Selecting Optimal Context Sentences for Event-Event Relation Extraction

Hieu Man[1], Nghia Trung Ngo[1], Linh Ngo Van[2], and Thien Huu Nguyen[3]

[1] VinAI Research, Vietnam
[2] Hanoi University of Science and Technology, Vietnam
[3]Department of Computer and Information Science, University of Oregon, USA
{v.hieumdt,v.nghiant66}@vinai.io, linhnv@soict.hust.edu.vn, thien@cs.uoregon.edu

Abstract

Understanding events entails recognizing the structural and temporal orders between event mentions to build event structures/graphs for input documents. To achieve this goal, our work addresses the problems of subevent relation extraction (SRE) and temporal event relation extraction (TRE) that aim to predict subevent and temporal relations between two given event mentions/triggers in texts. Recent state-of-the-art methods for such problems have employed transformer-based language models (e.g., BERT) to induce effective contextual representations for input event mention pairs. However, a major limitation of existing transformer-based models for SRE and TRE is that they can only encode input texts of limited length (i.e., up to 512 sub-tokens in BERT), thus unable to effectively capture important context sentences that are farther away in the documents. In this work, we introduce a novel method to better model document-level context with important context sentences for event-event relation extraction. Our method seeks to identify the most important context sentences for a given entity mention pair in a document and pack them into shorter documents to be consume entirely by transformer-based language models for representation learning. The REINFORCE algorithm is employed to train models where novel reward functions are presented to capture model performance, and context-based and knowledge-based similarity between sentences for our problem. Extensive experiments demonstrate the effectiveness of the proposed method with state-of-the-art performance on benchmark datasets.

Introduction

Understanding events is critical to natural language processing (NLP) due to their prevalence in texts. The major challenges to achieve this goal involve capturing multi-granular nature of events and their complex connections/relations (i.e., event structures) to deliver a coherent story for an input document (Wang et al. 2020). In information extraction (IE), these challenges are addressed in event-event relation extraction problems (EERE) that aims to recognize relations between pairs of event mentions/trigger words in an input document. In this work, we focus on two types of relations between events that provide important information to reveal event structures for documents, i.e., subevents and temporal orders. As such, given two event mentions in the same document, subevent relation extraction (SRE) seeks to determine if an event mention is a subevent (e.g., "*Parent-Child*") of the other while the goal of temporal relation extraction (TRE) is to identify the temporal order of the two event mentions (e.g., "*Before*", "*After*"). Based on such relations between events, an event graph/structure for each document can be obtained by using event mentions as the nodes and their subevent and temporal relations as the edges. In addition to the demonstration of event understanding, event structures find their applications in different downstream applications, including question answering, event prediction, timeline construction, and text summarization (Do, Lu, and Roth 2012; Chaturvedi, Peng, and Roth 2017; Han, Ning, and Peng 2019a).

The latest advances present transformer-based language models, e.g., BERT (Devlin et al. 2019), to encode input texts and deliver state-of-the-art performance for EERE problems (Ning, Subramanian, and Roth 2019; Han et al. 2019b; Wang et al. 2020; Ballesteros et al. 2020; Tran and Nguyen 2021). As the two event mentions of interest in EERE might appear in different sentences with long distances from each other, modeling document-level context for the event mentions is necessary for successful relation predictions. However, a critical issue with recent transformer-based language models for EERE is the limitation over the lengths of acceptable input texts. For instance, BERT can only encode input texts with up to 512 sub-tokens due to its quadratic self-attention complexity (Devlin et al. 2019). As such, given an input document, existing models for EERE have only constrained their operations to document context with a set of sentences that can fit into the length limits of transformer-based language models (Han, Ning, and Peng 2019a; Ballesteros et al. 2020; Wang et al. 2020). These models are thus unable to capture important context sentences for EERE that go beyond the length limit of the BERT-like models to boost the performance.

To address the length limit for transformer-based language models, prior methods for other NLP tasks have resorted to two major approaches. First, in Self-Attention Architecture Modification (Zaheer et al. 2020; Beltagy, Peters, and Cohan 2020; Kitaev, Kaiser, and Levskaya 2020), one can replace the vanilla self-attention of transformer networks with some variant architectures, e.g., sparse self-attention (Zaheer et al. 2020), that allow the modeling of larger document context while maintaining the same complexity as the original transformer. However, the transformer structures with variants of

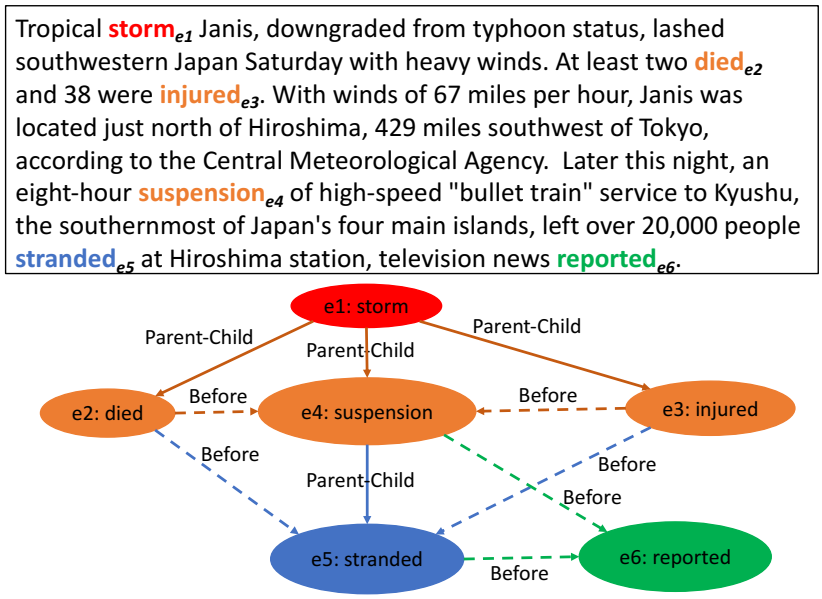

Figure 1: Event structure for an input document.

vanilla self-attention still suffer from a constraint of certain input length, thus failing to capture important context that are arbitrarily farther away from the event mention pair of interest in the input document for EERE. Moreover, the changes in the self-attention mechanism generally lead to poorer performance for NLP tasks compared to vanilla transformer (Beltagy, Peters, and Cohan 2020). The second approach involves Hierarchical Designs (Adhikari et al. 2019; Jörke et al. 2020) where the standard transformer-based language models are still leveraged to encode input texts with certain length limit. For larger input documents, another network architecture will be introduced to facilitate representation induction. For instance, (Adhikari et al. 2019) first splits an input document into multiple chunks that have shorter lengths than the limit of transformer-based models. Afterward, the BERT-based representation of each chunk is sent into a recurrent neural network for document modeling. However, in this case, the self-attention mechanism in the transformer-based language models cannot consume the entire input document to fully exploit its ability to capture long-range context dependencies in the whole document for improved representation learning. Above all, for both Self-Attention Architecture Modification and Hierarchical Designs, the transformer-based models are often used to encode a consecutive sequence of sentences in an input document without considering potential contribution of each sentence for the prediction tasks of interest. For EERE, this implies that irrelevant sentences for the relation prediction of event mentions might be included in the inputs for BERT-based models, potentially introducing noise into the representations and impairing the prediction performance.

To this end, to model document-level context for EERE, our intuition is to feed only on the important/relevant sentences in an input document into transformer-based language models to induce representation vectors for event relation prediction. As such, we propose to design models that can learn to select important context sentences for EERE to improve representation learning with BERT. On the one hand, sentence context selection helps compress an input document into a shorter one (with only important context) that can fit entirely into the length limit of transformer-based models to better leverage their representation learning capacity. Further, important sentences with arbitrary distances in the document can also be reached in this selection process to provide effective information for the predictions in EERE. Finally, context selection can avoid irrelevant sentences in the inputs for transformer-based models to reduce noise in the induced representations for EERE. In particular, starting with the host sentences of the two event mentions of interest, we will perform the sentence selection sequentially. The total length of the selected sentences will be constrained to not exceed the input limit in transformer-based models, thus allowing the entire consumption and encoding of the models for the selected context.

The major question in our models concerns the design of the sentence selection component to reveal important context for document-level representation learning with transformer-based models for EERE. To this end, our vision is to choose sentences in the input document that can be used to augment

the host sentences of the event mentions to improve the relation prediction performance of transformer-based models. As such, we propose to employ the performance of BERT-based models for EERE tasks as the reward to guide the important context selection. The policy-gradient method REINFORCE (Williams 1992) can thus be leveraged to facilitate the selection training. In addition, to enrich the selection reward, we introduce auxiliary rewards to capture the representational similarity between the context sentences and the event trigger host sentences. Our auxiliary rewards feature both contextual and background knowledge-based representations that focus on event information in the input documents to better serve EERE. Finally, our experiments for both subevent and temporal relation extractions demonstrate the state-of-the-art performance of the proposed method for EERE.

Model

Following (Wang et al. 2020), to build event structures with multi-faceted event-event relations for an input document, we focus on two tasks of EERE, i.e., subevent and temporal relation extraction. Given a pair of event mentions/triggers in the input document, both tasks aim to predict some relation between the two event mentions. For the subevent relation detection task, we follow the label set in prior work with four possible relations, i.e., `PARENT-CHILD`, `CHILD-PARENT`, `COREF`, and `NOREL` (Hovy et al. 2013; Glavaš et al. 2014). Based on the definition from (Hovy et al. 2013), an event c_2 is a child of the event c_1 if c_1 is a collector event with a sequence of activities where c_2 is one of the activity in the sequence and c_2 is spatially and temporally contained within c_1. For temporal event relation extraction (TRE), we use the label set with temporal relations/orders of `BEFORE`, `AFTER`, `EQUAL`, and `VAGUE` to be consistent with previous work (Ning, Feng, and Roth 2017; Ning, Subramanian, and Roth 2019; Han, Ning, and Peng 2019a; Wang et al. 2020).

Formally, let D be the input document with N sentences $S_1, S_2, \ldots, S_N$ (i.e., $D = [S_1, S_2, \ldots, S_N]$). In EERE, we are also given two event mention/trigger words e_1 and e_2 in D as the input for event relation prediction. For convenience, let S_i and S_j be the host sentences of e_1 and e_2 in D (respectively) with S_i as the earlier sentence, i.e., $i \leq j$. Here, i can be equal to j to indicate that the two event mentions e_1 and e_2 are presented in the same sentence.

To predict the relation between e_1 and e_2 in EERE, our model aims to induce effective representation vectors for the input document D. Transformer-based language models (e.g., BERT) will be leveraged to entirely consume important context sentences for e_1 and e_2 in D, thus fully exploiting their ability for representation learning. As such, our goal is to select a set of sentences C in D that contains the most important context for the relation prediction between e_1 and e_2, i.e., $C \subset S_{context} = \{S_k \in D | k \neq i, k \neq j\}$. The event host sentences S_i and S_j will be augmented with those in C to create a shorter document D' (i.e., $D' = \{S_i, S_j\} \cup C$) with important context information for e_1 and e_2 (i.e., the compressed document). In our approach, the number of words in D' is constrained to not exceed length limit of transformer-based language models, thus allowing the models to consume D' entirely to induce better representation vectors. Accordingly, any important context sentences for the relation prediction of e_1 and e_2 in D (i.e., including those that are far away from S_i and S_j) can be reached and packed into D' for effective BERT-based document encoding.

Event Relation Prediction Model

Given the selected important sentences in C, our event relation prediction model M^{EERE} first constructs the compressed document D' by concatenating the sentences in $\{S_i, S_j\} \cup C$. The order of the sentences in D' will follow their appearance order in D. For convenience, let $D' = w_1, w_2, \ldots, w_M$ be the concatenated word sequences with M words in D', i.e., $D' = w_1, w_2, \ldots, w_M$. Also, let i_1 and i_2 be the indexes of the event mentions/trigger words e_1 and e_2 (respectively) in D' (i.e., $e_1 = w_{i_1}$, $e_2 = w_{i_2}$). As the length of D' is constrained to follow the input limit of transformer-based language models, we can then send D' into a BERT-based language model to obtain representation vectors for the words w_i. In particular, following prior work (Wang et al. 2020), we employ the RoBERTa model (Liu et al. 2019) (with the limit of 512 sub-tokens for input texts) to encode D' in this work. Here, as each word $w_i \in D'$ might be split into multiple sub-tokens in RoBERTa, we use the vector v_i for the first sub-token of w_i in the last layer of RoBERTa as the representation vector for w_i. To this end, the compressed document D' is transformed into the vector sequence $V = [v_1, v_2, \ldots, v_M]$. Afterward, to perform event relation prediction, we form the overall representation vector $O = [v_{i_1}, v_{i_2}, max\_pool(v_1, v_2, \ldots, v_M)]$ to capture the document-level context information for e_1 and e_2 in D. Finally, O will be fed into a two-layer feed-forward network FF with softmax in the end to compute a distribution $P(\cdot|e_1, e_2, D) = FF(O)$ over the possible relations between e_1 and e_2 for our EERE problems. The negative log-likelihood $\mathcal{L}_{pred} = -\log P(y|e_1, e_2, D)$ will be utilized as the loss to train M^{EERE} in this work (y is the golden relation for e_1 and e_2).

Important Sentence Selection Model

The goal of this section is to select the most important context sentences C for the event relation prediction between e_1 and e_2 in D. As such, our major motivation is to directly rely on the prediction performance of M^{EERE} as the sentence selection guidance for C. In particular, a sentence $S_k \in D$ is considered to involve important context information for EERE if including S_k into the compressed document D' can lead to improved performance for the prediction of M^{EERE} over e_1 and e_2. To implement this idea, our model first seeks to obtain a representation vector x_k for each sentence $S_k \in S_{context}$ to facilitate the identification of important sentences. In our model, the representation of S_k is conditioned on the event host sentences S_i and S_j to achieve customization for the EERE problems. As such, S_k will be concatenated with S_i and S_j, following their order in D and using the token $[SEP]$ to separate sentences. The resulting sequence is then prepended with the special token $[CLS]$ and sent into RoBERTa. The vector for $[CLS]$ in the last layer will serve as the representation vector x_k for the

sentence S_k. In the next step, the representation vectors for the sentences in $S_{context}$, i.e., $X = \{x_k | S_k \in S_{context}\}$, will be employed by subsequent components to perform important sentence selection. Here, we note that the RoBERTa model in this selection component is different from those in the prediction model SM^{EERE} to allow the learning of task-specific information in our model.

Our sentence selection model follows an iterative process where a sentence in $S_{context}$ is chosen at each time step to be included in the sentence set C. In particular, C is empty at the beginning (step 0). At step $t+1$ ($t \geq 0$), given t sentences selected in previous steps, i.e., $C = \{S_{k_1}, S_{k_2}, \ldots, S_{k_t}\}$, we aim to choose a next sentence $S_{k_{t+1}}$ over the set of non-selected sentences $S^t_{context} = S_{context} \backslash \{S_{k_1}, S_{k_2}, \ldots, S_{k_t}\}$ to include into C. To summarize the selected sentences in prior steps, we run a Long Sort-Term Memory Network (LSTM) $LSTM$ over the representation vectors x_{k_i} of the selected sentences. The hidden vector h_t of $LSTM$ at step t will serve as the summarization vector of the previously selected sentences $S_{k_1}, S_{k_2}, \ldots, S_{k_t}$ (i.e., $h_0 = 0$ at the beginning). Afterward, the selection of $S_{k_{t+1}}$ at step $t+1$ will be conditioned on the selected sentences in prior steps via their sumarization vector h_t. In particular, for each non-selected sentence $S_u \in S^t_{context}$, a selection score sc^{t+1}_u is computed as a function of the representation vector x_u of S_u in X and the summarization vector h_t: $sc^{t+1}_u = sigmoid(G([x_u, h_t]))$ where G is a two-layer feed-forward network.

To this end, the sentence S_{u^*} with highest selection score, i.e., $S_{u^*} = \textit{argmax}_{S_u \in S^t_{context}} sc^{t+1}_u$, will be considered for selection at this step. In particular, if including S_{u^*} into C causes the the number of sub-tokens in the compressed document $D' = \{S_i, S_j\} \cup C$ to exceed the 512 length limit of RoBERTa (i.e., $|D'| = |S_i| + |S_j| + \sum_{q=1}^{t} |S_{k_q}| + |S_{u^*}| > 512$), the selection process will terminate and S_{u^*} will be discarded. Otherwise, the sentence selection will continue and S_{u^*} will be chosen for $S_{k_{t+1}}$ to be added into C. The representation vector x_{u^*} of S_{u^*} will then be fed into $LSTM$ to obtain the hidden vector h_{t+1} for the current step, i.e., $h_{t+1} = LSTM(h_t, x_{u^*})$, serving as the summarization vector for the next selection step. For convenience, we will consider C as the sequence of selected sentences from $S_{context}$ in the process, i.e., $C = S_{k_1} S_{k_2} \ldots S_{k_T}$ where T is the number of selected sentences.

Training Sentence Selection Model

To employ the relation prediction performance of M^{EERE} over e_1 and e_2 as the sentence selection guidance, we propose to utilize the REINFORCE algorithm (Williams 1992) that can treat the prediction performance as the reward function $R(C)$ for the selected sentence sequence C to train the selection processes for input documents. In addition, another benefit of REINFORCE involves its flexibility that facilitates the incorporation of different information sources from C to enrich the reward function $R(C)$ and provide more training signals for the selection model. As such, for EERE problems, we propose the following information sources to compute the reward function $R(C)$ for REINFORCE training:

• **Performance-based Reward** $R^{per}(C)$: We compute this reward via the relation prediction performance of the model M^{EERE} for the event mentions e_1 and e_2 in D. To condition on the selected sentence sequence C, M^{EERE} is applied on the compressed short document $D' = \{S_i, S_j\} \cup C$. As such, $R^{per}(C)$ is set to 1 if M^{EERE} correctly predict the relation between e_1 and e_2; and 0 otherwise.

• **Context-based Reward** $R^{context}(C)$: The motivation for this reward is that a sentence should be preferred to be included in C in the selection process if its contextual semantics is more similar to those for the event mentions e_1 and e_2 in the host sentences S_i and S_j (i.e., our target sentences). In particular, we expect that similar sentences to e_1 and e_2 are more likely to discuss the the same or related events (e.g., coreferring event mentions), thus providing more relevant contextual information to better understand the event mentions e_1 and e_2 and their relation in D. To this end, we propose to include the contextual similarity $R^{context}(C)$ between the given event mention pairs (e_1, e_2) and the selected sentence sequence C into the overall reward function $R(C)$ for enrichment. In particular, the representation vectors $h^{context}_e$ and $h^{context}_C$ for (e_1, e_2) and C are first computed by performing the max-pooling operation over the representation vectors of their corresponding event trigger words in V (i.e., obtained from the output of M^{EERE} over the compressed document D'): $h^{context}_e = max\_pool(v_{i_1}, v_{i_2})$ and $h^{context}_C = max\_pool(v_q | w_q \in C_{event})$ where C_{event} is the set of event mentions/trigger words presented in the sentences in C. Finally, the dot-product between $h^{context}_e$ and $h^{context}_C$ is used as the context-based reward $R^{context}(C)$ for our model: $R^{context}(C) = h^{context}_e h^{context}_C$.

• **Knowledge-based Reward** $R^{know}(C)$: This reward has the same motivation as the context-based reward $R^{context}(C)$ where similar sentences to e_1 and e_2 should be promoted for selection in C due to the potential to involve related events with helpful information for event-event relation prediction. However, instead of relying contextual semantics (i.e., via representation vectors) to obtain similarity measures as in $R^{context}(C)$, $R^{know}(C)$ seeks to exploit external knowledge resources to retrieve semantic word representations for the similarity-based reward (i.e., knowledge-based semantics). In particular, we propose to employ the commonsense knowledge graph ConceptNet (Speer, Chin, and Havasi 2017) to obtain the knowledge-based reward for our sentence selection model for EERE in this work. The graph structure in ConceptNet captures commonsense relations between concepts (including events such as *earthquake*, *tsunami*) from which some relations are directly related to our subevent and temporal extraction problems (Liu et al. 2020).

As such, to obtain representation vectors for words based on the commonsense knowledge in ConceptNet, we employ ConceptNet Numberbatch (CN) (Speer, Chin, and Havasi 2017), a set of embedding vectors for words that are trained over the commonsense connections between concepts in ConceptNet. Adjusted from Word2Vec and Glove, CN can encode commonsense knowledge/relation information into word embedding vectors to support knowledge-based similarity computation between words. For convenience, let $A = [a_1, a_2, \ldots, a_M]$ be the ConceptNet Num-

berbatch word embeddings for the words in the compressed document $D' = w_1, w_2, \ldots, w_M$ respectively. Here, if a word w_q does not have its corresponding embedding in CN, we simply set its vector a_q to zero. Based on such word embeddings, we then compute the representation vectors h_e^{know} and h_C^{know} for the input event pairs (e_1, e_2) and selected sequence C using the max-pooling operation as done with $R^{context}(C)$, i.e., $h_e^{know} = max\_pool(a_{i_1}, a_{i_2})$ and $h_C^{know} = max\_pool(a_q | w_q \in C_{event})$. Finally, the knowledge-based reward $R^{know}(C)$ will be computed via the similarity between the knowledge-based representation vectors h_e^{know} and h_C^{know}: $R^{know}(C) = h_e^{know} h_C^{know}$.

Consequently, the overall reward function $R(C)$ to train our context selection module with REINFORCE for EERE is: $R(C) = \alpha_{per} R^{per}(C) + \alpha_{context} R^{context}(C) + \alpha_{know} R^{know}(C)$ (α_{per}, $\alpha_{context}$, and α_{know} are trade-off parameters). Given the reward function, REINFORCE train the sentence selection model by minimizing the negative expected reward $R(C)$ over the possible choices of C: $\mathcal{L}_{select} = -\mathbb{E}_{C' \sim P(C'|e_1,e_2,D)}[R(C')]$. As such, the policy gradient can be estimated by: $\nabla \mathcal{L}_{select} = -\mathbb{E}_{C' \sim P(C'|e_1,e_2,D)}[(R(C') - b) \nabla \log P(C'|e_1, e_2, D)]$. Using one roll-out sample, we can further estimate $\nabla \mathcal{L}_{select}$ via the selected sequence C: $\nabla \mathcal{L}_{select} = -(R(C) - b) \nabla \log P(C|e_1, e_2, D)$ where b is the baseline to reduce variance. In our model, we obtain the baseline b via: $b = \frac{1}{|B|} \sum_{q=1}^{|B|} R(C^q)$, where $|B|$ is the mini-batch size and C^q is the selected sentence sequence for the q-th sample in the mini-batch. Finally, the probability of the selected sequence C is computed via: $P(C|e_1, e_2, D) = \prod_{t=0..T-1} P(S_{k_{t+1}}|e_1, e_2, D, S_{k \leq t})$ where $S_{k \leq t} = S_{k_1}, S_{k_2}, \ldots, S_{k_t}$ and $P(S_{k_{t+1}}|e_1, e_2, D, S_{k \leq t})$ is computed via the softmax function over the selection scores for the sentences in $S^t_{context}$ at selection step $t+1$: $P(S_{k_{t+1}}|e_1, e_2, D, S_{k \leq t}) = \exp(sc^{t+1}_{k_{t+1}}) / \sum_{S_u \in S^t_{context}} \exp(sc^{t+1}_u)$.

In this work, we train the relation prediction model M^{EERE} and the sentence selection component in an alternate training manner. At each update step with one batch of training data (i.e., one iteration), the current sentence selection component is used to choose the important sentence set C for each example (with an input document D and given event trigger words e_1 and e_2) in the batch, thus generating the compressed documents D'. The parameters for the relation prediction model M^{EERE} will then be updated using the gradient of $\mathcal{L}_{pred}$ over the compressed documents for the current batch. Afterward, the parameters of the selection component can be updated using the gradient of $\mathcal{L}_{select}$ in which the performance of the current prediction model M^{EERE} is employed to compute the reward function at this step.

Experiments

Datasets: For subevent relation extraction, we evaluate our models on the **HiEve** dataset (Glavaš et al. 2014) to make it consistent with prior work (Wang et al. 2020; Zhou et al. 2020). HiEve involves subevent and coreference relation annotation for events over 100 news articles. For temporal event relation extraction, we employ the popular dataset **MATRES** (Ning, Wu, and Roth 2018c) for model evaluation as in previous studies (Han et al. 2019b; Wang et al. 2020; Zhao, Lin, and Durrett 2021; Mathur et al. 2021). In particular, MATRES annotates 275 documents for four temporal relations, i.e., `BEFORE`, `AFTER`, `EQUAL`, and `VAGUE`. In addition, following recent work (Naik, Breitfeller, and Rose 2019; Mathur et al. 2021), we utilize the **TDDMan** and **TDDAuto** datasets in the TDDiscourse corpus (Naik, Breitfeller, and Rose 2019) to further evaluate the EERE models. TDDMan and TDDAuto are datasets for temporal event relation extraction on English articles that emphasize relations between event pairs with more than one sentence apart, thus making it critical to model global document-level context for successful predictions (Naik, Breitfeller, and Rose 2019).

Model	F1 score		
	PC	CP	Avg.
BigBird	65.6	53.7	59.6
Reformer	64.8	55.0	59.9
Longformer	65.7	54.5	60.1
Hierachical	63.7	57.1	60.4
Neighbor Sentences	66.8	58.9	62.8
Host Sentences-BERT	40.6	40.7	40.6
Host Sentences-RoBERTa	62.1	57.3	59.7
StructLR (Glavaš et al. 2014)	52.2	63.4	57.7
TACOLM (Zhou et al. 2020)	48.5	49.4	48.9
Joint Learning (Wang et al. 2020)	62.5	56.4	59.5
SCS-EERE (ours)	68.7	63.2	**65.9**

Table 1: Model performance on test data of HiEve for subevent relation extraction. We focus on the performance for `PARENT-CHILD` (PC), `CHILD-PARENT` (CP), and their micro-average to be consistent with prior state-of-the-art model (Wang et al. 2020).

For compatible comparison, we utilize the same data splits as in prior work for the considered datasets. In particular, for HiEve, we employ the split with 80 documents for training (with 35,001 event pairs) and 20 documents for testing (with 7,093 event pairs) as in (Wang et al. 2020). For MATRES, we apply the standard spit as in prior work (Han, Ning, and Peng 2019a; Ning, Subramanian, and Roth 2019; Wang et al. 2020), featuring 183/20 documents with 6332/827 event pairs for the training/test portions (respectively). MATRES also reserves 72 documents for development purpose (Han, Ning, and Peng 2019a; Wang et al. 2020). Finally, inherited from (Naik, Breitfeller, and Rose 2019; Mathur et al. 2021), our data splits involve 4000/650/1500 and 32609/1435/4258 event pairs in the training/development/test data for the TDDMan and TDDAuto datasets (respectively). We fine-tune the hyper-parameters in our model using the development set of the MATRES dataset. The selected values are applied for all datasets in this work.

Baselines: We compare our model for important sentence context selection for EERE (called SCS-EERE) with other variants of transformer-based language models that can encode input texts with longer length than the limit in RoBERTa. In particular, we consider three popular language models of

Model	P	R	F1
BigBird	74.4	84.4	79.1
Reformer	75.1	84.3	79.4
Longformer	76.2	83.8	79.8
Hierachical	74.2	83.1	78.4
Neighbor Sentences	74.7	86.5	80.2
Host Sentences-BERT	77.3	79.0	78.1
Host Sentences-RoBERTa	76.8	80.0	78.4
SP+ILP (Ning et al. 2017)	71.3	82.1	76.3
BiLSTM (Cheng and Miyao 2017)	59.5	59.5	59.5
CogCompTime (Ning et al. 2018b)	61.6	72.5	66.6
Perceptron (Ning et al. 2018c)	66.0	72.3	69.0
(Goyal and Durrett 2019)	-	-	68.6
BiLSTM+MAP (Han et al. 2019a)	-	-	75.5
CSE+ILP (Ning et al. 2019)	71.3	82.1	76.3
Joint Learning (Wang et al. 2020)	73.4	85.0	78.8
DEER (Han, Ren, and Peng 2020)	-	-	79.3
(Zhao, Lin, and Durrett 2021)	75.1	84.8	79.6
SMTL (Ballesteros et al. 2020)	-	-	81.6
TIMERS (Mathur et al. 2021)	81.1	84.6	82.3
SCS-EERE (ours)	78.8	88.5	**83.4**

Table 2: Model performance on test data of MATRES for temporal event relation extraction.

Model	TDD Man	TDD Auto
BigBird	43.3	65.3
Reformer	43.7	65.9
Longformer	44.2	66.8
Hierachical	42.3	64.9
Neighbor Sentences	44.7	67.1
Host Sentences-BERT	37.5	62.3
Host Sentences-RoBERTa	37.1	61.6
SP+ILP (Ning et al. 2017)	23.8	46.1
BiLSTM (Cheng and Miyao 2017)	24.3	51.8
BiLSTM+MAP (Han et al. 2019a)	41.1	57.1
Deep SSVM (Han et al. 2019b)	41.0	58.8
UCGraph+BERT (Liu et al. 2021)	43.4	61.2
TIMERS (Mathur et al. 2021)	45.5	71.1
SCS-EERE (ours)	**51.1**	**76.7**

Table 3: Model performance (F1) on test data of TDDMan and TDDAuto.

this type: **BigBird** (Zaheer et al. 2020) (using sparse self-attention), **Reformer** (Kitaev, Kaiser, and Levskaya 2020) (using locality-sensitive hashing to replace dot-product attention), and **Longformer** (Beltagy, Peters, and Cohan 2020) (using local self-attention with global task-aware attention). In addition, we also consider two typical approaches for document-context modeling with transformer-based language models as the baselines: **Hierarchical** (Adhikari et al. 2019) that splits a document into multiple chunks and encode them separately with RoBERTa. A BiLSTM model is then employed to aggregate the representations of the chunks to compute document representations; and **Neighbor Sentences** that augments the event host sentences (i.e., S_i and S_j) with the sentences immediately preceding and following the first and second input event triggers in the documents for RoBERTa. Note that for BigBird, Reformer, Longformer, and Neighbor Sentences, the sentences between the event host sentences will be first included into the input context for the models. The remaining quotas for input length of the models will then be distributed evenly for the preceding and following context of the event host sentences. Further, we include the baselines that only use the event host sentences S_i and S_j (i.e., via concatenation if $i \neq j$) as the input for the transformer-based models to learn representation vectors (called "Host Sentences"). Following (Mathur et al. 2021), we examine both RoBERTa (Liu et al. 2019) and BERT (Devlin et al. 2019) for these baselines (leading to **Host Sentences-RoBERTa** and **Host Sentences-BERT**) to serve as strong baselines. Finally, for each considered dataset, we also report the results of previous work that has reported their model performance on the datasets. In particular, the state-of-the-art performance on HiEve is due to the joint constrained learning method in (Wang et al. 2020) while the best reported performance for MATRES, TDDMan, and TDDAuto are recently achieved in the TIMERS system (Mathur et al. 2021).

Comparison: Tables 1, 2, and 3 show the performance of the models (F1 scores) on the HiEve, MATRES, TDDMan, and TDDAuto datasets. Here, the performance for the models in previous work (i.e., those accompanied with citations) is inherited from the original papers. An observation from the tables is that methods with document-level context modeling (i.e., BigBird, Reformer, Longformer, Hierarchical, Neighbor Sentences, SCS-EERE) tend to perform better than those with only event host sentence encoding (i.e., Host Sentence-RoBERTa) across different tasks and datasets. The performance gap between these models are larger in TDDMan and TDDAuto as they involve more sentences between the event host sentences S_i and S_j than other datasets. In all, such performance differences clearly demonstrate the benefits of capturing document-level context for EERE. In addition, among such document-level baseline models, the proposed method SCS-EERE achieves significantly better performance (i.e., $p < 0.01$) with substantial performance gap, thus highlighting the effectiveness of learning to select important context sentences in SCS-EERE. Finally, SCS-EERE significantly outperforms previous models (with $p < 0.01$) over different tasks and datasets, leading to the state-of-the-art performance on those datasets.

Ablation Study: In this section, we aim to ablate the major components in the SCS-EERE model and evaluate the performance of the remaining model to understand the components' contribution. In particular, we consider the following ablated models for SCS-EERE: (1) "**- Multiple-step Selection**": In our context selection component, multiple sentence selection steps are performed where the hidden vector of the $LSTM$ network at each step is treated as a summarization for previously selected sentences, providing a condition for sentence selection in the next step for C. To assess the necessity of the multi-step selection with LSTM, we instead examine a one-step selection strategy. In particular, we only

Model	MATRES	HiEve
SCS-EERE (full)	83.4	65.9
-Multiple-step Selection	80.4	60.4
-Performance-based Reward	80.9	63.4
-Context-based Reward	81.0	64.2
-Knowledge-based Reward	81.2	62.5
Most Context-based Similar	81.2	63.5
Most Knowledge-based Similar	80.8	62.3

Table 4: Performance (F1 scores) of the ablated models. The results on HiEve are the micro-average of PARENT-CHILD and CHILD-PARENT.

perform the sentence selection once where the the top Q sentences with highest selection scores from the first step (i.e., sc_u^1 for $S_u \in S_{context}$) are selected to form the context set C (thus eliminating LSTM). Here, Q is chosen such that the resulting compressed document D' can occupy the input length limit of RoBERTa as much as possible; and (2) "**- Performance-based Reward**", "**- Context-based Reward**", and "**- Knowledge-based Reward**": These models exclude the reward components $R^{per}(C)$, $R^{context}(C)$, and $R^{knowledge}(C)$ (respectively) from the overall reward $R(C)$ for sentence selection to study their effectiveness.

To further demonstrate the benefit of learning to select important sentences for EERE, we evaluate the typical heuristics to choose context sentences for the target event mentions e_1 and e_2 in a document D. The key property of such heuristics is that they directly suggest a set of context sentences C given the input event mentions, thus not involving any training step. The event host sentences S_i and S_j will then be concatenated with the suggested context sentence following their order in D and length limit in RoBERTa to produce D' for RoBERTa. As such, we explore the following suggestion heuristics for evaluation: (1) **Most Context-based Similar**: This baseline selects the top sentences $S_u \in S_{context}$ that bear the highest contextual similarity with the input event mentions e_1 and e_2. In particular, motivated by the context-based reward $R^{context}(C)$, for each sentence $S_u \in S_{context}$, we feed it into the RoBERTa model of M^{EERE} and the hidden vector of the $[CLS]$ token in the last layer is used as the representation vector for S_u. Afterward, the cosine similarity between S_u's representation vector and the input event mention representation $h_e^{context}$ will serve as the contextual similarity for sentence selection in this baseline; and (2) **Most Knowledge-based Similar**: This baseline also choose the top sentences in $S_{context}$ with highest similarity to e_1 and e_2; however, the representation vectors for similarity will be computed via the knowledge-based representations as in the reward $R^{know}(C)$ (i.e., h_e^{know} for the event mentions). In particular, the representation for each sentence in $S_u \in S_{context}$ will be based on the max-pooled vector of the ConceptNet Numberbatch embeddings for the event triggers in S_u.

Table 4 reports the performance of the ablated models on the test data of MATRES and HiEve. As can be seen, removing multi-step selection or any reward component (i.e., performance-, context-, and knowledge-based) significantly hurts the overall performance, thus clearly demonstrating their importance for sentence selection in SCS-EERE. The largest performance drop is due to the elimination of multi-step selection, suggesting that selecting sentences incrementally and conditioning on previously selected sentences are critical to document-context modeling for EERE. In addition, compared to heuristics-based selection methods (i.e., Most Context-based Similar and Most Knowledge-based Similar), the superior performance of SCS-EERE clearly highlights the advantage to learn to select context sentences with REINFORCE for EERE.

Related Work

Early methods for event temporal relation extraction have been rule-based where syntax, knowledge databases, regular expressions are leveraged to design temporal rules (Hagège and Tannier 2007; Strötgen and Gertz 2010; Llorens, Saquete, and Navarro 2010). Afterward, machine learning models have been applied to both event temporal relation extraction (Mani et al. 2006; Ning, Feng, and Roth 2017; Leeuwenberg and Moens 2017; Ning et al. 2018b; Tran, Nguyen, and Nguyen 2021) and subevent relation extraction (Glavaš et al. 2014; Araki et al. 2014; Aldawsari and Finlayson 2019) to exploit various contextual features for input texts (i.e., feature engineering). To alleviate feature engineering, recent works have explored deep learning models to induce representations for EERE from data, i.e., representation learning for TRE (Dligach et al. 2017; Tourille et al. 2017; Cheng and Miyao 2017; Meng, Rumshisky, and Romanov 2017) and SRE (Nguyen, Meyers, and Grishman 2016; Zhou et al. 2020; Tran, Phung, and Nguyen 2021). Approaches to improve deep learning models for EERE in prior work include joint inferring events and temporal relations (Han et al. 2019b; Han, Ning, and Peng 2019a) and leveraging transformer-based language models for input texts (Ning, Subramanian, and Roth 2019; Ross, Cai, and Min 2020; Wang et al. 2020; Phung, Nguyen, and Nguyen 2021; Phung et al. 2021). However, none of existing work explores context sentence selection to effectively encode long input documents with the limited input length of transformer-based language models for EERE as we do in this work.

Conclusion

We present a novel model for event-event relation extraction that learns to select the most important context sentences in a document and directly use them to induce representation vectors with transformer-based language models. Relevant context sentences are selected sequentially in our model that is conditioned on the summarization vector for the previously selected sentences in the sequence. We propose three novel reward functions to train our model with REINFORCE. Our extensive experiments show that the proposed model can select important context sentences that are far away from the given event mentions and achieve state-of-the-art performance for subevent and temporal event relation extraction. In the future, we plan to extend our proposed method to other related tasks in event structure understanding (e.g., for joint event and event-event relation extraction).

Acknowledgements

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112. 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.

References

Adhikari, A.; Ram, A.; Tang, R.; and Lin, J. 2019. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.

Aldawsari, M.; and Finlayson, M. 2019. Detecting Subevents using Discourse and Narrative Features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Araki, J.; Liu, Z.; Hovy, E.; and Mitamura, T. 2014. Detecting Subevent Structure for Event Coreference Resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*.

Ballesteros, M.; Anubhai, R.; Wang, S.; Pourdamghani, N.; Vyas, Y.; Ma, J.; Bhatia, P.; McKeown, K.; and Al-Onaizan, Y. 2020. Severing the Edge Between Before and After: Neural Architectures for Temporal Ordering of Events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Chaturvedi, S.; Peng, H.; and Roth, D. 2017. Story Comprehension for Predicting What Happens Next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Cheng, F.; and Miyao, Y. 2017. Classifying Temporal Relations by Bidirectional LSTM over Dependency Paths. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Dligach, D.; Miller, T.; Lin, C.; Bethard, S.; and Savova, G. 2017. Neural Temporal Relation Extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Do, Q.; Lu, W.; and Roth, D. 2012. Joint Inference for Event Timeline Construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Glavaš, G.; Šnajder, J.; Moens, M.-F.; and Kordjamshidi, P. 2014. HiEve: A Corpus for Extracting Event Hierarchies from News Stories. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*.

Goyal, T.; and Durrett, G. 2019. Embedding Time Expressions for Deep Temporal Ordering Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hagège, C.; and Tannier, X. 2007. XRCE-T: XIP Temporal Module for TempEval campaign. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*.

Han, R.; Hsu, I.-H.; Yang, M.; Galstyan, A.; Weischedel, R.; and Peng, N. 2019b. Deep Structured Neural Network for Event Temporal Relation Extraction. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*.

Han, R.; Ning, Q.; and Peng, N. 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)*.

Han, R.; Ren, X.; and Peng, N. 2020. DEER: A Data Efficient Language Model for Event Temporal Reasoning. *CoRR*, abs/2012.15283.

Hovy, E.; Mitamura, T.; Verdejo, F.; Araki, J.; and Philpot, A. 2013. Events are Not Simple: Identity, Non-Identity, and Quasi-Identity. In *Workshop on Events: Definition, Detection, Coreference, and Representation*.

Jörke, M.; Gillick, J.; Sims, M.; and Bamman, D. 2020. Attending to Long-Distance Document Context for Sequence Labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Leeuwenberg, A.; and Moens, M.-F. 2017. Structured Learning for Temporal Relation Extraction from Clinical Records. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Liu, J.; Liu, J.; Chen, Y.; and Zhao, J. 2020. Knowledge Enhanced Event Causality Identification with Mention Masking Generalizations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Liu, J.; Xu, J.; Chen, Y.; and Zhang, Y. 2021. Discourse-Level Event Temporal Ordering with Uncertainty-Guided Graph Completion. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692.

Llorens, H.; Saquete, E.; and Navarro, B. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Mani, I.; Verhagen, M.; Wellner, B.; Lee, C. M.; and Pustejovsky, J. 2006. Machine Learning of Temporal Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL-COLING)*.

Mathur, P.; Jain, R.; Dernoncourt, F.; Morariu, V.; Tran, Q. H.; and Manocha, D. 2021. TIMERS: Document-level Temporal Relation Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL) and the 11th International Joint Conference on Natural Language Processing (IJCNLP)*.

Meng, Y.; Rumshisky, A.; and Romanov, A. 2017. Temporal Information Extraction for Question Answering Using Syntactic Dependencies in an LSTM-based Architecture. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Naik, A.; Breitfeller, L.; and Rose, C. 2019. TDDiscourse: A Dataset for Discourse-Level Temporal Ordering of Events. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*.

Nguyen, T. H.; Meyers, A.; and Grishman, R. 2016. New York University 2016 System for KBP Event Nugget: A Deep Learning Approach. In *Proceedings of the Text Analysis Conference (TAC)*.

Ning, Q.; Feng, Z.; and Roth, D. 2017. A Structured Learning Approach to Temporal Relation Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ning, Q.; Subramanian, S.; and Roth, D. 2019. An Improved Neural Baseline for Temporal Relation Extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Ning, Q.; Wu, H.; and Roth, D. 2018c. A Multi-Axis Annotation Scheme for Event Temporal Relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ning, Q.; Zhou, B.; Feng, Z.; Peng, H.; and Roth, D. 2018b. CogCompTime: A Tool for Understanding Time in Natural Language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*.

Phung, D.; Minh Tran, H.; Nguyen, M. V.; and Nguyen, T. H. 2021. Learning Cross-lingual Representations for Event Coreference Resolution with Multi-view Alignment and Optimal Transport. In *Proceedings of the 1st Workshop on Multilingual Representation Learning (MRL)*.

Phung, D.; Nguyen, T. N.; and Nguyen, T. H. 2021. Hierarchical Graph Convolutional Networks for Jointly Resolving Cross-document Coreference of Entity and Event Mentions. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*.

Ross, H.; Cai, J.; and Min, B. 2020. Exploring Contextualized Neural Language Models for Temporal Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*.

Strötgen, J.; and Gertz, M. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Tourille, J.; Ferret, O.; Névéol, A.; and Tannier, X. 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)*.

Tran, H. M.; Phung, D.; and Nguyen, T. H. 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)*.

Tran, M. P.; Nguyen, M. V.; and Nguyen, T. H. 2021. Fine-grained Temporal Relation Extraction with Ordered-Neuron LSTM and Graph Convolutional Networks. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*.

Tran, M. P.; and Nguyen, T. H. 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)*.

Wang, H.; Chen, M.; Zhang, H.; and Roth, D. 2020. Joint Constrained Learning for Event-Event Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. In *Kluwer Academic*.

Zaheer, M.; Guruganesh, G.; Dubey, A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Zhao, X.; Lin, S.-T.; and Durrett, G. 2021. Effective Distant Supervision for Temporal Relation Extraction. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*.

Zhou, B.; Ning, Q.; Khashabi, D.; and Roth, D. 2020. Temporal Common Sense Acquisition with Minimal Supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.