Information Processing and Management 59 (2022) 102836

Contents lists available at ScienceDirect

Information Processing and Management

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

End-to-end event factuality prediction using directional labeled graph recurrent network

Xiao Liu [a], Heyan Huang [a,*], Yue Zhang [b,c]

[a] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081, China
[b] School of Engineering, Westlake University, Zhejiang, 310024, China
[c] Institute of Advanced Technology, Westlake Institute for Advanced Study, Zhejiang, 310024, China

ARTICLE INFO

Keywords:
Event factuality prediction
Event Anchor Detection
Joint modeling
Graph neural network
End-to-end
Syntactic information graph

ABSTRACT

Event factuality prediction is the task of assessing the degree to which an event mentioned in a sentence has happened. However, existing methods usually stack encoders to make factuality predictions given the gold positions of anchor words. In addition, the frequently used encoders, such as bidirectional LSTMS and graph convolution networks, ignore the directional labeled syntactic information while modeling the context. To fill the gap when facing plain text without identifying event anchor words in advance, we investigate the task of end-to-end EFP in this paper. We present the Directional Labeled Graph Recurrent Network, denoted as DLGRN, to solve Event Anchor Detection and Factuality Induction in a multi-task framework. Specifically, we represent sentences as syntactic information graphs. Then, to incorporate directional labeled information, we design edge-tied weights and edge-aware attention mechanism on top of a graph-based recurrently message passing encoder. We further propose to utilize multi-task learning to jointly model Event Anchor Detection and Factuality Induction by optimizing a mixed-objective learning function. We use four widely used factuality prediction benchmarks (i.e., FactBank, Meantime, UW, and UDS-IH2) to evaluate our framework. Our framework achieves state-of-the-art results in the two subtasks, averagely decreasing 17.12% MAE and raising 5.40% Pearson correlation r against the best baseline. In addition, experimental results show that our framework can capture the overall factuality score distributions, and incorporating directional and labeled syntactic information in EFP achieves better performances than the baselines.

1. Introduction

Event factuality prediction (EFP) is the task of estimating the factuality of events in texts. The goal is to recognize whether event mentions in texts represent actual situations in the world, situations that have not happened, or situations of uncertain interpretation (Saurí, 2008). For example, given the sentence "*Expert says the ground is too saturated*", the event "*Expert says*" is actually HAPPENED. In many cases, the factuality of events is conveyed by what we refer to as *event mention anchor words*, which are predicates including verbs, nouns, or adjectives denoting events. For example, in the case above, the event mention anchor word for the event is "*says*". Accurately predicting event factuality is essential for supporting downstream inferences that are based on the facts.

Recent work on EFP (Lee, Artzi, Choi, & Zettlemoyer, 2015; Rudinger, White, & Durme, 2018; Veyseh, Nguyen, & Dou, 2019) assumes that event mention anchor words are given in advance, predicting factuality of events by assigning labels to the anchor

* Corresponding author at: School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081, China.
E-mail addresses: xiaoliu@bit.edu.cn (X. Liu), hhy63@bit.edu.cn (H. Huang), yue.zhang@wias.org.cn (Y. Zhang).

https://doi.org/10.1016/j.ipm.2021.102836
Received 13 May 2021; Received in revised form 17 November 2021; Accepted 19 November 2021
Available online 8 December 2021

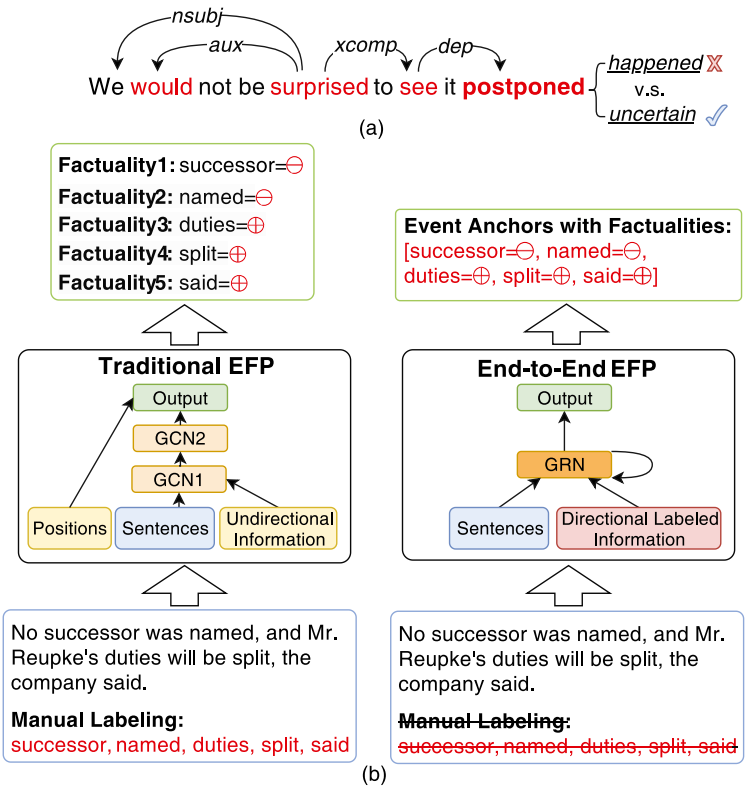


Fig. 1. (a) Example sentence with the dependency parse tree showing the factuality of the event represented by the bold red anchor. (b) Comparison between end-to-end EFP and traditional EFP. The blue rectangles are used to represent input examples. The black rectangles are typical models and features for both task settings. And the green rectangles represent output examples. ⊕ stands for HAPPENED and ⊖ means HAVE NOT HAPPENED. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

words according to their context. Event factuality prediction is a challenging task as there may be several words in the context jointly contributing to the factuality of the event, which can be far separated from each other and also from the event anchor words in the sentence. The state-of-the-art methods (Rudinger et al., 2018; Veyseh et al., 2019) stack neural network encoders like biLSTMs or graph convolution networks (GCNs) (Chen, Wei, Huang, Ding, & Li, 2020; Wei et al., 2019; Yang, Qiu, Song, Tao, & Wang, 2020; Yu, Wang, & Zhang, 2021) to model the context and make factuality predictions on hidden vectors of anchor words.

There are two main limitations of the abovementioned methods. First, in practice, given raw text data as input, these methods relies on event anchor word detection as a prerequisite task in a pipeline, errors in event anchor detection can propagate to the traditional EFP task, leading to decreased performance. Second, existing methods only model coarse-grained graph information using undirected edges, but do not consider directional or labeled syntactic information when modeling the context, which also limits the use of syntactic knowledge (Liao et al., 2019; Tao et al., 2020; Wu, Chen, & Wan, 2018). As illustrated in Fig. 1(a), in the sentence "*We would not be surprised to see it postponed*", if we want to predict the factuality of the event mention anchor word "*postponed*", it is difficult to choose the correct factuality category between the two situations, HAPPENED or UNCERTAIN. We can find that a fine-grained dependency path from "*would*" to "*postponed*" is $would \xleftarrow{\text{aux}} surprised \xrightarrow{\text{xcomp}} see \xrightarrow{\text{dep}} postponed$, which indicates a subjunctive and hypothetical mood for the event "*postponed*". If we take the fine-grained syntactic knowledge into account, it would be easier to predict the factuality to be UNCERTAIN instead of HAPPENED.

To address these issues, we consider a fine-grained graph neural network for end-to-end EFP, performing Event Anchor Detection and Factuality Induction jointly. As shown in Fig. 1(b), different from traditional EFP, whose input is sentences and positions, and output is factuality scores at given positions, end-to-end EFP takes plain text sentences as the input and identifies event anchor words along with predicted factuality scores. We present a graph neural multi-task framework for modeling the directional and labeled syntactic information, which is named Directional Labeled Graph Recurrent Network (DLGRN). DLGRN first constructs a message passing graph for each input sentence and then learns shared word states and graph states for Event Anchor Detection and Factuality Induction. The states are learnt in a graph-based message passing process, implemented by extending graph recurrent network (GRN) (Song, Zhang, Wang, & Gildea, 2018; Yin et al., 2019; Zhang, Liu, & Song, 2018) with edge-tied weights and edge-aware attention mechanism. The message passing process contains two steps, context computation and state transition, which is recurrently stacked to model the long dependencies with the help of the parameter-sharing mechanism. At last, the shared word states and graph states are fed into task-specific layers to make word-level predictions and form the output.

Results on four widely used factuality prediction benchmarks (i.e., FactBank, Meantime, UW, and UDS-IH2) show that our framework provides better Event Anchor Detection and Factuality Induction performances compared with both the state-of-the-art EFP methods using biLSTM or GCN and the state-of-the-art graph encoding methods like GAT-LSTM and GRAN. We also conduct experiments and analysis on the overall score distribution and sentence length. The findings further show that our framework is able to capture the overall factuality score distributions like the gold ones. Additionally, thanks to the parameter-sharing mechanism, our framework can model long sentences better than the baselines. To our knowledge, we are the first to exploit end-to-end EFP and the interactions of the directional labeled information in neural syntactic event factuality prediction.

We make several significant contributions which can be summarized as follows:

- **An end-to-end EFP task setting.** We propose a practical task setting for Event Factuality Prediction, namely end-to-end EFP, by performing Event Anchor Detection and Factuality Induction jointly in a single model. To our knowledge, we are the first to study end-to-end EFP.
- **A graph neural framework.** We propose a novel graph neural multi-task framework DLGRN, leveraging directional and labeled syntactic information by extending the recurrent message passing process with edge-tied weights and edge-aware attention mechanism. The proposed techniques are also off-the-shelf and portable to other graph encoders. To our knowledge, we are the first to make use of directional labeled information for neural EFP.
- **Extensive experiments.** We achieve state-of-the-art results on FactBank, Meantime, UW, and UDS-IH2 for both Event Anchor Detection and Factuality Induction against the extended traditional EFP methods and the strong graph encoders. The results show the effectiveness and the generalizability of DLGRN.

2. Related work

EFP is a fundamental task in information extraction. Much work have been done for EFP, including rule-based approaches (Lotan, Stern, & Dagan, 2013; Nairn, Condoravdi, & Karttunen, 2006; Saurí, 2008), statistical approaches with manually designed features (Diab et al., 2009; Lee et al., 2015; de Marneffe, Manning, & Potts, 2012; Prabhakaran, Rambow, & Diab, 2010), hybrid approaches (Qian, Li, & Zhu, 2015; Saurí& Pustejovsky, 2012; Stanovsky, Eckle-Kohler, Puzikov, Dagan, & Gurevych, 2017) and deep learning methods integrating the semantic and syntactic information (Rudinger et al., 2018; Veyseh et al., 2019). Nairn et al. (2006) propose a deterministic algorithm based on associating certain clause-embedding verbs with implication signatures. Lotan et al. (2013) build a recursive rule-based system using implication signatures and other lexical- and dependency tree-based features. Diab et al. (2009) and Prabhakaran et al. (2010) use support vector machine (SVM) and conditional random field (CRF) over lexical and dependency features for predicting author belief commitments, which they treat as a sequence tagging problem. Lee et al. (2015) train an SVM on lexical and dependency path features. Sauríand Pustejovsky (2012) and Stanovsky et al. (2017) train support vector models over the outputs of rule-based systems.

Our work follows neural network methods, which is a new and emerging branch on event factuality. Qian, Li, Zhu, and Zhou (2019) employ Generative Adversarial Networks (GANs) for leveraging rich features. Rudinger et al. (2018) stack biLSTMs for sequential modeling and tree LSTM for combining dependency representations of the input sentences. Veyseh et al. (2019) propose a method to integrate syntactic and semantic structures of sentences using biLSTM to capture sequential context and stacking GCN for encoding the syntactic information. This line of work studies EFP while under the assumption that event anchor words are identified in advance. In contrast, we study end-to-end EFP in a multi-task framework (El-allaly, Sarrouti, Ennahnahi, & Alaoui, 2021; Zaporojets, Deleu, Develder, & Demeester, 2021) consisting of Event Anchor Detection and Factuality Induction, investigating directional labeled information for neural EFP.

For either traditional EFP or end-to-end EFP, the predictions can be performed in word-level, current state-of-the-art graph encoders including GCN (Chen et al., 2021; Pedronette & Latecki, 2021; Ragesh, Sellamanickam, Iyer, Bairi, & Lingam, 2021; Wei et al., 2019), GAT-LSTM (Tao et al., 2020; Wu et al., 2018) and GRAN (Liao et al., 2019) can also be applied to encode context for EFP. There is also a line of work on incorporating syntactic information with graph neural networks (Balali, Asadpour, Campos, & Jatowt, 2020; Zhang, He, & Zhang, 2021). However, compared with these graph encoders, our framework is additionally able to incorporate directional labeled information to recurrent graph message passing process.

3. Research objectives

As described in Section 1, this paper has two research objectives. The first is to propose a task setting of end-to-end Event Factuality Prediction. In contrast to the existing work on EFP (Lee et al., 2015; Rudinger et al., 2018; Veyseh et al., 2019), the practical input of an NLP application system is usually raw text data, and errors in automatically detecting event anchor words can propagate to that line of methods, leading to decreased performance. To this end, our end-to-end Event Factuality Prediction aims at performing Event Anchor Detection and Factuality Induction jointly on plain texts for predicting event factualities. Fig. 1(b) shows the comparison between traditional EFP and our end-to-end EFP. The input of traditional EFP is sentences and positions, and the output is factuality scores at given positions. On the contrary, end-to-end EFP takes plain text sentences as the input and outputs event anchor words and corresponding predicted factuality scores.

Additionally, the second research objective is to leverage directional and labeled syntactic information in graph modeling methods of Event Factuality Prediction. Despite the existing graph encoding methods (Liao et al., 2019; Veyseh et al., 2019; Wu et al., 2018) limits the use of syntactic knowledge and do not consider directional or labeled syntactic information, the sentence in

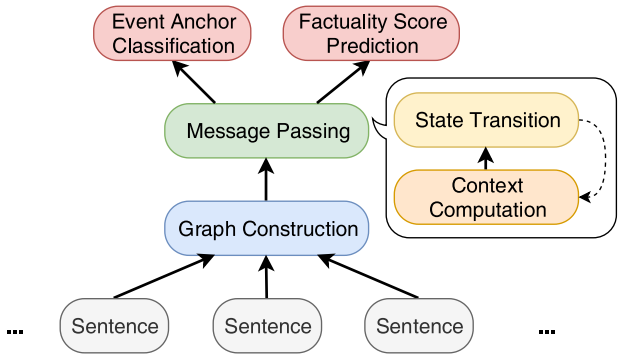


Fig. 2. Framework overview of DLGRN.

Fig. 1(a) strongly indicates the necessities of directional and labeled syntactic information as salient differentiating factors in this task setting. Therefore, we present a graph neural multi-task framework modeling the directional and labeled syntactic information, named Directional Labeled Graph Recurrent Network (DLGRN), for end-to-end EFP. DLGRN first constructs a message passing graph for the input plain sentence. And then DLGRN uses a graph-based message passing process to learn the shared word states and graph states by extending graph recurrent network (GRN) (Song et al., 2018; Yin et al., 2019; Zhang et al., 2018). To model the directional and labeled information, we design edge-tied weights and edge-aware attention mechanism in DLGRN. Finally, the shared states are used in a multi-task learning framework to jointly predicting event anchor words and event factualities.

To show the effectiveness and generalizability of the proposed DLGRN framework, we evaluate it on four benchmark datasets, i.e., FaceBank (Saurí& Pustejovsky, 2009), Meantime (Minard et al., 2016), UW (Lee et al., 2015) and UDS-IH2 (Rudinger et al., 2018; White, Rudinger, Rawlins, & Durme, 2018). Experimental results show that DLGRN achieves better performances than state-of-the-art systems in all subtasks of end-to-end EFP. Specifically, DLGRN decreases 17.12% MAE and raises 5.40% Pearson correlation r against the best baseline in end-to-end EFP averagely on all the datasets. We also conduct development experiments and ablation studies on our framework to show the influence of hyper-parameters and individual parts. Besides, we present an analysis on factuality scores' overall distributions, showing the generalizability of DLGRN to fit the biased long-tail scores. And we conduct experiments on the impact of sentence length to show that our method performs better than the baselines.

4. DLGRN framework

4.1. Problem definition

Before the work of Lee et al. (2015), the EFP task was formulated as a multi-label classification task. Recently, following the work of Rudinger et al. (2018) and Stanovsky et al. (2017), researchers focus on a regression formulation that aims to predict a floating number score in the range of $[-3, +3]$ to quantify the occurrence possibility of a given event mention. The floating score provides more meaningful information for the downstream tasks than the classification settings. Therefore, we follow the regression setting in factuality induction.

Formally, the input of end-to-end EFP is a plain text sentence $s = \{w_0, w_1, w_2, \ldots, w_n\}$ as a sequence of n words and the output is a list $\{(p_i, f_i)|p_i \in s, f_i \in [-3, +3]\}$ of event anchor word p_i along with their factuality score f_i. End-to-end EFP consists of two tasks, namely Event Anchor Detection and Factuality Induction. The former identifies whether a word w_i is an event anchor in the sentence, and the latter predicts a floating-point factuality score for an event anchor.

4.2. Overall framework

The overall framework of Directional Labeled Graph Recurrent Network (DLGRN) is shown in Fig. 2. It consists of three stages. First, we construct the message passing graph G_s for the sentence s, modeling each word as a node and initializing the representation $\boldsymbol{e}_i$ for each word w_i. Then, we compute the directional labeled context and update the word states and the graph state by iterating the graph propagation. At last, the final word states and the graph state are the input to task-specific layers, which make predictions for event anchor words and factuality scores.

4.3. Graph construction

We construct a message passing graph G_s with words as nodes for the sentence s, which guides how to propagate the information among words.

4.3.1. Edge initialization

Following recent work on modeling events (Liu, Luo, & Huang, 2018; Marcheggiani & Titov, 2017; Nguyen & Grishman, 2018; Veyseh et al., 2019), the dependency parse tree T_s is applied as the basic graph structure for each sentence s, keeping the directions and labels for modeling the syntactic meanings. We combine the directional edges in the dependency parse trees with sequential edges and self-loops to maintain the ability to model local n-grams. Formally, we define the adjacent matrix $\boldsymbol{a}$, which indicates the specific edge labels of the sentence s as follows:

$$\boldsymbol{a}_{ij} = \begin{cases} T_s(w_i, w_j), & \langle i \to j \rangle \in T_s \\ \texttt{[seq]}, & j \equiv i+1 \ \& \ \langle i \to j \rangle \notin T_s \\ \texttt{[self]}, & i \equiv j \end{cases} \tag{1}$$

where $T_s(w_i, w_j)$ represents the label of the edge $\langle i \to j \rangle$ in T_s and `[seq]` and `[self]` are two additional labels for denoting the sequential edges and self-loops, respectively.

4.3.2. Node representation

The input sentence s is first tokenized into a word piece sequence and then fed into a pre-trained BERT model (Devlin, Chang, Lee, & Toutanova, 2019). We select the contextualized embeddings $\boldsymbol{e} = \{\boldsymbol{e}_0, \boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_n\}$ of each word in the last Transformer encoder layer (Vaswani et al., 2017) for further computation. The first word piece embedding,[1] denoted as $\boldsymbol{e}_i$, is used for each word w_i.

4.4. Message passing

We adopt Graph Recurrent Network (GRN) (Song et al., 2018; Yin et al., 2019; Zhang et al., 2018) as our graph encoder. A GRN uses a set of hidden vectors to represent word states $\boldsymbol{h}_i$ and the graph state $\boldsymbol{g}$ while updating them using GRU cells (Cho et al., 2014) through recurrent steps, which natively encodes multi-hop paths by parameter sharing. It can be seen as a recurrent network counterpart to GCN (Kipf & Welling, 2017) and GAT (Velickovic et al., 2018) for graph representation.

We enhance the original GRN with directional labeled information by introducing edge-tied weights and edge-aware attention mechanism, dividing the message passing process of each recurrent step $t \in \{1, 2, \dots, T\}$ into two stages.

4.4.1. Context computation

The outgoing context $\overrightarrow{\boldsymbol{m}}_i^t$ and the incoming context $\overleftarrow{\boldsymbol{m}}_i^t$ of word w_i are both represented as the weighted sums of the previous word states $\boldsymbol{h}^{t-1}$ and the edge type embeddings $\boldsymbol{k}$

$$\overrightarrow{\boldsymbol{m}}_i^t = \sum_{j \in \{j | \boldsymbol{a}_{ij} \neq 0\}} \overrightarrow{\boldsymbol{w}}_{ij} [\boldsymbol{h}_j^{t-1}; \boldsymbol{k}_{\text{type}(i \to j)}] \tag{2}$$

$$\overleftarrow{\boldsymbol{m}}_i^t = \sum_{j \in \{j | \boldsymbol{a}_{ji} \neq 0\}} \overleftarrow{\boldsymbol{w}}_{ij} [\boldsymbol{h}_j^{t-1}; \boldsymbol{k}_{\text{type}(j \to i)}] \tag{3}$$

where $\boldsymbol{a}_{ij} \neq 0$ means that there is non-empty type of edge $\langle i \to j \rangle$ in the graph, $\overrightarrow{\boldsymbol{w}}_{ij}$ and $\overleftarrow{\boldsymbol{w}}_{ij}$ are the outgoing and incoming weights of the specific edge $\langle i \to j \rangle$ and $\langle j \to i \rangle$, respectively, $\text{type}(i \to j)$ is the type label of the edge $\langle i \to j \rangle$ and $[\ ;\ ]$ denotes the concatenation operation of vectors. Please note that the word states $\boldsymbol{h}^0$ and the graph state $\boldsymbol{g}^0$ are initialized as zeros, which will be explained later.

The weights $\overrightarrow{\boldsymbol{w}}_{ij}$ and $\overleftarrow{\boldsymbol{w}}_{ij}$ are calculated using the bilinear attention to incorporate edge-aware information as

$$\overrightarrow{\boldsymbol{w}}_{ij} = \underset{j \in \{j | \boldsymbol{a}_{ij} \neq 0\}}{\text{softmax}} \left(\frac{\boldsymbol{h}_i^{t-1\top} \boldsymbol{W}^a_{\text{type}(i \to j)} \boldsymbol{h}_j^{t-1}}{\sqrt{d_{\boldsymbol{h}}}} \right) \tag{4}$$

$$\overleftarrow{\boldsymbol{w}}_{ij} = \underset{j \in \{j | \boldsymbol{a}_{ji} \neq 0\}}{\text{softmax}} \left(\frac{\boldsymbol{h}_i^{t-1\top} \boldsymbol{W}^b_{\text{type}(j \to i)} \boldsymbol{h}_j^{t-1}}{\sqrt{d_{\boldsymbol{h}}}} \right) \tag{5}$$

where $d_{\boldsymbol{h}}$ is the dimension of the node states $\boldsymbol{h}$, $\top$ denotes the vector transposition and $\boldsymbol{W}^a$ and $\boldsymbol{W}^b$ are model parameters.

4.4.2. State transition

Following the approaches of Yin et al. (2019) and Zhang et al. (2018), a GRU cell is used to model the state transition process. We allow information exchange between a word and directly connected words in G_s.

The word state $\boldsymbol{h}_i^{t-1}$ is updated by aggregating the node representation, the directional labeled context, and the graph state while being controlled by the gates in the GRU cell to avoid gradient vanishing or explosion, as

$$\boldsymbol{\zeta}_i^t = [\boldsymbol{e}_i; \overrightarrow{\boldsymbol{m}}_i^t; \overleftarrow{\boldsymbol{m}}_i^t; \boldsymbol{g}^{t-1}] \tag{6}$$

[1] We found that the last one, max-pooling or average-pooling of the word piece embeddings for a word, was not any better or worse for evaluation. Kondratyuk and Straka (2019) report similar findings for dependency parsing.

$$\boldsymbol{r}_i^t = \text{sigmoid}(\boldsymbol{W}^r \boldsymbol{\zeta}_i^t + \boldsymbol{U}^r \boldsymbol{h}_i^{t-1}) \tag{7}$$

$$\boldsymbol{z}_i^t = \text{sigmoid}(\boldsymbol{W}^z \boldsymbol{\zeta}_i^t + \boldsymbol{U}^z \boldsymbol{h}_i^{t-1}) \tag{8}$$

$$\boldsymbol{\mu}_i^t = \tanh(\boldsymbol{W}^\mu \boldsymbol{\zeta}_i^t + \boldsymbol{U}^\mu (\boldsymbol{r}_i^t \odot \boldsymbol{h}_i^{t-1})) \tag{9}$$

$$\boldsymbol{h}_i^t = (1 - \boldsymbol{z}_i^t) \odot \boldsymbol{h}_i^{t-1} + \boldsymbol{z}_i^t \odot \boldsymbol{\mu}_i^t \tag{10}$$

where $\odot$ denotes the element-wise multiplication.

Similarly, the graph state $\boldsymbol{g}^{t-1}$ is updated by the average word states under the control of another GRU cell as

$$\boldsymbol{c}^t = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{h}_i^t \tag{11}$$

$$\bar{\boldsymbol{r}}^t = \text{sigmoid}(\boldsymbol{W}^{\bar{r}} \boldsymbol{c}^t + \boldsymbol{U}^{\bar{r}} \boldsymbol{g}^{t-1}) \tag{12}$$

$$\bar{\boldsymbol{z}}^t = \text{sigmoid}(\boldsymbol{W}^{\bar{z}} \boldsymbol{c}^t + \boldsymbol{U}^{\bar{z}} \boldsymbol{g}^{t-1}) \tag{13}$$

$$\bar{\boldsymbol{\mu}}^t = \tanh(\boldsymbol{W}^{\bar{\mu}} \boldsymbol{c}^t + \boldsymbol{U}^{\bar{\mu}} (\bar{\boldsymbol{r}}^t \odot \boldsymbol{g}^{t-1})) \tag{14}$$

$$\boldsymbol{g}^t = (1 - \bar{\boldsymbol{z}}^t) \odot \boldsymbol{g}^{t-1} + \bar{\boldsymbol{z}}^t \odot \bar{\boldsymbol{\mu}}^t \tag{15}$$

where $\boldsymbol{W}^x, \boldsymbol{U}^x (x \in \{r, z, \mu, \bar{r}, \bar{z}, \bar{\mu}\})$ are model parameters.

In this way, each word absorbs context information through its neighbors in a recurrent step t. Through multiple recurrent steps, word states achieve multi-hop feature integration. After T steps, we obtain the final word state $\boldsymbol{h}_i^T$ and the final graph state $\boldsymbol{g}^T$ for further usage. Before the first recurrent step, we initialize both the word states $\boldsymbol{h}^0$ and the graph state $\boldsymbol{g}^0$ as zero for two main reasons. First, in each recurrent step, the node representation $\boldsymbol{e}_i$ of each word w_i is fed into the GRU state transition, offering necessary word information. Second, initializations as zero will result in the weight $\overline{\boldsymbol{w}}_{ij}$ and $\overline{\boldsymbol{w}}_{ij}$ to be equivalent for the outgoing and incoming edges in the first step, respectively, offering balanced features for each edge.

4.5. Multi-task prediction

Event Anchor Detection is treated as a word-level binary classification task, and Factuality Induction is also treated as a word-level regression task. The final word state $\boldsymbol{h}_i$ and the final graph state $\boldsymbol{g}$ are first used to compute a shared word-level representation $\boldsymbol{q}_i$. Then $\boldsymbol{q}_i$ are fed to task-specific layers to make word-level predictions for both Event Anchor Detection and Factuality Induction.

$$\boldsymbol{o}_{ij} = \underset{j=1}{\overset{n}{\text{softmax}}}\left(\frac{\boldsymbol{h}_i^\top \boldsymbol{h}_j}{\sqrt{d_{\boldsymbol{h}}}}\right) \tag{16}$$

$$\boldsymbol{q}_i = [\sum_{j=1}^{n} \boldsymbol{o}_{ij} \boldsymbol{h}_j; \boldsymbol{g}] \tag{17}$$

The task-specific layers are two-layer fully-connected networks. For Event Anchor Detection, the word-level output $\hat{\boldsymbol{y}}_i^d$ is a two-dimensional vector for the binary classification. Additionally, for Factuality Induction, the output $\hat{y}_i^r$ is only a scalar.

4.6. Training

For Event Anchor Detection, the binary cross-entropy is used as the loss function for all the words. For Factuality Induction, we follow previous work (Lee et al., 2015; Rudinger et al., 2018; Stanovsky et al., 2017; Veyseh et al., 2019) to calculate the smooth L1, i.e. Huber loss with $\delta = 1$ for words with labeled factuality scores. The final loss function is a weighted sum of the individual losses for the two tasks as

$$\mathcal{L} = \lambda^d \mathcal{L}^d + \lambda^r \mathcal{L}^r \tag{18}$$

where the task weights λ^d and λ^r are hyper-parameters reflecting the importances of the two tasks.

We follow Devlin et al. (2019), using Adam (Kingma & Ba, 2015) with learning rate $2e{-}5$ for the BERT part and $1e{-}3$ for the other part, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L_2 weight decay of $1e{-}4$, learning rate warmup over the first 10% steps, and linear decay of the learning rate. Additionally, we preprocess the labeled factuality scores by linearly rescaling them to the range of $[0, 1]$ during training and do the reversed transformation with clipping to $[-3, +3]$ before output.

5. Experiments

5.1. Datasets

Following previous work (Rudinger et al., 2018; Stanovsky et al., 2017; Veyseh et al., 2019), we conduct experiments on four benchmark datasets with public access: **FaceBank** (Sauríi& Pustejovsky, 2009), **Meantime** (Minard et al., 2016), **UW** (Lee et al., 2015) and **UDS-IH2** (Rudinger et al., 2018; White et al., 2018).

Table 1
Statistics of the datasets. The first four columns represent the number of event anchor words and the corresponding proportion in that data split. The last column stands for the average sentence lengths.

Dataset	Train	Dev	Test	Total	Avg. len.
FactBank	6636 (12.87%)	2462 (12.64%)	663 (13.03%)	9761 (12.82%)	23.95
Meantime	967 (14.15%)	210 (13.51%)	218 (16.09%)	1395 (14.32%)	19.07
UW	9422 (12.66%)	3358 (12.63%)	864 (12.08%)	13644 (12.62%)	25.08
UDS-IH2	22108 (10.81%)	2642 (10.51%)	2539 (10.12%)	27289 (10.71%)	15.33

5.1.1. FactBank

FactBank (Saurí& Pustejovsky, 2009) is built on top of TimeBank (Pustejovsky et al., 2003). The factuality annotations of event mentions are discrete and categorized into four classes: Factual (CT+/−), Probable (PR+/−), Possible (PS+/−), and Unknown (Uu/CTu). The annotations achieved a relatively high inter-annotator agreement (IAA), $\kappa = 0.81$, a positive result when considered against similar annotation efforts.

5.1.2. Meantime

Meantime (Minard et al., 2016), the NewsReader MEANTIME corpus, consists of 120 English news articles and their translations in Spanish, Italian, and Dutch. The factuality annotations are also discrete, such as Fact, Counterfact, Possibility (uncertain), and Possibility (future).

5.1.3. UW

UW (Lee et al., 2015) is built for event detection and factuality, reusing sentences from the TempEval-3 corpus (UzZaman et al., 2013). The factuality annotations are continuous in $[-3, +3]$ and completed by crowdsourcing. The IAA is also strong, achieving 92.6% F1 for detection and 83.1% correlation for factuality.

5.1.4. UDS-IH2

UDS-IH2 (Rudinger et al., 2018; White et al., 2018) covers all predicates in English Universal Dependencies (EUD) v1.2 treebank.[2] Thirty-two unique crowdsource annotators through Amazon's Mechanical Turk are recruited to annotate whether the events have happened and the confidence scores in $[0, 4]$. The raw IAA for event factuality annotations is 0.84.

5.1.5. Preprocessing

The instance statistics of the four datasets are shown in Table 1. The four datasets provide word-level event mention and factuality score annotations in sentences. For first three datasets, we follow Stanovsky et al. (2017) and transform the annotated factuality scores to $[-3, +3]$. The python package `SpaCy`[3] is used to produce the labels of dependency parse trees for the sentences in FactBank, Meantime and UW. For the UDS-IH2 dataset, we follow Rudinger et al. (2018) to transform the factuality scores of multiple annotators to $[-3, +3]$, and align the sentences with the gold dependency parse labels in EUD v1.2.

5.2. Implementation details

Most of the common hyper-parameters are determined by grid search on the development experiments on each dataset according to the Pearson correlation (r) of Factuality Induction. For UW, FactBank, and UDS-IH2, we train models with at most eight epochs with 16 sentences in a mini-batch with a 0.1 dropout rate and a L_2 weight decay of $1e{-}4$, we use $\text{BERT}_{\text{BASE}}$ for all the tasks, 768 hidden units for word state, 300 hidden units for graph state and 50 hidden units for edge type embeddings. We use 300 hidden units for the task-specific layer in Event Anchor Detection and 600 hidden units in Factuality Induction. The maximum recurrent step number is 4. For Meantime, we use batch size 4 and L_2 weight decay $6e{-}3$.

5.3. Comparative methods

We extend the following three categories of baseline methods for traditional EFP into multi-task setting and compare DLGRN with them.

The first category is:

- **BERT**: A word-level model directly making predictions using BERT embeddings.

The second category includes two extended state-of-the-art methods:

- **H-biLSTM**: A hybrid model concatenating the hidden states of two-layer linear chain biLSTM and two-layer tree biLSTM (Rudinger et al., 2018) with BERT.

[2] https://universaldependencies.org/.
[3] https://spacy.io/.

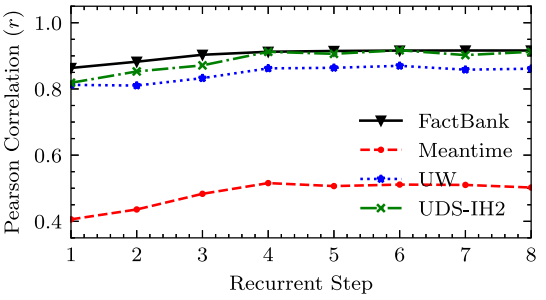


Fig. 3. Performances of Factuality Induction with different maximum recurrent step numbers T.

Table 2
Results of Event Anchor Detection. The underline marker denotes $p < 0.01$ on a paired t-test of F_1 values against DLGRN.

Method	FactBank			Meantime			UW			UDS-IH2			#params
	Acc	F_1	MCC	Acc	F_1	MCC	Acc	F_1	MCC	Acc	F_1	MCC	
BERT	0.90	0.87	0.611	0.88	0.89	0.662	0.89	0.86	0.702	0.89	0.87	0.630	109.5M
H-biLSTM	0.90	0.88	0.637	0.88	0.89	0.680	0.89	0.86	0.727	0.90	0.87	0.692	133.1M
GCN	0.91	0.90	0.778	0.94	0.91	0.734	0.91	0.88	0.819	0.92	0.89	0.736	125.9M
GAT-LSTM	0.91	0.90	0.772	0.93	0.91	0.791	0.90	0.89	0.799	0.93	0.89	0.768	120.8M
GRAN	0.92	0.90	0.769	0.93	0.90	0.713	0.90	0.89	0.812	0.93	0.88	0.722	120.6M
DLGRN	**0.93**	**0.92**	**0.874**	**0.95**	**0.93**	**0.895**	**0.92**	**0.91**	**0.883**	**0.95**	**0.91**	**0.869**	119.7M

- **GCN**: A three-layer GCN model integrating semantic and syntactic weights to compute *undirectional* context (Veyseh et al., 2019) on top of BERT.

The third category includes two state-of-the-art graph encoders with BERT:

- **GAT-LSTM**: A graph encoder extending the LSTM with graph attention structure in the input-to-state and state-to-state transitions (Wu et al., 2018).
- **GRAN**: A graph network incorporating undirectional edge information with unnormalized gates (Liao et al., 2019).

Please note that the first two categories of baselines are designed for traditional EFP, which are fed with ground truth positions of event anchor words. To ingratiate with end-to-end EFP, we slightly modify the baseline frameworks and extend them by adding a binary classification layer parallel to the existing regression layer under the multitask setting. As for the third category, both a binary classification layer and a regression layer are added after the baseline graph encoders for end-to-end EFP.

5.4. Evaluation metrics

For Event Anchor Detection, we use the Accuracy (Acc), F_1 score and Matthews Correlation Coefficient (MCC) (Matthews, 1975) as the evaluation metrics. MCC is applied because it avoids label bias due to data skew as reported in Table 1. For Factuality Induction, following Lee et al. (2015), Rudinger et al. (2018), Stanovsky et al. (2017) and Veyseh et al. (2019), we use the mean absolute error (MAE) and Pearson correlation (r) as the evaluation metrics.[4] An event anchor word is correctly identified if its position matches the ground truth. We only take into account the factuality scores of those correctly identified event anchor words.

6. Results and discussion

6.1. Development experiments

The maximum recurrent step number T is an important hyper-parameter in our framework. We analyze the influence of T on the performance of Factuality Induction according to the Pearson correlation (r) on the development sets. Fig. 3 shows the results.

We observe that different datasets have different optimal T values, where r reaches the highest point. In Meantime and UDS-IH2, there are significant improvements when T increases from 1 to 4, showing the effectiveness of our framework. The result decreases when T exceeds 4. As for FactBank and UW, when increasing T from 1 to 6, there are constant increments of r. However, the increase of T from 6 to 8 does not lead to further improvements. The F_1 scores of Event Anchor Detection with different T show similar trends. Therefore, considering the results and the running time, we choose $T = 4$ for all subsequent experiments.

[4] MAE measures the total error, while Pearson correlation measures the linear correlation between gold and predicted scores, which is more likely to reflect the dataset variance.

Table 3
Results of Factuality Induction. The underline marker denotes $p < 0.01$ on a paired t-test of r values against DLGRN.

Method	FactBank		Meantime		UW		UDS-IH2		#params
	MAE	r	MAE	r	MAE	r	MAE	r	
BERT	0.392	0.782	0.401	0.387	0.492	0.733	0.902	0.782	109.5M
H-biLSTM	0.381	0.850	0.389	0.394	0.475	0.752	0.895	0.804	133.1M
GCN	0.315	0.890	0.350	0.452	0.451	0.828	0.730	0.905	125.9M
GAT-LSTM	0.327	0.862	0.366	0.431	0.456	0.812	0.772	0.835	120.8M
GRAN	0.324	0.874	0.362	0.422	0.468	0.796	0.762	0.827	120.6M
DLGRN	**0.240**	**0.912**	**0.279**	**0.516**	**0.346**	**0.863**	**0.722**	**0.912**	119.7M

Table 4
Ablation study on UW.

Method	Acc	$\Delta_{\text{Acc}}\%$	F_1	$\Delta_{F_1}\%$	MCC	$\Delta_{\text{MCC}}\%$	MAE	$\Delta_{\text{MAE}}\%$	r	$\Delta_r\%$
DLGRN	**0.92**	–	**0.91**	–	**0.883**	–	**0.346**	–	**0.863**	–
- Multi-task	0.91	1.09	0.90	1.10	0.824	6.68	0.352	1.73	0.844	2.20
- Sequential	0.91	1.09	0.90	1.10	0.805	8.83	0.408	17.92	0.801	7.18
- Directional	0.90	2.17	0.90	1.10	0.772	12.57	0.415	19.94	0.796	7.76
- Labeled	0.90	2.17	0.89	2.20	0.745	15.63	0.422	21.97	0.788	8.69

6.2. Main results

Tables 2 and 3 show the overall results for Event Anchor Detection and Factuality Induction. The underline markers denote that the p-values of paired t-test against DLGRN is smaller than 0.01. We achieve the best performances in all the datasets for both Event Anchor Detection and Factuality Induction. By comparing DLGRN with BERT, we find a significant performance gain, showing the importance of the message passing process with graphs. By comparing DLGRN with H-biLSTM and GCN, we can find that modeling directional context is more powerful than encoding undirectional structures. This coincides with finds of Song et al. (2018) for AMR. Additionally, iterating over recurrent steps helps to capture longer-hop path information. By comparing DLGRN with GAT-LSTM and GRAN, we can see that incorporating directional labeled information helps to encode structural context for end-to-end EFP. Finally, the parameter number of DLGRN is smaller than all the baselines, which shows fewer risks of overfitting. There is also an exciting finding that our multi-task reimplementation of the baselines performs better than the origin factuality results reported in Rudinger et al. (2018) and Veyseh et al. (2019) with gold event anchor words, which shows the effectiveness of multi-task setting and shared representation learning.

6.3. Ablation study

To investigate the impacts of submodules of DLGRN for end-to-end event factuality prediction, we consider four main aspects in ablation study: sequential context modeling, directional context modeling, labeled information, and multi-task framework. For simplicity, we report the Event Anchor Detection and Factuality Induction results of UW in Table 4. The most important feature is the labeled information in the edge-tied weights and edge-aware attention mechanism, which brings a 2.20% decline in F_1, 15.63% decline in MCC, 21.97% decline in MAE and 8.69% raise in r. The construction of sequential edges and directional edges is also essential and offers 8.83% and 12.57% reduction in MCC, 17.92% and 19.94% reduction in MAE and 7.18% and 7.76% lift in r, respectively. The reason is that it offers the ability to model fine-grained interactions between words and capturing longer-hop information. Additionally, we removed the multi-task learning framework, degenerating DLGRN to a pipeline framework, which will first find possible positions of event mention anchor words and then predict factuality scores in traditional position-based form. The performances of DLGRN with the pipeline framework not only decrease 1.09% Acc, 1.10% F_1, 6.68% MCC and 2.2% r but also increase 1.7% MAE. We think the shared representations for Event Anchor Detection and Factuality Induction is more effective and thus beneficial to each subtasks.

6.4. Analysis on overall score distribution

We analyze the differences between the overall distributions of gold factuality scores and predicted ones. Fig. 4 shows the distributions of both gold and predicted factuality scores on the test sets of all four datasets. As we can see from the figure, in both UDS-IH2 and UW, DLGRN can fit the score distribution well, which also explains the high Pearson correlation r in Table 3. However, due to the limited size of FactBank and Meantime compared with UW and UDS-IH2, the distributions of factuality scores seem highly biased, which obstruct DLGRN from learning the diversity of FactBank and Meantime. In addition, we can see that the ranges of the predicted scores are narrower than the gold, indicating that there is a challenge to produce the scores near the two ends. This finding can also support the results of higher MAE on UDS-IH2, which may be a potential direction in future studies. There is also an exciting finding that the DLGRN learns the long-tail factuality scores in $[-3, 0]$ on UW well, which also shows the effectiveness of our framework.

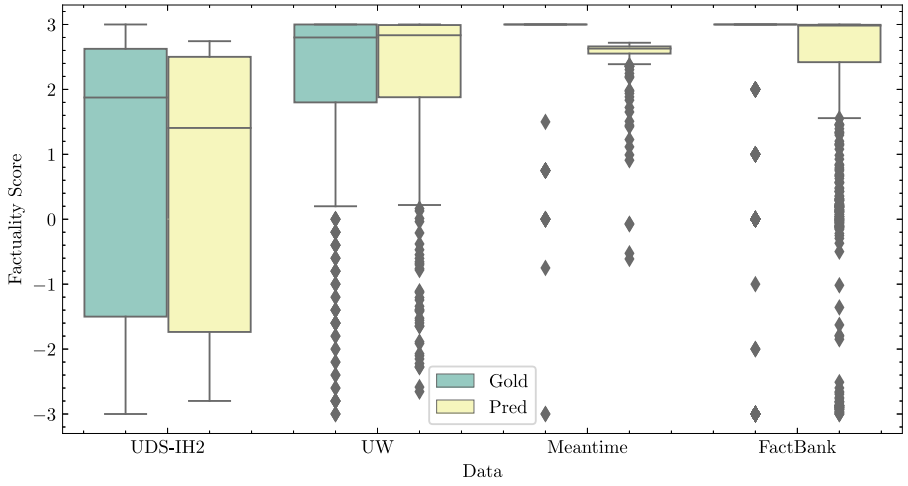


Fig. 4. The distributions of gold and predicted factuality scores on all datasets.

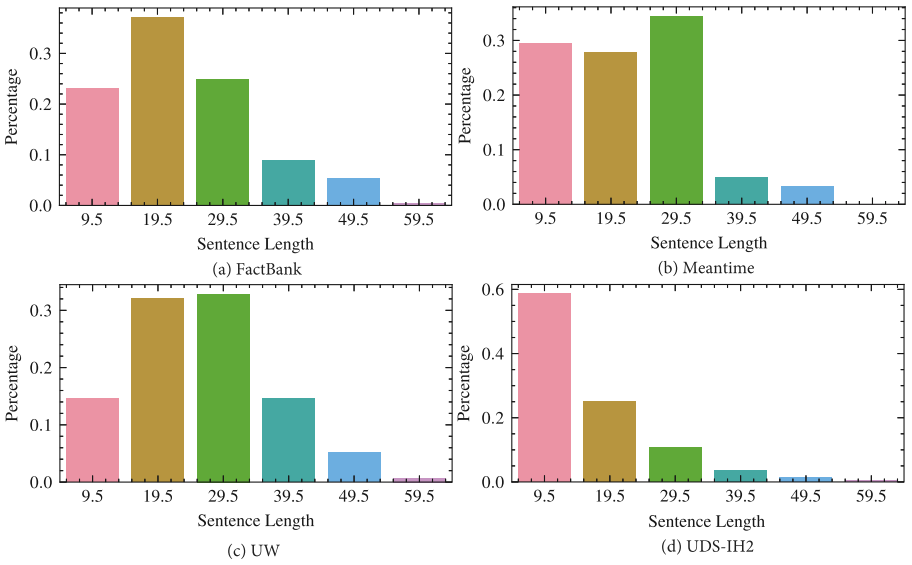


Fig. 5. The distributions of the sentence lengths of the datasets in the six disjoint buckets. The datasets are (a) FactBank, (b) Meantime, (c) UW and (d) UDS-IH2.

6.5. *The influence of sentence length*

We find that the optimal maximum recurrent step T value may be affected by the sentence length. Therefore, we analyze the impacts in different sentence lengths on the test sets of the four datasets, i.e., FactBank, Meantime, UW, and UDS-IH2. First, we divide the range of sentence lengths into six disjoint buckets with equal widths, showing the counting percentages in Fig. 5. By also referring to the average sentence lengths reported in Table 1, we can see that the sentence length distributions of FactBank and UW are the closest, so as their average lengths. The first three buckets of Meantime seem to contain homogeneous quantities, while UDS-IH2 has the shortest sentence length.

We then compare the performances of DLGRN with GCN, which is extended from a state-of-the-art traditional EFP baseline, and GAT-LSTM, which is developed from a strong graph encoder, in each bucket of all the datasets. The Pearson correlation r values are shown in Fig. 6. As we can see from the results, DLGRN outperforms GCN and GAT-LSTM in each bucket, showing the effectiveness of our proposed framework. It is worth noticing that the gap of Pearson correlation r between DLGRN and other baselines in the last two buckets is more significant than in the rest buckets, suggesting that DLGRN can work better in longer sentences thanks to the possible more giant maximum recurrent step T.

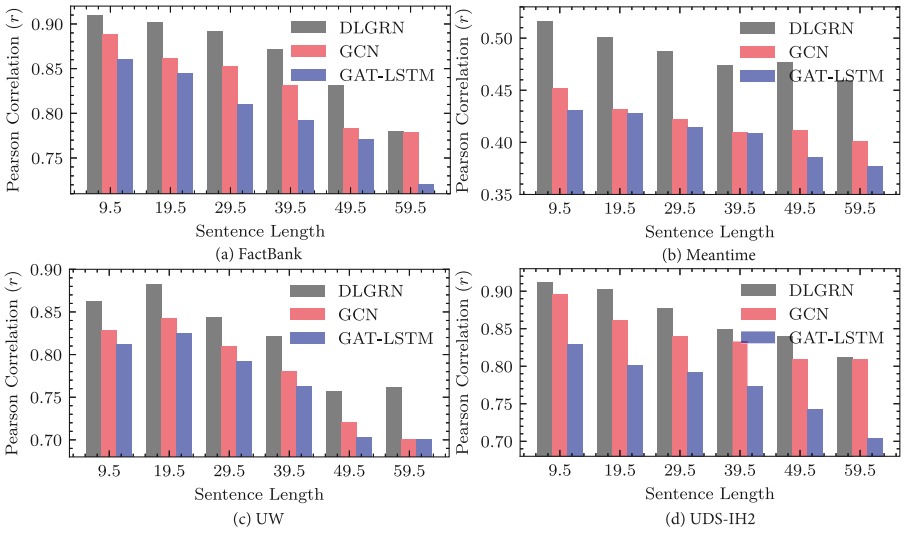
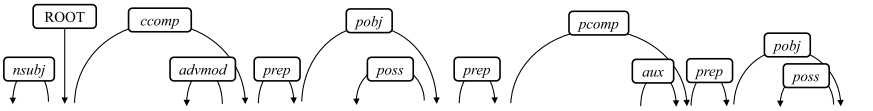


Fig. 6. The Pearson correlation r in the disjoint buckets of the four datasets, (a) FactBank, (b) Meantime, (c) UW and (d) UDS-IH2.

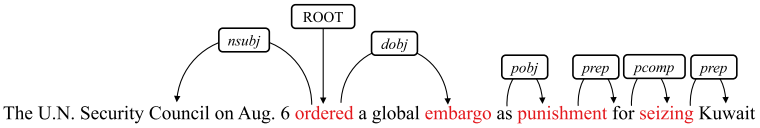
(a)



Mr. Alito said his office " just responded to an attorney 's question about whether we would go after attorney 's fees "

#Example 1	Mr. Alito said his office " just responded to an attorney 's question about whether we would go after attorney 's fees "
Gold Factuality	said@2=3.0, responded@7=3.0, question@12=3.0, go@17=0.0
GCN	said@2=2.99, responded@7=2.99, question@12=2.99, go@17=-1.02, fee@21=0.53
GAT-LSTM	said@2=2.98, responded@7=2.97, go@17=1.56
DLGRN	said@2=2.99, responded@7=2.99, question@12=2.99, go@17=-0.16

(b)



The U.N. Security Council on Aug. 6 ordered a global embargo as punishment for seizing Kuwait

#Example 2	The U.N. Security Council on Aug. 6 ordered a global embargo as punishment for seizing Kuwait
Gold Factuality	ordered@7=3.0, embargo@10=0.0, punishment@12=0.0, seizing@14=3.0
GCN	ordered@7=2.99, embargo@10=2.98, punishment@12=-0.84, seizing@14=2.67
GAT-LSTM	ordered@7=2.99, punishment@12=1.24, seizing@14=2.15
DLGRN	ordered@7=2.99, embargo@10=1.32, punishment@12=0.66, seizing@14=2.72

Fig. 7. Predicted and gold factualities of two example sentences along with involved directional and labeled syntactic information. The red words are gold event anchor words, and the blue words are false positive. The format of each factuality is `TOKEN@POS=Value`. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.6. Implications for research

Our research introduces the task setting of end-to-end EFP, where Event Anchor Detection and Factuality Induction should perform jointly on plain texts for predicting factualities of events. We also introduce the method of incorporating directional and labeled syntactic information with graph neural networks. The experiment results demonstrated that our proposed framework exposes excellent potential to effectively detecting event anchors and inducing event factualities by achieving state-of-the-art results

on four benchmark datasets, i.e., FactBank, Meantime, UW, and UDS-IH2, comparing with solid baselines. Our findings also showed that our framework could capture the overall factuality score distributions like the gold ones. In addition, findings showed that our framework could model longer dependencies better than the baselines. Because the methods of leveraging directional and labeled syntactic information are portable, we believe that it will also significantly improve the performance of other natural language processing tasks like natural language inference, named entity recognition, and relation extraction.

6.7. Case study

Fig. 7 shows two examples from FactBank. We compare the gold factuality results with the output of DLGRN and two strong baselines, GCN and GAT-LSTM. In example 1, the baseline GCN improperly predicts a false positive event anchor word "*fee*" with the index 21. In contrast, the baseline GAT-LSTM leaves out a positive event anchor word "*question*", which is indexed at 12. Additionally, DLGRN predicts more accurate factuality scores than GCN and GAT-LSTM, which shows the effectiveness of our graph neural multi-task framework. In example 2, only GAT-LSTM omits a positive event anchor word "*embargo*" indexed at 10, while other models predict the correct event anchor words. DLGRN also predicts factuality scores closer to the gold scores, especially for the event anchor word "*punishment*". One possible reason is that DLGRN considers directional and labeled syntactic information, which is ignored in GCN and GAT-LSTM.

7. Conclusion

We investigated end-to-end Event Factuality Prediction, a more realistic task setting in practice, performing Event Anchor Detection and Factuality Induction jointly in a single model. A novel graph neural multi-task framework with the directional labeled information, DLGRN, was investigated in this task. This framework identifies event anchor words and factuality scores by extending the recurrent message passing process with edge-tied weights and edge-aware attention mechanism. Results on four benchmark datasets, i.e., FactBank, Meantime, UW, and UDS-IH2, show that our framework gives better performances on Event Anchor Detection and Factuality Induction than the extended traditional state-of-the-art EFP methods and the graph encoders. We also find that our framework is capable of capturing the overall score distributions. In addition, incorporating directional and labeled syntactic information in EFP performs better than the baselines for long sentences. To our knowledge, we are the first to study end-to-end EFP and make use of directional labeled information for neural EFP, which is also portable to other graph encoders.

CRediT authorship contribution statement

Xiao Liu: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Heyan Huang:** Supervision, Funding acquisition, Project administration. **Yue Zhang:** Supervision, Investigation, Writing – review & editing, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful and constructive comments. This work was supported by the Joint Funds of the National Natural Science Foundation of China (Grant No. U19B2020), the Funds of the Integrated Application Software Project, China, and the Funds from *Rxhui Inc.*, China (https://rxhui.com).

References

Balali, A., Asadpour, M., Campos, R., & Jatowt, A. (2020). Joint event extraction along shortest dependency paths using graph convolutional networks. *Knowledge-Based Systems*, *210*, Article 106492. http://dx.doi.org/10.1016/j.knosys.2020.106492.

Chen, C., Cai, F., Hu, X., Zheng, J., Ling, Y., & Chen, H. (2021). An entity-graph based reasoning method for fact verification. *Information Processing & Management*, *58*, Article 102472. http://dx.doi.org/10.1016/j.ipm.2020.102472.

Chen, M., Wei, Z., Huang, Z., Ding, B., & Li, Y. (2020). Simple and deep graph convolutional networks. In *Proceedings of Machine Learning Research*: *vol. 119*, *Proceedings of the 37th international conference on machine learning* (pp. 1725–1735). URL: http://proceedings.mlr.press/v119/chen20v.html.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1724–1734). http://dx.doi.org/10.3115/v1/d14-1179.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 4171–4186). http://dx.doi.org/10.18653/v1/n19-1423.

Diab, M. T., Levin, L. S., Mitamura, T., Rambow, O., Prabhakaran, V., & Guo, W. (2009). Committed belief annotation and tagging. In *Proceedings of the third linguistic annotation workshop* (pp. 68–73). URL: https://www.aclweb.org/anthology/W09-3012/.

El-allaly, E., Sarrouti, M., Ennahnahi, N., & Alaoui, S. O. E. (2021). MTTLADE: A multi-task transfer learning-based method for adverse drug events extraction. *Information Processing & Management*, *58*, Article 102473. http://dx.doi.org/10.1016/j.ipm.2020.102473.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd international conference on learning representations*. URL: http://arxiv.org/abs/1412.6980.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th international conference on learning representations*. URL: https://openreview.net/forum?id=SJU4ayYgl.

Kondratyuk, D., & Straka, M. (2019). 75 Languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 2779–2795). http://dx.doi.org/10.18653/v1/D19-1279.

Lee, K., Artzi, Y., Choi, Y., & Zettlemoyer, L. (2015). Event detection and factuality assessment with non-expert supervision. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1643–1648). http://dx.doi.org/10.18653/v1/d15-1189.

Liao, R., Li, Y., Song, Y., Wang, S., Hamilton, W. L., Duvenaud, D., et al. (2019). Efficient graph generation with graph recurrent attention networks. In *Proceedings of the advances in neural information processing systems 32: Annual conference on neural information processing systems 2019* (pp. 4257–4267). URL: https://proceedings.neurips.cc/paper/2019/hash/d0921d442ee91b896ad95059d13df618-Abstract.html.

Liu, X., Luo, Z., & Huang, H. (2018). Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1247–1256). http://dx.doi.org/10.18653/v1/d18-1156.

Lotan, A., Stern, A., & Dagan, I. (2013). Truthteller: Annotating predicate truth. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 752–757). URL: https://www.aclweb.org/anthology/N13-1091/.

Marcheggiani, D., & Titov, I. (2017). Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1506–1515). http://dx.doi.org/10.18653/v1/d17-1159.

de Marneffe, M., Manning, C. D., & Potts, C. (2012). Did it happen? the pragmatic complexity of veridicality assessment. *Computational Linguistics*, *38*, 301–333. http://dx.doi.org/10.1162/COLI_a_00097.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta*, *405*, 442–451. http://dx.doi.org/10.1016/0005-2795(75)90109-9.

Minard, A., Speranza, M., Urizar, R., Altuna, B., van Erp, M., Schoen, A., et al. (2016). Meantime, the newsreader multilingual event and time corpus. In *Proceedings of the tenth international conference on language resources and evaluation* (pp. 4417–4422). URL: http://www.lrec-conf.org/proceedings/lrec2016/summaries/488.html.

Nairn, R., Condoravdi, C., & Karttunen, L. (2006). Computing relative polarity for textual inference. In *Proceedings of the 5th international workshop on inference in computational semantics*. URL: https://www.aclweb.org/anthology/W06-3907.

Nguyen, T. H., & Grishman, R. (2018). Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the thirty-second AAAI conference on artificial intelligence* (pp. 5900–5907). URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16329.

Pedronette, D. C. G., & Latecki, L. J. (2021). Rank-based self-training for graph convolutional networks. *Information Processing & Management*, *58*, Article 102443. http://dx.doi.org/10.1016/j.ipm.2020.102443.

Prabhakaran, V., Rambow, O., & Diab, M. T. (2010). Automatic committed belief tagging. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 1014–1022). URL: https://www.aclweb.org/anthology/C10-2117/.

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., et al. (2003). The timebank corpus. In *Corpus linguistics, Vol. 2003* (p. 40). URL: http://ucrel.lancs.ac.uk/publications/cl2003/papers/pustejovsky.pdf.

Qian, Z., Li, P., & Zhu, Q. (2015). A two-step approach for event factuality identification. In *Proceedings of the 2015 international conference on asian language processing* (pp. 103–106). http://dx.doi.org/10.1109/IALP.2015.7451542.

Qian, Z., Li, P., Zhu, Q., & Zhou, G. (2019). Document-level event factuality identification via adversarial neural network. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 2799–2809). http://dx.doi.org/10.18653/v1/n19-1287.

Ragesh, R., Sellamanickam, S., Iyer, A., Bairi, R., & Lingam, V. (2021). Hetegcn: Heterogeneous graph convolutional networks for text classification. In *Proceedings of the 14th ACM international conference on web search and data mining* (pp. 860–868). http://dx.doi.org/10.1145/3437963.3441746.

Rudinger, R., White, A. S., & Durme, B. V. (2018). Neural models of factuality. In *Proceedings of the 2018 conference of the North American Chapter of the association for computational linguistics: Human language technologies* (pp. 731–744). http://dx.doi.org/10.18653/v1/n18-1067.

Saurí, R. (2008). *A factuality profiler for eventualities in text*. Brandeis University, http://dx.doi.org/10.5555/1415229.

Saurí, R., & Pustejovsky, J. (2009). Factbank: a corpus annotated with event factuality. *Language Resources and Evaluation*, *43*, 227–268. http://dx.doi.org/10.1007/s10579-009-9089-9.

Saurí, R., & Pustejovsky, J. (2012). Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, *38*, 261–299. http://dx.doi.org/10.1162/COLI_a_00096.

Song, L., Zhang, Y., Wang, Z., & Gildea, D. (2018). A graph-to-sequence model for amr-to-text generation. In *Proceedings of the 56th annual meeting of the association for computational linguistics* (pp. 1616–1626). http://dx.doi.org/10.18653/v1/P18-1150.

Stanovsky, G., Eckle-Kohler, J., Puzikov, Y., Dagan, I., & Gurevych, I. (2017). Integrating deep linguistic features in factuality prediction over unified datasets. In *Proceedings of the 55th annual meeting of the association for computational linguistics* (pp. 352–357). http://dx.doi.org/10.18653/v1/P17-2056.

Tao, Z., Wei, Y., Wang, X., He, X., Huang, X., & Chua, T. (2020). Mgat: multimodal graph attention network for recommendation. *Information Processing & Management*, *57*, Article 102277. http://dx.doi.org/10.1016/j.ipm.2020.102277.

UzZaman, N., Llorens, H., Derczynski, L., Allen, J. F., Verhagen, M., & Pustejovsky, J. (2013). Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of the 7th international workshop on semantic evaluation, SemEval@NAACL-HLT 2013* (pp. 1–9). URL: https://www.aclweb.org/anthology/S13-2001/.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Proceedings of the advances in neural information processing systems 30: Annual conference on neural information processing systems 2017* (pp. 5998–6008). URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *Proceedings of the 6th international conference on learning representations*. URL: https://openreview.net/forum?id=rJXMpikCZ.

Veyseh, A. P. B., Nguyen, T. H., & Dou, D. (2019). Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *Proceedings of the 57th conference of the association for computational linguistics* (pp. 4393–4399). http://dx.doi.org/10.18653/v1/p19-1432.

Wei, Y., Wang, X., Nie, L., He, X., Hong, R., & Chua, T. (2019). MMGCN: multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia* (pp. 1437–1445). http://dx.doi.org/10.1145/3343031.3351034.

White, A. S., Rudinger, R., Rawlins, K., & Durme, B. V. (2018). Lexicosyntactic inference in neural models. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4717–4724). http://dx.doi.org/10.18653/v1/d18-1501.

Wu, T., Chen, F., & Wan, Y. (2018). Graph attention lstm network: A new model for traffic flow forecasting. In *Proceedings of the 5th international conference on information science and control engineering* (pp. 241–245). http://dx.doi.org/10.1109/ICISCE.2018.00058.

Yang, Y., Qiu, J., Song, M., Tao, D., & Wang, X. (2020). Distilling knowledge from graph convolutional networks. In *Proceedings of the 2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 7072–7081). http://dx.doi.org/10.1109/CVPR42600.2020.00710.

Yin, Y., Song, L., Su, J., Zeng, J., Zhou, C., & Luo, J. (2019). Graph-based neural sentence ordering. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 5387–5393). http://dx.doi.org/10.24963/ijcai.2019/748.

Yu, X., Wang, S., & Zhang, Y. (2021). Cgnet: A graph-knowledge embedded convolutional neural network for detection of pneumonia. *Information Processing & Management*, *58*, Article 102411. http://dx.doi.org/10.1016/j.ipm.2020.102411.

Zaporojets, K., Deleu, J., Develder, C., & Demeester, T. (2021). DWIE: an entity-centric dataset for multi-task document-level information extraction. *Information Processing & Management*, *58*, Article 102563. http://dx.doi.org/10.1016/j.ipm.2021.102563.

Zhang, J., He, Q., & Zhang, Y. (2021). Syntax grounded graph convolutional network for joint entity and event extraction. *Neurocomputing*, *422*, 118–128. http://dx.doi.org/10.1016/j.neucom.2020.09.044.

Zhang, Y., Liu, Q., & Song, L. (2018). Sentence-state LSTM for text representation. In *Proceedings of the 56th annual meeting of the association for computational linguistics* (pp. 317–327). http://dx.doi.org/10.18653/v1/P18-1030.