Does It Happen? Multi-hop Path Structures for Event Factuality Prediction with Graph Transformer Networks

Duong Minh Le[1] **and Thien Huu Nguyen**[2]
[1] VinAI Research, Vietnam
[2] Dept. of Computer and Information Science, University of Oregon, Eugene, OR, USA
v.duonglm1@vinai.io, thien@cs.uoregon.edu

Abstract

The goal of Event Factuality Prediction (EFP) is to determine the factual degree of an event mention, representing how likely the event mention has happened in text. Current deep learning models has demonstrated the importance of syntactic and semantic structures of the sentences to identify important context words for EFP. However, the major problem with these EFP models is that they only encode the one-hop paths between the words (i.e., the direct connections) to form the sentence structures. In this work, we show that the multi-hop paths between the words are also necessary to compute the sentence structures for EFP. To this end, we introduce a novel deep learning model for EFP that explicitly considers multi-hop paths with both syntax-based and semantic-based edges between the words to obtain sentence structures for representation learning in EFP. We demonstrate the effectiveness of the proposed model via the extensive experiments in this work.

1 Introduction

In Information Extraction (IE), an event mention is represented via an anchor/trigger word that evokes an event in the input sentence. We study the problem of Event Factuality Prediction (EFP) that aims to identify the degrees of uncertainty/factuality for event mentions in text. Among others, EFP finds its applications in knowledge base construction to differentiate between factual and non-factual event mentions. In this work, we follow the recent regression formulation for EFP that seeks to predict a real-valued score in the range of [-3,3] to indicate the occurrence possibility for a given event mention (Stanovsky et al., 2017; Rudinger et al., 2018). For instance, in the sentence "*He cannot go to the restaurant.*", "*go*" is the trigger word for an event mention with the factuality score of -3 (i.e., certainly not happened).

In order to predict the factuality scores for the event mentions, the EFP models need to locate the important context words in the sentences (i.e., the cue words) and combine them appropriately to reveal the factuality for the event triggers. As the important context words might be distributed at different positions in the sentences, the current state-of-the-art deep learning models for EFP have relied on the sentence structures to facilitate the identification of the cue words. In particular, the sentence structures in the EFP models can be represented via the importance score matrices that involve cells to quantify the contribution of a context word for the representation vector computation of the current word for EFP (Veyseh et al., 2019a). The sentence structures would then be used to induce the representation vectors for the words to perform factuality prediction. Both syntactic and semantic structures of the sentences have been exploited in the deep learning models for EFP. As such, the syntactic structures are based on the direct connections between the words in the dependency parsing trees of the input sentences while the contextual similarities between the words are employed to form the semantic structures (Veyseh et al., 2019a).

Despite their success, a major limitation of the current deep learning models for EFP is their inability to capture the multi-hop paths between the words to produce the importance scores in the sentence structures for EFP. In particular, the current deep learning models for EFP have only focused on the direct connection/relation (i.e., one-hop path) between a pair of words to determine the importance score for the words in the structures. For example, the syntactic structures in (Veyseh et al., 2019a) involve an binary importance score matrix where a cell is only set to 1 if the two words corresponding to that cell are directly connected in the dependency tree. This is not desirable as based on our analysis, the multi-hop paths between the words are also important and should be considered

Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text, pages 46–55
November 11, 2021. ©2021 Association for Computational Linguistics

to generate better importance scores for the structures for EFP. Consider the trigger word "*involvement*" in the following sentence as an example:

It was confirmed that Hagai, Basir's brother, had been a key member of al-Qaeda while their representatives constantly denied the ***involvement*** *of any Basir's family members in this terrorist group.*

The important context words to correctly predict the factuality score +3 (i.e., actually happened) for "*involvement*" in this case involve "*any Basir's family members*", "*Basir's brother*", "*Hagai*", "*a key member*", and "*confirmed*". In the deep learning models for EFP, these important words should be encoded into the representation vector for "*involvement*" to perform factuality prediction. Using the dependency tree for this sentence, ones can use the one-hop paths (i.e., the edges) to link "*involvement*" with the information in '*any Basir's family members*". Similarly, the direct semantic similarity between "*involvement*" and "*member*" can also be used to connect "*a key member*" to "*involvement*" for representation learning in this case. However, it is very challenging for the one-hop paths to directly connect "*involvement*" with the other important context words (i.e., "*Basir's brother*", "*Hagai*", and "*confirmed*") (either syntactically or semantically) due to the far distances/differences between them in the sentence. Fortunately, by considering the multi-hop paths between the words, we can rely on "*any Basir's family members*" to link "*involvement*" with "*Basir's brother*" (i.e., with the semantic connection between "*family members*" and "*brother*") that can be further extended to "*Hagai*", "*member*", and "*confirmed*" via the dependency connections for representation learning. Besides the multi-hop nature, we also note that the edges along the multi-hop path in this example contains both syntactic and semantic connections (i.e., heterogeneous edge types) that are necessary to identify the important context words for EFP.

Motivated by this limitation, in this work, we propose to learn the importance scores for the sentence structures, leveraging Graph Transformer Networks (GTN) (Yun et al., 2019) to facilitate the emergence of the effective multi-hop paths with heterogeneous edge types for EFP. In particular, we propose to first generate the initial sentence structures for EFP based on both the syntactic and semantic information. These initial sentence structures are then combined by the GTN model via the weighted sums, serving as the intermediate structures that are able to capture both syntactic and semantic one-hop connections between the words for EFP. Afterward, the intermediate structures are multiplied to induce the final structures that enable the modeling of the multi-hop paths with heterogeneous edge types to compute the importance scores for the structures (Yun et al., 2019). As illustrated by our example, we expect that these multi-hop paths between the words can help to produce more effective representation vectors for the deep learning models to achieve better performance for EFP.

Finally, in order to improve the generalization of the proposed model for EFP, we propose a novel inductive bias for the GTN model based on the Information Bottleneck technique (Tishby et al., 2000). In particular, the rich combined structures from the syntactic and semantic information might offer the proposed GTN model with the high capacity for representation learning to encode the detailed information in the input sentences. As the training datasets for EFP are generally small, such high capacity might eventually lead to the overfitting of the GTN model where all the context information in the input sentences, including the irrelevant ones, is preserved in the induced representation vectors. To this end, we propose to promote the GTN model in this work as an information bottleneck so the GTN-produced representations are trained to not only have good factuality prediction performance but also maintain a minimal mutual information with the input sentences (Belghazi et al., 2018). The extensive experiments on four benchmark datasets demonstrate the benefits of the proposed model, yielding the state-of-the-art performance for EFP in this work.

2 Related Work

Various methods have been proposed to solve EFP, including the early rule-based approaches (Nairn et al., 2006; Saurí, 2008; Lotan et al., 2013), the feature-based machine learning approaches (Diab et al., 2009; Prabhakaran et al., 2010; De Marneffe et al., 2012; Lee et al., 2015), and the hybrid methods (Saurí and Pustejovsky, 2012; Qian et al., 2015). The recent work has featured deep learning as the state-of-the-art method for EFP. In particular, (Qian et al., 2018) presents a model based on Generative Adversarial Networks (GAN) while (Rudinger et al., 2018) applies Long-short Term Memory Networks (LSTM) over both the sequential order and the dependency tree of the input sentences for factuality prediction. The best performance for EFP so far is reported by (Veyseh et al., 2019a) that linearly combines the syntactic and semantic structures for Graph Convolutional Neural Networks (GCN). We also employ syntactic and semantic structures for EFP in this work; however, our model presents novel techniques with trigger-based structure customization, GTNs to learn the sentence structures with the multi-hop path reasoning, and information bottleneck to improve the generalization for EFP. Model-wise, our work bears some similarity with other NLP models that leverage syntactic structures and GCNs to encode input texts for different NLP tasks, including relation extraction (Zhang et al., 2018), joint information extraction (Nguyen et al., 2021), metaphor detection (Le et al., 2020), and rumor detection (Veyseh et al., 2019b). Finally, we also note some related tasks for EFP that seek to classify event trigger words in texts, including event detection (Nguyen and Grishman, 2015; Chen et al., 2015; Lai et al., 2020; Veyseh et al., 2021), event realis classification (Mitamura et al., 2015; Nguyen et al., 2016) and uncertainty detection (Adel and Schütze, 2017).

3 Model

We formalize EFP as a regression problem in this work. In particular, given an input sentence $W = w_1, w_2, \ldots, w_N$ of N words/tokens (i.e., w_i is the i-th token) and an event mention with the trigger word located at the k-th position (i.e., w_k), we need to predict a real-valued score between -3 and +3 to indicate the factual degree for w_k.

In order to achieve a fair comparison with the prior work for EFP (Veyseh et al., 2019a), we first apply the BERT$_{base}$ model in (Devlin et al., 2019) to obtain a pre-trained embedding vector x_i for each word $w_i \in W$. In particular, we run the BERT$_{base}$ model over the input sentence W and use the hidden vector for the first wordpiece of w_i in the last layer of BERT as the embedding vector x_i (of 768 dimensions) for w_i. This encoding step transforms W into a sequence of embedding vectors $X = x_1, x_2, \ldots, x_N$ (called the input vectors) for the neural computation in the next steps. The EFP model in this work involves three major components: (i) structure generation, (iii) structure combination, and (iii) representation regularization. We will explain the details of these components in the following sections.

3.1 Structure Generation

The goal of this section is to generate the initial sentence structures that would be combined in the next steps to generate richer structures for representation learning in EFP. Formally, the sentence structures in this work can be seen as the importance score matrices of size $N \times N$. Each cell (i, j) in these matrices contains a score to represent the importance of the contextual information from w_j for the representation vector of w_i if this vector is used to create the features for factuality prediction (called the importance score for the pair (w_i, w_j)). Following the previous work for EFP, we consider two types of sentence structures for EFP in this work, i.e., the syntactic structures and the semantic structures (Veyseh et al., 2019a).

Syntactic Structures: As presented in the introduction, the syntactic structures would leverage the information in the dependency tree T of W to compute the syntactic importance scores for EFP. The simplest approach for the syntactic structures is to directly use the binary adjacency matrix $A^{syn} = \{a^{syn}_{i,j}\}_{i,j=1..N}$ of T for the importance score matrix as in (Veyseh et al., 2019a): $a^{syn}_{i,j} = 1$ if w_i and w_j are connected in T or $i = j$. This approach is based on the motivation that the syntactic neighboring words of w_i in T would be the most informative words to reveal the contextual semantics of w_i for EFP (Veyseh et al., 2019a). However, one problem with this syntactic structure is its ignorance of the trigger word $w_k \in W$ (i.e., A^{syn} is not dependent on w_k). As w_k is the focused word in EFP, in this work, we argue that the syntactic structures should be conditioned on the trigger word w_k to produce more effective structures for representation learning in EFP. To this end, we propose to customize the syntactic structures for the event triggers in EFP, leveraging the intuition that the closer words to w_k in the dependency tree T would provide more contextual information for the representation vectors in EFP than the farther ones (e.g., the words "*Basir's family members*" in our running example). The syntactic neighboring words of w_k in T should thus be assigned with higher importance scores in the syntactic sentence structures for EFP, serving as the main method to achieve trigger-based customization for the syntactic structures in this work. In particular, to generate the task-specific syntactic structures, we first compute the length d_i of the shortest path between w_i and the trigger word w_k (i.e., the distance) in T for all

$1 \leq i \leq N$. Afterward, we obtain the customized syntactic structure $A^{syn} = \{a_{i,j}^{syn}\}_{i,j=1..N}$ via:

$$a_{i,j}^{syn} = \sigma(FF([d_i, d_j, d_i * d_j, d_i + d_j, |d_i - d_j|])) \quad (1)$$

where [] is the vector concatenation, FF is a two-layer feed-forward network to convert a vector to a scalar, and σ is the sigmoid function. We expect that learning the syntactic structures in this way would introduce the flexibility to infer effective structures for EFP.

Semantic Structure: The importance score for a pair of words (w_i, w_j) in the semantic structures would be based on the contextual semantics of w_i and w_j in the sentence (Veyseh et al., 2019a). As such, to capture the contextual semantic for $w_i \in W$, we directly utilize the embedding vector x_i from the BERT model of the encoding step. As BERT is a deep model that has been trained on a large corpus, we expect that the BERT-based vectors x_i would provide effective semantic representations for the importance scores in this case. Concretely, given the semantic vectors x_i and x_j for w_i and w_j, the semantic importance scores $a_{i,j}^{sem}$ for the semantic structure $A^{sem} = \{a_{i,j}^{sem}\}_{i,j=1..N}$ can be learned via $a_{i,j}^{sem} = f(x_i, x_j)$ where f is some learnable function to fuse x_i and x_j to produce a score. A simple version of the function f for the semantic importance scores is presented in (Veyseh et al., 2019a):

$$\begin{aligned} x'_i &= tanh(W_1^{sem} x_i) \\ a_{i,j}^{sem} &= \sigma(W_2^{sem}[x'_i, x'_j]) \end{aligned} \quad (2)$$

where W_1^{sem} and W_2^{sem} are the weight matrices and the biases are omitted for brevity.

Similar to the simple syntactic structure A^{syn}, a problem for this version of f is that the semantic scores $a_{i,j}^{sem}$ are not dependent on the trigger word w_k, potentially causing the lack of necessary context (i.e., the trigger word) to obtain the sentence structures for EFP. To this end, we propose to improve the semantic score function f in (Veyseh et al., 2019a) by additionally including the embedding vector x_k of the trigger word w_k into the computation of the semantic structure A^{sem} for EFP. In particular, we first employ the embedding vector x_k of the trigger word w_k to generate a task-specific control vector c^{syn}. This control vector would then be used to filter the information in the embedding vectors x_i of the words in W so only the relevant information for the trigger word w_k in EFP is preserved. This trigger-based filtering will serve as the main mechanism to customize the semantic structures for the trigger words for EFP in this work. Finally, to obtain the task-specific semantic structures, the filtered vectors would be sent to the same function in (Veyseh et al., 2019a) to compute the importance scores $a_{i,j}^{sem}$:

$$\begin{aligned} c^{syn} &= tanh(W_3^{sem} x_k) \\ x'_i &= tanh(W_4^{sem} x_i), x''_i = c^{syn} \odot x'_i \\ a_{i,j}^{sem} &= \sigma(W_5^{sem}[x''_i, x''_j]) \end{aligned} \quad (3)$$

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

3.2 Structure Combination

In this work, we consider the customized sentence structures A^{syn} and A^{sem} as two different types of relations between the pairs of words in W (called the relation word types). For these structures, the importance score in the cell (i, j) is intended to capture the degree of connection between w_i and w_j based on their direct interaction/edge (i.e., the one-hop path (w_i, w_j)) and the corresponding relation type (i.e., syntactic relations for A^{syn} and semantic relations for A^{sem}). Given this interpretation for the structures, this component aims to combine A^{syn} and A^{sem} to generate richer sentence structures for EFP. In particular, instead of only relying on the direct interactions between a pair of word (w_i, w_j) to compute the importance scores, the combined structures should be able to model the multi-hop interactions between w_i and w_j that possibly involve the other words in W (i.e., the multi-hop reasoning paths between w_i and w_j). In addition, the multi-hop reasoning paths between w_i and w_j are also expected to enable the appearance of the direct edges/connections between the words that belong to different relation types in the initial structures (i.e., heterogeneous edge types with syntactic and semantic relations). As illustrated in the introduction, both the multi-hop reasoning paths and the heterogeneous edge types are necessary for factuality score prediction in our problem. Consequently, in this work, we propose to further feed A^{syn} and A^{sem} into the Graph Transformer Networks (GTN) (Yun et al., 2019) that are able to generate rich sentence structures with multi-hop reasoning paths and heterogeneous edge types, thus fitting well with our intuition for EFP.

In particular, to learn the multi-hop paths at different lengths, following (Yun et al., 2019), we first include the identity matrix I (of size $N \times N$) into the set $\mathcal{A}$ of the initial structures for EFP, i.e., $\mathcal{A} = [A^{syn}, A^{sem}, I] = [\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3]$. The GTN model in this work would then process these initial structures via C channels to learn richer structures for EFP. At the i-th channel of GTNs ($1 \leq i \leq C$), M intermediate structures $Q^i_1, Q^i_2, \ldots, Q^i_M$ of size $N \times N$ (i.e., amounting to $M-1$ layers in GTNs) are computed via the weighted sums of the initial structures in $\mathcal{A}$: $Q^i_j = \sum_{v=1..3} \alpha^i_{j,v} \mathcal{A}_v$ for all $1 \leq j \leq M$ ($\alpha^i_{j,v}$ are the learnable weights). Note that similar to (Veyseh et al., 2019a), the weighted sums help to combine A^{syn} and A^{sem}, enabling the intermediate structures Q^i_j to reason with any of the two relation types (i.e., syntactically with A^{syn} or semantically with A^{sem}) for EFP. Afterward, to capture the multi-hop paths for the importance scores, the intermediate structures at the i-th channel are multiplied to obtain a single sentence structure Q^i for this channel: $Q^i = Q^i_1 \times Q^i_2 \times \ldots \times Q^i_M$. The resulting structures Q^i at the GTN channels serve as the final structures that can model any multi-hop reasoning paths with lengths up to M and edges of heterogeneous relation types (i.e., syntactic or semantic) for the importance scores (as demonstrated in (Yun et al., 2019)).

In the next step, the final structures $Q^1, Q^2, \ldots, Q^C$ of GTN would be used as the adjacency matrices in a Graph Convolutional Network (GCN) model (Kipf and Welling, 2017; Nguyen and Grishman, 2018) over the input vector sequence X to induce more abstract representation vectors for the words in W for EFP. In particular, the GCN model in this work involves G layers to compute the representation vectors at different abstract levels for the words. For the j-th final structure Q^j, the representation vector $h^{j,t}_i$ for the word w_i in the t-th GCN layer is computed via:

$$h^{j,t}_i = ReLU(U^t \sum_{v=1..N} \frac{Q^j_{i,v} h^{j,t-1}_v}{\sum_{u=1..N} Q^j_{i,u}}) \qquad (4)$$

where U^t is the weight matrix for the t-th GCN layer and the input vectors $h^{j,0}_i$ for the GCN model are obtained from BERT-generated vectors x_i (i.e., $h^{j,0}_i = h_i$ for all $1 \leq j \leq C, 1 \leq i \leq N$).

Afterward, the hidden vectors in the last GCN layer for w_i for all the final structures (i.e., $h^{1,G}_i, h^{2,G}_i, \ldots, h^{C,G}_i$) are concatenated to form the final representation vector h'_i for w_i in the proposed model: $h'_i = [h^{1,G}_i, h^{2,G}_i, \ldots, h^{C,G}_i]$. Finally, in order to predict the factuality score for w_k in W, we create an overall representation vector R based on the hidden vectors from the GCN model via: $R = [h'_k, MaxPool(h'_1, h'_2, \ldots, h'_N)]$. This vector is then fed into a two-layer feed-forward network to produce the factuality score for the regression model. Following (Rudinger et al., 2018; Veyseh et al., 2019a), we use the Huber loss L_{pred} with $\delta = 1$ to train the models in this work.

3.3 Representation Regularization

Due to the high learning capacity with rich syntactic and semantic structures, the proposed GTN model might overfit to the training data by memorizing the irrelevant information from the input sentences in the induced representation vectors for EFP (as described in the introduction). In order to improve the generalization of the GTN model, we propose to regularize the representation vectors obtained by the GTN model so only the effective information for EFP is preserved in the representation vectors for factuality prediction. To this end, we introduce the Information Bottleneck (IB) framework (Tishby et al., 2000) into the GTN model so the GTN-produced representation vectors $H' = h'_1, h'_2, \ldots, h'_N$ would be simultaneously trained for two objectives: (1) retain the effective information to predict the factuality score for EFP (i.e., the high prediction capacity), and (2) achieve a small Mutual Information (MI)[1] with the representation vectors from the earlier layers of the model (i.e., the minimality of the representations) (Belghazi et al., 2018). In this work, on the one hand, we follow the common practice to accomplish the high prediction capacity by training the GTN representation vectors to directly perform the prediction task of interest (i.e., the factuality score prediction in our case of EFP). On the other hand, we propose to achieve the minimality of the representations for the GTN model by explicitly minimizing the MI between the GTN-produced vectors $H' = h'_1, h'_2, \ldots, h'_N$ and the BERT-produced hidden vectors $X = x_1, x_2, \ldots, x_N$ from sentence encoding. By enforcing a small MI between X and H', we expect that only the relevant information for EFP in X is passed through the GTN bottleneck to be recorded in H' for better generalization.

In order to facilitate the MI estimation between X and H', we first aggregate them into a single summarization vector via the max-pooling function: $x = MaxPool(x_1, x_2, \ldots, x_N)$ and $h' = MaxPool(h'_1, h'_2, \ldots, h'_N)$. We would then evaluate the MI between x and h' and include it in the

[1] In information theory, MI measures the information we know about one random variable if the value of another variable is revealed.

overall loss function for minimization. Note that the MI between x and h' is the KL divergence between the joint and marginal distributions of these variables. Unfortunately, the direct computation for the MI between x and h' is prohibitively expensive due to their high dimensions. Consequently, in this work, we propose to apply the mutual information neural estimation (MINE) method in (Belghazi et al., 2018) to approximate the MI with its lower bound. In particular, motivated by (Hjelm et al., 2019), we further approximate the lower bound of the MI between two the vectors/variables x and h' via the adversarial approach using the loss function of a variable discriminator. The goal of the discriminator is to differentiate the vectors that are sampled from the joint distribution and those from the product of the marginal distributions of the variables. In our case, we sample from the joint distribution for x and h' by directly concatenating the two vectors (i.e., $[h', x]$) and treat it as the positive example. To obtain the sample from the product of the marginal distributions, we first obtain another sentence $\hat{W}$ from the same batch with the current sentence W during training. Afterward, we compute the aggregated vector $\hat{x}$ (i.e., via max-pooling) of the BERT-produced vectors for the words in $\hat{W}$. The concatenation vector $[h', \hat{x}]$ would then be used as the sampled vector for the product of the marginal distributions (the negative example). These positive and negative examples are then fed into a two-layer feed-forward network D (i.e., the discriminator) to produce a scalar score, serving as the probability to perform a binary classification for the variables. Afterward, we use the logistic loss for the discriminator $\mathcal{L}_{disc}$ as an estimation for the MI between x and h' and add it into the overall loss function for minimization: $\mathcal{L}_{disc} = \log(1+e^{(1-D([h',x]))})+\log(1+e^{D([h',\hat{x}])})$.

Finally, the overall loss function $\mathcal{L}$ to train the model in this work would be: $\mathcal{L} = \mathcal{L}_{pred} + \alpha_{disc}\mathcal{L}_{disc}$ where α_{disc} is a trade-off parameter.

4 Experiments

Datasets & Parameters: Following the previous work (Stanovsky et al., 2017; Rudinger et al., 2018; Veyseh et al., 2019a), we evaluate the proposed model on four datasets for EFP: FactBank (Saurí and Pustejovsky, 2009), UW (Lee et al., 2015), Meantime (Minard et al., 2016) and UDS-IH2 (Rudinger et al., 2018). The factuality scores for the first three datasets (i.e., FactBack, UW, and Meantime) are unified and scaled to the values in [-3, +3] based on their original annotations by (Stanovsky et al., 2017). The scaling of the factuality scores for UDS-IH2, on the other hand, is done with the procedure described in (Rudinger et al., 2018) (i.e., the scores are also between -3 and +3 in this case). In order to achieve a fair comparison, we obtain the same scaled and preprocessed versions of these datasets (i.e., with dependency trees) from (Veyseh et al., 2019a) where the training/development/test data is provided for each dataset.

We use the development datasets to tune the hyper-parameters for the models in this work. The values suggested by this tuning process include: 300 dimensions for the hidden vectors in the layers of the GCN model and all the feed-forward networks (i.e., to compute $a^{syn}_{i,j}$ for the customized syntactic structures, and to consume the overall representation vector R), $G = 2$ layers for the GCN model, $C = 2$ channels for the GTN model with $M = 3$ intermediate structures in each layer, and a learning rate of $1e$-5 for the Adam optimizer. For the trade-off parameter α_{disc} in the loss function $\mathcal{L}$, the best values based on the development data is 0.1 for the FactBank, UW, and UDS-IH2 datasets, and 0.5 for Meantime.

Comparing with the State of the Art: This part compares the proposed model (called "*SynSem-Customization+MultiHop+GCN+IB+BERT*") with the previous models for EFP. In particular, we consider both the traditional feature-based models (Lee et al., 2015; Stanovsky et al., 2017) and the recent deep learning methods (Rudinger et al., 2018; Veyseh et al., 2019a) as the baselines for EFP. Note that the model in (Veyseh et al., 2019a) (called "*SynSemLinearCombine+GCN+BERT*") currently has the best reported performance on the datasets. Table 1 reports the test set performance of the models, using Mean Absolute Error (i.e., MAE) and Pearson Correlation (i.e., r) as the performance measures.

Similar to the prior work (Rudinger et al., 2018; Veyseh et al., 2019a), we consider two methods to train the models in this work: (i) training and evaluating the models on separate datasets (i.e., the rows with * in the table), and (ii) training the models on the union of FactBank, UW and Meantime, leading to a single model to be evaluated on the test data of the individual datasets (i.e., the rows with ** in the table). As we can see from the table, for both training methods, the proposed model significantly

Models	FactBank		UW		Meantime		UDS-IH2	
	MAE	r	MAE	r	MAE	r	MAE	r
(Lee et al., 2015)*	-	-	0.511	0.708	-	-	-	-
(Stanovsky et al., 2017)*	0.590	0.710	0.420	0.660	0.340	0.470	-	-
Models reported in (Rudinger et al., 2018)								
L-biLSTM(2)-S*	0.427	0.826	0.508	0.719	0.427	0.335	0.960	0.768
L-biLSTM(2)-MultiBal**	0.391	0.821	0.496	0.724	0.278	0.613	-	-
L-biLSTM(1)-MultiFoc**	0.314	0.846	0.502	0.710	0.305	0.377	-	-
L-biLSTM(2)-MultiSimp w/UDS-IH2**	0.377	0.828	0.508	0.722	0.367	0.469	0.965	0.771
H-biLSTM(1)-MultiSimp**	0.313	0.857	0.528	0.704	0.314	0.545	-	-
H-biLSTM(2)-MultiSimp w/UDS-IH2**	0.393	0.820	0.481	0.749	0.374	0.495	0.969	0.760
Models reported in (Veyseh et al., 2019a)								
L-biLSTM(2)-S+BERT*	0.381	0.850	0.475	0.752	0.389	0.394	0.895	0.804
L-biLSTM(2)-MultiSimp w/UDS-IH2+BERT**	0.343	0.855	0.476	0.749	0.358	0.499	0.841	0.841
H-biLSTM(1)-MultiSimp+BERT**	0.310	0.821	0.495	0.771	0.281	0.639	0.822	0.812
H-biLSTM(2)-MultiSimp w/UDS-IH2+BERT**	0.330	0.871	0.460	0.798	0.339	0.571	0.835	0.802
SynSemLinearCombine+GCN+BERT*	0.315	0.890	0.451	0.828	0.350	0.452	0.730	0.905
SynSemLinearCombine+GCN+BERT**	0.310	0.903	0.438	0.830	0.204	**0.702**	0.726	0.909
Models proposed in this work								
SynSemCustomization+MultiHop+GCN+IB+BERT*	0.257	0.914	0.392	0.850	0.197	0.619	0.511	0.915
SynSemCustomization+MultiHop+GCN+IB+BERT**	**0.239**	**0.920**	**0.389**	**0.852**	**0.190**	0.685	**0.482**	**0.918**

Table 1: Test set performance. * denotes the models trained on separate datasets while ** indicates those trained on multiple datasets. The smaller values are better for MAE while the correlation r prefers the larger values.

Models	UW		UDS-IH2	
	MAE	r	MAE	r
The proposed model	**0.389**	**0.852**	**0.482**	**0.918**
- A^{syn}	0.448	0.842	0.590	0.909
- A^{sem}	0.449	0.839	0.580	0.904

Table 2: The contribution of the initial structures.

outperforms the baseline models across different performance measures and datasets (except for r on Meantime). In fact, the separate dataset performance of the proposed model is also significantly better than the performance of the other models with the union of the datasets for training. The proposed model achieves the state-of-the-art performance when trained on multiple datasets, clearly demonstrating the benefits of the model in this work for EFP. As UW and UDS-IH2 are the two largest datasets among the four considering datasets, we will focus on them in the following model analysis.

Structure Analysis: The proposed model for EFP has two major sentence structures in the initial set $\mathcal{A} = [A^{syn}, A^{sem}]$ based on the syntactic and semantic information (i.e., A^{syn} and A^{sem}). This part investigates the effectiveness of the individual structures by evaluating the performance of the remaining models when each of these structures is eliminated from the overall proposed model. Table 2 presents the performance of the models[2]. It is clear from the table that the model performance is significantly worse when we remove any of the initial structures in $\mathcal{A}$, thus testifying to the benefits of the initial structures for the proposed model.

Ablation Study: There are three major components in the proposed models for EFP, i.e., the structure customization, the structure combination with GTN, and the representation regularization with information bottleneck. In order to analyze the contribution of these components, this part seeks to remove each of them from the overall model and evaluate performance of the remaining models. In particular, we consider two ablated models for the structure customization: (i) avoiding the trigger-based customization for the syntactic structure A^{syn} (i.e., instead of using Equation 1, the adjacency matrix of the dependency tree T is directly used for A^{syn} as in (Veyseh et al., 2019a)) (called "**- SyntaxCustom**"), and (ii) avoiding the trigger-based customization for the semantic structure A^{sem} (i.e., instead of using Equation 3, the function in Equation 2 is employed to compute the syntactic structure A^{sem} as in (Veyseh et al., 2019a)) (called "**- SemanticCustom**").

The main benefit of the GTN models in the second component for structure combination is to combine the initial syntactic and semantic structures

[2] Note that we train the models in analysis experiments with the multiple dataset setting (i.e., FactBank, UW and Meantime); however, the same trends for the models also hold for the setting with separate dataset training.

	UW		UDS-IH2	
Models	MAE	r	MAE	r
The proposed model	**0.389**	**0.852**	**0.482**	**0.918**
- SyntaxCustom	0.429	0.838	0.578	0.901
- SemanticCustom	0.409	0.844	0.565	0.909
- GTN	0.461	0.828	0.610	0.894
- Multi-Hop	0.402	0.836	0.587	0.905
- IB	0.450	0.842	0.602	0.907
- IB + BERT in R	0.419	0.830	0.564	0.910

Table 3: The ablation study.

to generate richer structures with multi-hop path reasoning. Consequently, we examine two ablated versions for this component: (i) completely removing the GTN model for structure combination and directly running the GCN model on the initial structures in $\mathcal{A}$ (so the intermediate and final structures are not computed) (called "**- GTN**"), and (ii) only generating the intermediate structures and avoiding the intermediate structure multiplications for multi-hop path reasoning in each channel of GTN. The final structures are thus not computed and the GCN model is applied directly over the intermediate structures in this case (called "**- Multi-Hop**")[3].

Finally, for third component with representation regularization, the introduction of the information bottleneck (IB) leads to the inclusion of the loss term $\mathcal{L}_{disc}$ in the overall loss function $\mathcal{L}$. The removal of this regularization loss $\mathcal{L}_{disc}$ from $\mathcal{L}$ amounts to the ablated model "**- IB**" for this component. In addition, as this component relies on the MI between the hidden vectors computed by the BERT and GTN models for the words, we further evaluate another version for the overall model in which the regularization loss $\mathcal{L}_{disc}$ is also removed from $\mathcal{L}$, but the hidden vectors from the BERT model $X = x_1, x_2, \ldots, x_N$ are incorporated into the final representation vector R for prediction (i.e., $R = [x_k, x, h'_k, h']$) (called "**- IB + BERT in** R"). The performance of the models for this ablation study is shown in Table 3.

The most important observation from the table is that all the components are important for the proposed model to ensure the highest performance. In particular, the customization for the syntactic and semantic structures are necessary as eliminating any of them would reduce the performance significantly. The removal of the GTN model or its multi-hop path reasoning for the structures also makes the performance worse, thus highlighting the benefits of the structure combination with multi-hop paths for the structures for EFP in this work. Finally, the better performance of the proposed model over "- IB" and "-IB + BERT in R" clearly demonstrates the ability of the IB-based regularization technique to improve the generalization of the proposed model in this work.

Error Analysis: In order to better understand the errors made the proposed model for EFP, we analyze the outputs of the model on the test set of the UDS-IH2 dataset (i.e., the largest dataset in our case). In particular, we examine the examples for which the absolute values of the differences between the predicted factuality scores and the golden ones are greater than 1 (i.e., focusing on the examples with the largest prediction errors). A notable insight from our analysis is that among 118 examples selected in this way, 71.4% of the examples involves the same signs for the predicted and golden factuality scores. This suggests that although the proposed model has large prediction error on these examples, it can still capture the correct factuality polarity (i.e., positive or negative) for a great portion of the examples (i.e., 71.4%). In other words, a main source of errors for the proposed model has to do with the difficulty to identify the degrees of factuality (i.e., the fine-grained distinction with the real-valued factuality scores) for the events, not with the factuality polarity.

In addition, among the 28.6% of the examples with both large prediction errors and different signs for the predicted and golden scores, we find that a major portion of the examples (i.e., 62.5%) involves important context words that are not present in the training data (i.e., unknown word issue). Some examples of this type are shown below where the unknown and important context words are highlighted (the trigger words are in bold):

Israel-Syrian ***talks*** *have been cut off for two years.* (Predicted score: 2.78, Golden Score: -3).

A man who was accused of faking his ***death*** *last summer pleaded guilty to a conspiracy charge ...* (Predicted score: 2.45, Golden Score: -3).

Based on this observation, we hypothesize that even with the contextualized word embeddings (e.g., BERT) and the wordpiece tokenization to encode the input sentences, unknown words still constitute a challenging problem for EFP. In par-

[3] Note that for the ablated models in this component, we also re-tune the numbers of intermediate structures and channels for the GTN model (i.e., M and C), and the number of layers for the GCN model (i.e., G) on the the development sets, leading to $M = 3$, $C = 2$, and $G = 2$.

ticular, as the unknown words do not appear in the training data, the model does not have sufficient training signals to adapt the initial language models (i.e., BERT) to appropriately encode the unknown words for EFP. Future research can focus on these directions to improve the performance for EFP.

5 Conclusion

We present a novel deep learning model for EFP that combines the customized sentence structures (i.e., based on both syntactic and semantic information) to learn effective representation vectors. Our model features GTNs to infer rich sentence structures with multi-hop reasoning paths for the importance scores and information bottleneck to improve the generalization. We perform extensive experiments to demonstrate the effectiveness of the proposed model. In the future, we plan to extend the proposed model to the related tasks of EFP.

Acknowledgments

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112 and the NSF grant CNS-1747798 to the IUCRC Center for Big Learning. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Heike Adel and Hinrich Schütze. 2017. Exploring different dimensions of attention for uncertainty detection. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Marie-Catherine De Marneffe, Christopher D Manning, and Christopher Potts. 2012. Did it happen? the pragmatic complexity of veridicality assessment. *Computational linguistics*, 38(2):301–333.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Mona T Diab, Lori Levin, Teruko Mitamura, Owen Rambow, Vinodkumar Prabhakaran, and Weiwei Guo. 2009. Committed belief annotation and tagging. In *Proceedings of the Third Linguistic Annotation Workshop*.

Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Duong Minh Le, My Thai, and Thien Huu Nguyen. 2020. Multi-task learning for metaphor detection with graph convolutional neural networks and word sense disambiguation. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Kenton Lee, Yoav Artzi, Yejin Choi, and Luke Zettlemoyer. 2015. Event detection and factuality assessment with non-expert supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Amnon Lotan, Asher Stern, and Ido Dagan. 2013. Truthteller: Annotating predicate truth. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

A-L Minard, Manuela Speranza, Ruben Urizar, Begona Altuna, MGJ van Erp, AM Schoen, CM van Son, et al. 2016. Meantime, the newsreader multilingual event and time corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*.

Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2015. Overview of tac kbp 2015 event nugget track. In *Proceedings of Text Analysis Conference (TAC)*.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of the fifth international workshop on inference in computational semantics (icos-5)*.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Thien Huu Nguyen, Adam Meyers, and Ralph Grishman. 2016. New york university 2016 system for kbp event nugget: A deep learning approach. In *Proceedings of Text Analysis Conference (TAC)*.

Vinodkumar Prabhakaran, Owen Rambow, and Mona Diab. 2010. Automatic committed belief tagging. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Zhong Qian, Peifeng Li, Yue Zhang, Guodong Zhou, and Qiaoming Zhu. 2018. Event factuality identification via generative adversarial networks with auxiliary classification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Zhong Qian, Peifeng Li, and Qiaoming Zhu. 2015. A two-step approach for event factuality identification. In *Asian Language Processing (IALP), 2015 International Conference on*, pages 103–106. IEEE.

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Roser Saurí. 2008. A factuality profiler for eventualities in text. *Unveröffentlichte Dissertation, Brandeis University. Zugriff auf http://www. cs. brandeis. edu/~ roser/pubs/sauriDiss*, 1.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227.

Roser Saurí and James Pustejovsky. 2012. Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, 38(2):261–299.

Gabriel Stanovsky, Judith Eckle-Kohler, Yevgeniy Puzikov, Ido Dagan, and Iryna Gurevych. 2017. Integrating deep linguistic features in factuality prediction over unified datasets. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. In *arXiv preprint physics/0004057*.

Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Unleash GPT-2 power for event detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019a. Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Amir Pouran Ben Veyseh, My T. Thai, Thien Huu Nguyen, and Dejing Dou. 2019b. Rumor detection in social networks via deep contextual modeling. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.