International Journal on Digital Libraries
https://doi.org/10.1007/s00799-021-00310-1

CNN-based framework for classifying temporal relations with question encoder

Yohei Seki[1] · Kangkang Zhao[1] · Masaki Oguni[1] · Kazunari Sugiyama[2]

Received: 1 May 2021 / Revised: 6 September 2021 / Accepted: 6 September 2021

Abstract
Temporal-relation classification plays an important role in the field of natural language processing. Various deep learning-based classifiers, which can generate better models using sentence embedding, have been proposed to address this challenging task. These approaches, however, do not work well due to the lack of task-related information. To overcome this problem, we propose a novel framework that incorporates prior information by employing awareness of events and time expressions (time–event entities) with various window sizes to focus on context words around the entities as a filter. We refer to this module as "question encoder." In our approach, this kind of prior information can extract task-related information from simple sentence embedding. Our experimental results on a publicly available *Timebank-Dense* corpus demonstrate that our approach outperforms some state-of-the-art techniques, including CNN-, LSTM-, and BERT-based temporal relation classifiers.

Keywords Temporal-relation classification · Neural networks · Event and time expressions · Question encoder · Timebank

1 Introduction

With the rapid development of information technology, the number of distributed news services on the Internet has been growing exponentially. Thus, quickly searching for news relevant to each user's interests is becoming more and more difficult. To address this problem, temporal-relation classification is a promising approach to constructing timelines that allows a search engine to provide much more relevant results and tips for users [9]. In addition, to achieve better fact-checking, the informative events from news media should be arranged chronologically. For example, in the 2020 U.S. Presidential Election, Mr. Trump's lawyers claimed that the election was fraudulent. Afterward, the office of Pennsylvania's attorney general has said that there is no evidence to support the claims.

By contrast, as COVID-19 spreads worldwide, a vast number of clinical papers have been published online [33], and informative events from clinical papers should be arranged chronologically to monitor the effect of various treatments [28]. To extract and collect the most informative parts from clinical papers, effective information extraction [30] is an essential technique. To observe disease progression and some longitudinal effects of medications [14], it is also important to improve the accuracy of temporal-relation classification for the extracted time and event entities.

Temporal-relation classification aims to identify the relations (e.g., "BEFORE," "OVERLAP," and "AFTER,") between event and time expressions (i.e., time–event entities). For example, the following sentence is an example of the "BEFORE" relation between events "**established**" and "**believed**":

Example 1 He said he ***believed*** the "conditions for a meeting" between Mr. Trump and Mr. Rouhani "in the next few weeks" had been ***established***.

In recent years, several feature-based methods have been proposed to address temporal-relation classification [1,12, 21,22]. However, most of them rely on the manual annotation of features and rules, and this is very time-consuming and labor-intensive. Following the recent success of neural networks (NNs), various NN-based models, including convolutional NN (CNN) [4,5,15], recurrent NN (RNN) [2,29,32], and contextual embedding [16] using Bidirectional Encoder Representations from Transformers (BERT) [3] have been proposed to achieve better performance with less manual work in temporal-relation classification. Other researchers have proposed related works [26,31] for general relation

✉ Yohei Seki
yohei@slis.tsukuba.ac.jp

[1] University of Tsukuba, Kasuga, Tsukuba 305-8550, Japan

[2] Kyoto University, Kyoto 606-8501, Japan

Published online: 13 October 2021

classification using BERT, but they did not focus on temporal-relation classification.

In many NN-based models proposed thus far, the classification module generates labels from sentence embedding without any prior information. However, lacking prior information causes some problems. For example, the decoder generates irrelevant labels even with well-trained sentence embedding because classifiers cannot choose a necessary feature among dense features for specific tasks.

Researchers have encoded input sentences into high-dimensional vectors that contain the semantic information required for classification. For example, as a variant of RNN, long short-term memory (LSTM) can automatically choose what to remember or forget when modeling long sequences using four specially designed "gate" structures (input modulation gate, input gate, forget gate, and output gate) [8]. By contrast, CNNs manipulate word tokens sequentially using sliding windows, resulting in a loss of long dependency information, but they extract local semantic features from pretrained word embeddings with convolutional filters [6]. For temporal-relation classification tasks, we assume that we do not need to capture semantic information completely because the clues of temporal-relation classification tend to appear locally around the time–event entities.

We first published our work in [25] and proposed a novel framework to classify temporal relations with a "question encoder" using the context of time–event entities as prior information. In this paper, we explore a new CNN-based framework and update our method with the detailed motivation to elaborate our question encoder module. We also add the new baselines with the CNN-based method [4] and the methods [26,31] using BERT [3]. Our contributions are summarized as follows.

1. We update our extractor module named "question encoder" to incorporate various window sizes to focus on context words around the entities. This module is used to extract the required information from sentence embeddings for classification using expressions for time and an event. In contrast, we update our sentence encoder to keep simple using convolutional layers with a filter window size of 3. We also update our optimizer using the learning rate optimizer AdamW [18]: Adam with weighted decay, which was proposed recently instead of Adam [11].
2. We conduct comparative experiments on the *Timebank-Dense* corpus [1]. We update our baselines including the CNN-based method [4] and the methods [26,31] using BERT [3]. We also discuss the effectiveness of the question encoder module. Experimental results show that our question encoder module can significantly improve the performance, not only with CNNs but also with LSTMs. We also describe how to set the hyperparameters in our model.
3. To solve the problem with the lack of training data, we expand some of the *Timebank-Dense* dataset by including the reversed examples, and this significantly improves the performance, especially for a smaller number of training samples. Finally, our proposed model with expanded training data demonstrates significantly improved performance to 0.699 and 0.732 in Macro and Micro F1 scores, respectively.

2 Related work

2.1 Feature-based methods for temporal-relation classification

In earlier works, traditional feature-based machine learning approaches have achieved acceptable performance in temporal-relation classification. For example, Chambers et al. [1] used a maximum entropy classifier with the lexical features such as token, lemma, POS tag of event, tense, aspect of an event, or syntactic features such as syntactic parse tree path between the event and time. Mirza and Tonelli [22] employed L2-regularized logistic regression to classify temporal relations by incorporating word-embedding features, demonstrating its effectiveness.

However, to make machine learning algorithms work much better, the unstructured texts need to be converted into numeric representations that can be understood by the algorithms. In this framework, laborious feature engineering is required.

In addition, challenges remain, and human-annotated features do not guarantee acceptable performance due to the impact of errors from the subjective judgment in the process.

2.2 Bidirectional LSTM-based methods for temporal-relation classification

In temporal-relation classification, the shortcomings described in the previous subsection indicate that the traditional approaches do not work well, motivating researchers to employ NNs, which can automatically extract effective features, instead of complicated feature engineering.

At the same time, LSTM-based models can learn the rules automatically and also achieve higher performance by simply giving more input data. For example, Cheng and Miyao [2] employed LSTM in a bidirectional form (Bi-LSTM) by taking dependency paths as the input, resulting in better temporal-relation classification. To enhance their work, Zhang et al. [32] proposed using deep Bi-LSTM, demonstrating that the deep neural approach can learn representations more semantically.

On the other hand, Liu et al. [17] leveraged an attention mechanism [19] to improve the system performance of neu-

ral network models. Word-level attention weights could be interpreted as importance measures in given contexts, i.e., temporal-relation indicators for each relation instance of a sentence. They implemented the attention mechanism on top of the LSTM or GRU model.

2.3 CNN-based methods for temporal-relation classification

By contrast, several researchers have proposed temporal-relation classification methods using CNNs [4,5]. Do and Jeong [5] proposed a CNN architecture for temporal-relation classification. They used lexical features for window processing and contextual features for convolution and max-pooling operations. Their approach, however, did not outperform state-of-the-art methods. Dligach et al. [4] found that CNN models outperform LSTM models for temporal-relation extraction tasks, although their dataset differs from ours introduced in Sect. 4.1. In their CNN-based model, the embedding layer was followed by a convolution layer that applied convolving filters of various sizes to extract n-gram-like features that were then pooled by a max-pooling layer. The output of the max-pooling layer was fed into a fully connected dense layer that was followed by the final softmax layer outputting the probability distribution over the possible classes for the input. The filter sizes of CNN models in [4] were 2, 3, 4, and 5.

From these related works, we design our framework to focus on context words around the entities on top of the CNN model and weight them with the matrix–matrix product of input sentence embeddings and question embeddings, which is similar to the dot-product attention mechanism [19].

2.4 BERT-based methods for temporal-relation classification

Finally, some systems [16,26,31] have been proposed based on contextualized word representations called BERT [3]. Note that two systems [26,31] are developed for general relation classification as SemEval-2010 Task 8 [7], and they did not focus on temporal-relation classification. Lin's system [16] focused on "contains" relation and "contains by" relation only for classification, which was similar to the "overlap" relation. Wu and He's approach [31] was based on the concatenation of a [CLS] token[1] vector and two averaged entity vectors for relationship. Soares et al. [26] and Lin et al. [16] focused on hidden states of BERT for the symbol at the starting and ending positions of entities called entity markers.

Building on previous work for temporal-relation classification, we take the CNN model as our base system with the question encoder and compare our proposed model with some state-of-the-art systems for temporal-relation classification including BERT-based ones in Sect. 4.3.

[1] [CLS] stands for classification. It is added at the beginning to represent the meaning of the entire sentence.

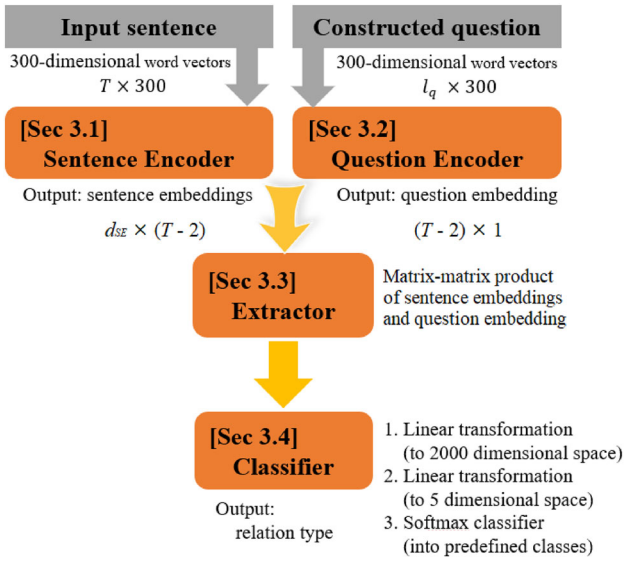

Fig. 1 Architecture of our proposed framework

3 Our proposed model

In this section, we propose a CNN-based framework for temporal-relation classification. As shown in Fig. 1, our CNN-based framework consists of the following four main components: sentence encoder, question encoder, extractor, and classifier (see Sects. 3.1, 3.1, 3.3 and 3.4, respectively). The sentence encoder encodes sentence information using CNN, and the question encoder encodes the context information around the entities. This information is combined in the extractor module with matrix–matrix product operation, which is similar to the dot-product operation in the attention mechanism [19].

3.1 Sentence encoder

The sentence encoder module encodes input sentences into high-dimensional embedding. Here, we employ a variant of CNN [10] to encode input sentences, demonstrating its ability to encode the semantic features of sentences.

Given a sentence $S = \{x_1, x_2, \ldots, x_T\}$ where T denotes the length of the sentence, the objective of the word-embedding layer is to map each word x_t into a high-dimensional vector e_t. Global Vectors for Word Representation (GloVe) capture sublinear relationships of words in the vector space [24]. GloVe word vectors are commonly used as pretrained word representations trained by the unsupervised learning algorithm. In general, GloVe outperforms the

word2vec [20] algorithm in the word analogy tasks. Hence, in this work, we initialize the word embeddings with the publicly available 300-dimensional GloVe [24] word vectors. Then, we use the convolutional layers to obtain the feature map from word embeddings. Afterward, we apply a max-pooling algorithm to identify the most important feature in each feature map by taking the maximum value as the feature for filters.

Hyperparameter settings

We use convolutional layers with a filter window size of 3 because the smaller size is popular and because the odd window size will be symmetric around the word.[2] In the standard CNN approach, various window sizes were incorporated to allow the network to capture wider ranges of n-grams [4,23]. In our approach, however, we suppose that the sentence encoder should keep simple to apply matrix–matrix product operation to the output of the question encoder (prior information). In contrast, the question encoder should incorporate various window sizes to focus on context words around the entities. The length of the sentence T should be defined according to the dataset. We also determine the number of filters (output channels) d_{SE} by investigating the optimal number using a validation dataset. The details of parameter tuning are described in Sect. 4.1. The results are a matrix with $d_{\mathrm{SE}} \times T - 2$ size to obtain the semantic sentence embedding.

3.2 Question encoder

The sentence encoder can encode input sentences into high-dimensional vectors that contain semantically rich information. However, in previous work, classification modules decode labels from sentence embedding without any prior information. Lacking prior information results in problems with the decoder. For example, it generates irrelevant labels even with well-trained sentence embedding because there are too many features.

Our framework employs a question encoder to incorporate prior information to achieve better temporal-relation classification. Li et al. [13] noted that just two entities in a sentence can be viewed as forming a pseudo-question when casting relation extraction as a question-answering problem, even if the sentence is not necessarily grammatical. For example, as shown in Table 1, the pseudo-question about Example 1 in Sect. 1 is "**believed**," "**established**," and their contexts. Note that "<pad> " is used as the padding symbol within the context window size = 2 in Table 1.

Question encoder takes the pseudo-question as inputs, given by: $Q = \{x_{e1-l_{\mathrm{CW}}} : x_{e1+l_{\mathrm{CW}}}, x_{e2-l_{\mathrm{CW}}} : x_{e2+l_{\mathrm{CW}}}\}$, where $e1$ and $e2$ denote the index of two entities, and l_{CW} is the window size.

Figure 2 shows the architecture of our question encoder.

Hyperparameter settings

We use three different filter window sizes (3, 4, and 5) in the convolutional layers with the number of filters (output channels) of d_{QE} to obtain feature maps from the constructed question embeddings. We exclude filter size 2 because it will not return the data with a peak centered around the word. Afterward, we apply a max-pooling algorithm to identify the most important feature in each feature map by taking the maximum value as the feature for filters. The results are merged as a $(d_{\mathrm{QE}} * 6)$-dimensional vector[3] to obtain the semantic sentence embedding, as shown in Fig. 2. Finally, we apply a fully connected layer and unsqueeze operation to it and obtain a matrix with the size of $(T - 2) \times 1$ for the matrix–matrix product with the output from the sentence encoder.

3.3 Extractor

In this step, we combine sentence embeddings and pseudo-question embeddings to generate the necessary information representation for classification.

We calculate a batch matrix–matrix product of sentence embeddings and question embedding with which we can only extract the information necessary for the task. Therefore, we can obtain a matrix to represent necessary information with the size of $d_{\mathrm{SE}} \times 1$.

3.4 Classifier

Thus far, we have obtained the necessary information representation for classification. Then, we apply a linear transformation to magnify the representation to the 2000-dimensional space that followed the dropout function with a ratio of 0.2. We also use another linear transformation to map this vector to a 5 (number of total classes)-dimensional space. Five classes are predefined in Sect. 4.1. The classifier allows the model to output temporal relations for input examples. We simply employ a $Softmax$ classifier to classify input sentences into predefined classes.

4 Experiments

We conduct comparative experiments to verify the effectiveness of our proposed approach. We first introduce the dataset,

[2] https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363.

[3] Note that "*" symbol means the scalar multiplication operator.

Table 1 Example of the context for questioned time–event entities in the *Timebank-Dense* corpus

Constructed question		
said he **believed** the conditions; had been **established** <pad> <pad>		
Entity 1	Entity 2	Window size
believed	**established**	2

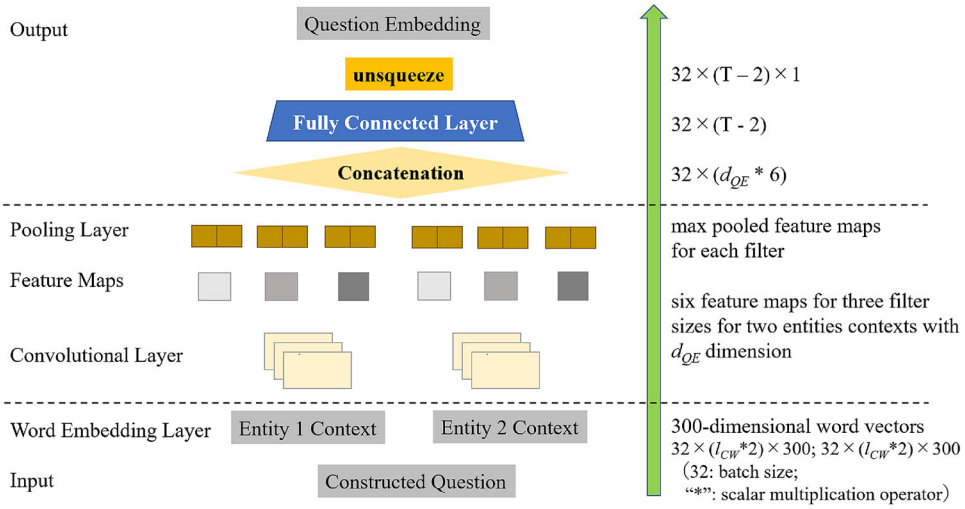

Fig. 2 Architecture of our question encoder

hyperparameters, and baseline systems used in our experiments. Then, we compare the performance of our model with some baselines. To evaluate further the effects of our proposed model, we also replace the CNN part with Bi-LSTM. We also show our experimental results with different sizes of training data in Sect. 5.

4.1 Dataset

We conduct experiments on the *Timebank-Dense* corpus [1], which contains 36 documents, including 12,715 examples. We divide the 36 documents into 22, 5, and 9 documents for training, validation, and testing, respectively.

The *Timebank-Dense* corpus is constructed to identify temporal relations between events and times in terms of the following four combinations: event and event (E-E), time and time (T-T), event and time (E-T), and event and document creation time (E-D). This dataset contains the following six temporal-relation types: "AFTER," "BEFORE," "SIMULTANEOUS," "INCLUDES," "IS_INCLUDED," and "VAGUE."

As in previous work, we skip the "SIMULTANEOUS" relation type because it has only a small number of instances.

Hyperparameter tuning using validation dataset

In our proposed method, we optimized the learning rate as 2e−5 and the number of epochs as 112 by finding the maximum F1 score using a validation dataset. As described in Sect. 1, we used the AdamW optimizer [18]: Adam with weighted decay. We set the batch size to 32. We also set the length of the sentence T to 130 by considering the maximum length (145) and average length (51.7) of the training dataset with avoiding redundant padding symbols.

Furthermore, we set hyperparameters d_{SE} and d_{QE} to 32 by comparing the F1 score using a validation dataset, as shown in Tables 2 and 3. Note that the F1 score was not sensitive to d_{SE} so much, but d_{QE} should be set between 32 and 128.

Finally, we set context window size l_{CW} in the question encoder to 5 by comparing the F1 score using the validation dataset, as shown in Table 4.

Table 2 F1 score using validation dataset with hyperparameter d_{SE}

d_{SE}	8	16	32	64	128	256	512
F1	0.670	0.662	**0.682**	0.675	0.680	0.675	0.673

Table 3 F1 score using validation dataset with hyperparameter d_{QE}

d_{QE}	8	16	32	64	128	256	512
F1	0.664	0.648	**0.682**	0.661	0.681	0.648	0.455

Table 4 F1 score using validation dataset with context window size l_{CW}

l_{CW}	3	4	5	6	7	8	9	10
F1	0.661	0.669	**0.682**	0.661	0.680	0.672	0.675	0.678

4.2 Baseline systems

To investigate the effectiveness of our model, we compare our model with the following six state-of-the-art models:
MIRZA [22]. This system comprises four classifiers: a rule set for T-T pairs and three L2-regularized logistic regression classifiers for E-D, E-T, and E-E pairs.
CHENG [2]. This system comprises two dependency path-based Bi-LSTM classifiers: one for E-E and E-T pairs and one for E-D pairs.
ZHANG [32]. This is a multilayer neural Bi-LSTM model. This model classifies temporal relations for E-E pairs only.
DLIGACH [4] This system is based on the CNN model. We implemented their system by setting the hyperparameters as described in [4]. We, however, replaced the optimizer from RMSprop with AdamW [18]: Adam with weighted decay because of the effectiveness of AdamW and the fairness of the comparison with our proposed method. We optimized the number of epochs to 151 by maximizing the F1 score using the validation data.
WU [31]. For comparison, reflecting on the recent trend of state-of-the-art methods, we prepared the system based on BERT [3] for general relation classification. Wu and He used a BERT model transformer with a sequence classification named "BertForSequenceClassification" and AdamW optimizer [18], following the original implementation. BERT is fine-tuned using the "bert-large-uncased" model, which is pretrained on a large and uncased corpus [3]. We optimized the hyperparameters using validation data with a learning rate of 4e−5 and setting the number of epochs to five.
SOARES [26]. This is another version of a BERT-based system for general relation classification. We used a bare BERT model transformer named "BertModel," outputting raw hidden states without any specific head on top and Adam optimizer [11], following the original implementation. BERT is fine-tuned using a pretraining model on "bert-large-uncased." We optimized the hyperparameters using the validation data with a learning rate of 1e−4 and setting the number of epochs to 10.

4.2.1 LSTM model with question encoder

Finally, we add one more comparison system based on our question encoder. In our proposed framework, the sentence encoder part plays an important role because the sentence embeddings affect the final classification accuracy. We demonstrate the impact of applying Bi-LSTM to encoding sentences. We replace CNN with Bi-LSTM in our proposed framework. That is, we apply Bi-LSTM to encode sentence embeddings in the sentence encoder part (in Fig. 2), while we keep the other parts (question encoder, extractor, and classifier in Fig. 1) remain unchanged. To conduct comparative experiments, we first feed the embedded sequence $E = \{e_1, e_2, \ldots, e_T\}$ to a forward LSTM from the beginning to the end and then to a backward LSTM from the end to the beginning. Then, the forward $\overrightarrow{h_t}$ and backward $\overleftarrow{h_t}$ results of each word x_t are combined as $[\overrightarrow{h_t} \oplus \overleftarrow{h_t}]$ by a concatenation operation. Next, we use the same operations (see Sect. 3.3) to construct pseudo-questions and extract the required information representation for classification. Finally, the model generates labels with the same classifier (see Sect. 3.4. For the Bi-LSTM layer, we set each hidden layer to 256-dimensional and train our model for up to 50 epochs with a learning rate of 0.01.

4.3 Overall results

We now report our experimental results on the *Timebank-Dense* corpus.

Table 5 compares the results obtained by our model with those obtained by three state-of-the-art models, DLIGACH [4], CHENG [2], and MIRZA [22]. We observe that our proposed model significantly improves the best state-of-the-art model, CHENG [2], by an F1 score of 0.147 (28.3%), indicating the effectiveness of our model.[4]

We compare the results by relation types for all pairs and for E–E pairs in Tables 6 and 7 including other baselines.[5] From the two tables, we conclude that our approach is more effective for E–E pairs. This is because the context words around the entities, which are focused on in our question encoder, affect E–E pairs more than other pairs for temporal-relation classification.

In Table 7, note that our CNN-based proposed model outperforms the other seven models for all types of relations with statistical significance (using a two-tailed t test), at a significance level of 5% for LSTM with question encoder, DLIGACH, and ZHANG and 1% for CHENG, MIRZA, WU, and SOARES in macro F1, particularly for the "INCLUDES" type. The limited amount of data for the "INCLUDES" type (5% of all data) always makes it the most difficult to find temporal relations. However, it is remarkable that our proposed model improves the F1 score by 0.254 (125.7%) for this type compared with the best state-of-the-art model, WU [31]. We observed that the best baseline model is WU's model [31] based on BERT. Our proposed model, however, still improved the micro F1 score by 0.104 (17.3%) compared with WU's model.

Tables 6 and 7 show that our CNN-based method (CNN+QE) outperforms the Bi-LSTM-based model (LSTM+QE)

[4] Note that these results were computed based on the weighted average of the results for E-D, E–E, and E–T pairs in Table 1 of [2]. The weight was computed based on the distribution in Table 7 of [1].

[5] Note that ZHANG classifies temporal relations between E–E pairs only, and WU and SOARES's approaches were developed to classify general E–E pair relationships.

Table 5 Overall comparison between our proposed framework and the three state-of-the-art models, DLIGACH [4], CHENG [2], and MIRZA [22], which are a CNN model, a Bi-LSTM-based model, and a feature-based model, respectively

Systems	Proposed	DLIGACH [4]	CHENG [2]	MIRZA [22]
Micro F1	0.667	0.497	0.520	0.512

Table 6 Comparison by relation types for all pairs

Methods	Approach	AFTER	BEFORE	INCLUDES	IS_ INCLUDED	VAGUE	F1 Score		
							Macro	*P* value	Micro
Proposed	CNN+QE	0.738	0.730	0.409	0.371	0.710	**0.592**	–	**0.667**
	LSTM+QE	0.688	0.670	0.385	0.372	0.669	0.557	*	0.622
DLIGACH [4]	CNN	0.429	0.423	0.073	0.178	0.710	0.363	*	0.497
CHENG [2][4]	LSTM	0.454	0.391	0.216	0.309	0.623	0.399	*	0.520
MIRZA [22]	LR	0.44	0.51	0.11	0.47	0.58	0.422	0.083	0.518

"*"denotes that the difference between our proposed CNN-based approach (bold score) and all the other four models in macro F1 is statistically significant for $p < 0.05$

Table 7 Comparison by relation types for E–E pairs

Methods	Approach	AFTER	BEFORE	INCLUDES	IS_ INCLUDED	VAGUE	F1 Score		
							Macro	*P* value	Micro
Proposed	CNN+QE	0.747	0.732	0.456	0.419	0.741	**0.619**	–	**0.705**
	LSTM+QE	0.665	0.672	0.427	0.419	0.711	0.579	*	0.661
DLIGACH [4]	CNN	0.440	0.444	0.096	0.143	0.737	0.372	*	0.546
CHENG [2]	LSTM	0.440	0.460	0.025	0.170	0.624	0.344	**	0.529
ZHANG [32]		0.526	0.503	0.106	0.325	0.626	0.417	*	0.548
MIRZA [22]	LR	0.430	0.471	0.049	0.250	0.613	0.363	**	0.519
WU [31]	BERT	0.536	0.613	0.202	0.234	0.656	0.448	**	0.601
SOARES [26]		0.297	0.308	0.067	0.102	0.311	0.217	**	0.444

"*" and "**"denote that the difference between our proposed CNN-based approach (bold score) and all the other seven models in macro F1 is statistically significant for $p < 0.05$ and $p < 0.01$, respectively

in our proposed framework for "AFTER," "BEFORE," "INCLUDES," and "VAGUE" types, indicating that CNN is effective in our proposed framework. As discussed in Sect. 1, we assume that we do not need to capture semantic information completely using Bi-LSTMs because the clues of temporal-relation classification tend to appear locally around the time–event entities. These results demonstrate that our hypothesis is correct because CNN can take full advantage of convolutional filters to extract local semantic features from pretrained word embeddings.

For E–E pairs classification, thanks to question encoder, our proposed model improves the micro F1 score by 0.159 (29.1%) compared with the ordinary CNN approach (DLIGACH [4]). In addition, our LSTM model with question encoder (LSTM+QE) improves the micro F1 score by 0.132 (25.0%) and 0.113 (20.6%) compared with the existing LSTM approaches (CHENG [2] and ZNANG [32]). Note that the difference in macro F1 is statistically significant for $p < 0.01$ (CHENG [2]) and $p < 0.05$ (ZHANG [32]). From these results, we can conclude that our question encoder module is quite effective, not only with CNNs but also with Bi-LSTMs. LSTM can automatically choose what to remember or forget when modeling long sequences using four gates (input modulation gate, input gate, forget gate, and output gate). Inevitably, it still keeps or forgets the wrong information required for identifying temporal relations because of the small training dataset. The extractor in our proposed framework, however, can select the necessary information to decide the relevant temporal relation based on constructed pseudo-questions.

5 Discussion

In this section, we discuss whether the size of the training data affects the results. NN-based models always give only slight improvement due to the small training datasets in supervised learning. To overcome this issue, we expand some of the

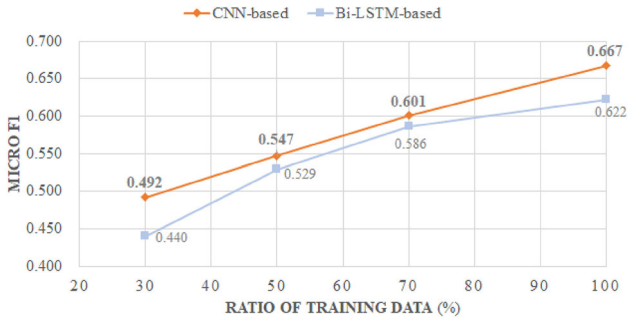

Fig. 3 Comparison between the CNN-based and the Bi-LSTM-based models by varying the ratio of the training data

Timebank-Dense dataset by including the reversed examples and report experimental results conducted on the expanded *Timebank-Dense* dataset.

5.1 The effect of training data size

Figure 3 compares our experimental results obtained by varying the ratio of training data by 30%, 50%, 70%, and 100% in our CNN-based and Bi-LSTM-based models. We observe that the CNN-based model in any ratio gives better results. Furthermore, by employing our proposed framework, the two models achieve satisfactory results with a limited size of data, and the CNN-based model is comparable with some of the state-of-the-art models with only 50% training samples.

Table 8 also compares our experimental results of each relation type obtained by varying the ratio of training data. We note that in the relation type "IS_INCLUDED," our CNN-based method cannot improve the F1 scores sufficiently as the size of the training data increases. In Sect. 5.2, we discuss how to improve the accuracy of these types with a smaller number of training samples.

5.2 Effect of data expansion

Due to the limited size of data, it can be more difficult to classify the temporal relation for the "INCLUDES" and "IS_INCLUDED" types. In our proposed model, thanks to the question encoder, we can differentiate two event entities in the temporal relation. We observe that the training data could be expanded by reversing the temporal relation between entity A and entity B from "AFTER" ("INCLUDES") and "BEFORE" ("IS_INCLUDED"), and *vice versa*.

Based on this observation, we can expand the *Timebank-Dense* dataset by including the reversed training examples for "AFTER," "BEFORE," "INCLUDES," and "IS_INCLUDED." This framework enables us to double the training data available for relations other than the "VAGUE" type.

Tables 9 and 10 show examples of our data expansion method and its statistics in each temporal relation, respectively. For the original instance, there is a BEFORE relation between the event "**established**" and the event "**believed**," whereas the relation between the event "**believed**" and the event "**established**" is AFTER.

Because of limited data in Table 10, it is more difficult to classify temporal relations for the "INCLUDES" and "IS_INCLUDED" types. As shown in Table 11, the F1 score is improved with statistical significance (using a two-tailed t test at a significance level of 5% in macro F1) as the amount of available training data increases, especially for "INCLUDES" and "IS_INCLUDED" types.

For comparison, we applied the expanded data to WU [31] and SOARES [26] using BERT, as introduced in Sect. 4.2. Micro F1 scores obtained by WU [31] and SOARES [26] are 0.633 and 0.439, respectively. Compared with the results using the original data in Table 7, our approach improved by 9.7% in micro F1 score, while WU's approach improved by 4.6% only and SOARES's approach decreased. From these results, our question encoder approach is more effective to improve the estimation with the expanded data compared with the contextualized embedding approaches.

Note that the document genre of the *Timebank-Dense corpus* is newspaper articles [1], and the sentence expressions do not diversify and depend on individual style so much. The document genre dependency is the limitation of our data expansion approach. We should investigate the generality of our data expansion approach with another document genre such as clinical domain [27].

6 Conclusion and future work

In this paper, we designed a CNN-based framework for temporal-relation classification with a question encoder to incorporate various window sizes to focus on context words around the entities. We assumed that task-related information can be extracted by introducing pseudo-questions as prior information and then by classifying labels through a classifier. Our proposed model was more interpretable and robust through the constructed questions.

Experimental results on the *Timebank-Dense* corpus demonstrated that our CNN-based model with question encoder significantly outperforms the Bi-LSTM-based model with it. Our proposed model also outperforms state-of-the-art systems including CNN-, Bi-LSTM-, and BERT-based models. It could classify comparably with those baselines even with a small training dataset (i.e., 50%). In addition, we demonstrated that expanding the training data by reversing the temporal relation improved the accuracy effectively for the relation types with a limited number of training datasets.

Table 8 Comparison of relation types in the CNN-based and the Bi-LSTM-based models by varying the ratio of training data

Relation	CNN-based				Bi-LSTM-based			
	30%	50%	70%	100%	30%	50%	70%	100%
AFTER	0.471	0.574	0.632	0.738	0.413	0.566	0.637	0.688
BEFORE	0.503	0.584	0.582	0.730	0.386	0.562	0.647	0.670
INCLUDES	0.157	0.288	0.386	0.409	0.150	0.222	0.275	0.385
IS_INCLUDED	0.218	0.302	0.418	0.371	0.160	0.298	0.400	0.372
VAGUE	0.610	0.614	0.676	0.710	0.591	0.597	0.624	0.669
Macro F1	0.392	0.472	0.539	**0.592**	0.340	0.449	0.517	0.557
Micro F1	0.492	0.547	0.601	**0.667**	0.440	0.529	0.586	0.622

Table 9 Example of original and expanded data instances

	Text	Label	Relation
Original instance	He said he **believed** the "conditions for a meeting" between Mr. Trump and Mr. Rouhani "in the next few weeks" had been **established**	Before	**established** ⇒ **believed**
Expanded instance	He said he **believed** the "conditions for a meeting" between Mr. Trump and Mr. Rouhani "in the next few weeks" had been **established**	After	**believed** ⇒ **established**

Table 10 Comparison between the expanded and original training data sizes

Relation	# Expanded training data	# Original training data
AFTER	4316	1889
BEFORE	4316	2427
INCLUDES	1733	695
IS_INCLUDED	1733	1038
VAGUE	442	442

Table 11 Experimental results conducted on expanded data

Relation	Expanded training data	Original training data
AFTER	0.795	0.738
BEFORE	0.807	0.730
INCLUDES	0.617	0.409
IS_INCLUDED	0.558	0.371
VAGUE	0.767	0.718
Macro F1	**0.699***	0.592
Micro F1	**0.732**	0.667

Learning sentence representation, however, remains a core issue for temporal-relation classification. In future work, we plan to enhance the word-embedding and sentence-encoding approach based on contextualized word representations such as BERT [3] to achieve much better classification accuracy. We also plan to evaluate our approach in the clinical domain using the THYME corpus [27] and investigate the effectiveness of our model for monitoring the effect of various treatments for the COVID-19 pandemic chronologically.

Acknowledgements This work was partially supported by a Japanese Society for the Promotion of Science Grant-in-Aid for Scientific Research (B) (#19H04420).

References

1. Chambers, N., Cassidy, T., McDowell, B., Bethard, S.: Dense event ordering with a multi-pass architecture. Trans. Assoc. Comput. Linguist. **2**, 273–284 (2014)
2. Cheng, F., Miyao, Y.: Classifying temporal relations by bidirectional LSTM over dependency paths. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), pp. 1–6 (2017)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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, Minneapolis, MN, USA, pp. 4171–4186 (2019)
4. Dligach, D., Miller, T., Lin, C., Bethard, S., Savova, G.: Neural temporal relation extraction. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, vol. 2, pp. 746–751 (2017)
5. Do, H.W., Jeong, Y.S.: Temporal relation classification with deep neural network. In: Proceedings of the 2016 International Conference on Big Data and Smart Computing (BigComp), Hong Kong, China, pp. 454–457 (2016)
6. He, D., Zhang, H., Hao, W., Zhang, R., Hao, H.: An attention-based hybrid neural network for document modeling. IEICE Trans. Inf. Syst. E100.D(6), 1372–1375. https://doi.org/10.1587/transinf.2016EDL8231 (2017)
7. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Séaghdha, Ó., D, Padó S, Pennacchiotti M, Romano L, Szpakowicz S.: SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Uppsala, Sweden, pp. 33–38. https://www.aclweb.org/anthology/S10-1006 (2010)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Jin, P., Lian, J., Zhao, X., Wan, S.: Tise: a temporal search engine for web contents. Intell. Inf. Technol. Appl. (2008). https://doi.org/10.1109/IITA.2008.132
10. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1746–1751 (2014)
11. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of 3rd International Conference on Learning Representations, ICLR: San Diego, CA, USA (2015)
12. Laokulrat, N., Miwa, M., Tsuruoka, Y., Chikayama, T.: UTTime: temporal relation classification using deep syntactic features. In: Second Joint Conference on Lexical and Computational Semantics, Proceedings of the Seventh International Workshop on Semantic Evaluation, vol. 2, Atlanta, Georgia, USA, pp. 88–92 (2013)
13. Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., Zhou, M., Li, J.: Entity-relation extraction as multi-turn question answering. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), pp. 1340–1350 (2019)
14. Lin, C., Dligach, D., Miller, T., Bethard, S., Savova, G.: Multilayered temporal modeling for the clinical domain. J. Am. Med. Inform. Assoc. **23** (2016)
15. Lin, C., Miller, T., Dligach, D., Bethard, S., Savova, G.: Representations of time expressions for temporal relation extraction with convolutional neural networks. In: Proceedings of the 2017 Biomedical Natural Language Processing Workshop, pp. 322–327. https://doi.org/10.18653/v1/W17-2341 (2017)
16. Lin, C., Miller, T., Dligach, D., Bethard, S., Savova, G.: A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In: Proceedings of the 2nd Clinical Natural Language Processing Workshop, Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 65–71. https://doi.org/10.18653/v1/W19-1908. https://aclanthology.org/W19-1908 (2019)
17. Liu, S., Wang, L., Chaudhary, V., Liu, H.: Attention neural model for temporal relation extraction. In: Proceedings of the 2nd Clinical Natural Language Processing Workshop, Minneapolis, Minnesota, USA, pp. 134–139. https://doi.org/10.18653/v1/W19-1917 (2019)
18. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of Seventh International Conference on Learning Representations, ICLR: New Orleans. LA, USA (2019)
19. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, pp. 1412–1421. https://doi.org/10.18653/v1/D15-1166. https://aclanthology.org/D15-1166 (2015)
20. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
21. Mirza, P., Tonelli, S.: Classifying temporal relations with simple features. In: Proceedings of the 2014 Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014), pp. 308–317 (2014)
22. Mirza, P., Tonelli, S.: Classifying temporal relations with simple features. In: Proceedings of the 2014 Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014), pp. 308–317 (2016)
23. Nguyen, T.H., Grishman, R.: Relation extraction: perspective from convolutional neural networks. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Association for Computational Linguistics, Denver, Colorado, pp. 39–48. https://doi.org/10.3115/v1/W15-1506. https://aclanthology.org/W15-1506 (2015)
24. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1532–1543 (2014)
25. Seki, Y., Zhao, K., Oguni, M., Sugiyama, K.: A framework for classifying temporal relations with question encoder. In: Digital Libraries at Times of Massive Societal Transition—Proceedings of 22nd International Conference on Asia-Pacific Digital Libraries (ICADL 2020) , Springer, Lecture Notes in Computer Science, vol 12504, pp. 20–32 (2020)
26. Soares, L.B., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: distributional similarity for relation learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 2895–2905. https://doi.org/10.18653/v1/P19-1279. https://www.aclweb.org/anthology/P19-1279 (2019)
27. Styler, W.F., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P.C., Erickson, B., Miller, T., Lin, C., Savova, G., Pustejovsky. J. Temporal annotation in the clinical domain. Trans. Assoc. Comput. Linguist. **2**, 143–154 (2014)
28. Tikkinen, K.A.O., Malekzadeh, R., Schlegel, M., Rutanen, J., Glasziou, P.: COVID-19 clinical trials: learning from exceptions in the research chaos. Nat. Med. **26**, 1671–1672 (2020)
29. Tourille, J., Ferret, O., Névéol, A., Tannier, X.: Neural architecture for temporal relation extraction: a bi-LSTM approach for detecting narrative containers. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (volume 2: short papers), pp. 224–230. https://doi.org/10.18653/v1/P17-2035 (2017)
30. Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., Zeng, Y., Mehrabi, S., Sohn, S., Liu, H.: Clinical information extraction applications: a literature review. J. Biomed. Inform. **77**, 34–49 (2018)

31. Wu, S., He, Y.: Enriching pre-trained language model with entity information for relation classification. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19), Beijing, People's Republic of China, pp. 2361–2364. https://doi.org/10.1145/3357384.3358119. https://dl.acm.org/doi/10.1145/3357384.3358119 (2019)
32. Zhang, Y., Li, P., Zhou, G.: Classifying temporal relations between events by deep bilstm. In: Proceedings of the 2018 International Conference on Asian Language Processing (IALP 2018), pp. 267–272 (2018)
33. Zheng, N., Wang, J., Zhang, H., Cui, W., Kang, Z., Yang, T., Lou, B., Chi, Y., Long, H., Ma, M., Yuan, Q., Zhang, S., Zhang, D., Ye, F., Xin, J.: Predicting COVID-19 in China using hybrid AI model. IEEE Trans. Cybern. **50**(7), 2891–2904 (2020)