Qian Z, Li PF, Zhu QM *et al.* Document-level event factuality identification via reinforced semantic learning network. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(6): 1248-1268 Nov. 2024. DOI: 10.1007/s11390-024-2655-1

Document-Level Event Factuality Identification via Reinforced Semantic Learning Network

Zhong Qian[1] (钱　忠), *Member, CCF*
Pei-Feng Li[1, 2] (李培峰), *Senior Member, CCF, Member, ACM, IEEE*
Qiao-Ming Zhu[1, 2] (朱巧明), *Distinguished Member, CCF*
and Guo-Dong Zhou[1, 2] (周国栋), *Distinguished Member, CCF*

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *AI Research Institute, Soochow University, Suzhou 215006, China*

E-mail: qianzhong@suda.edu.cn; pfli@suda.edu.cn; qmzhu@suda.edu.cn; gdzhou@suda.edu.cn

Received July 12, 2022; accepted April 9, 2024.

Abstract This paper focuses on document-level event factuality identification (DEFI), which predicts the factual nature of an event from the view of a document. As the document-level sub-task of event factuality identification (EFI), DEFI is a challenging and fundamental task in natural language processing (NLP). Currently, most existing studies focus on sentence-level event factuality identification (SEFI). However, DEFI is still in the early stage and related studies are quite limited. Previous work is heavily dependent on various NLP tools and annotated information, e.g., dependency trees, event triggers, speculative and negative cues, and does not consider filtering irrelevant and noisy texts that can lead to wrong results. To address these issues, this paper proposes a reinforced multi-granularity hierarchical network model: Reinforced Semantic Learning Network (RSLN), which means it can learn semantics from sentences and tokens at various levels of granularity and hierarchy. Since integrated with hierarchical reinforcement learning (HRL), the RSLN model is able to select relevant and meaningful sentences and tokens. Then, RSLN encodes the event and document according to these selected texts. To evaluate our model, based on the DLEF (Document-Level Event Factuality) corpus, we annotate the ExDLEF corpus as the benchmark dataset. Experimental results show that the RSLN model outperforms several state-of-the-arts.

Keywords document-level event factuality identification, hierarchical reinforcement learning, attention network, multi-granularity encoding

1 Introduction

Event factuality identification (EFI) aims to predict the factual nature of a given event in texts, i.e., whether the event is evaluated as a fact, a counterfact, or a possibility. EFI mainly consists of two subtasks: 1) sentence-level event factuality identification (SEFI), which predicts the factuality of an event only considering the current sentence containing this event, and 2) document-level event factuality identification (DEFI), which is defined as identifying the factuality of an event based on a document, from which the event is derived. This paper focuses on the DEFI task exemplified by Fig.1, where we can observe the following aspects.

1) The sentences S1.1, S1.2, S1.3, S1.6, and S1.7 contain the event mentions (i.e., the event trigger "cancel") of the event E1 directly, while S1.4 and S1.5 refer to the event mention of E1 indirectly. 2) S1.3, S1.4, and S1.6 hold negative positions CT− with regard to E1, mainly according to negative cues

Regular Paper
The work was supported by the National Natural Science Foundation of China under Grant Nos. 62006167, 62276177, 62376181, and 62376178, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No. 24KJB520036, and the Project Funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions.

(S1.1) Due to COVID-19 in 2021, International Olympic Committee and Tokyo Olympic Organizing Committee are facing a tough decision whether to cancel the Olympic Games after a year's delay.

(S1.2) Japan's government has privately concluded the Tokyo Olympics will have to be cancelled because of COVID-19, The Times reported, citing an unnamed senior member of the ruling coalition.

(S1.3) "We clearly deny the report that we will cancel the Games," Deputy Chief Cabinet Secretary told a news conference.

(S1.4) "It is very disappointing to see that the Times is developing such a tabloid-like story with an untrustworthy source," a source from the organizing committee told Reuters.

(S1.5) IOC President Thomas Bach reaffirmed his commitment to holding the Games this year in an interview.

(S1.6) "We, the IOC, will never abandon the athletes, so therefore, a cancellation for us was not really an option," Bach said.

(S1.7) About 80% of people in Japan are *pessimistic* about the Games, and hope to cancel it, recent opinion polls show, as the country's COVID-19 situation worsens.

(S1.8) Japan has been hit less severely by COVID-19, but a recent surge in cases has forced it to close its borders to non-resident foreigners and declare a state of emergency in Tokyo and other cities.

Fig.1. Example for document-level event factuality, where the event E1 is "Tokyo Olympics is canceled in 2021" with the factuality of CT−. Event triggers are green, speculative cues are blue, and negative cues (including negative sentimental tokens) are red. The token "pessimistic" in S1.7 can be regarded as both a speculative cue and a negative sentimental token expressing the semantics of incomplete negation (PS−).

"deny" (S1.3), "untrustworthy" (S1.4), "not" (S1.6), and negative sentimental words "disappointing" (S1.4). 3) Other sentences express different factuality. For example, S1.1 and S1.7 evaluate E1 as "possible positive"/PS+, while S1.2 thinks that E1 is "certain positive"/CT+. 4) In addition, S1.8 mentions another irrelevant CT+ event "Japan closes its borders to non-resident foreigners", rather than E1. Hence, S1.8 offers unrelated factual information for E1, and may mislead it to be predicted as CT+ by mistake.

In term of document-level factuality, the value of E1 is unique, i.e., "certain negative"/CT−. According to the core semantics of the document, sentences S1.3, S1.4, and S1.6 are responsible for determining the factuality of E1. Other sentences may filter negative information and mislead E1 to be identified as CT+ or PS+ mistakenly. Therefore, to tackle the inconsistency of sentence-level information and to understand texts correctly and comprehensively with respect to (w.r.t.) the event, we should design a DEFI model that can select the most relevant and meaningful sentences and tokens.

Up to now, previous EFI work mainly considered sentence-level task SEFI and employed neural networks[1–5]. Nevertheless, DEFI is in the preliminary stage. Related work[6, 7] designed complex methods capturing syntactic and semantic features from parsed trees and sentences with event mentions, or extracting local and global information for event triggers based on graph convolution networks (GCN)[8]. However, the limitations of these studies are that they depend on annotated information, and encode the whole document directly without discarding irrelevant and noisy texts.

Based on the analysis above, the main challenges of DEFI are summarized as follows.

End-to-End Modeling Formulation. End-to-end DEFI should be defined clearly, and corresponding solution needs to be proposed for practical and real-world application. It is required that the DEFI model only relies on the event and document, and no other explicitly annotated information (e.g., event triggers, speculative and negative cues in Fig.1) is needed, since upstream tasks detecting other information may result in cascade errors and performance degradation.

Comprehensive Encoding. The DEFI model is required to be able to learn contextual information, and capture interactions between events and documents to understand the semantics of the event-related texts comprehensively. For example, it should be inferred that E1 is negated by the document in Fig.1.

Text Selection. It is required that a DEFI model can select the most relevant and meaningful sentences and tokens, meanwhile discard those irrelevant and noisy ones, since noise is likely to result in wrong prediction. As illustrated in Fig.1, the model needs to select S1.3, S1.4, S1.6, and ensures the predicted results are not influenced by other sentences.

To address these challenges, we develop a novel model named Reinforced Semantic Learning Network (RSLN). In summary, our core contributions and main work are as follows.

1) We define the end-to-end DEFI task and design the RSLN model as the solution. To the best of our knowledge, this is the first end-to-end framework on DEFI to address the end-to-end modeling formulation.

2) We design sentence and document-level encoders with hierarchical structures to extract semantics from the event and document at various levels of granularity, aiming at building a comprehensive encoding method.

3) We integrate policy networks that can select relevant and useful sentences and tokens to the pro-

posed RSLN model with the mechanism of hierarchical reinforcement learning, which can tackle the problem of applying text selection mechanism.

4) We construct the ExDLEF (Extended version of Document-Level Event Factuality) corpus that is suitable for the DEFI defined by this paper. Experimental results on both English and Chinese sub-corpus (*MacroF*1/*MicroF*1: 67.42/77.48 and 75.74/78.29, respectively) demonstrate that our RSLN model is better than several state-of-the-art baselines.

The rest of this paper is organized as follows. Related work is discussed in Section 2. Then we present the formalized definition of DEFI in Section 3. We provide a detailed description of the proposed RSLN model in Section 4. Section 5 introduces the ExDLEF corpus and analyzes its differences from the related dataset DLEF. Experimental results and analysis are demonstrated in Section 6. Finally, Section 7 concludes this paper.

2 Related Work

2.1 Event Factuality Identification

SEFI. With the wide and successful application in NLP (natural language processing), neural networks have been applied to SFEI. Some work aimed to learn semantical and lexical information from texts. Sheng *et al.*[4] devised a convolutional neural network (CNN) model with linguistic features such as event selected predicates, negative words and degree words. Rudinger *et al.*[1] developed long short-term memory (LSTM) models with the versions of linear chain and dependency tree. Qian *et al.*[2] used a hybrid network with LSTM and CNN working on sentences and syntactic paths. To further improve the performance of CT−, PS+, and PR+ that are in minority, Qian *et al.*[3] designed a generative adversarial network (GAN) to produce more syntactic features. Veyseh *et al.*[5] presented a graph neural network exploiting syntactic and semantic structures of sentences to model the contexts.

DEFI. This task is still in its preliminary stage. Qian *et al.*[6] constructed the first corpus, Document-Level Event Factuality (DLEF), which is annotated with both sentence-level and document-level event factuality, and also includes event triggers, and speculative and negative cues. Based on DLEF, Qian *et al.*[6] designed a multi-layer LSTM neural network to capture both intra- and inter-sequence information from dependency paths and sentences. Similarly, Huang *et al.*[7] employed a double-layer LSTM network to encode sentences. Cao *et al.*[8] developed an uncertain local-to-global network to model the uncertainty of local information and to leverage global structure for integrating. We find that the main limitations of these studies are that they depend on annotated information, and do not discard noisy texts. Therefore, we re-define the DEFI task and devote to an end-to-end paradigm.

2.2 Hierarchical Reinforcement Learning (HRL)

HRL integrates two-level policy networks with reinforcement learning to capture information of different levels. Liu *et al.*[9] devised a goal-oriented dialogue system, where the high-level policy guides the conversation to the final goal, and the low-level policy reaches sub-goals by generating the corresponding utterance for response. Wang *et al.*[10] incorporated clause and word selection to tackle the problem of data noise in document-level aspect sentiment classification. Xiao *et al.*[11] proposed an HRL framework for summarization switching between copying and rewriting sentences. Wan *et al.*[12] built an HRL model encoding historical knowledge and structured action space, and achieved improvements on relation and entity link prediction.

This paper considers HRL to select the most relevant sentences and tokens with regard to the event by policy networks. Some work[10–12] designed policy networks with simple structures. To extract sentences and tokens more accurately, we apply sentence encoding layer and document encoding layer to both sentence and token policy networks for multi-granularity and hierarchical encoding. Additionally, we use similar encoders in classification networks and policy networks to ensure the homogeneity so that all of them can capture speculative and negative syntactic and semantic features.

2.3 Advanced Attention Networks

Co-Attention. This is a bi-directional mechanism computing weights for two sequences, and mainly covers parallel and alternating co-attention. Zhou *et al.*[13] employed a co-attention enhanced hierarchical MRC (Machine Reading Comprehension) model to capture interactions between the article and questions, thus guided the decoder to produce more consistent and relevant distractors. Wu *et al.*[14] exploit-

ed a decision tree based explainable claim verification model integrating co-attention to make the evidence and claims interact with each other. Wu *et al.*[15] devised multi-modal co-attention network for fake news detection to fuse textual and visual features.

Gated Attention. It selects elements with gates for aggregation to alleviate unnecessary calculation on unattended elements. In NLP field, Lai *et al.*[16] used a gated self-attention memory network for the answer selection task. Xue *et al.*[17] proposed a dynamically gated attention network and achieved satisfactory results on several sentence classification tasks. Liu *et al.*[18] devised a multi-classification sentiment analysis model based on attention with gated linear units.

Given the above advantages, we integrate various attentions into our RSLN model, including self-attention, multi-head attention, co-attention, gated attention, and hierarchical attention.

2.4 Differences Between DEFI and FEVER

Fact extraction and verification (FEVER) concentrates more on information retrieval including claim verification and evidence selection, which are a bit similar to DEFI. We compare them from the following perspectives.

Task Definitions. FEVER requires claim verification labelled as "Supported/Refuted/NotEnoughInfo" and evidential sentence selection. DEFI focuses on identifying event factuality based on a given event and a document. Due to the combination of modality and polarity associated with speculation and negation, the definition of event factuality values is more complicated, as defined in Table 1. Therefore, the classification task in DEFI is more difficult than that in FEVER.

Table 1. Event Factuality Values Used in This Paper

	Positive/+	Negative/−	Underspecified/u
Certain/CT	**CT+***	**CT−***	CTu
Possible/PS	**PS+***	**PS−***	N/A
Underspecified/U	N/A	N/A	**Uu***

Note: The applicable values in the ExDLEF corpus are highlighted in bold and marked with *. CT: certain. PS: possible. +: positive. −: negative. U: modality. u: polarity.

Resources. Annotation of FEVER requires several documents, while for an event in ExDLEF, we annotate its label of factuality according to one document. Moreover, FEVER consists of 185 445 claims crawled from Wikipedia, whose size is larger than that of ExDLEF focusing on news. Hence, the main differences of resources are annotated information, genre, and scale.

Methods. Some work[19, 20] on FEVER constructed a pipeline system comprising document retrieval, sentence-level evidence selection, and textual entailment, and designed networks to fuse evidences and then verify the claim. Other studies[21, 22] organized evidence selection and claim verification into multi-task learning frameworks. Compared with FEVER, DEFI does not aim at retrieving documents or evidential sentences or tokens precisely.

As a conclusion, we argue that FEVER and DEFI are different tasks. Compared with the models on FEVER, our RSLN is absorbed in the document-level solution, integrates both sentence and token selection, focuses on factual and non-factual information, and defines more comprehensive and detailed labels.

3 Task Formulation

This section gives definitions and formulations of the dataset, DEFI, and factuality values.

Dataset. The whole dataset C can be defined as $C = \{(y, E, D)\}_{i=0}^{N_c-1}$, where N_c is the total number of samples (i.e., events). Each event is denoted as a triple sample (y, E, D). For each sample, event E (usually a sentence) is associated with a ground-truth label of its document-level factuality value y and a document D from which y can be inferred. In this paper, we only consider one event in each document.

DEFI. This task is defined as predicting the factuality value of event E according to document D, from which E is derived. Given E and D as input, a DEFI model M aims to learn the event-specific representation $\boldsymbol{h}_E$ according to D specified to E, and the probability of y is calculated by softmax:

$$\boldsymbol{h}_E = M(E, D|\Theta), \tag{1}$$

$$p(y|E, D) = \text{softmax}(\boldsymbol{h}_E|\Theta), \tag{2}$$

where Θ is the set of parameters of the model. A document D can be denoted as a set of sentences, $D = \{S_0, S_1, \ldots, S_{I-1}\}$.

Event Factuality Values. As previous work[6, 23], event factuality values are characterized as the combination of modality and polarity, where modality conveys the certainty degree of events, mainly including three applicable values, certain (CT), probable (PR), and possible (PS), while polarity expresses whether the event happened or not, mainly containing positive (+) and negative (−). There is one default value for both modality and polarity, which is underspecified (U/u, where U for modality, and u for polarity),

meaning unknown or uncommitted. We utilize the factuality values in Table 1[6]. PR and PS are merged into PS in both the DLEF and ExDLEF corpus, because of similar semantics on modality expressing "not totally certainty". Grammatically speaking, PSu and U+/− are not applicable (N/A). Although applicable, no events can be annotated as CTu, since it means "partially underspecified" that is extremely rare in news texts. Therefore, there are five applicable event factuality values in the benchmark dataset ExDLEF corpus: ⟨underspecified, underspecified⟩/Uu, ⟨certain, negative⟩/CT−, ⟨possible, negative⟩/PS−, ⟨possible, positive⟩/PS+, and ⟨certain, positive⟩/CT+.

4 Approach: RSLN for DEFI

This section introduces the proposed reinforced semantic learning network (RSLN) in detail. For clear description, we first give the overview architecture, and then present the structures of main sub-networks. Finally, we explain the optimization of RSLN.

4.1 Overview

The architecture of RSLN is shown in Fig.2. Overall, RSLN is composed of three sub-networks: 1) classification network (CNet) ϕ^c that outputs the results of DEFI, and produces rewards for policy networks; 2) sentence selection policy network (SPNet) π^s that selects sentences from the document D; and 3) token selection policy network (TPNet) π^t that selects tokens from each sentence S_i. Based on these sub-networks, the RSLN model is presented as the set of them: $\text{Model}/M = \{\text{CNet}/\phi^c, \text{SPNet}/\pi^s, \text{TPNet}/\pi^t\}$. The advantages of RSLN can be characterized as follows.

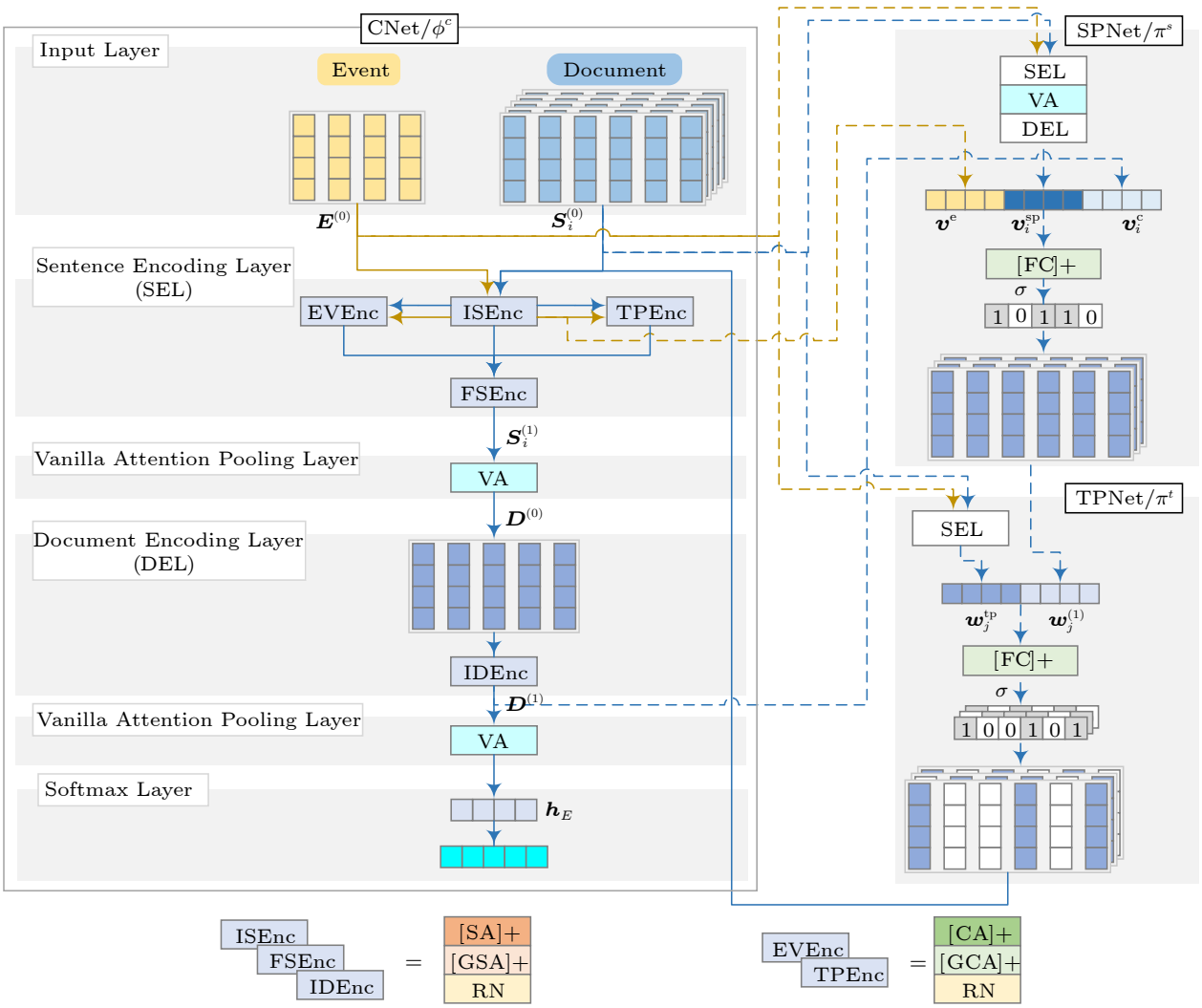


Fig.2. Overall architecture of our RSLN model, below which we give the legends for encoders in SEL and DEL. The regular expression operator "[ ]+" means this sub-network is used and stacked more than once.

Reinforcement. RSLN enables policy networks to select sentences and tokens with HRL for information extraction and refinement.

Multi-Granularity. RSLN captures semantics of events and documents at multi-levels of granularity. It learns intra- and inter-sentence information from events, topics, and sentences.

Hierarchy. RSLN integrates both the sentence and document encoding layers hierarchically to understand texts comprehensively, rather than concatenating all the sentences into one sequence, since sentences may hold different factuality values for events.

Attention. RSLN devises various attention sub-networks to capture the most meaningful information among events and sentences.

Next, modules of each sub-network, CNet, SPNet, and TPNet, are described as follows.

4.2 Input Layer of CNet

This layer produces the embedding of each token t_j^i in event E and each sentence S_i. For t_j^i, we mainly consider the information of the word embedding (WE) provided by a pre-trained model GloVe[24], and the position embedding (PE). The embedding of token t_j^i can be denoted as the sum of them:

$$\boldsymbol{t}_j^i = WE(t_j^i) + PE(t_j^i).$$

Then the matrix representations of E and S_i are denoted as $\boldsymbol{E}^{(0)}$ and $\boldsymbol{S}_i^{(0)}$, respectively, where $\boldsymbol{E}^{(0)} \in \mathbb{R}^{d_{\text{model}} \times |E|}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}^{d_{\text{model}} \times |S_i|}$, and d_{model} is the dimension of our attention sub-networks. In CNet, $\boldsymbol{E}^{(0)}$ and $\boldsymbol{S}_i^{(0)}$ are fed into the following encoding layers.

4.3 Sentence Encoding Layer (SEL)

SEL is an important and fine-grained module of CNet, SPNet, and TPNet, which aims at extracting information of event E and each sentence S_i. SEL firstly updates hidden states of tokens, and then learns the vector representations for E and S_i. Encoders and networks of SEL are defined as follows.

4.3.1 Intra-Sentence Encoder (ISEnc)

Based on self-attention, ISEnc is used for intra-sentence encoding, i.e., encodes sentence-level semantics in event E and each sentence S_i:

$$\boldsymbol{E}^{\text{IS}} = RN(GSA(SA(\boldsymbol{E}^{(0)}))),$$
$$\boldsymbol{S}_i^{\text{IS}} = RN(GSA(SA(\boldsymbol{S}_i^{(0)}))),$$

where SA, GSA, and RN are self-attention, gated self-attention, and residual network, respectively. As mentioned by previous work[25], for any input $\boldsymbol{U}^{\text{in}}$, SA (i.e., $\boldsymbol{U}^{\text{out}} = SA(\boldsymbol{U}^{\text{in}})$) is defined as:

$$\boldsymbol{U}^{(0)} = FFN(MHA(\boldsymbol{U}^{\text{in}}, \boldsymbol{U}^{\text{in}}, \boldsymbol{U}^{\text{in}})),$$

where MHA is multi-head attention, and FFN is a position-wise fully connected feed-forward network.

Considering not all the tokens are related and beneficial to DEFI, in addition to conventional self-attention, we equip SA with gate mechanism to learn meaningful high-level representations, and adopt one variant of gated self-attention (i.e., $\boldsymbol{U}^{\text{out}} = GSA(\boldsymbol{U}^{\text{in}})$) that is formally calculated as:

$$\boldsymbol{U}^{(0)} = MHA(\boldsymbol{U}^{\text{in}}, \boldsymbol{U}^{\text{in}}, \boldsymbol{U}^{\text{in}}),$$
$$\boldsymbol{G}^{(0)} = \sigma(FC^{(0)}(\boldsymbol{U}^{\text{in}}) + FC^{(1)}(\boldsymbol{U}^{(0)})),$$
$$\boldsymbol{U}^{\text{out}} = FFN(\boldsymbol{G}^{(0)} \odot \boldsymbol{U}^{\text{in}} + (1 - \boldsymbol{G}^{(0)}) \odot \boldsymbol{U}^{(0)}),$$

where $FC^{(0/1)}$ are fully-connected layers with tanh as the activation, σ is the sigmoid function, and $\odot$ is element-wise multiplication operator.

By integrating attentions, our RSLN model is quite deep and probably exposed to the degradation. The solution is a stacked layer of residual networks (RN) used to control the output, and one variant is computed as:

$$\boldsymbol{U}^{(0)} = GELU(LN(LL(\boldsymbol{U}^{\text{in}}))), \tag{3}$$

$$\boldsymbol{U}^{\text{out}} = GELU(LN(LL(\boldsymbol{U}^{(0)})) + \boldsymbol{U}^{\text{in}}), \tag{4}$$

where $GELU$ is the Gaussian error linear unit employed as activation function, LN is a normalization layer, and LL is a linear layer. Since the DEFI is a document-level task, we also consider learning inter-sentence information, and further employ the following sub-encoders.

4.3.2 Event Encoder (EVEnc)

Event E is the basic clue to guide the model to identify document-level factuality of E. To avoid DEFI becoming trivial, events are kept concise and brief during annotation, and hence contain fundamental information for event-specific DEFI. For example, in Fig.1, E1 includes the event trigger "canceled", the event argument "Tokyo Olympics", and the time stamp "2021". Therefore, we integrate events into D in order to capture event-related semantics, and define EVEnc as:

$$\boldsymbol{S}_i^{\text{EV-1}},\ \boldsymbol{E}^{\text{EV-1}} = CA(\boldsymbol{S}_i^{\text{IS}},\ \boldsymbol{E}^{\text{IS}}),$$
$$\boldsymbol{S}_i^{\text{EV-2}},\ \boldsymbol{E}^{\text{EV-2}} = GCA(\boldsymbol{S}_i^{\text{EV-1}},\ \boldsymbol{E}^{\text{EV-1}}),$$
$$\boldsymbol{S}_i^{\text{EV}} = RN(\boldsymbol{S}_i^{\text{EV-2}}),$$

where CA and GCA are co-attention and gated co-attention, respectively. For any input $\boldsymbol{U}^{\text{in}}$ and $\boldsymbol{V}^{\text{in}}$, CA (i.e., $\boldsymbol{U}^{\text{out}},\ \boldsymbol{V}^{\text{out}} = CA(\boldsymbol{U}^{\text{in}},\ \boldsymbol{V}^{\text{in}})$) is calculated as:

$$\boldsymbol{V}^{\text{out}} = MHA(\boldsymbol{V}^{\text{in}},\ \boldsymbol{U}^{\text{in}},\ \boldsymbol{U}^{\text{in}}),$$
$$\boldsymbol{U}^{\text{out}} = FFN(MHA(\boldsymbol{U}^{\text{in}},\ \boldsymbol{V}^{\text{out}},\ \boldsymbol{V}^{\text{out}})).$$

To filter unrelated information propagating in previous attention layers, we exploit gates as well, and compute gated co-attention GCA (i.e., $\boldsymbol{U}^{\text{out}},\ \boldsymbol{V}^{\text{out}} = GCA(\boldsymbol{U}^{\text{in}},\ \boldsymbol{V}^{\text{in}})$) as follows:

$$\boldsymbol{V}^{(0)} = MHA(\boldsymbol{V}^{\text{in}},\ \boldsymbol{U}^{\text{in}},\ \boldsymbol{U}^{\text{in}}),$$
$$\boldsymbol{G}_V^{(0)} = \sigma(FC^{(0)}(\boldsymbol{V}^{\text{in}}) + FC^{(1)}(\boldsymbol{V}^{(0)})),$$
$$\boldsymbol{V}^{\text{out}} = FFN(\boldsymbol{G}_V^{(0)} \odot \boldsymbol{V}^{\text{in}} + (1 - \boldsymbol{G}_V^{(0)}) \odot \boldsymbol{V}^{(0)}),$$
$$\boldsymbol{U}^{(0)} = MHA(\boldsymbol{U}^{\text{in}},\ \boldsymbol{V}^{\text{out}},\ \boldsymbol{V}^{\text{out}}),$$
$$\boldsymbol{G}_U^{(0)} = \sigma(FC^{(2)}(\boldsymbol{U}^{\text{in}}) + FC^{(3)}(\boldsymbol{U}^{(0)})),$$
$$\boldsymbol{U}^{\text{out}} = FFN(\boldsymbol{G}_U^{(0)} \odot \boldsymbol{U}^{\text{in}} + (1 - \boldsymbol{G}_U^{(0)}) \odot \boldsymbol{U}^{(0)}).$$

4.3.3 Topic Encoder (TCEnc)

In order to get important information as much as possible from some brief texts, we integrate the topic sentence into other sentences in document D, where the topic sentence is the first sentence in the main body of D. Due to the characteristics of news, topic sentences usually summarize the main or core semantics of D, and may contain more information about the events than other sentences, including event triggers and arguments, time stamps, etc. Therefore, we can extract beneficial and event-related information from topic sentences as well. Take the event E1 in Fig.1 as an example again. The sentence S1 involves several arguments with regard to E1, e.g., event trigger "cancel", named entities "COVID-19" and "Tokyo Olympic Organizing Committee", and time stamp "2021". Similar to the events, we encode the topic sentence into other sentences by CA to learn interactive information among them,

$$\boldsymbol{S}_i^{\text{TC-1}},\ \boldsymbol{S}_0^{\text{TC-1}} = CA(\boldsymbol{S}_i^{\text{IS}},\ \boldsymbol{S}_0^{\text{IS}}),$$
$$\boldsymbol{S}_i^{\text{TC-2}},\ \boldsymbol{S}_0^{\text{TC-2}} = GCA(\boldsymbol{S}_i^{\text{TC-1}},\ \boldsymbol{S}_0^{\text{TC-1}}),$$
$$\boldsymbol{S}_i^{\text{TC}} = RN(\boldsymbol{S}_i^{\text{TC-2}}),$$

where $i = 1,\ \ldots,\ I-1$.

4.3.4 Fusion Encoder (FSEnc)

We have got $\boldsymbol{S}_i^{\text{IS}}$, $\boldsymbol{S}_i^{\text{EV}}$, and $\boldsymbol{S}_i^{\text{TC}}$ that are the output of ISEnc, EVEnc, and TCEnc, respectively. To eliminate the manifold differences, this encoder is devised to fuse these representations and extract higher-level semantic information, as computed by the follows:

$$\boldsymbol{S}_i^{\text{FS-1}} = \tanh(LL^{\text{FS-0}}(\boldsymbol{S}_i^{\text{IS}}) + LL^{\text{FS-1}}(\boldsymbol{S}_i^{\text{EV}}) + LL^{\text{FS-2}}(\boldsymbol{S}_i^{\text{TC}})),$$
$$\boldsymbol{S}_i^{\text{FS}} = RN(GSA(SA(\boldsymbol{S}_i^{\text{FS-1}}))),$$

where $\text{LL}^{\text{FS-}j}$ $(j = 1,\ 2,\ 3)$ are linear layers. FSEnc uses fully-connected layers to transform the dimension of the representation of S_i back to d_{model}, for the reason of keeping the consistency of hidden units and reducing the complexity of the model. And then it employs self-attention to learn high-level abstract semantics based on the output of previous encoders ISEnc, EVEnc, and TCEnc. We use matrix $\boldsymbol{S}_i^{(1)} = \boldsymbol{S}_i^{\text{FS}}$ to denote the matrix representation of S_i encoded by FSEnc in SEL.

4.3.5 Output of SEL

According the above encoders, the document $D = \{S_0,\ S_1,\ \ldots,\ S_{I-1}\}$ can be denoted as $\{\boldsymbol{S}_0^{(1)},\ \boldsymbol{S}_1^{(1)},\ \ldots,\ \boldsymbol{S}_{I-1}^{(1)}\}$. Next, we plan to capture interactive semantics among sentence from the level of the document. Therefore, we learn the vector representation $\boldsymbol{h}_i^{(0)}$ for each sentence S_i: $\boldsymbol{h}_i^{(0)} = VA(\boldsymbol{S}_i^{(1)})$, where $i = 1,\ \ldots,\ I-1$. VA is the Vanilla attention pooling operation (i.e., $\boldsymbol{u}^{\text{out}} = VA(\boldsymbol{U}^{\text{in}})$) that is applied to learning vector representation of any matrix input $\boldsymbol{U}^{\text{in}}$:

$$\boldsymbol{\alpha} = \text{softmax}(\boldsymbol{u}_s^{\text{T}} \tanh(\boldsymbol{U}^{\text{in}})),$$
$$\boldsymbol{u}^{\text{out}} = \boldsymbol{U}^{\text{in}} \boldsymbol{\alpha}^{\text{T}},$$

where $\boldsymbol{u}_s$ is the parameter. Then the representation of D contains vectors of $\{S_i\}$, i.e., $\boldsymbol{D}^{(0)} = \{\boldsymbol{h}_0^{(0)},\ \boldsymbol{h}_1^{(0)},\ \ldots,\ \boldsymbol{h}_{I-1}^{(0)}\}$, which is fed into the document encoding layer in Subsection 4.4.

4.4 Document Encoding Layer (DEL)

DEL is responsible for learning the representation of document D, and includes one encoder and one intra-document encoder (IDEnc). This encoder learns the intra-document information among sentences, and calculates the representation of D by a stack of several SA, GSA, and RN networks:

$$\boldsymbol{D}^{(1)} = RN(GSA(SA(\boldsymbol{D}^{(0)}))).$$

To acquire the final vector representation $\boldsymbol{h}_E$ in (1) of D with regard to E, we apply VA pooling on $\boldsymbol{D}^{(1)}$ and get the output of DEL, $\boldsymbol{h}_E = VA(\boldsymbol{D}^{(1)})$.

4.5 Sentence Selection Policy Network (SPNet)

SPNet, or denoted as $\pi^s(a_i^s|\boldsymbol{s}_i^s, \theta_p^s)$ formally, is responsible for selecting sentences from document D. Since SPNet is significant for promoting the performance of DEFI, we plan to establish close semantic correlation between CNet and SPNet, e.g., utilizing the representations encoded by CNet as the input for SPNet. Concretely, the state, action, and reward of SPNet are defined as follows.

1) *State*. The state of each sentence S_i is represented as $\boldsymbol{s}_i^s$, which is comprised of three vectors:

- $\boldsymbol{v}^{\mathrm{e}} = VA(\boldsymbol{E}^{\mathrm{IS}})$, which means $\boldsymbol{v}^{\mathrm{e}}$ is the vector of event E and is computed based on $\boldsymbol{E}^{\mathrm{IS}}$ using VA pooling, where $\boldsymbol{E}^{\mathrm{IS}}$ is encoded by ISEnc in CNet;
- $\boldsymbol{v}_i^{\mathrm{c}} = VA(\boldsymbol{D}^{(1)})$, where $\boldsymbol{v}_i^{\mathrm{c}}$ is the vector representation of sentence S_i, and $\boldsymbol{D}^{(1)}$ is computed by SEL and IDEnc in CNet;
- $\boldsymbol{v}_i^{\mathrm{sp}} = VA(\boldsymbol{D}^{\mathrm{SP}})$, where $\boldsymbol{D}^{\mathrm{SP}}$ is encoded by SEL and DEL in SPNet with $\boldsymbol{E}^{(0)}$ and $\boldsymbol{S}_i^{(0)}$ as the input, respectively.

We make use of VA in SEL. Thus, the state of each sentence S_i is denoted as the concatenation ($\oplus$) of the above vectors with tanh as the activation function:

$$\boldsymbol{s}_i^s = \tanh(\boldsymbol{v}^e \oplus \boldsymbol{v}_i^{\mathrm{c}} \oplus \boldsymbol{v}_i^{\mathrm{sp}}).$$

2) *Action*. We set the action a_i^s of SPNet as a binary value, i.e., $a_i^s \in \{0, 1\} \sim \pi^s$, where 1 means sentence S_i is selected, while 0 means not. Formally, the possibility distribution of a_i^s is computed based on state $\boldsymbol{s}_i^s$ according to $\pi^s(a_i^s|\boldsymbol{s}_i^s, \theta_p^s)$, which is based on a stack of several fully connected layers FC^s with GELU as the activation function:

$$\pi^s(a_i^s|\boldsymbol{s}_i^s, \theta_p^s) = \sigma(\boldsymbol{W}_p^s FC^s(\boldsymbol{s}_i^s) + \boldsymbol{b}_p^s),$$

where each $FC_k^s \in FC^s$ can be calculated as:

$$FC_k^s(\boldsymbol{s}_i^s) = GELU(\boldsymbol{W}_k^s \boldsymbol{s}_i^s + \boldsymbol{b}_k^s).$$

3) *Reward*. The reward of SPNet for each sentence S_i is represented as r_i^s that is used to guide π^s to select sentences, and is computed as:

$$r_i^s = \epsilon_0^s \log p_{\theta_c}(y|\boldsymbol{h}_i^{(0)}) - \epsilon_1^s \frac{I^*}{I},$$

where the first term is a delay reward of the sentence S_i provided by CNet, and can be obtained as follows. After π^s completes all the actions, we feed each sentence whose representation is $\boldsymbol{h}_i^{(0)} \in \boldsymbol{D}^{(0)}$ encoded by SEL into the softmax of CNet to compute the probability according to the annotated label y of E. I^* and I are the numbers of the selected sentences and total sentences, respectively.

4.6 Token Selection Policy Network (TPNet)

TPNet selects tokens from each sentence S_i, and can be formally represented as $\pi^t(a_j^t|\boldsymbol{s}_j^t, \theta_p^t)$. Similar to SPNet, we also aim to exploit the token-level knowledge learned from CNet. Hence, the input of TPNet is derived from CNet. Formally, the state, action, and reward of TPNet are defined as follows.

1) *State*. In sentence S_i, the state of each token t_j^i is denoted as $\boldsymbol{s}_j^t$. To compute $\boldsymbol{s}_j^t$, we mainly employ the following vectors:

- $\boldsymbol{w}_j^{(1)} \in \boldsymbol{S}_i^{(1)}$, where $\boldsymbol{w}_j^{(1)}$ is the representation of the token $t_j^i \in S_i$, and $\boldsymbol{S}_i^{(1)}$ is computed by SEL in CNet, as described in Subsection 4.3;
- $\boldsymbol{w}_j^{\mathrm{tp}} \in \boldsymbol{S}_i^{\mathrm{TP}}$, where $\boldsymbol{S}_i^{\mathrm{TP}}$ is calculated by SEL in TPNet with the input of $\boldsymbol{E}^{(0)}$ and $\boldsymbol{S}_i^{(0)}$.

Then we can get the state of each token t_j^i as follows:

$$\boldsymbol{s}_j^t = \tanh(\boldsymbol{w}_j^{(1)} \oplus \boldsymbol{w}_j^{\mathrm{tp}}).$$

2) *Action*. Similar to SPNet, the action of TPNet is set as a binary integer $a_j^t \in \{0, 1\} \sim \pi^t$, and 1 denotes token t_j^i is selected, while 0 denotes t_j^i is discarded. The possibility distribution of a_j^t is calculated by π^t with state $\boldsymbol{s}_j^t$ as the input, where FC^t is a stack of fully connected layers as well:

$$\pi^t(a_j^t|\boldsymbol{s}_j^t, \theta_p^t) = \sigma(\boldsymbol{W}_p^t FC^t(\boldsymbol{s}_j^t) + \boldsymbol{b}_p^t).$$

3) *Reward*. The reward of TPNet is denoted as r_i^t that is used to guide π^t to select tokens. Similar to previous work[10, 26], in order to reduce the variance, we compute r_i^t based on the vector representation of each sentence rather than those of each token:

$$r_j^t = \epsilon_0^t \log p_{\theta_c}(y|VA(\boldsymbol{S}_i^{(1)})) - \epsilon_1^t \frac{J^*}{J},$$

where $\log p_{\theta_c}$ in the first term is the output layer of CNet/ϕ^c, y is the annotated label of the event E

based on document D. As for terms, 1) the first term is a delay reward of tokens produced by CNet, and can be calculated as the following steps: VA pooling is applied to $\boldsymbol{S}^{(1)}$ (selected by SPNet) to obtain the vector representation $\boldsymbol{h}_i^{(0)}$ for each S_i, and $\boldsymbol{h}_i^{(0)}$ is fed into to the softmax of CNet to compute the probability; 2) the second term is mainly comprised of J^* and J that are the numbers of the selected tokens and total tokens, respectively.

4.7 Output of CNet

Finally, $\boldsymbol{h}_E$, which is the vector representation of document D with regard to the specific event E originally defined by (1) in Section 3, and is also the output of the document encoding layer in Subsection 4.4, is fed into the softmax layer to compute the possibility distribution of the event factuality:

$$\boldsymbol{p} = \text{softmax}(\boldsymbol{W}_s \boldsymbol{h}_E + \boldsymbol{b}_s).$$

4.8 Model Optimization

Based on the components and architectures of the RSLN model defined above, the whole encoding procedures mainly include the following steps: 1) encoding each sentence S_i by the sentence encoding layer in CNet, 2) selecting sentences from the document D by SPNet, 3) selecting tokens from each sentence S_i by TPNet, and 4) learning the vector representation $\boldsymbol{h}_E$ of document D by the document encoding layer in CNet. The whole optimization of the RSLN model is presented in Algorithm 1, where an additional warm start is adopted by selecting all the sentences and tokens to train the RSLN.

According to the settings of the classification network CNet and policy networks SPNet and TPNet, the total parameters Θ ((1) and (2) in Section 3) of the RSLN model can be mainly classified as two sets:

1) θ_p^s and θ_p^t of policy networks SPNet/π^s and TPNet/π^t, respectively, including the parameters in SEL and DEL of them, and those parameters in the fully connected layers used to compute the states and actions;

2) θ_c of CNet/ϕ^c, including those parameters in the embedding layer, SEL, DEL, and the softmax layer.

For the optimization of policy networks, we update θ_p^s and θ_p^t by the REINFORCE algorithm[27] and policy gradients[28] to maximize the expected rewards.

$$\nabla_{\theta_p^s} J(\theta_p^s) = \sum_{i=0}^{I-1} R_i^s \nabla_{\theta_p^s} \log \pi^s(a_i^s | \boldsymbol{s}_i^s, \theta_p^s),$$

$$\nabla_{\theta_p^t} J(\theta_p^t) = \sum_{j=0}^{J-1} R_j^t \nabla_{\theta_p^t} \log \pi^t(a_j^t | \boldsymbol{s}_j^t, \theta_p^t),$$

where $R_i^s = r_i^s - b(\tilde{r}^s)$ and $R_j^t = r_j^t - b(\tilde{r}^t)$ are designed to estimate the reward of sentence and token selection $(r_j^s,\ r_j^t)$, respectively. The baseline values $b(\tilde{r}^s)$ and $b(\tilde{r}^t)$ are approximated by the average of all the previous rewards. The setting of advantage estimate R_j^s and R_i^t using baseline values can minimize the variance of the individual weight changes of original rewards over time without altering the expectation theoretically[27].

Algorithm 1. Optimization of Reinforced Semantic Learning Network (RSLN)

Input: corpus $C = \{(y,\ E,\ D)\}_{i=0}^{N_c-1}$; each event sample E has two types of input:

1) an event E (usually a sentence);

2) a document $D = \{S_0,\ S_1,\ \ldots,\ S_{I-1}\}$ with I sentences.

Output: the trained RSLN model

```
Initialize the parameters θ_c, θ_p^t, θ_p^s randomly;
Phase 1:
  Warm start, i.e., train CNet/φ^c, TPNet/π^t, SPNet/π^s, and update θ_c, θ_p^t, θ_p^s by selecting all the tokens and sentences;
Phase 2:
for a document D ∈ C do
    Encode all the sentences {S_i} in D by CNet;
    for a sentence S_i ∈ D do
        Calculate the state s_i^s and sample the action a_i^s ~ π^s for S_i;
        Determine whether to select S_i or not;
        Calculate the reward r_i^s of S_i for π^s;
    end for
    for a sentence S_i ∈ D do
        for a token t_j^i ∈ S_i, j ∈ [0, J) do
            Calculate the state s_i^t and sample the action a_j^t ~ π^t for t_j^i;
        end for
        Select tokens of S_i;
        Calculate the reward r_j^t of {t_j^i} for π^t;
    end for
    Calculate the vector representation h_E of D by DEL in CNet;
    Update θ_c, θ_p^t, θ_p^s;
end for
```

For the optimization of the classification network CNet/ϕ^c, the parameter set θ_c is updated by back propagation, and the objective function is computed as:

$$L(\theta) = -\frac{1}{N}\sum_{n=0}^{N-1}\log p(y_n|\theta_c),$$

where y_n is the annotated label of the event E w.r.t D, and N is the number of samples.

5 Corpus

To evaluate our model, we utilize ExDLEF as the benchmark dataset whose English and Chinese sub-corpora are the extended versions of the DLEF-v2[23]. Concretely, we give statistics of ExDLEF in Table 2. Labeling the document-level event factuality requires for comprehensive semantic understanding the event-related document. The differences between DLEF and ExDLEF (including the DLEF-v2) mainly lie in the following aspects.

Table 2. Statistics of the ExDLEF Corpus

Sub-Corpus	Uu	CT−	PS−	PS+	CT+	Total
English	42	745	51	660	3 532	5 030
Chinese	22	1 504	42	953	2 629	5 150

Event Expressions. In DLEF, events are represented as triggers that are words or phrases. Hence, trigger mentions are annotated explicitly in the sentences containing them, and the models on DLEF can make use of them directly. In ExDLEF, an event is a sentence summarized from the document without annotated triggers. Therefore, the task defined on ExDLEF are more difficult than that on DLEF.

Annotated Information. The DLEF corpus annotates various information for each sentence-level event, i.e., speculative and negative cues, event triggers, and its sentence-/document-level factuality. DLEF-v2 further annotates a document-level event (usually a sentence) for each document, and ExDLEF is comprised of more documents than DLEF-v2.

Input of Models. The tasks and models defined and designed by previous work[6, 8] usually rely on a variety of annotated elements, including event triggers, speculative and negative cues. Since event mentions and triggers in sentences are given explicitly, previous models employ these sentences directly, rather than extracting sentences with event mentions. On the contrary, we re-define DEFI as an end-to-end task, which only relies on the event, document, and factuality, without other explicitly annotated information. Therefore, compared with previous research, the task defined in this paper is more difficult, but our RSLN model is more suitable for practical scenario and can be applied to real-world applications directly.

Size of Corpora. The sizes of the Chinese sub-corpora are nearly the same in DLEF and ExDLEF (4 649 vs 5 150). However, in the DLEF corpus, Chinese documents are much more than English ones (4 649 vs 1 727), which means it is less fair to evaluate our model on English texts. Actually, due to the minority of CT− and PS+, both RSLN and RMHAN (Reinforced Multi-Granularity Hierarchical Attention Network)[23] get low results on them ($F1$-score<41), leading to lower $MacroF1<55$, where $MacroF1$ means macro-averaged $F1$-score. Therefore, based on the original documents, we annotate more English samples (up to 5 030 from China Daily) in ExDLEF. CT+ events occupy the majority because of the characteristics of news texts. To avoid the extreme imbalance between CT+ and non-CT+, we pay attention to collecting more CT− and PS+ events during annotation.

6 Experimentation

In this section, we introduce the experimental settings, which are evaluation metrics, implementation details, and baselines. Then, we report the performance of our proposed RSLN model compared with baselines, and present experimental analysis.

6.1 Evaluation Metrics

In the experiments evaluating our model, we focus on the performance of the three main applicable factuality values, CT−, PS+, and CT+, since the events with these values occupy 98.15%/98.76% in English/Chinese sub-corpus. The results of Uu and PS− are excluded from consideration, mainly owing to their extremely small proportions, which is similar to previous work[6–8]. We employ $F1$-score as the main metrics to describe the performance of each applicable value.

Moreover, both macro-averaged and micro-averaged $F1$-scores are utilized to describe the overall performance. The former averages $F1$-scores of each category. The latter, first collects together the decisions (true positives) for all the categories in a single contingency table, and then applies the measure over them:

$$MacroF1 = \frac{1}{3}\sum_{X} F1(X),$$

$$MicroPrecision = \frac{\sum_X TP(X)}{\sum_X TP(X) + \sum_X FP(X)},$$

$$MicroRecall = \frac{\sum_X TP(X)}{\sum_X TP(X) + \sum_X FN(X)},$$

$$MicroF1 = \frac{2 \times MicroPrecision \times MicroRecall}{MicroPrecision + MicroRecall},$$

where X={CT−, PS+, CT+}, and TP, FP, and FN mean "true positives", "false positives", and "false negatives", respectively.

6.2 Implementation Details

For fair comparison, 10-fold cross validation is performed on both the English and Chinese sub-corpus. Word embeddings are pre-trained by GloVe[24] with the dimension of 300, the same as the hidden units of the attention layers in RSLN. In term of training, we exploit a two-phase strategy defined in Algorithm 1: the first phase is set as a warm start, and all the sentences and tokens are selected during training the model, where the Adam algorithm[29] is used as the optimizer. Then, the second phase switches to consider both sentence and token selection, and continues to update the model by the stochastic gradient descent algorithm[30].

6.3 Baselines

We mainly use the following methods as baselines that can be organized as several groups.

6.3.1 *SEFI Model*

SGCN[5] is a sentence-level GCN model that works on sentences and syntactic paths. This model requires that the event triggers are given, and utilizes a simple voting mechanism to decide document-level factuality.

6.3.2 *Pipeline DEFI Models*

These models employ pipeline architectures, and may suffer from errors produced by upstream tasks (detection of event triggers, speculative and negative cues).

LSTM-A[6] employs multi-layer LSTM with vanilla attention pooling including both intra- and inter-sequence attentions to model dependency paths and sentences, and considering adversarial training.

BERT-MSF[31] firstly detects speculation and negation scopes, and then fuses them with the sentences containing events using the model based on BERT.

ULGN[8] represents an uncertain local-to-global network that models local uncertainty and global structure with graph convolution networks.

The network detecting event triggers, and speculative and negative cues is illustrated by Fig.3, whose performance is reported in Table 3. To be in line with complicated attentions employed by this paper, we leverage BERT[32] as the backbone with each sentence I as the input: $\boldsymbol{H}_0 = BERT(I)$, where $\boldsymbol{H}_0 \in \mathbb{R}^{d_{\text{BERT}} \times |I|}$, d_{BERT} is the dimension of BERT, and $|I|$ is the number of tokens. Then we exploit three residual networks to encode the output of BERT for the three tasks, i.e., detection of event triggers, speculative cues, and negative cues:

$$\boldsymbol{H}_1 = RN_1(\boldsymbol{H}_0),$$
$$\boldsymbol{H}_2 = RN_2(\boldsymbol{H}_0),$$
$$\boldsymbol{H}_3 = RN_3(\boldsymbol{H}_0),$$

where residual networks are defined in (3) and (4). Fi-

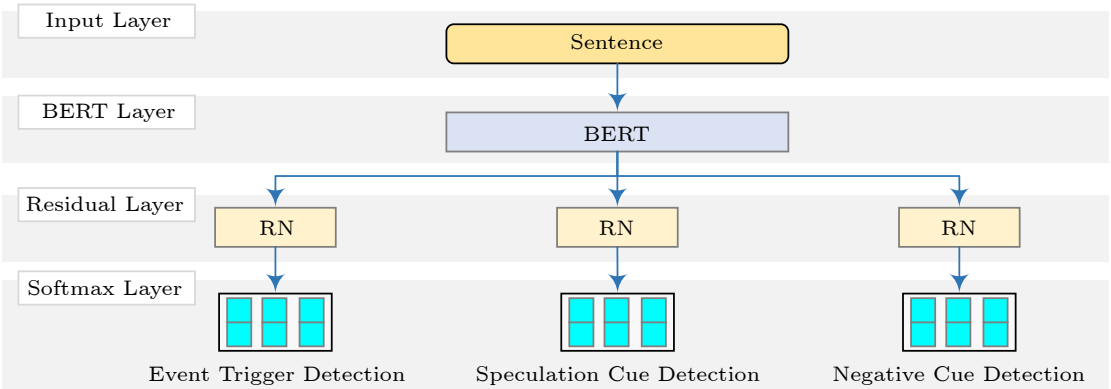


Fig.3. Overall architecture of the networks for the detection of event triggers, speculative cues, and negative cues.

Table 3. Performance of the Detection of Event Triggers, Speculative Cues, and Negative Cues

Sub-Corpus	Task	P(%)	R(%)	*F*1
English	Event trigger detection	87.62	82.70	85.09
	Speculative cue detection	65.91	76.42	70.65
	Negative cue detection	73.79	81.08	77.24
Chinese	Event trigger detection	84.08	77.98	80.87
	Speculative cue detection	72.25	65.41	68.62
	Negative cue detection	69.57	74.52	71.93

nally, the representations of the three tasks are fed into softmax to compute the possibility distributions of corresponding labels for each token, respectively:

$$\begin{aligned}\boldsymbol{p}_{s1} &= \text{softmax}(\boldsymbol{W}_{s1}\boldsymbol{H}_1 + \boldsymbol{b}_{s1}),\\ \boldsymbol{p}_{s2} &= \text{softmax}(\boldsymbol{W}_{s2}\boldsymbol{H}_2 + \boldsymbol{b}_{s2}),\\ \boldsymbol{p}_{s3} &= \text{softmax}(\boldsymbol{W}_{s3}\boldsymbol{H}_3 + \boldsymbol{b}_{s3}),\end{aligned}$$

where $\boldsymbol{W}_{s1}, \boldsymbol{W}_{s2}, \boldsymbol{W}_{s3} \in \mathbb{R}^{2\times d_{\text{BERT}}}$, $\boldsymbol{b}_{s1}, \boldsymbol{b}_{s2}, \boldsymbol{b}_{s3} \in \mathbb{R}^2$. The objective functions $L_s(\theta_s)$ is designed as:

$$L_s(\theta_s) = \frac{1}{3}((L_{s1}(\theta_s) + L_{s2}(\theta_s) + L_{s3}(\theta_s)),$$

$$L_{s1}(\theta_s) = -\frac{1}{N}\sum_{n=0}^{N-1}\log p(y_n^{(s1)}|\theta_s),$$

$$L_{s2}(\theta_s) = -\frac{1}{N}\sum_{n=0}^{N-1}\log p(y_n^{(s2)}|\theta_s),$$

$$L_{s3}(\theta_s) = -\frac{1}{N}\sum_{n=0}^{N-1}\log p(y_n^{(s3)}|\theta_s),$$

where $y_n^{(s1)}$, $y_n^{(s2)}$, and $y_n^{(s3)}$ are the annotated labels indicating whether a token is an event trigger, a speculative cue or a negative cue, respectively. N is the number of samples.

6.3.3 *Target-Dependent Classification Models*

TEND-C[33] and TEND-T[34] are designed for learning target-dependent document representations, where the event is the target in this paper. TEND-C utilizes LSTM with vanilla attention to compose context-sensitive sentence representations, while TEND-T considers more attention modules and integrates word-to-word alignment scheme.

RLSTM[10] is an LSTM neural network with hierarchical reinforcement learning (HRL) and incorporates both token and sentence selection.

6.3.4 *Large-Scale Pre-Trained Attention Model*

BERT-B[32] is the base version of BERT, i.e., BERT-Base-Uncased, with the input set as the linear concatenation of the event and all the sentences.

6.3.5 *End-to-End DEFI Model*

RMHAN[23] is our conference version that considers token selection before sentence selection, and designs stacks of sentence encoders, each of which integrates intra-sentence, topic, and event sub-encoders.

6.3.6 *Variants of Our RSLN Model*

CNet is the supervised classification network in RSLN. It does not consider policy networks or hierarchical reinforcement learning.

RSLN-L is a linear version of RSLN. It concatenates all the sentences into a single long sequence and considers token selection only.

RSLN-tk2sp is an extended version of RSLN-L. It first selects tokens in each sentence, and then extracts the text span with regard to the event. The suffix can be abbreviated as "from token to span (tk2sp)".

RSLN-sp2tk contains the different order of policies compared with RSLN-tk2sp. It first selects the text span, and then selects tokens within the span. Therefore it has the abbreviation of suffix "from span to token (sp2tk)".

RSLN-tk2st shares the same structure with the RSLN model in Section 4 but has a different order of policies. It first selects tokens in each sentence, and then selects sentences, whose suffix can be abbreviated as "from token to sentence (tk2st)".

RSLN-cpnet has collapsed policy networks compared with RSLN. For both TPNet and SPNet, RSLN-cpnet employs neither SEL nor DEL, and only considers fully connected layers.

RSLN-st2tk, or RSLN for short, is the model proposed in Section 4. In order to distinguish from other variants, RSLN and RSLN-st2tk are equivalent to each other unless particularly stated. Apparently, the suffix "st2tk" means from sentence to token.

6.4 Overall Results and Analysis

Table 4 summarizes the overall performance of the RSLN model compared with several baselines on the end-to-end DEFI task. To display a set of more persuasive and meaningful results, Table 5 also reports the performance of our RSLN model and several representative baselines on the DLEF-v2 corpus. Tables

Table 4. Performance ($F1$-Score) of Various Models on the ExDLEF Corpus for the DEFI Task

Sub-Corpus	Model	CT−	PS+	CT+	*Macro*F1	*Micro*F1
English	SGCN	46.72	43.04	78.67	56.14	69.01
	LSTM-A	44.19	43.45	81.17	56.27	69.21
	BERT-MSF	46.65	43.72	83.09	57.82	71.87
	ULGN	47.38	45.06	82.26	58.23	71.74
	TEND-C	44.74	44.12	82.48	57.11	70.03
	TEND-T	49.46	48.87	83.25	60.53	72.01
	RLSTM	51.40	50.34	83.88	61.87	73.52
	BERT-B	53.92	52.88	83.46	63.42	74.99
	RMHAN	57.25	55.41	84.59	65.75	76.33
	CNet	52.67	52.13	82.80	62.53	74.08
	RSLN-L	50.59	49.87	81.60	60.69	72.64
	RSLN-cpnet	56.66	54.06	83.92	64.88	74.92
	RSLN-tk2sp	51.49	51.55	81.73	61.59	72.63
	RSLN-sp2tk	56.85	54.61	84.05	65.17	75.56
	RSLN-tk2st	57.72	56.59	84.41	66.24	76.58
	RSLN	**59.23**	**57.88**	**85.17**	**67.42**	**77.48**
Chinese	SGCN	64.02	50.34	77.90	64.09	68.46
	LSTM-A	62.52	52.71	79.85	65.03	69.19
	BERT-MSF	63.29	52.88	78.28	64.82	68.93
	ULGN	64.43	54.90	78.98	66.10	69.99
	TEND-C	63.08	53.79	79.62	65.51	69.44
	TEND-T	65.17	56.31	80.08	67.19	70.92
	RLSTM	66.35	57.17	80.59	68.04	71.73
	BERT-B	72.44	62.25	82.31	72.33	75.62
	RMHAN	74.78	65.27	82.53	74.19	77.02
	CNet	70.06	61.58	82.02	71.22	74.64
	RSLN-L	71.83	61.17	81.55	71.52	74.56
	RSLN-cpnet	73.11	63.36	82.49	72.99	75.91
	RSLN-tk2sp	69.26	58.61	80.35	69.41	73.07
	RSLN-sp2tk	75.09	64.82	**83.17**	74.36	77.05
	RSLN-tk2st	74.96	66.28	82.65	74.62	77.38
	RSLN	**76.12**	**67.97**	83.14	**75.74**	**78.29**

4 and 5 show that RSLN performs superior to other models, proving that both comprehensive encoding and text selection is meaningful and effective. According to the inherent adaptive advantages of RSLN, we analyze the comparison with baselines in the following aspects.

Design of Document-Level Model. The performance of SGCN is much lower than that of the other end-to-end document-level models, including RMHAN and RSLN. According to Section 3, document-level event factuality requires the comprehensive understanding of semantics. However, SGCN identifies event factuality in each sentence separately, and uses the most frequent sentence-level value as the document-level one. But this simple voting is not consistent with the definition of DEFI.

Performance of Various Categories of Factuality Values. For all the models, the results of CT+ are higher than CT− and PS+ due to the majority of CT+ events. It is not surprising, since DEFI models can learn more information from CT+ samples compared with other values. Hence, the performance of a model mainly depends on CT− and PS+, and our RSLN achieves more improvements on CT− and PS+ than other baselines.

Discrepancy of Pipeline and End-to-End Framework. Previous sentence-level (SGCN) and document-level (LSTM-A, BERT-MSF, and ULGN) models focus on the task different from this paper, as analyzed in Section 5 , and they usually rely on annotated information. We also employ them as baselines due to the minority of relevant DEFI models. It is worth noting that our RSLN is an end-to-end model. For fair comparison with SGCN, LSTM-A, and ULGN, we first launch upstream tasks to detect various factors (e.g., event triggers, speculative cues, and negative cues) whose performance is presented in Table 3, and then predict document-level factuality. Apart from errors predicted by those DEFI models, the main wrong cases are due to cascade errors propagat-

Table 5. Performance (F1-Score) of Various Models on the DLEF-v2 Corpus for the DEFI Task

Sub-Corpus	Model	CT−	PS+	CT+	*MacroF*1	*MicroF*1
English	SGCN	45.48	40.80	77.71	54.66	67.20
	ULGN	45.87	43.05	81.87	56.93	70.55
	RLSTM	49.89	48.80	82.04	60.24	72.39
	RMHAN	56.43	55.13	84.35	65.30	76.38
	RSLN-sp2tk	56.20	54.88	84.26	65.11	76.23
	RSLN-tk2st	57.76	55.67	84.37	65.93	76.61
	RSLN	**59.12**	**57.24**	**85.49**	**67.29**	**77.47**
Chinese	SGCN	60.78	50.63	76.82	62.74	66.52
	ULGN	61.07	49.58	76.49	62.38	66.27
	RLSTM	64.64	54.83	77.92	65.80	69.51
	RMHAN	73.83	65.55	82.60	73.99	77.07
	RSLN-sp2tk	74.08	64.52	83.26	73.96	77.13
	RSLN-tk2st	74.81	65.67	**83.41**	74.63	77.58
	RSLN	**75.67**	**67.23**	83.33	**75.41**	**78.22**

ed from upstream tasks.

Settings of Complicated Attention Networks. The RSLN model obviously outperforms those baselines with plain and simple structures, e.g., TEND-C, TEND-T, and RLSTM, proving the stronger ability of encoding of attentions (including multi-head attention) compared with other methods (e.g., LSTM). Due to the framework of attention that is able to ascertain the most useful and relevant texts, RSLN can not only learn significant internal semantics within sentences, but also capture interactions at various levels of granularity among events, topics, and documents.

Usefulness of Hierarchical Encoding. Some baselines ignore the structured characteristic among events and documents, and simply concatenate all the input into one sequence, e.g., BERT-B, RSLN-L. Our RSLN obtains higher results than them, manifesting that linear input and encoding is not conducive to distinguish different semantics of sentences. We argue that a hierarchical model is more capable of capturing meaningful features from sentences, especially from the core ones of the document, since sentences probably hold different factualtiy of the event, i.e., inconsistency of sentence-level factuality. Meanwhile, it is the hierarchical encoding that makes the model to select the most useful sentences by SPNet.

Effectiveness of Reinforcement Learning. 1) Firstly, compared with those models without RL for text selection, RSLN is able to achieve better performance, demonstrating that policy networks can select the most relevant and meaningful sentences and tokens based on RL. 2) Secondly, under the framework of sentence and token selection, RSLN is superior to RSLN-tk2st, which means selecting sentences firstly is more effective than selecting tokens firstly. The main reason is that if we first launch token selection, the incomplete token sequence can affect the semantical integrity of sentences and may have side effects on the sentence selection. 3) Thirdly, we also compare the sentence selection and span selection. Table 4 shows that RSLN-st2tk outperforms RSLN-sp2tk, mainly because selecting continuous spans with relatively complete semantics is more difficult than selecting sentences, because a sentence usually holds one specific factuality of the event, while span may include several factuality values inconsistent with the document-level one. Among the variants of RSLN, RSLN-tk2sp gains relatively lower results, indicating that it is not wise to perform token selection before span selection. The reason is probably that selecting spans relies on a few key boundary tokens that may be abandoned during token selection and cause wrong detection of spans. Therefore, we mainly focus on sentence and token selection in the RSLN model.

6.5 Ablation Study

This subsection aims to verify the impact of each key component of the RSLN model, which is ablated into several simplified models as follows.

1) w/o Gate denotes that our RSLN model does not utilize the Gate mechanism in attention sub-networks.

2) w/o ISEnc/EVEnc/TPEnc/IDEnc means it does not consider ISEnc/EVEnc/TPEnc/IDEnc in CNet or policy networks SPNet/TPNet.

3) w/o FSEnc means it only considers the fully connected layer to fuse the output of ISEnc, EVEnc, and TPEnc without attention and residual layers in

FSEnc.

4) w/o SPNet denotes this model does not incorporate sentence selection policy network (SPNet), and encodes all the sentences indiscriminately.

5) w/o TPNet denotes this model does not utilize token selection policy network (TPNet), and encodes all the tokens.

The performance of the ablation study is shown in Table 6, which can be discussed from the following components in detail.

All the Components. We mainly investigate TPNet, SPNet, encoders in SEL (ISEnc, EVEnc, TCEnc, FSEnc), and DEL (IDEnc). The removal of each component all weaken the performance of the RSLN model, and the improvement of macro-/micro-averaged F1 vary among $[-6.97, -1.35]/[-7.33, -1.27]$ and $[-7.91, -0.85]/[-7.94, -0.84]$ on English and Chinese sub-corpus, respectively. Thus, we can deem that all the sub-networks contribute to the RSLN model in positive ways, and can prove their organic integrity.

Gate Mechanism in Attention Sub-Networks. We get lower performance if we neglect gates in attention sub-networks, which can manifest that gates are effective and helpful for DEFI. Actually, gates can filter irrelevant information, which are in line with text selection and refinement in RSLN, and then can offer supplementary clues for policy networks.

Sentence Encoding Layer in CNet. Encoders in SEL are mainly comprised of ISEnc, EVEnc, TCEnc, and FSEnc. 1) Firstly, w/o ISEnc underperforms RSLN, certifying the necessity of self-attention for discovering valuable information (e.g., speculation and negation) at the sentence level. 2) Secondly, w/o EVEnc/TCEnc leads to more degradation in performance, due to the property of events and topics. Events are brief but indispensable clues with fundamental event-related information, e.g., event triggers and arguments. While topic sentences usually summarize the core idea of the document, and have richer semantics than events. Therefore, our model can boost more performance by using TCEnc than EVEnc. 3) Thirdly, w/o FSEnc causes lower drops on results than other encoders, which manifests its validity. FSEnc is employed to eliminate the manifold differences among representations learned by ISEnc/EVEnc/TCEnc, and extract higher-level semantics. Thus, w/o FSEnc causes less loss of input texts than other encoders.

Document Encoding Layer in CNet. Table 6 exhibits that the performance of the RSLN model declines if neglecting IDEnc in DEL, especially on CT− and PS+, whose inconsistency between the sentence-level and document-level factuality is relatively obvious. The primary function of IDEnc is extracting interactive knowledge from sentences, especially those with speculative and negative meanings. Hence, IDEnc can determine whether speculation and negation propagates to the entire document and affect its

Table 6. Performance of Ablation Study for the RSLN Model

Sub-Corpus	Model	CT−	PS+	CT+	*MacroF*1	*MicroF*1
English	RSLN	59.23	57.88	85.17	67.42	77.48
	w/o Gate	−1.86	−2.70	+0.47	−1.36	−0.11
	w/o ISEnc	−4.18	−4.91	−3.63	−4.23	−4.77
	w/o EVEnc	−7.42	−7.31	−6.19	−6.97	−7.33
	w/o TCEnc	−6.39	−8.43	−5.15	−6.65	−6.92
	w/o FSEnc	−2.84	−2.17	−1.31	−2.10	−2.31
	w/o IDEnc	−3.51	−3.84	−2.49	−3.27	−3.29
	w/o TPNet	−2.10	−1.66	−0.30	−1.35	−1.27
	w/o SPNet	−5.66	−3.70	−2.86	−4.07	−4.10
Chinese	RSLN	76.12	67.97	83.14	75.74	78.29
	w/o Gate	−2.14	−2.81	−1.66	−2.20	−2.22
	w/o ISEnc	−4.20	−4.69	−4.06	−4.31	−4.57
	w/o EVEnc	−8.49	−8.21	−6.15	−7.61	−7.55
	w/o TCEnc	−8.87	−7.49	−7.38	−7.91	−7.94
	w/o FSEnc	−2.74	−3.38	−2.25	−2.79	−3.19
	w/o IDEnc	−5.68	−4.79	−2.84	−4.43	−4.21
	w/o TPNet	−1.16	−1.65	+0.24	−0.85	−0.84
	w/o SPNet	−4.95	−5.14	−3.83	−4.64	−4.71

Note: The values are *F*1-scores for the complete RSLN model, and improvements of *F*1-scores for other models without (w/o) some components.

factuality, implying that IDEnc plays an important role in deciding which of the selected sentences have greatest impacts on correct results, especially for non-CT+ events.

Policy Networks Based on Hierarchical Reinforcement Learning. From Table 6, we can observe that the performance degrades more when considering w/o SPNet than w/o TPNet, which confirms that sentence selection is more beneficial and significant than token selection. The principal reason is that sentences are basic units with complete semantics. Compared with discontinuous tokens, sentences can convey more accurate meanings. On the one hand, if we do not consider sentence selection and encode all of them, those noisy sentences, which are not related to the event or hold different factuality with the document, may mislead our model to gain wrong results. On the other hand, if we ignore token selection and feed all of them into CNet, the tokens that are useless and ineffective for the events have lower impacts on the understanding of document-level factuality, because they attain smaller attention weights computed by vanilla attention pooling.

Therefore, the above analysis of ablation is able to validate the components of our RSLN model.

6.6 Case Study

As described above, the RSLN model considers both sentence and token selection. To interpret the predicted results more convincingly, this subsection displays qualitative analysis on event E1 and E2 by Fig.4. We also consider RSLN-sp2tk that can also achieve better results than most other models for comparison in Fig.5.

6.6.1 *Case Study for RSLN*

As analyzed in Section 1 , the document-level factuality of E1 in Fig.4(a) is CT−, and we need to extract negative information and determine whether they can negate E1. For token selection, we can see that the selected tokens that are the most helpful for DEFI can be classified into two types: 1) negative tokens and cues, of course, e.g., "deny", "untrustworthy", and "not"; 2) event triggers and arguments that convey the elementary and essential information of the event E1, e.g., "cancel", "Tokyo", "Olympics", and "COVID-19". Consequently, these tokens can confirm the ability of SEL and TPNet.

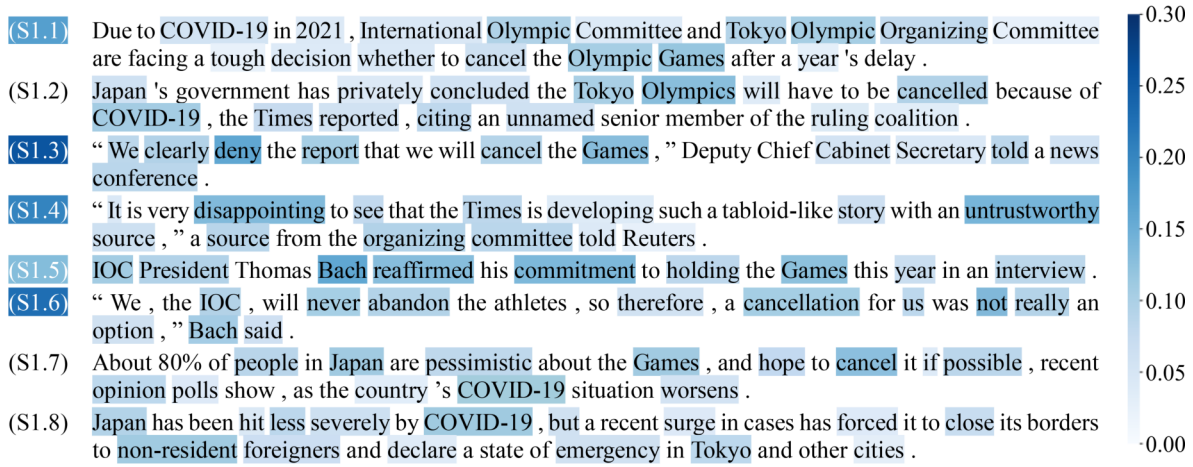
In term of sentence selection, we observe that those selected sentences (especially S1.3, S1.4, and S1.6) contain negative cues, event triggers, and arguments, and they can summarize the main idea of this document with regard to E1, as analyzed in Section 1, certifying the effectiveness of EVEnc, SPNet, and VA. Next, we also notice that although stating speculation rather than negation, S1 is also captured, since it conveys richer information about event arguments, which can validate capability of TCEnc. Finally, selected sentences vary among factuality, i.e., speculation/PS+ (S1.1), negation/CT− (S1.3, S1.4, and S1.6), and uncommitted (S1.5), and our model gives the correct result CT−, which is owing to IDEnc capturing event-related and meaningful interactive features among sentences.

Similarly, we examine the visualization of a PS+ event E2 as shown in Fig.4(b). We can see that valuable captured tokens mainly cover speculative words ("plan", "possibly"), event triggers ("return"), and arguments ("NASA", "Moon"), which are key syntactic and semantical elements contributing to the factuality. Based on semantics, selected sentences also fall into two categories: 1) the ones that have triggers and arguments of E2, regardless of their sentence-level factuality, and 2) those related speculative ones that narrate the core semantics of the document with regard to E2.

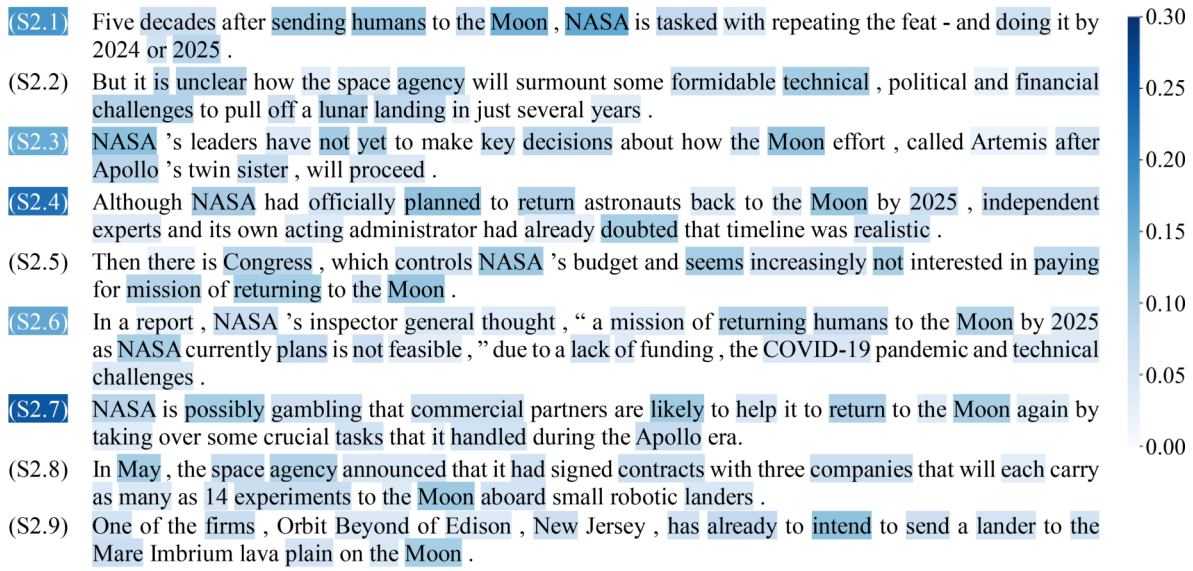
6.6.2 *Case Study for RSLN-sp2tk*

To clarify the advantages of sentence selection compared with span extraction, we display the selected spans and attention weights of tokens for the events E1 and E2 computed by RSLN-sp2tk in Fig.5. RSLN-sp2tk makes a correct prediction CT− for E1 in Fig.5(a), because the extracted span comprises the mention of it, where "deny" holds the negative position without disturbance from other clauses.

However, the PS+ event E2 is predicted as false value CT− in Fig.5(b). We infer that it is mainly due to the negative semantics from the negative cues "not" in the extracted spans "not interested in paying for..." and "... is not feasible", although speculation is in the span and selected speculative cue "possible" has been assigned a significant weight. The first "not" negates another event "Congress is interested in paying for returning to the Moon", while the second "not" denies E2. As for the result, the negative semantics has more impacts on E2 than speculation in

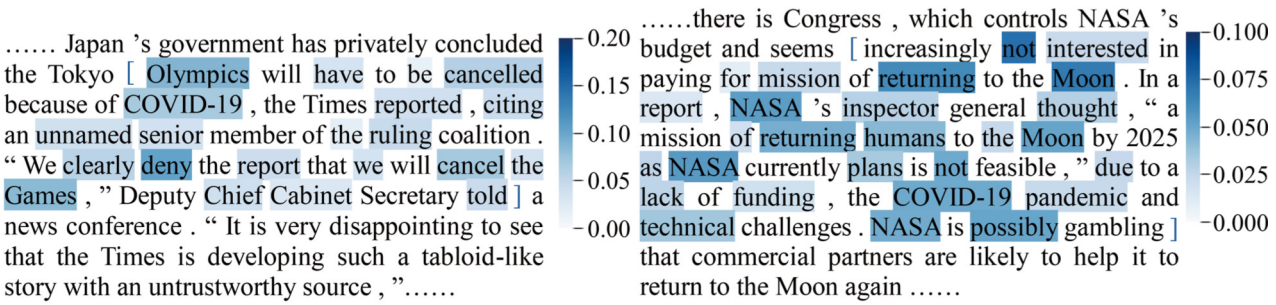


(a)



(b)

Fig.4. Visualizations of the sentences and tokens selected by RSLN (RSLN-st2tk). Th events are (a) the CT− event E1 "Tokyo Olympics is canceled in 2021" and (b) the PS+ event E2 "NASA returns humans to the Moon". Attention weights are computed by vanilla attention (VA) pooling, and are visualized as background colors of sentence IDs and tokens, while no background color means this sentence or token has been discarded. According to the encoding procedure of RSLN, unselected sentences have no selected tokens. To be consistent with selected sentences, we also launch token selection for unselected sentences by the trained TPNet.



(a) (b)

Fig.5. Visualizations of the span and tokens selected by RSLN-sp2tk for the events (a) E1 "Tokyo Olympics is canceled in 2021" (CT−/CT−) and (b) E2 "NASA returns humans to the Moon" (PS+/CT−), where the formats of labels are (Annotated/Predicted) ones, and the extracted spans are in square brackets.

the span, which is distinguished from the document-level factuality.

Hence, these two cases in Fig.5 show that spans may be comprised of various speculation and negation that can cause interference to the identification of event factuality, and illustrate that sentence selection is more reasonable and effective than span extraction.

6.7 Error Analysis

As mentioned in Subsection 6.4, Table 4 reveals that the performance of CT− and PS+ is usually lower than that of CT+ due to the minority of speculative and negative samples. Therefore, the wrong results mainly come from speculation and negation, and can be classified as three types, as exemplified in Fig.6.

CT+ events are predicted as non-CT+ (e.g., CT−, PS+). This type of error is mainly due to the interference from irrelevant speculation and negation. For example, in Fig.6(a), the event E3 "Phil Valentine died from COVID-19" is a fact CT+ decided by the sentence S3.4. However, some selected sentences evaluate E3 as non-CT+ values, e.g., S3.1 and S3.3 commit to it as PS+ according to the speculative cues "a chance of" and "possibility", and S3.2 holds PS−, because E3 in S3.2 is governed by the speculative cue "probably" and negative cue "not". Actually, S3.1, S3.2, and S3.3 are before the current event on the timeline, and cannot affect E3 semantically. But our model fails to discard these non-CT+ mentions leading to the wrong results.

Non-CT+ events are predicted as CT+. It is primarily because our model fails to extract corresponding speculative or negative for CT− or PS+ event. In Fig.6(b), the event E4 is PS+ inferred from the sentences S4.2 and S4.3, but our model predicts E4 as

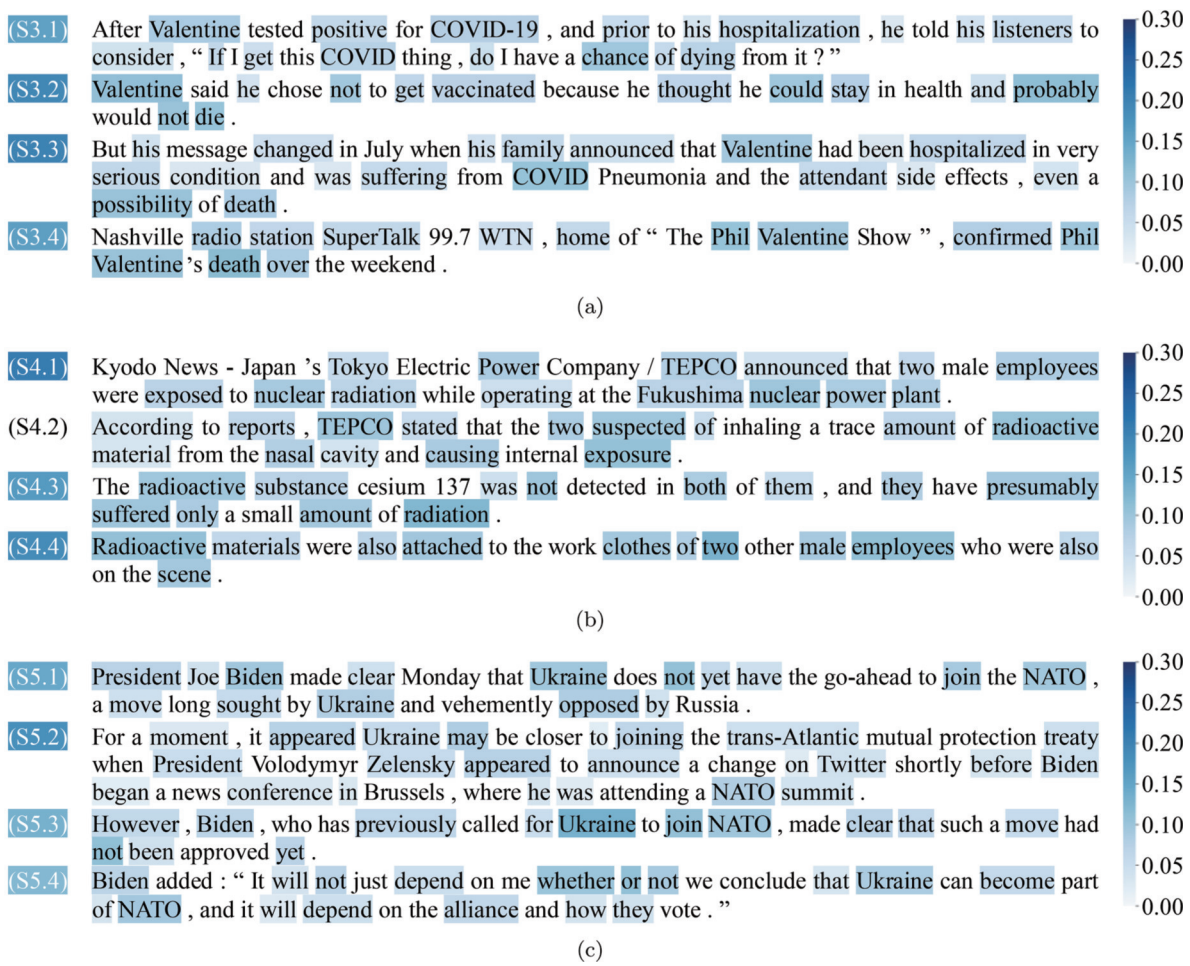


Fig.6. Visualizations of error cases, in which the sentences and tokens are selected by the RSLN model. The events are (a) E3 "Phil Valentine died from COVID-19" (CT+/PS+), (b) E4 "Two employees of Fukushima nuclear power plant were exposed to nuclear radiation" (PS+/CT+), and (c) E5 "Ukraine joins NATO" (CT−/PS+). The formats of labels are (Annotated / Predicted) ones. For simplification, only several representative sentences (e.g., those selected by SPNet, or containing event mentions) are listed.

CT+, mainly owing to S4.1. Fig.6(b) visualizes that S4.1 (the first sentence in the document) is assigned relatively high attention weight and holds the value CT+ about E4. S4.4 mentions another CT+ event "Radiation were attached to two other male employees". As for S4.2 and S4.3 with speculative information that are primary clues for PS+, S4.2 is not selected by RSLN, and S4.3 has lower attention weight than S4.1. We conjecture that the speculative cues "suspected" in S4.2 and "presumably" in S4.3 appear less frequently than other ones (like "may", "likely", and "possible"), and cannot learn enough information to get a higher weight from the training set.

Non-CT+ (mainly CT−, PS+) events are predicted as another non-CT+ value. The main reason is that our model is confused by various speculation and negation, and cannot determine which one to concentrate on. In Fig.6(c), the event E5 "Ukraine joins NATO" is negated by S5.1 and S5.3 that state the core semantics of the document, and is annotated as CT−. But there are also other sentences (S5.2 and S5.4) holding PS+ accounting for the wrong result, and are not filtered out. Moreover, S5.2 and S5.4 obtain higher weights, especially S5.2 with speculative cues "appear" and "may".

These error cases exemplify the significance of speculation and negation in the DEFI task. We need to design an appropriate DEFI model that can not only extract speculative and negative information, but also determine whether this information can govern the event from the view of the document.

7 Conclusions

This paper is devoted to end-to-end document event factuality identification (DEFI), and can be concluded as the following aspects.

We presented a clear definition of the end-to-end DEFI task that only considers the event, document, and factuality as input for training.

We proposed a reinforced multi-granularity hierarchical network model named Reinforced Semantic Learning Network (RSLN) as the solution. RSLN can not only capture semantics using encoders with hierarchical structure at different levels of granularity, but also select relevant and meaningful sentences and tokens employing policy networks with hierarchical reinforcement learning. Therefore, the RSLN model can solve the problems including end-to-end modeling formulation, comprehensive encoding network, and text selection mechanism.

We contributed a novel corpus called ExDLEF to assess our RSLN model, and this dataset is in line with the end-to-end task. Experimental results manifest that RSLN outperforms the state-of-the-arts.

In the future work, we plan to launch more fine-grained DEFI tasks, e.g., identifying factuality of several events simultaneously, and extracting evidential sentences of them to explore high-level interpretability. Furthermore, we will also explore cross-document event factuality identification.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Rudinger R, White A S, Van Durme B. Neural models of factuality. In *Proc. the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2018, pp.731–744. DOI: 10.18653/v1/n18-1067.

[2] Qian Z, Li P, Zhou G, Zhu Q. Event factuality identification via hybrid neural networks. In *Proc. the 25th International Conference on Neural Information Processing*, Dec. 2018, pp.335–347. DOI: 10.1007/978-3-030-04221-9_30.

[3] Qian Z, Li P, Zhang Y, Zhou G, Zhu Q. Event factuality identification via generative adversarial networks with auxiliary classification. In *Proc. the 27th International Joint Conference on Artificial Intelligence*, Jul. 2018, pp.4293–4300. DOI: 10.24963/ijcai.2018/597.

[4] Sheng J, Zou B, Gong Z, Hong Y, Zhou G. Chinese event factuality detection. In *Proc. the 8th CCF International Conference on Natural Language Processing and Chinese Computing*, Oct. 2019, pp.486–496. DOI: 10.1007/978-3-030-32236-6_44.

[5] Veyseh A P B, Nguyen T H, Dou D. Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *Proc. the 57th Conference of the Association for Computational Linguistics*, Oct. 2019, pp.4393–4399. DOI: 10.18653/v1/p19-1432.

[6] Qian Z, Li P, Zhu Q, Zhou G. Document-level event factuality identification via adversarial neural network. In *Proc. the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2019, pp.2799–2809. DOI: 10.18653/v1/n19-1287.

[7] Huang R, Zou B, Wang H, Li P, Zhou G. Event factuality detection in discourse. In *Proc. the 8th CCF International Conference on Natural Language Processing and Chinese Computing*, Oct. 2019, pp.404–414. DOI: 10.1007/978-3-030-32236-6_36.

[8] Cao P, Chen Y, Yang Y, Liu K, Zhao J. Uncertain local-to-global networks for document-level event factuality identification. In *Proc. the 2021 Conference on Empirical*

Methods in Natural Language Processing, Nov. 2021, pp.2636–2645. DOI: 10.18653/v1/2021.emnlp-main.207.

[9] Liu J, Pan F, Luo L. GoChat: Goal-oriented chatbots with hierarchical reinforcement learning. In *Proc. the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2020, pp.1793–1796. DOI: 10.1145/3397271.3401250.

[10] Wang J, Sun C, Li S, Wang J, Si L, Zhang M, Liu X, Zhou G. Human-like decision making: Document-level aspect sentiment classification via hierarchical reinforcement learning. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov. 2019, pp.5580–5589. DOI: 10.18653/v1/D19-1560.

[11] Xiao L, Wang L, He H, Jin Y. Copy or rewrite: Hybrid summarization with hierarchical reinforcement learning. In *Proc. the 34th AAAI Conference on Artificial Intelligence*, Feb. 2020, pp.9306–9313. DOI: 10.1609/AAAI.V34I05.6470.

[12] Wan G, Pan S, Gong C, Zhou C, Haffari G. Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning. In *Proc. the 29th International Joint Conference on Artificial Intelligence*, Jul. 2020, pp.1926–1932. DOI: 10.24963/ijcai.2020/267.

[13] Zhou X, Luo S, Wu Y. Co-attention hierarchical network: Generating coherent long distractors for reading comprehension. In *Proc. the 34th AAAI Conference on Artificial Intelligence*, Feb. 2020, pp.9725–9732. DOI: 10.1609/AAAI.V34I05.6522.

[14] Wu L, Rao Y, Zhao Y, Liang H, Nazir A. DTCA: Decision tree-based co-attention networks for explainable claim verification. In *Proc. the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp.1024–1035. DOI: 10.18653/v1/2020.acl-main.97.

[15] Wu Y, Zhan P, Zhang Y, Wang L, Xu Z. Multimodal fusion with co-attention networks for fake news detection. In *Proc. the 2021 Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Aug. 2021, pp.2560–2569. DOI: 10.18653/v1/2021.findings-acl.226.

[16] Lai T M, Tran Q H, Bui T, Kihara D. A gated self-attention memory network for answer selection. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Nov. 2019, pp.5953–5959. DOI: 10.18653/v1/D19-1610.

[17] Xue L, Li X, Zhang N L. Not all attention is needed: Gated attention network for sequence data. In *Proc. the 34th AAAI Conference on Artificial Intelligence*, Feb. 2020, pp.6550–6557. DOI: 10.1609/AAAI.V34I04.6129.

[18] Liu L, Chen H, Sun Y. A multi-classification sentiment analysis model of Chinese short text based on gated linear units and attention mechanism. *Trans. Asian and Low-Resource Language Information Processing*, 2021, 20(6): Article No. 109. DOI: 10.1145/3464425.

[19] Chen Z, Hui S C, Zhuang F, Liao L, Li F, Jia M, Li J. EvidenceNet: Evidence fusion network for fact verification. In *Proc. the 2022 ACM Web Conference*, Apr. 2022, pp.2636–2645. DOI: 10.1145/3485447.3512135.

[20] Chen J, Bao Q, Sun C, Zhang X, Chen J, Zhou H, Xiao Y, Li L. LOREN: Logic-regularized reasoning for interpretable fact verification. In *Proc. the 36th AAAI Conference on Artificial Intelligence*, Feb. 2022, pp.10482–10491. DOI: 10.1609/AAAI.V36I10.21291.

[21] Ma J, Gao W, Joty S, Wong K F. Sentence-level evidence embedding for claim verification with hierarchical attention networks. In *Proc. the 57th Annual Meeting of the Association for Computational Linguistics*, Jul. 2019, pp.2561–2571. DOI: 10.18653/v1/p19-1244.

[22] Chen J, Zhang R, Guo J, Fan Y, Cheng X. GERE: Generative evidence retrieval for fact verification. In *Proc. the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2022, pp.2184–2189. DOI: 10.1145/3477495.3531827.

[23] Qian Z, Li P, Zhu Q, Zhou G. Document-level event factuality identification via reinforced multi-granularity hierarchical attention networks. In *Proc. the 31st International Joint Conference on Artificial Intelligence*, Jul. 2022, pp.4338–4345. DOI: 10.24963/ijcai.2022/602.

[24] Pennington J, Socher R, Manning C D. GloVe: Global vectors for word representation. In *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing*, Oct. 2014, pp.1532–1543. DOI: 10.3115/v1/d14-1162.

[25] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I. Attention is all you need. In *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.5998–6008.

[26] Zhang T, Huang M, Zhao L. Learning structured representation for text classification via reinforcement learning. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, Feb. 2018, pp.6053–6060. DOI: 10.1609/AAAI.V32I1.12047.

[27] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992, 8(3): 229–256. DOI: 10.1007/BF00992696.

[28] Sutton R S, McAllester D, Singh S, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In *Proc. the 12th International Conference on Neural Information Processing Systems*, Nov. 1999, pp.1057–1063.

[29] Kingma D P, Ba J. Adam: A method for stochastic optimization. In *Proc. 3rd International Conference on Learning Representations*, 2015. DOI: 10.48550/arXiv.1412.6980.

[30] Robbins H, Monro S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 1951, 22(3): 400-407. DOI: 10.1214/aoms/1177729586.

[31] Zhang H, Qian Z, Zhu X, Li P. Document-level event factuality identification using negation and speculation scope. In *Proc. the 28th International Conference on Neural Information Processing*, Dec. 2021, pp.414–425. DOI: 10.1007/978-3-030-92185-9_34.

[32] Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pre-

training of deep bidirectional transformers for language understanding. In *Proc. the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2019, pp.4171–4186. DOI: 10.18653/v1/n19-1423.

[33] Duan J, Zhang Y, Ding X, Chang C Y, Liu T. Learning target-specific representations of financial news documents for cumulative abnormal return prediction. In *Proc. the 27th International Conference on Computational Linguistics*, Aug. 2018, pp.2823–2833.

[34] Duan J, Ding X, Zhang Y, Liu T. TEND: A target-dependent representation learning framework for news document. *IEEE/ACM Trans Audio, Speech, and Language Processing*, 2019, 27(12): 2313–2325. DOI: 10.1109/TASLP.2019.2947364.

Zhong Qian received his B.S. and Ph.D. degrees in computer science and technology from Soochow University, Suzhou, in 2012 and 2018, respectively. Currently, he is an associate professor in the School of Computer Science and Technology, Soochow University, Suzhou. His main research interest is information extraction in natural language processing.

Pei-Feng Li received his B.S., M.S., and Ph.D. degrees all in computer science from Soochow University, Suzhou, in 1994, 1997, and 2006, respectively. Currently, he is a professor at the School of Computer Science and Technology, and AI Research Institute, Soochow University, Suzhou. His current research interests include Chinese information processing, machine learning, and information extraction.

Qiao-Ming Zhu received his Ph.D. degree in computer science and technology from Soochow University, Suzhou, in 2008. Currently, he is a professor at the School of Computer Science and Technology, and AI Research Institute, Soochow University, and acts as the director of Department of Science, Technology and Industry in Soochow University, Suzhou. His research interests include natural language processing, information extraction, and embedded systems.

Guo-Dong Zhou received his Ph.D. degree in computer science from the National University of Singapore, Singapore, in 1999. Currently, he is a professor at the School of Computer Science and Technology, and AI Research Institute, Soochow University, Suzhou. His research interests include natural language processing, information extraction, and machine learning.