Fine-grained Temporal Relation Extraction with Ordered-Neuron LSTM and Graph Convolutional Networks

Minh Phu Tran[1]*, **Minh Van Nguyen**[2]*, **and Thien Huu Nguyen**[2]
[1] VinAI Research, Vietnam
[2] Dept. of Computer and Information Science, University of Oregon, Eugene, OR, USA
v.minhtp@vinai.io, {minhnv,thien}@cs.uoregon.edu

Abstract

Fine-grained temporal relation extraction (FineTempRel) aims to recognize the durations and timeline of event mentions in text. A missing part in the current deep learning models for FineTempRel is their failure to exploit the syntactic structures of the input sentences to enrich the representation vectors. In this work, we propose to fill this gap by introducing novel methods to integrate the syntactic structures into the deep learning models for FineTempRel. The proposed model focuses on two types of syntactic information from the dependency trees, i.e., the syntax-based importance scores for representation learning of the words and the syntactic connections to identify important context words for the event mentions. We also present two novel techniques to facilitate the knowledge transfer between the subtasks of FineTempRel, leading to a novel model with the state-of-the-art performance for this task.

1 Introduction

An important step in event understanding involves identifying the temporal relations between events (i.e., TempRel), finding its applications in different natural language processing (NLP) systems such as question answering and timeline construction. A large volume of the prior works has focused on the classification setting for this problem where categorical temporal relations should be predicted for pairs of event-referring and/or time-referring expressions in text (i.e., categorical TempRel) (Dligach et al., 2017; Cheng and Miyao, 2017; Ning et al., 2019). For instance, in the sentence "*The meeting to discuss the possible merger of the two financial companies lasted for two hours, eventually leading to their union yesterday.*", a system for TempRel should be able to realize that the "*discussion*" event happens before the "*union*" event (i.e., the categorical label of *BEFORE*).

*The first two authors contribute equally to this paper.

However, a major problem with the classification setting is its inability to capture the fine-grained distinction between the temporal structures of the events (i.e., the duration information). For example, the classification setting would not be able to specify the amount of time between the end time of an event and the start time of a later one in the timeline. To this end, (Vashishtha et al., 2019) presents the first work on fine-grained temporal relation extraction (FineTempRel) that seeks to distribute event pairs in a real-valued relative timeline. In particular, in FineTempRel, given a pair of events, the systems need to predict the start and end times (thus the durations) of the events so the relative orders between these times of the events can be revealed. Our work follows this fine-grained setting for TempRel, aiming to introduce a novel model to improve the performance for this problem.

The current state-of-the-art methods for FineTempRel have involved deep learning models (Vashishtha et al., 2019); however, one problem with these deep learning models is that they fail to exploit the syntactic structures of the input sentences (i.e., the parsing trees) to further advance the performance. Consequently, in the current work, we seek to fill in this gap by extracting useful knowledge from the syntactic structures to help the deep learning models learn better representations for FineTempRel. In particular, based on the dependency parsing trees, we envision two major types of syntactic information that can be complementarily beneficial for the deep learning models for FineTempRel in this work, i.e., the syntax-based importance scores and the syntactic word connections for representation learning. First, for the syntax-based importance scores, our intuition is that the closer words to the words along the shortest dependency paths between the two event mentions of interest would involve more useful context information for FineTempRel than the farther ones. For instance,

Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text, pages 35–45

in our running example, the word "*leading*" along the shortest dependency path between "*meeting*" and "*union*" is an important word to reveal the temporal relations between these two events, suggesting the distances from the words to the shortest dependency path between the event mentions as an useful feature for FineTempRel. Consequently, in this work, we propose to use such dependency distances to obtain a score to represent the contextual importance for every word in the sentence (i.e., the importance scores). These importance scores would then be introduced into the deep learning models to improve the representation learning for FineTempRel.

How can we integrate these syntax-based importance scores into the deep learning models for FineTempRel? In this work, we propose to employ the representation vectors for the words in the deep learning models to compute a model-based importance score for each word in the sentence. Afterward, we propose to introduce the information from the syntax-based importance scores into the models for FineTempRel by enforcing the similarity/consistency between the syntax-based and model-based importance scores for the words in the sentence. The motivation is to leverage the importance score consistency to guide the representation learning process of the deep learning models (using the extracted syntactic information) so more effective representation vectors for FineTempRel can be induced. In order to implement this idea, we utilize the Ordered-Neuron Long Short-Term Memory Networks (ON-LSTM) (Shen et al., 2019; Veyseh et al., 2020a) to facilitate the computation of the model-based importance scores and the effective integration of the syntax-based scores for better representation vectors for FineTempRel.

For the second type of syntactic information, the main motivation is to leverage the syntactic dependency connections between the words to identify the important context words that should be encoded to compute effective representation vectors for the event mentions in the sentences. In particular, following (Vashishtha et al., 2019), we decompose FineTempRel into two subtasks that would be solved jointly for a given pair of events in this work, i.e., event duration prediction (i.e., predicting the durations of the events) and temporal relation prediction (i.e., predicting the start and end times of the events). First, for event duration prediction, we argue that the important context words for the representation vectors of the event mentions involve the syntactic neighboring words of the event mentions in the dependency trees. For instance, in our example, the words "*lasted*" and "*two hours*" are crucial to determine the duration for the event mention "*meeting*". Note that although these words are far away from "*meeting*" in the sentence, they are directly connected to "*meeting*" in the dependency tree (i.e., the syntactic neighboring words). Second, for temporal relation prediction, our intuition is based on (Cheng and Miyao, 2017) and (Goyal and Durrett, 2019) that use the shortest dependency paths between the event mentions to capture the important context words for categorical TempRel (e.g., the word "*leading*" in our example). Motivated by these benefits of the dependency trees for FineTempRel, in this work, we propose to run Graph Convolutional Neural Networks (GCN) (Kipf and Welling, 2017; Nguyen and Grishman, 2018) over the dependency structures of the sentences to facilitate the incorporation of the dependency-based important context words into the representation vectors for FineTempRel. To our knowledge, this is the first work on using GCNs for TempRel in the literature.

In particular, we propose to employ two separate GCN models (i.e., one for for each task in FineTempRel) to induce more flexible representation vectors for the two subtasks in FineTempRel. Afterward, we introduce a mechanism to allow the layers of the two GCN models to interact with each other so the knowledge learned for one task in each GCN layer can be used to improve the representation vectors for the other task (i.e., transfer learning). Finally, a novel inductive bias is proposed to further connect the two GCN models, seeking to enforce the similarity of the representation vectors learned by the two models for the same input sentences. We perform extensive experiments to demonstrate the effectiveness of the proposed model, yielding the state-of-the-art performance for FineTempRel on the benchmark datasets.

2 Related Work

Most of the previous work on TempRel has focused on the categorical setting using the TimeML standard for the datasets (i.e., TimeBank, TimeBank-Dense, Richer Event Description (RED)) (Pustejovsky et al., 2003; UzZaman et al., 2013; Cassidy et al., 2014; Minard et al., 2016; O'Gorman et al., 2016; Hong et al., 2016; Ning et al., 2018b,c).

(Vashishtha et al., 2019) is the first work to consider the fine-grained distinction for the temporal relations for events.

Regarding TempRel methods, the early approaches has involved feature-based models (Mani et al., 2006; Bethard, 2013; Lin et al., 2015), the hybrid methods (D'Souza and Ng, 2013), sieve-based methods (i.e., CAEVO (Chambers et al., 2014), CATENA (Mirza and Tonelli, 2016)), structured learning methods (Ning et al., 2017), and Interger Linear Programming (Ning et al., 2018). Recently, deep learning models have been developed and shown promising results for TempRel (Dligach et al., 2017; Tourille et al., 2017; Cheng and Miyao, 2017; Meng and Rumshisky, 2018; Ning et al., 2019; Han et al., 2019a). The closet work to ours is (Vashishtha et al., 2019) that presents an attention-based deep learning model for FineTempRel; however, it does not capture the syntactic structures of the sentences as we do in this work. Some previous works have also considered event duration modeling in text (Pan et al., 2007; Gusev et al., 2011; Williams and Katz, 2012; Filatova and Hovy, 2001) although they do not tie duration and temporal relations as we do. Finally, we also note related tasks that concern other types of relations between events/entities, including event coreference resolution (Lu et al., 2016; Nguyen et al., 2016; Lu and Ng, 2017; Tran et al., 2021), event causaltiy identification (Liu et al., 2020; Tran and Nguyen, 2021), and event argument extraction (Veyseh et al., 2020b; Nguyen et al., 2021).

3 Model

FineTempRel can be formulated as a regression problem. Formally, given a sentence W of N words: $W = w_1, w_2, \ldots, w_N$ with w_{m_1} and w_{m_2} ($1 \leq m_1 < m_2 \leq N$) as the trigger words for the two event mentions of interest, we need to predict the start and end times for the two event mentions to reflect the temporal order and durations of the events in the timeline. Following (Vashishtha et al., 2019), we aim to predict the start and end times b_1 and e_1 for w_{m_1} and b_2 and e_2 for w_{m_2} using a reference interval $[0, 1]$ (i.e., $0 \leq b_1 \leq e_1 \leq 1$ and $0 \leq b_2 \leq e_2 \leq 1$). Also, let d_1 and d_2 be the durations for w_{m_1} and w_{m_2} respectively. Note that as the input event mentions might belong to different sentences in FineTempRel, we follow (Vashishtha et al., 2019) to combine these sentences into a single sequence of words for W in those cases.

To prepare the input sentence for the deep learning models and to achieve a fair comparison with (Vashishtha et al., 2019), we first send W into the pre-trained language model ELMo (Peters et al., 2018) to produce a sequence of hidden vectors $X = x_1, x_2, \ldots, x_N$ for W. Note that the hidden vector x_i for $w_i \in W$ is the concatenation of the hidden vectors for w_i in three layers of ELMo.

3.1 Syntax-Model Consistency

The first component in our model for FineTempRel aims to exploit the consistency between the syntax-based and model-based importance scores for the words in the input sentence to improve the representation vectors in the deep learning models for FineTempRel. In particular, the syntax-based importance scores are supposed to evaluate the potential contributions of the words in W for the representation vectors in FineTempRel. As the closer words to the shortest dependency path *SDP* between between w_{m_1} and w_{m_2} in the dependency tree T of W are considered to be more important, we first compute the distance d_i^{syn} between every word $w_i \in W$ to *SDP* via: $d_i^{syn} = \min\{L(w_i, w) | w \in \textit{SDP}\}$ where $L(w_i, w)$ denotes the length of the shortest path between the words w_i and w in the dependency tree T. Afterward, the syntax-based importance score s_i^{syn} for the word w_i would be computed by: $s_i^{syn} = \frac{\exp(-d_i^{syn})}{\sum_{j=1..N} \exp(-d_j^{syn})}$.

In order to implement the importance score consistency, we additionally need to obtain the model-based importance scores $s_1^{mod}, s_2^{mod}, \ldots, s_N^{mod}$ for the words in W based on the representation vectors of the deep learning models for FineTempRel. While the computation of the model-based scores will be described later, the syntax-model consistency between the importance scores in this work is achieved by including the KL divergence $\mathcal{L}_{const}$ between the importance scores into the overall loss function for minimization during the training of the models: $\mathcal{L}_{KL} = -\sum_i s_i^{mod} \frac{s_i^{mod}}{s_i^{syn}}$.

Model-based Importance Scores: As presented in the introduction, we propose to compute the model-based importance scores for FineTempRel based on the Ordered-Neuron Long Short-Term Memory Networks (ON-LSTM) (Shen et al., 2019), an extension of the popular Long Short-Term Memory Networks (LSTM). In particular, given the vector sequence $X = x_1, x_2, \ldots, x_N$ as the input, a LSTM layer returns a sequence of

hidden vectors $H = h_1, h_2, \ldots, h_N$ via:

$$\begin{aligned} f_i &= \sigma(W_f x_i + U_f h_{i-1} + b_f) \\ i_i &= \sigma(W_i x_i + U_i h_{i-1} + b_i) \\ o_i &= \sigma(W_o x_i + U_o h_{i-1} + b_o) \\ \hat{c}_i &= tanh(W_c x_i + U_c h_{i-1} + b_c) \\ c_i &= f_i \circ c_{i-1} + i_i \odot \hat{c}_i, h_i = o_i \odot tanh(c_i) \end{aligned} \quad (1)$$

where h_0 is the zero vector, $\odot$ is the element-wise multiplication, and f_t, i_t and o_t are called the forget, input, and output gates respectively. In ON-LSTM, two additional gates (i.e., the master forget gate $\hat{f}_t$ and the master input gate $\hat{h}_t$) (Shen et al., 2019) are introduced into the LSTM cell via:

$$\begin{aligned} \hat{f}_i &= cummax(W_{\hat{f}} x_i + U_{\hat{f}} h_{i-1} + b_{\hat{f}}) \\ \hat{i}_i &= 1 - cummax(W_{\hat{i}} x_i + U_{\hat{i}} h_{t-1} + b_{\hat{i}}) \\ \bar{f}_i &= \hat{f}_i \odot (f_i \hat{i}_i + 1 - \hat{i}_i), \bar{i}_i = \hat{i}_i \odot (i_t \hat{f}_i + 1 - \hat{f}_i) \\ c_i &= \bar{f}_i \odot c_{i-1} + \bar{i}_i \odot \hat{c}_i \end{aligned} \quad (2)$$

where $cummax$ is an activation function: $cumax(x) = cumsum(softmax(x))$[1].

The difference between the forget and input gates in LSTM (i.e., f_t and i_t) and the master forget and input gates in ON-LSTM (i.e., $\hat{f}_t$ and $\hat{i}_t$) is that the dimensions/neurons of the hidden vectors in the gates of LSTM are considered equally important, being active/used for every word in W. This is in contrast to the master gates in ON-LSTM that impose an importance hierarchy over the neurons in the hidden vectors, enabling the neurons to be active for only a portion of the words in the sentence (i.e., neurons with higher rankings would be active for more words in the sentence). ON-LSTM achieves such neuron hierarchy and activity limitation via the function $cumax(x)$ that aggregates the softmax-produced result of the input vector x along its dimensions. The output of $cummax(x)$ would represent the expectation of a binary vector of the form $(0, \ldots, 0, 1, \ldots, 1)$ (i.e., two consecutive segments of 0's and 1's). At one word w_i, the 1's segment in its master gate vector correspond to the neurons that are activated for that word. In ON-LSTM, a word w_i is more contextually important than another word w_j for representation learning if the master gates for w_i have more active neurons than those for w_j (Shen et al., 2019). Consequently, in order to compute the model-based importance score for a word w_i, we employ the numbers of active neurons in the master gates for w_i that in turn can be estimated via the sums of the weights of the neurons in the gates of ON-LSTM. Following (Shen et al., 2019), we use the hidden vectors for the master forget gate to obtain the model-based importance scores s_i^{sem} for the words w_i:

$$d_i^{sem} = 1 - \sum_{i=1..D} \hat{f}_{ij}, s_i^{sem} = \frac{\exp(d_i^{sem})}{\sum_{j=1..N} \exp(d_j^{sem})} \quad (3)$$

where D is the dimensionality of the hidden vectors for ON-LSTM and $\hat{f}_{ij}$ is the weight of the j-th dimension of the master forget gate $\hat{f}_i$ at w_i.

Consequently, by promoting the syntax-model consistency with the loss $\mathcal{L}_{KL}$, we expect that the syntactic information from the syntax-based importance scores can deeply interfere with the internal computation/structure of the ON-LSTM cell (via the neurons of the master gates) to potentially lead to better representation vectors for FineTempRel. For convenience, we also use $H = h_1, h_2, \ldots, h_N$ to denote the hidden vectors returned by ON-LSTM over the input sequence vector X for the next components (called the ON-LSTM hidden vectors).

3.2 Graph Convolutional Networks

In the previous component, the syntax-model consistency has attempted to enrich the representation vectors for FineTempRel by encouraging them to capture the contextual importance scores for the words in the sentence. This component seeks to further improve the representation vectors for the event mentions by identifying and encoding the important context words for the temporal relation prediction between w_{m_1} and w_{m_2}. In particular, as presented in the introduction, we aim to leverage the words in the syntactic neighbors of w_{m_1} and w_{m_2} and their shortest dependency path in the dependency tree T of W for this purpose. As such, in order to encode these dependency-based important words for representation learning, we propose to feed the ON-LSTM hidden vectors H into the Graph Convolutional Neural Networks (GCN) that structure the computations over T. In particular, a GCN model in this work involves several layers (i.e., G layers in our case) to compute the representation vectors for the words in W with different abstract levels. At the $(k+1)$-th layer ($0 \leq k < G$), the representation vector $\bar{h}_i^{k+1}$ for the word $w_i \in W$ is computed via:

$$\bar{h}_i^{k+1} = \sigma\left(\frac{\Sigma_{w_j \in \mathcal{N}(i)} W_k \bar{h}_j^k}{|\mathcal{N}(i)|}\right) \quad (4)$$

Here, $\mathcal{N}(i)$ is the set of the words in W that are directly connected to w_i (including itself) in T, W_k is

[1] $cumsum(u_1, u_2, \ldots, u_n) = (u'_1, u'_2, \ldots, u'_n)$ where $u'_i = \sum_{j=1..i} u_j$.

the weight matrix (biases are omitted for simplicity in this work), and σ is the sigmoid function. The input vector $\bar{h}_i^0$ for GCNs is set to the ON-LSTM hidden vector h_i in this case (i.e., $\bar{h}_i^0 = h_i$ for all $1 \leq i \leq N$). Note that if there are more than one sentences in the input, following (Cheng and Miyao, 2017), we unify their dependency trees by introducing a new root node to be the parent of the roots of these dependency trees.

GCN Interaction: The straightforward application of GCNs in our FineTempRel problem is to run a single GCN model over H whose representation vectors are used for both event duration prediction (EDP) and temporal relation prediction (TRP). However, a problem with this approach is that the induced representations for the event mentions from the single GCN model might be confused between the representation learning for the two subtasks. In particular, EDP needs to focus on the modeling of the individual event mentions for their duration while jointly encoding the two event mentions is crucial for the TRP between them. Due to such distinction, a single GCN model might not be able to customize its computation for the different modeling expectations required by the two subtasks. Consequently, in this work, we propose to employ two different GCN models; each of them aims to learn the representation vectors for one subtask of FineTempRel. Among others, the benefit of the two GCN models is the improved flexibility to enable the models to tailor their operation toward the specific subtasks for FineTempRel. Note that the two GCN models share the architecture and are only different in terms of their weights.

Despite its flexibility, one limitation with the model so far is that the GCN models for EDP and TRP are operating separately, lacking the effective interactions to benefit from the knowledge transfer between the two subtasks. In particular, as shown in (Vashishtha et al., 2019), using the representation vectors for one subtask as one of the inputs to make prediction for the other task is helpful to improve the performance for FineTempRel on unseen test data. This demonstrates the relatedness of the two subtasks in FineTempRel, suggesting the knowledge transfer between the models for the two subtasks to enhance the representation vectors. To this end, we propose a novel interaction mechanism for the two GCN models of the two subtasks for FineTempRel in which the representation vectors in the current layer for one GCN model are additionally conditioned on the representation vectors from the previous layer of the other GCN model. In particular, let GCN^{dur} and GCN^{rel} be the GCN models for event duration and temporal relation prediction respectively. Also, let π_i^k and r_i^k ($1 \leq k \leq G$) be the hidden vectors for the word w_i at the k-the layers of GCN^{dur} and GCN^{rel} respectively ($\pi_i^0 = r_i^0 = h_i$ for all $1 \leq i \leq N$). The hidden vector π_i^{k+1} of GCN^{dur} ($1 \leq k < G$) in this work would then be computed based on the hidden vectors from the previous layer (i.e., the k-layer) of GCN^{rel} (in addition to the hidden vectors in the k-layer of GCN^{dur} itself):

$$\begin{aligned} \pi_i^{k+1} &= \sigma\left(\Sigma_{w_j \in \mathcal{N}(i)} \alpha_j^k W_k^\pi \pi_j^k\right) \\ \alpha_j^k &= \exp(u^k \pi_j^k) / \sum_{w_t \in \mathcal{N}(i)} \exp(u^k \pi_t^k) \\ u^k &= W_k^u [r_{m_1}^k, r_{m_2}^k] \end{aligned} \tag{5}$$

where W_k^π and W_k^u are the weight matrices at the k-layer for GCN^{dur}.

The rationale for this formula is to use the representation vectors for the event mentions at the k-layer of GCN^{rel} (i.e., $r_{e_1}^k$ and $r_{e_2}^k$) to compute a query vector u^k that would then be used to determine the weight α_j^k for each neighboring word $w_j \in \mathcal{N}(i)$ in the representation computation for π_i^{k+1}. A similar formula is employed to compute the hidden vector r_i^{k+1} of GCN^{rel} in this work:

$$\begin{aligned} r_i^{k+1} &= \sigma\left(\Sigma_{w_j \in \mathcal{N}(i)} \beta_j^k W_k^r r_j^k\right) \\ \beta_j^k &= \exp(v^k r_j^k) / \sum_{w_t \in \mathcal{N}(i)} \exp(v^k r_t^k) \\ v^k &= W_k^v [\pi_{m_1}^k, \pi_{m_2}^k] \end{aligned} \tag{6}$$

Representation Regularization: Finally, to provide an additional channel for the two GCN models to transfer their knowledge, we introduce a novel inductive bias that encourages the two GCN models to generate similar representation vectors for the input sentence for FineTempRel. As the two GCN models use the similar network architecture to learn the representations for the two related temporal subtasks of FineTempRel, we expect that the overall representation vectors for the same input sentence induced by these two GCN models should also be similar (although the representation vectors for the event mentions in the input sentence from each network might be more task-specific). To this end, we first obtain the overall representation vectors π^W and r^W for W based on the hidden vectors in the last layers of GCN^{dur} and GCN^{rel} (respectively) via: $\pi^W = max\_pool(\pi_1^G, \pi_2^G, \ldots, \pi_N^G)$

and $r^W = max\_pool(r_1^G, r_2^G, \ldots, r_N^G)$. Afterward, we enforce the similarity between these representation vectors for W by including their mean square error (MSE) $\mathcal{L}_{dif}$ into the overall loss function for minimization: $\mathcal{L}_{dif} = MSE(\pi^W, r^W)$.

On the one hand, $\mathcal{L}_{dif}$ serves as a regularizer to improve the representation vectors for the model. On the other hand, $\mathcal{L}_{dif}$ introduces a novel communication channel between the two GCN models so the knowledge from this GCN model can be transferred to improve the representation learning of the other GCN. Overall, our model performs deep and layer-wise interactions between the two GCN models for FineTempRel (i.e., the interactions are done for all the GCN layers), enabling the knowledge transfer to occur at different abstraction levels for potentially better representations for FineTempRel.

3.3 Prediction

In order to predict the durations for w_{m_1} and w_{m_2}, we assemble the overall representation vectors R_1^{dur} and R_2^{dur} for them based on hidden vectors in the last layer of GCN^{dur}: $R_1^{dur} = [\pi_{m_1}^G, \pi^W]$, $R_2^{dur} = [\pi_{m_2}^G, \pi^W]$. These overall vectors are then sent to a two-layer feed-forward network FF_1 to predict the durations $\hat{d}_1$ and $\hat{d}_2$ for w_{m_1} and w_{m_2}: $\hat{d}_1 = FF_1(R_1^{dur})$, $\hat{d}_2 = FF_1(R_2^{dur})$. Afterward, the loss function for EDP would be: $\mathcal{L}_{dur} = (d_1 - \hat{d}_1)^2 + (d_2 - \hat{d}_2)^2$.

Similarly, for TRP, we create an overall representation vector R^{rel} for this task by: $R^{rel} = [r_{m_1}^G, r_{m_2}^G, r^W]$. This vector is then consumed by another two-layer feed-forward network FF_2 to predict the start and end times $\hat{b}_1$, $\hat{e}_1$, $\hat{b}_2$ and $\hat{e}_2$ for w_{m_1} and w_{m_2}: $[\hat{b}_1, \hat{e}_1, \hat{b}_2, \hat{e}_2] = FF_2(R^{rel})$. Following (Vashishtha et al., 2019), we use the following loss function for temporal relation prediction: $\mathcal{L}_{rel} = |(b_1 - b_2) - (\hat{b}_1 - \hat{b}_2)| + |(e_1 - b_2) - (\hat{e}_1 - \hat{b}_2)| + |(e_2 - b_1) - (\hat{e}_2 - \hat{b}_1)| + |(e_1 - e_2) - (\hat{e}_1 - \hat{e}_2)|$.

To summarize, the overall loss function to train the model in this work is: $\mathcal{L} = \mathcal{L}_{dur} + \gamma_{rel}\mathcal{L}_{rel} + \gamma_{KL}\mathcal{L}_{KL} + \gamma_{dif}\mathcal{L}_{dif}$ where γ_{rel}, γ_{KL} and γ_{dif} are the trade-off parameters.

4 Experiments

• **Datasets & Parameters**: To evaluate the performance of the models, we use the Universal Decompositional Semantics Time (UDS-T) dataset introduced in (Vashishtha et al., 2019) for FineTempRel. UDS-T is annotated on top of the Universal Dependencies English Web Treebank (Bies et al., 2012) whose sentences are associated with the gold standard Universal Dependency parses (to be used for the dependency trees). There are 32,302 events and 70,368 event-event relations (i.e., examples) in UDS-T where the same data split for training data, development data and test data in (Vashishtha et al., 2019) is used to ensure a fair comparison. The development dataset of UDS-T is used to fine-tune the hyper-parameters for the models in this work. This fine-tuning process leads to the following values for the hyper-parameters in this work: 1 layer for the ON-LSTM model with $D = 256$ dimensions for the hidden vectors, 2 layers for the GCN models (also with 256 dimensions for the hidden vectors), 64 hidden units for the FF_1 and FF_2 networks for duration and temporal relation prediction, and $\gamma_{rel} = 0.5$, $\gamma_{KL} = 0.1$, and $\gamma_{dif} = 0.1$ for the trade-off parameters in the overall loss function. Also, we use the Adam optimizer with the learning rate of $1e$-5 to train the models in this work.

In addition, for the TE3 and TD datasets with the transfer learning experiment, similar to (Vashishtha et al., 2019), we employ the `sklearn 0.20.0` package to train the SVM classifier with Gaussian kernel and the same hyper-parameters. Finally, we use the same procedure as in (Vashishtha et al., 2019) to pre-process the datasets in this experiment (e.g., using the `Stanford CoreNLP 3.9.2` toolkit for data pre-processing).

In addition, similar to (Vashishtha et al., 2019), we further examine the models on the categorical TempRel datasets, including the TempEval3 (TE3, (UzZaman et al., 2013), Task C-relation only) and TimeBank-Dense (TD, (Cassidy et al., 2014)) datasets. We follow the same preprocessing procedure for these datasets and the same training and tuning procedures for the models as in (Vashishtha et al., 2019) to achieve a fair comparison. In particular, we use a transfer learning approach where the best-performing model on the UDS-T development set is first used to obtain the overall representation vector $R^{rel} = [r_{m_1}^G, r_{m_2}^G, r^W]$ for each pair of annotated event-event relations in TE3 and TD. Afterward, we employ this vector as the input features for a SVM classifier with a Gaussian kernel that is trained on the training sets and evaluated on the test sets for these datasets. Specifically, the training set for TE3 involves the union of the TimeBank (Pustejovsky et al., 2003) and AQUAINT (Graff, 2002) datasets provided in the TE3 workshop (UzZaman et al., 2013) while the training data and

Model	Duration			Timeline		
	ρ	rank diff	R1(rank diff)	Absolute ρ	Relative ρ	R1(MAE)
System1 (Vashishtha et al., 2019)	32.63	1.86	8.59	77.91	68.00	2.82
System2 (Vashishtha et al., 2019)	37.75	1.75	13.73	77.87	67.68	2.35
System3 (Vashishtha et al., 2019)	38.38	1.75	13.85	77.82	67.73	2.58
System4 (Vashishtha et al., 2019)	38.12	1.75	13.68	78.12	68.22	2.96
ON-LSTM-GCN (with ELMo)	**50.77**	**1.30**	**37.58**	**84.41**	**80.41**	**23.33**
ON-LSTM-GCN (with BERT)	**54.37**	**1.27**	**39.14**	**84.91**	**81.67**	**24.71**

Table 1: The performance on the UDS-T test set. Except for the rank diff for duration prediction, all metrics prefer large values.

development data in TD (i.e., TD-train, TD-dev) are combined to form the training set for TD. We follow the same hyper-parameter tuning process for the SVM classifier as in (Vashishtha et al., 2019) for these datasets where a grid-search with 4-fold cross-validation is done over the training sets. Finally, the performance of the models is reported on the test sets of the datasets (i.e., TE3-Platinum (TE3-PT) for TE3 and TD-test for TD).

For the performance measures for FineTempRel on the UDS-T dataset, following (Vashishtha et al., 2019), we use three metrics for the duration prediction task (i.e., Spearman correlation (ρ), mean rank difference (*RD*), and the proportion of rank difference explained (R1(*RD*))), and three metrics for temporal relation prediction (i.e., Spearman correlation between the normalized values of the actual start and end times and the predicted ones (*absolute* ρ), the Spearman correlation between the actual and predicted values for $\mathcal{L}_{rel}$ (*relative* ρ), and the proportion of mean absolute error (MAE) explained (R1(MAE))). Note that for a performance metric A, the proportion of A explained is computed by: R1(A) = 1 - $A_{model}/A_{baseline}$ where A_{model} is the performance metric A computed for the model and $A_{baseline}$ is those computed for the model that always guesses the median.

• **Comparison on UDS-T**: We compare the FineTempRel model in this work (called **ON-LSTM-GCN**) with the best-reported models on the UDS-T dataset in (Vashishtha et al., 2019). In particular, we use the top four models in (Vashishtha et al., 2019) (called System1, System2, System3, and System4) as the baselines in this work. Table 1 reports the performance of the models. Note that in addition to the ELMo embeddings as in (Vashishtha et al., 2019), we also show the performance of the ON-LSTM-GCN model when the BERT embeddings (Devlin et al., 2019) (i.e., the base model) are employed to encode the sentences. Both ELMo and BERT are fine-tuned during training in this work.

As we can see, using the same ELMo embeddings, the proposed model ON-LSTM-GCN significantly outperforms all the models in (Vashishtha et al., 2019) with substantial performance gap over different performance metrics and the two subtasks (i.e., event duration prediction and temporal relation prediction). This clearly demonstrates the effectiveness of the proposed model for FineTempRel. We also see that replacing ELMo with the BERT embeddings can help to improve the performance for both subtasks, suggesting the application of BERT for FineTempRel in the future research.

• **Comparison on TE3 and TD**: Table 2 shows the performance of the models on the TE3 and TD datasets. For both datasets, similar to (Vashishtha et al., 2019), we only evaluate on the Event-Event (E-E) relations as we only capture those in the model. Note that this is different from some of the prior works on TempRel where the performance metrics are reported for all relations (i.e., including timex-timex, and event-timex relations), making them not directly comparable to ours. For instance, (Ning et al., 2017) reports a F1-score of 0.672 for all relations on the TE3 test set, but it is not directly comparable to our model as we only evaluate on event-event relations. Also, as this is a transfer learning experiment from UDS-T, the most comparable baselines to ours in this case is from (Vashishtha et al., 2019) for both TE3 and TD. For a reference, we also include the best-reported performance on TD from the recent work (i.e., (Cheng and Miyao, 2017; Han et al., 2019a,b)). Note that we follow the previous work to report the temporal awareness scores (F1) for TE3 and the F1 micro-average scores for TD (Vashishtha et al., 2019).

In Table 2, we explicitly indicate the pre-trained word embeddings (i.e., ELMo or BERT) for the recent models on TempRel. The first observation from the table is that among the models with ELMo embeddings, the proposed model ON-LSTM-GCN is significantly better than the baseline models for

System	Evaluation Data	F1 (E-E)
(Vashishtha et al., 2019) (with ELMo)	TE3-PT	0.498
ON-LSTM-GCN (with ELMo)	TE3-PT	**0.551**
ON-LSTM-GCN (with BERT)	TE3-PT	**0.596**
CAEVO	TD-test	0.494
CATENA	TD-test	0.519
(Cheng and Miyao, 2017)	TD-test	0.529
(Vashishtha et al., 2019) (with ELMo)	TD-test	0.566
(Han et al., 2019a) (with BERT)	TD-test	0.632
(Han et al., 2019b) (with BERT)	TD-test	0.645
ON-LSTM-GCN (with ELMo)	TD-test	**0.620**
ON-LSTM-GCN (with BERT)	TD-test	**0.658**

Table 2: Test performance on TE3 and TD for our transfer learning experiment (only for the Event-Event relations).

	Duration			Timeline		
Model	ρ	RD	R1 (RD)	Abs ρ	Rel ρ	R1 (MAE)
ON-LSTM-GCN	**55.4**	**1.26**	**38.8**	**85.7**	**82.5**	**26.0**
- $\mathcal{L}_{KL}$	53.4	1.33	35.3	83.9	80.3	20.8
- ON-LSTM	41.1	1.50	27.2	80.2	75.7	15.7
repw/ LSTM	44.2	1.42	30.9	82.9	79.0	19.8
- GCN Interact	46.7	1.34	35.0	84.1	80.5	22.2
- $\mathcal{L}_{dif}$	51.6	1.31	36.2	84.6	81.4	23.6
- Interact - $\mathcal{L}_{dif}$	46.1	1.38	32.5	83.1	79.6	20.7
One GCN	45.0	1.43	30.6	82.2	78.1	19.3
- GCN	42.7	1.49	27.6	80.9	76.7	16.2

Table 3: Models' performance on the UDS-T development set. Except for the rank diff (RD) metric for duration prediction, all the metrics prefer large values.

both datasets TE3 and TD, including the previous best-reported model in (Vashishtha et al., 2019) with $p < 0.01$. Second, for the models with BERT embeddings, ON-LSTM-GCN also significantly outperforms the previous best-reported models on TD (i.e., (Han et al., 2019a,b)). In fact, ON-LSTM-GCN achieves the state-of-the-art performance with the BERT embeddings over both datasets, clearly testifying to the advantages of the proposed model for TempRel.

• **Model Analysis and Ablation Study**: This section investigates different variations of the two major components in ON-LSTM-GCN (i.e., ON-LSTM and GCN) to demonstrate their benefits.

ON-LSTM: First, we consider the following variations for the ON-LSTM component: (i) "**-** $\mathcal{L}_{KL}$": this is the ON-LSTM-GCN model where the syntax-model consistency loss $\mathcal{L}_{KL}$ is not included in the overall loss function, (ii) "**- ON-LSTM**": this model completely removes the ON-LSTM component from ON-LSTM-GCN (so the consistency loss $\mathcal{L}_{KL}$ is not used and the input vector sequence X is directly sent to the GCN models), and (iii) "**repw/ LSTM**": this model replaces ON-LSTM with the traditional LSTM model in ON-LSTM-GCN (the $\mathcal{L}_{KL}$ loss is also not employed in this case as LSTM does not support the neuron hierarchy for the model-based importance scores).

GCN: Second, for the GCN component, we evaluate the following variations for ON-LSTM-GCN: (i) "**- GCN Interact**": this model does not apply the interaction mechanism for the two GCN models in Equations 5 and 6 for ON-LSTM-GCN; it instead uses Equation 4 for the computation of both GCN models, (ii) "**-** $\mathcal{L}_{dif}$": this model eliminates the regularization loss $\mathcal{L}_{dif}$ from ON-LSTM-GCN, (iii) "**- Interact -** $\mathcal{L}_{dif}$": this model removes both the GCN interaction mechanism in Equations 5 and 6 and the loss $\mathcal{L}_{dif}$ from ON-LSTM-GCN (but it still has two GCN models), (iv) "**One GCN**": instead of using two GCN models for the two subtasks of FineTempRle, this model only utilize a single GCN model for both tasks (the GCN interaction in Equations 5 and 6 and the regularization loss $\mathcal{L}_{dif}$ are thus not used in this case as well), and (v) "**- GCN**": this model completely removes the GCN component (thus also excluding the GCN interaction and the loss $\mathcal{L}_{dif}$); the ON-LSTM hidden vectors H are directly exploited to perform the duration and relation predictions in this case. Table 3 presents the models' performance (using ELMo) on the UDS-T development set

There are several important observations from this table. First, regarding ON-LSTM, we see that both the ON-LSTM and syntax-model consistency loss $\mathcal{L}_{KL}$ are necessary for the proposed model as eliminating any of them or replacing ON-LSTM with LSTM would significantly hurt the performance. Second, for the GCN component, it is clear that the GCN interaction mechanism in Equations 5 and 6 and the regularization loss $\mathcal{L}_{dif}$ are crucial for ON-LSTM-GCN to achieve its highest performance, clearly verifying the benefits of knowledge transferring between duration and relation prediction for FineTempRel. In addition, the better performance of the full model and the "- Interact - $\mathcal{L}_{dif}$" model over "One GCN" demonstrates the necessity to employ different GCN models for the two subtasks in FineTempRel to enhance the representation customization capacity for the tasks. The model's performance becomes the worst when GCNs are completely excluded (i.e., "- GCN"), confirming the effectiveness of GCNs for FineTempRel.

5 Conclusion

We introduce a novel deep learning model for FineTempRel that exploits the syntactic structures

of the sentences to improve the representation vectors. We focus on two type of syntactic information for FineTempRel in this work, i.e., the syntax-based importance scores for the representations of the words that are injected into the models via ON-LSTM, and the dependency connections between the words that are exploited in GCN models. Two novel transferring learning methods are presented for the two GCN models for duration and relation predictions. Comprehensive experiments are performed to demonstrate the advantages of the proposed model for FineTempRel.

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 IU-CRC 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

Steven Bethard. 2013. ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English web treebank. In *Linguistic Data Consortium*.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. In *Transactions of the Association for Computational Linguistics (TACL)*.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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)*.

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Jennifer D'Souza and Vincent Ng. 2013. Classifying temporal relations with rich linguistic knowledge. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Elena Filatova and Eduard Hovy. 2001. Assigning time-stamps to event-clauses. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing*.

Tanya Goyal and Greg Durrett. 2019. Embedding time expressions for deep temporal ordering models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

David Graff. 2002. The aquaint corpus of english news text. In *Linguistic Data Consortium (LDC2002T31)*.

Andrey Gusev, Nathanael Chambers, Divye Raj Khilnani, Pranav Khaitan, Steven Bethard, and Dan Jurafsky. 2011. Using query patterns to learn the duration of events. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS)*.

Rujun Han, I-Hung Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, and Nanyun Peng. 2019b. Deep structured neural network for event temporal relation extraction. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019a. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yu Hong, Tongtao Zhang, Tim O'Gorman, Sharone Horowit-Hendler, Heng Ji, and Martha Palmer. 2016. Building a cross-document event-event relation corpus. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*.

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

Chen Lin, Dmitriy Dligach, Timothy A Miller, Steven Bethard, and Guergana K Savova. 2015. Multilayered temporal modeling for the clinical domain. In *Journal of the American Medical Informatics Association*.

Jian Liu, Yubo Chen, and Jun Zhao. 2020. Knowledge enhanced event causality identification with mention masking generalizations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*.

Jing Lu and Vincent Ng. 2017. Joint learning for event coreference resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jing Lu, Deepak Venugopal, Vibhav Gogate, and Vincent Ng. 2016. Joint inference for event coreference resolution. In *Transactions of the Association for Computational Linguistics (TACL)*.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yuanliang Meng and Anna Rumshisky. 2018. Context-aware neural model for temporal information extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Anne-Lyse Minard, Manuela Speranza, Ruben Urizar, Begoña Altuna, Marieke van Erp, Anneleen Schoen, and Chantal van Son. 2016. MEANTIME, the NewsReader multilingual event and time corpus. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Paramita Mirza and Sara Tonelli. 2016. CATENA: CAusal and TEmporal relation extraction from NAtural language texts. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

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, , Adam Meyers, and Ralph Grishman. 2016. New york university 2016 system for kbp event nugget: A deep learning approach. In *Text Analysis Conference*.

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)*.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qiang Ning, Hao Wu, and Dan Roth. 2018c. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Qiang Ning, Zhongzhi Yu, Chuchu Fan, and Dan Roth. 2018b. Exploiting partially annotated data in temporal relation extraction. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. 2007. Modeling and learning vague event durations for temporal reasoning. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and et al 2003. 2003. The timebank corpus. In *Corpus linguistics, volume 2003, page 40*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Julien Tourille, Olivier Ferret, Aurélie Névéol, and Xavier Tannier. 2017. Neural architecture for temporal relation extraction: A bi-LSTM approach for detecting narrative containers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hieu Minh Tran, Duy Phung, and Thien Huu Nguyen. 2021. Exploiting document structures and cluster consistencies for event coreference resolution. 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)*.

Minh Phu Tran and Thien Huu Nguyen. 2021. Graph convolutional networks for event causality identification with rich document-level structures. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020a. Exploiting the syntax-model consistency for neural relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020b. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*.

Jennifer Williams and Graham Katz. 2012. Extracting and modeling durations for habits and events from twitter. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.