_{n=1}^{N_m^e} P(w_{m,n}^e | w_{m,1:n-1}^e, s_{1:m-1}^e),
\end{aligned} \tag{12}$$

where s_m^e denotes the *m-th* subevent of the *e-th* event and $w_{m,n}^e$ is the *n-th* word in the subevent s_m^e. E_{train} denotes the number of events in the training set. Batch gradient descent was adopted for optimization.

During test time, we use a beam-search (size=1) [31] for forward prediction until ⟨end⟩ is predicted in the decoding process. Beam search selects the word with the largest conditional probability as a new predicted word, which is subsequently combined with preceding output words for next word prediction using the decoder LSTM.

IV. EXPERIMENTS

In this section, we performed comprehensive experiments to evaluate our model for future subevent prediction.

A. DATASET

Following [15], we tested on a Chinese news event dataset from Sina News[2] covering various news series from 1999 to 2016. This dataset contains 15,254 news series, each composed of a sequence of news articles in chronological order reporting on the same event. In average, the number of articles within a news series is 50. For each news article, we only use its headline sine it summarizes the core idea of the news article in a concise way. We further segment each news series with a window of size 5 and get non-overlapping events (partitions). Each news article within an event are viewed as a subevent. We set the size of the window to 5, as we observe from the data that there is little dependency beyond more than 5 continuous subevents. Afterwards, we get 155,358 events overall. We performed tokenization with the existing tool ICTCLAS.[3] Besides, we filtered out stopwords and the words with frequency less than 100 documents. Finally, we get a vocabulary of 4,515 unique words including a special end symbol ⟨end⟩. Each subevent contains about five words on average. We randomly splited the events into 80%, 10% and 10% for training, validation and test, respectively. We show the statistics of the dataset in Table 1.

TABLE 1. Statistics of our dataset.

	Training	Validation	Test
Events	124,288	15,535	15,535
Subevents	607,090	75,802	75,957

1) EXPERIMENTAL SETTING AND EVALUATION METRICS

The hyper-parameters in the models are determined through experiments. We tried different parameter settings and evaluated them on the validation set. Then we selected the best hyper-parameters for evaluation in the independent test set. The optimal parameter settings are as follows.

1) LSTM parameters and word embedding were initialized with a uniform distribution between [−0.08, 0.08];
2) The batch size is set as 32;
3) The learning rate is set as 0.1;
4) The dropout rate is set as 0.2;
5) The dimension of hidden vector D is set as 400, and the dimension of word embeddings and topic embeddings is set as 100;
6) The number of hidden layers of LSTM is set as 2;
7) The topic number = 1,000.

Evaluation Metrics: We select two metrics for evaluating the effectiveness of our proposed models, named *perplexity* and *word error-rate*. Perplexity [29] is closely related with cross entropy loss between the model and test data, which can be viewed as exponential of average per-word entropy of the test data. In this task, it shows how well our model fits the data. Lower perplexity is better. We define per-word perplexity under two different cases. In the first case, we consider

[2] http://news.sina.com.cn/zt/
[3] http://ictclas.nlpir.org/

the subevents from 2 to M in a test event (consisting of M subevents) and compute the perplexity as follows:

$$\text{Perp} = \exp(-\frac{1}{N_w}\sum_{e=1}^{|E_{test}|}\sum_{m=2}^{M}\log P(s_m^e|s_{1:m-1}^e)) \ , \quad (13)$$

where N_w denotes the number of words in the test cases, $|E_{test}|$ represents the number of events in the test set, s_m^e is the *m-th* subevent in the *e-th* event. The perplexity is computed based on all the subevents of all the events in the test data.

In the second case, we consider only the last (*M-th*) subevent in a test event and thus we can get:

$$\text{Perp@last} = \exp(-\frac{1}{N_w}\sum_{e=1}^{|E_{test}|}\log P(s_M^e|s_{1:M-1}^e)) \ , \quad (14)$$

Perp@last is computed based on the last subevents of all the events in the test data.

Following [28], we also adopted the word classification error (i.e., *word error-rate*) for evaluation. It directly measures the quality of the predicted subevents by counting the number of mismatching words between the predicted events and the ground-truth.[4] Like the definition of perplexity, we also consider two cases. In the first case, we consider the subevents from 2 to M in a test event.

$$\text{Error\_Rate} = \frac{1}{N_w}\sum_{e=1}^{|E_{test}|}\sum_{m=2}^{M}\sum_{n=1}^{N_m^e} I(w_{m,n}^{'e} \neq w_{m,n}^e), \quad (15)$$

where $I()$ represnets an indicator function. When a predicted word is not the true word, i.e., $w_{m,n}^{'e} \neq w_{m,n}^e$, it equals to 1; otherwise 0. $w_{m,n}^{'e}$ denotes the predicted word and $w_{m,n}^e$ is the true word. N_m^e denotes the number of words in the subevent s_m^e.

In the second case, we consider only the last subevents of the events in the test data.

$$\text{Error\_Rate@last} = \frac{1}{N_w}\sum_{e=1}^{|E_{test}|}\sum_{n=1}^{N_M^e} I(w_{M,n}^{'e} \neq w_{M,n}^e), \quad (16)$$

A model with lower word classification error is preferred.

B. EXPERIMENTAL RESULTS

To demonstrate the advance of our proposed models, we compared them with both state-of-the-art language models and neural network baselines. Three well-established *n-gram language models*, namely Backoff n-gram, Modified Kneser-Ney and Witten-Bell Discounting n-gram model, were implemented with SRILM [32]. Specifically, we set $n = 5$ for these n-gram language models. In addition, we also compared the proposed models with the basic LSTM model, HLSTM model (i.e., considering hierarchical sequential structure of an event) and CH-LSTM (incorporating contextual information to HLSTM).

[4]For a correct predicted word, its position should also be correct.

1) OVERALL RESULTS

We report our results in terms of Perp, Error_Rate, Perp@last and Error_Rate@last as shown in Table 2. As can be seen from the table, all neural models outperform state-of-the-art n-gram models with respect to all evaluation metrics.

The n-gram models get high word error rate on the prediction of the last subevent (Error_Rate@last), which demonstrates that they are not proper for the task of future subevent prediction. The n-gram models consider only previous $n-1$ words and thus cannot exploit the overall information within previous subevents. The good performance of HLSTM, CH-LSTM and CH-LSTM-Att proves that considering hierarchical sequential structures of events improves the performance in terms of all measures. From Table 2, we can see it improves by around 70 perplexity points and 5 percentage points of word error-rate compared to the LSTM prediction model, showing the advance of the proposed models. We also observe that CH-LSTM model further outperforms HLSTM, which demonstrates the importance of topics for subevent prediction. Furthermore, we can see that CH-LSTM-Att model with a hierarchical attention mechanism improves the CH-LSTM model, which implies that the hierarchical attention mechanism considering different influences of the previous subevents and words within the subevents is important for future subevent prediction.

2) COMPLEXITY ANALYSIS

With the trained neural models, the complexity of the decoding is largely dominated by the computation of the output probabilities, giving $O(nD|\mathcal{V}|)$, where n is the generated text length of the subevents, D is the dimensionality of output word embedding.

3) QUALITATIVE ANALYSIS

Given the previous subevents $s_{1:M-1}$, we adopted beam-search to approximate the most possible next subevents s_M. In Fig. 3, we presented a case study about travel issues on China's National day. There are hundreds of millions of people traveling during the holiday. We listed four predicted next subevents which are generated by four neural models plus the ground truth subevents. In this case, we can see that the result of LSTM prediction model is entirely not relevant. Although HLSTM model achieves better results, it is still slightly different from the truth. The resutls of our proposed CH-LSTM and CH-LSTM-Att are almost the same with the ground truth.

4) VISUALIZATION OF ATTENTION

We implement attention visualization graph based on a case chosen from the test dataset, as shown in Fig. 4. This figure illustrates how the proposed attention mechanism puts emphasis on valued information during each step in the decoding stage. When generating the word "China and South Korea", the model stresses subevents s_1, s_2, s_4. Meanwhile, the word tokens "Chinese Team", "China and South Korea",

TABLE 2. The average test perplexity and word error-rate of five runs.

Model	Perp	Error_Rate	Perp@last	Error_Rate@last
Backoff N-Gram	268.08	76.40%	264.07	93.03%
Modified Kneser-Ney	260.50	75.70%	257.24	93.06%
Witten-Bell Discounting N-Gram	259.44	76.05%	255.48	92.60%
LSTM	211.97 ± 0.11	75.67%±0.01%	201.59 ± 0.38	75.22% ± 0.02%
HLSTM	187.32 ± 0.25	74.86%±0.02%	129.44 ± 0.23	71.06% ± 0.02%
CH-LSTM	182.13 ± 0.11	74.64%±0.05%	127.74 ± 0.21	70.02% ± 0.01%
CH-LSTM-Att	**178.53± 0.12**	**73.50% ± 0.03%**	**120.56 ± 0.13**	**68.01% ± 0.02%**

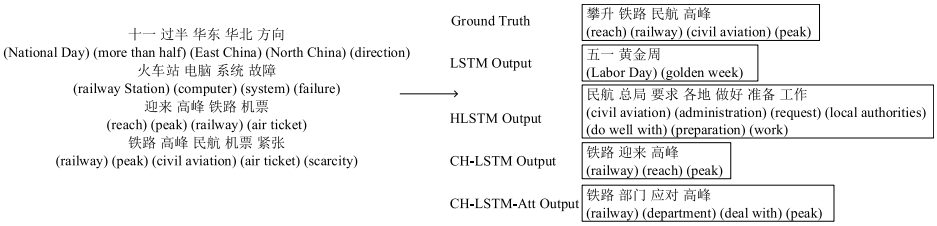

FIGURE 3. An example of model outputs. We show the observed previous subevents in the left and show the ground truth and model outputs in the right.

FIGURE 4. Visualization of attention. Previous subevents $s_{1:4}$ and our model's output is placed in the left. The right bar denotes for the attention weight. The stressed subevents and word tokens have been labeled in blue.

"Dual Match" and "Report" are emphasised by the attention mechanism. This visualization graph shows that the proposed attention mechanism is capable of capturing important information from both word-level and subevent-level to enhance the next subevent decoding.

a: NEXT SUBEVENT RANKING

We further evaluate our model on the next subevent ranking task, which aims to find out the most likely next subevent of a sequence of existing subevents from candidate subevents. As shown in Fig. 5, the problem can be formalized as: given a model with parameter Θ and a sequence of $m-1$ subevents $s_1, \ldots, s_{m-1}$ (with the topics $z_1, \ldots, z_{m-1}$), find out the most possible next subevent s_m from candidates S:

$$s_m = \arg\max_{s \in S} logP(s|s_{1:m-1}, z_{1:m-1}, \Theta), \tag{17}$$

We generated the dataset from our test set. Specifically, we randomly separated the test set into non-overlapping subsets with size 50. After processing the test set with 15,535 events, we obtain 312 non-overlapping subsets among which 311 subsets contain 50 events and one subset contains 35 events (i.e., $15{,}535 = 311*50 + 35$). Our goal is to choose the most likely last subevent of an event given its previous subevents. The candidates are the last subevents of all the events in its corresponding subset. We use the metric hits@n which represents the proportion of correct subevents in the top ranked n candidates for evaluation. The results are shown in Table 3.

TABLE 3. Performance of different models on next subevent ranking.

Model	hits@1	hits@5	hits@10
Random	2.00%±0.10%	10.00%±0.15%	20.00%±0.20%
LSTM	21.96%± 0.12%	49.73%±0.16%	66.31%±0.21%
HLSTM	25.11%±0.10%	54.49%±0.17%	70.22%±0.18%
CH-LSTM	25.79 %± 0.10%	55.68%±0.18%	71.57%±0.20%
CH-LSTM-Att.	**26.58%± 0.11%**	**56.64%± 0.15%**	**72.42 %± 0.18%**

As shown in Table 3, all the models achieve better results than the random method. The proposed HLSTM model and CH-LSTM model futher improves the performance, which demonstrates the effectiveness of considering the hierarchical event structure and topic information. Finally, we can observe that the proposed CH-LSTM-Att model incorporating a hierarchical attention mechanism significantly outperforms all the models.

V. RELATED WORK

Our related work include event prediction and neural language models.

A. EVENT PREDICTION

The task of event prediction is to predict the occurrence of a future event. The work can be divided into two categories. On one hand, some work learn the causal relations of two events [1], [4], [16] for prediction [24]. For instance, Radinsky et al. [24] extracted generalized causality relations of two events (i.e., "x causes y") from past news and applied

FIGURE 5. Next subevent ranking example.

them to predict the next possible event given a current event. Granroth-Wilding et al. [10] extracted typical sequences of events from texts [3] and used a compositional neural network to learn the coherence score of two events. They aim to predict the next event by learning the strength of association between two event.

On the other hand, some work focus on modeling event sequences for prediction. For example, Radinsky et al. [25] extracted event chains from news documents for predicting the happening of target events. In this work, we focus on non-targeted event prediction. Along this line, Manshadi et al. [18] learned a *n*-gram language model of event sequences from Internet Web log stories. Pichotta and Mooney [23] developed a LSTM based model for learning scripts which represents knowledge of prototypical event sequences. They represented an event as a predicate with several arguments and are limited to predict the events from candidates.

In this work, we propose a novel end-to-end generative model which automatically predicts the next event by generating the words describing it.

B. NEURAL LANGUAGE MODELS

Neural networks have been widely applied in a variety of tasks, ranging from information retrieval [12], [30], language modeling [17], [19], [22] to machine translation [6], [33]. Neural language models have different network architectures including feed-forward [2] and RNN [19]. To address the vanishing gradient problem of RNN, variants such as LSTM [11], [26] and Gated Recurrent Unit (GRU) [5] were proposed. Recent research efforts on RNN models further exploited hierarchical structures [14] in many applications such as query suggestion [31], movie dialogue modeling [28] and video representation [21]. Some other efforts improved the RNN models using attention [7], [14], [35] and additional contextual features [8], [20].

Different from the existing work, we propose a novel hierarchical attention based hierarchical LSTM network combining topic information for prediction.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a hierarchical attention based generative hierarchical LSTM model which automatically generates a short text description of a potential future subevent given the texts describing previous subevents. Our proposed model can capture hierarchical sequential structures of a event with a hierarchical LSTM. In addition, it incorporates the topic information for improving the prediction of future subevents. Furthermore, it can pay attention to critical previous subevents for future subevent prediction with a hierarchical attention mechanism. The experimental results on a real-world dataset demonstrate the superiority of our model for future subevent prediction over several state-of-the-art models.

In future work, we will explore to extend our model to predict not only the content of next subevent but also the exact time and place of the subevent.

REFERENCES

[1] S. Acharya and B. S. Lee, "Incremental causal network construction over event streams," *Inf. Sci.*, vol. 261, pp. 32–51, May 2014.
[2] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.
[3] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," in *Proc. 46th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2008, pp. 789–797.
[4] B. Chikhaoui, S. Wang, T. Xiong, and H. Pigot, "Pattern-based causal relationships discovery from event sequences for modeling behavioral user profile in ubiquitous environments," *Inf. Sci.*, vol. 285, pp. 204–222, May 2014.
[5] K. Cho, B. van Merrienboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.
[6] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: https://arxiv.org/abs/1406.1078
[7] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based LSTM and semantic consistency," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2045–2055, Sep. 2017.
[8] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck, "Contextual LSTM (CLSTM) models for large scale NLP tasks," in *Proc. KDD Workshop Large-Scale Deep Learn. Data Mining (DL-KDD)*, 2016.
[9] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, pp. 1012–1014, Feb. 2009.
[10] M. Granroth-Wilding and S. Clark, "What happens next? Event prediction using a compositional neural network model," in *Proc. 30th Conf. Artif. Intell. AAAI*, 2016, pp. 2727–2733.
[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[12] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for Web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 2333–2338,

[13] Y. Kim, Y. Jernite, D. Sontag, and M. A. Rush, "Character-aware neural language models," in *Proc. AAAI 30th Conf. Artif. Intell.*, 2016, pp. 2741–2749.
[14] J. Li, M.-T. Luong, and D. Jurafsky, "A hierarchical neural autoencoder for paragraphs and documents," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, 2015, pp. 1106–1115.
[15] H. Linmei, L. Juanzi, N. Liqiang, L. Xiao-Li, and S. Chao, "What happens next? Future subevent prediction using contextual hierarchical LSTM," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3450–3456.
[16] L. Liu, S. Wang, G. Su, B. Hu, Y. Peng, Q. Xiong, and J. Wen, "A framework of mining semantic-based probabilistic event relations for complex activity recognition," *Inf. Sci.*, vol. 418, pp. 13–33, Nov. 2017.
[17] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2871–2875.
[18] M. Manshadi, R. Swanson, and A. S. Gordon, "Learning a probabilistic model of event sequences from Internet Weblog stories," in *Proc. 21st Int. Florida Artif. Intell. Res. Soc. Conf.*, 2008, pp. 159–164.
[19] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Int. Conf. Spoken Lang. Process.*, vol. 2, 2010, p. 3.
[20] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *Proc. 4th IEEE Workshop Spoken Lang. Technol. (SLT)*, Dec. 2012, pp. 234–239.
[21] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1029–1038.
[22] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, vol. 28, 2013, pp. 1310–1318.
[23] K. Pichotta and J. R. Mooney, "Learning statistical scripts with LSTM recurrent neural networks," in *Proc. AAAI 30th Conf. Artif. Intell.*, 2016, pp. 2800–2806.
[24] K. Radinsky, S. Davidovich, and S. Markovitch, "Learning causality for news events prediction," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 909–918.
[25] K. Radinsky and E. Horvitz, "Mining the Web to predict future events," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 255–264.
[26] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. INTERSPEECH*, 2014, pp. 338–342.
[27] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: Real-time event detection by social sensors," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 851–860.
[28] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. AAAI 30th Conf. Artif. Intell.*, 2016.
[29] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.
[30] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. 23rd ACM Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 101–110.
[31] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. G. Simonsen, and J.-Y. Nie, "A hierarchical recurrent encoder-decoder for generative context-aware query suggestion," in *Proc. 24th Conf. Inf. Knowl. Manage.*, 2015, pp. 553–562.
[32] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Proc. INTERSPEECH*, 2002, p. 2002.
[33] I. Sutskever, O. Vinyals, and V. Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf.*, 2014, pp. 3104–3112.
[34] S. V. Canneyt, S. Schockaert, and B. Dhoedt, "Categorizing events using spatio-temporal and user features from Flickr," *Inf. Sci.*, vol. 328, pp. 76–96, Jun. 2016.
[35] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn. JMLR Workshop Conf.*, 2015, pp. 2048–2057.
[36] Z. Xu, Y. Liu, N. Yen, L. Mei, X. Luo, X. Wei, and C. Hu, "Crowdsourcing based description of urban emergency events using social media big data," *IEEE Trans. Cloud Comput.*, to be published.

LINMEI HU was born in Jiangxi, China, in July, 1992. She received the Ph.D. degree in computer science and technology from Tsinghua University in January, 2018.

She currently works as an Assistant Professor with the Beijing University of Posts and Telecommunications, China. She has published many articles in top conferences such as AAAI, ACL, EMNLP, and KDD. Her research interests include natural language processing and knowledge graphs.