# NA-Aware Machine Reading Comprehension for Document-Level Relation Extraction

Zhenyu Zhang, Bowen Yu, Xiaobo Shu, and Tingwen Liu (✉)

Institute of Information Engineering, Chinese Academy of Sciences, Beijing. China
School of Cyber Security, University of Chinese Academy of Sciences, Beijing. China
{zhangzhenyu1996, yubowen, shuxiaobo, liutingwen}@iie.ac.cn

**Abstract.** Document-level relation extraction aims to identify semantic relations between target entities from the document. Most of the existing work roughly treats the document as a long sequence and produces target-agnostic representation for relation prediction, limiting the model's ability to focus on the relevant context of target entities. In this paper, we reformulate the document-level relation extraction task and propose a NA-aware machine Reading Comprehension (NARC) model to tackle this problem. Specifically, the input sequence formulated as the concatenation of a head entity and a document is fed into the encoder to obtain comprehensive target-aware representations for each entity. In this way, the relation extraction task is converted into a reading comprehension problem by taking all the tail entities as candidate answers. Then, we add an artificial answer `NO-ANSWER` (NA) for each query and dynamically generate a NA score based on the decomposition and composition of all candidate tail entity features, which finally weighs the prediction results to alleviate the negative effect of having too many no-answer instances after task reformulation. Experimental results on DocRED with extensive analysis demonstrate the effectiveness of NARC.

**Keywords:** Document-level relation extraction · Machine reading comprehension · No-answer query.

## 1 Introduction

Reading text to identify and extract relational facts in the form of (*head entity, relation, tail entity*) is one of the fundamental tasks in data mining and natural language processing. For quite some time, researchers mainly focus on extracting facts from a sentence, i.e., sentence-level relation extraction [8, 34, 35], However, such an ideal setting makes it powerless to handle a large number of inter-sentence relational triples in reality. To move relation extraction forward from sentence-level to document-level, the DocRED dataset is proposed recently [31], in which each document is annotated with a set of named entities and relations. In Figure 1, we show an example in DocRED development set to illustrate the challenging yet practical extension: for the extraction of relational fact (`U Make Me Wanna`, `performer`, `Blue`), one has to first identify the fact that `U Make Me`

**> One Love (Blue album)**
[1] *One Love* is the second studio album by English boy band *Blue*, released on *4 November 2002* in the *United Kingdom* and on *21 October 2003* in the *United States*. [2] The album peaked at number one on the *UK Albums Chart*, where it stayed for one week. On *20 December 2003* it was certified *4 ×Platinum* in the UK. ... [4] Three singles were released from the album: "*One Love*", which peaked at number three, "*Sorry Seems to Be the Hardest Word*", featuring *Elton John*, which peaked at number one, and "*U Make Me Wanna*", which peaked at number four.

**Subject:** *One Love*, *Sorry Seems to Be the Hardest Word*, *U Make Me Wanna*
**Object:** *4 November 2002*, *21 October 2003*
**Relation:** *publication date*

**Subject:** *One Love*, *Sorry Seems to Be the Hardest Word*, *U Make Me Wanna*
**Object:** *Blue*
**Relation:** *performer*

**Fig. 1.** An example from DocRED. Word spans with the same color indicate the same named entity, and the key clues for relation inference are underlined.

`Wanna` is a music single in `One Love` from sentence 4, then identify the facts `One Love` is an album by `Blue` from sentence 1, and finally infer from these facts that the `performer` of `U Make Me Wanna` is `Blue`.

In recent times, there are considerable efforts devoted to document-level relation extraction. Some popular techniques in sentence-level relation extraction (e.g., attention mechanism, graph neural networks, and pre-trained language models) are introduced and make remarkable improvements [17, 24, 32]. Specifically, most of them take the document as a long sequence and generate target-agnostic representations, then perform relation classification for each entity pair. Despite the great success, we argue that learning general representation is sub-optimal for extracting relations between specific target entities from the long document, since some target-irrelevant words could introduce noise and cause confusion to the relation prediction.

Inspired by the current trend of formalizing NLP problems as machine reading comprehension (MRC) style tasks [7, 13, 30], we propose NARC, a NA-aware MRC model, to address this issue. Instead of treating document-level relation extraction as a simple entity pair classification problem, NARC first formulates it as a MRC task by taking the given head entity as query and all tail entities in the document as candidate answers, then performing relation classification for each candidate tail entity. Specifically, the input sequence of NARC is organized as the concatenation of the head entity and the document in the form of "`[CLS]+Head Entity+[SEP]+ DOCUMENT+[SEP]`", and then fed into the query-context encoder, which is made up of a pre-trained language model followed by a simple entity graph. The former serves the target-aware context encoding, while the later is constructed to perform multi-hop reasoning.

However, one barrier in such task formulation is the troublesome No-Answer (NA) problem. Considering a document with $n$ entities, MRC-style formulation requires $n$ enumerations for a complete extraction, in which many queries have no correct answer (65.5% in DocRED) since there are a great number of entity pairs in a document that do not hold pre-defined relations. To fill this gap, we append a special candidate `NO-ANSWER` for each query. As a result, the number of

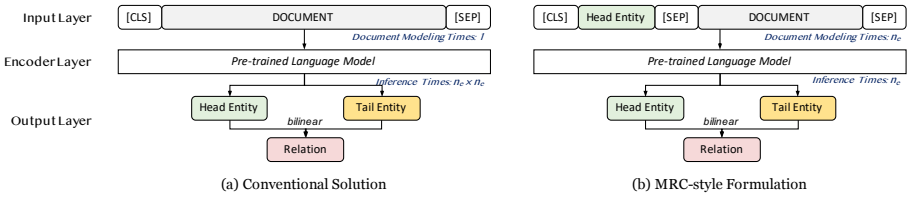


**Fig. 2.** Diagrams of the (a) conventional and (b) MRC-style paradigms for document-level relation extraction with pre-trained language models.

queries pointing to `NO-ANSWER` often exceeds that of other queries with valid answers, causing extremely imbalanced data distribution. To mitigate this adverse effect, we introduce a novel answer vector assembler module after task reformulation, which firstly integrates features from different layers of the encoder as the final representation of each entity, then vectorizes the human-made candidate `NO-ANSWER` with a decomposition-composition strategy, where each candidate tail entity vector is first decomposed into the relevant and irrelevant components with respect to the head entity, and then composed to a query-specific NA vector. Finally, this vector is projected into a NA score, which weighs the predicted relation scores to take the probability of `NO-ANSWER` into account.

Experiments conducted on DocRED, the largest public document-level relation extraction dataset, show that the proposed NARC model achieves superior performance over previous competing approaches. Extensive validation studies demonstrate the effectiveness of our MRC-style task formulation and the NA-aware learning strategy.

## 2 Task Formulation

In this section, we first briefly recall some basic concepts and classic baselines for document-level relation extraction, and then describe the task transformation.

Formally, given a document $\mathcal{D} = \{w_i\}_{i=1}^{n_w}$ and its entity set $\mathcal{E} = \{e_i\}_{i=1}^{n_e}$, where $e_i$ is the $i$-th entity with $n_m^i$ mentions $\mathcal{M}_i = \{m_j\}_{j=1}^{n_m^i}$. The goal of document-level relation extraction is to predict all the relations $\mathcal{R}' \in \mathcal{R} = \{r^i\}_{i=1}^{n_r}$ between every possible entity pair. Named entity mentions corresponding to the same entity have been assigned with the same entity id in the annotation. Considering that many relational facts express in multiple sentences, the document-level task is more complicated than the traditional sentence-level task. The model is expected to have a powerful ability to extract relational evidence from the long text and eliminate the interference of noise information.

**Conventional Solution.** Previous work usually takes the document as a long sequence, and converts it into hidden states with kinds of encoders. Typically, Wang et al. [26] packed the input sequence to "`[CLS]+DOCUMENT+[SEP]`" and employed BERT [6] as the encoder. Then, the document-level relation ex-

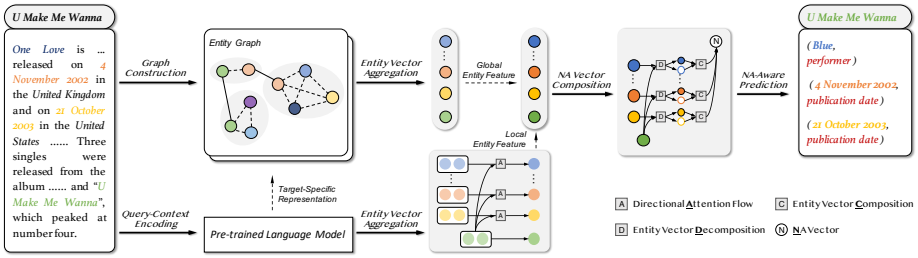


**Fig. 3.** Overview of the NARC model. The model receives a document and a target *head entity* at a time, and outputs all the related (*tail entity, relation*) pairs. Here we take *U Make Me Wanna* (symbols with green color) as an example.

traction task is treated as a multi-label classification problem. For each target entity pair, it gathers entity representations from the hidden states and employs the Sigmoid function to compute the probability of each relation. Obviously, this practice encodes the document once to produce target-agnostic representations, and the classification enumerates all possible entity pairs by $n_e \times n_e$ times.

**MRC-style Formulation.** Different from the conventional solution, we propose that document-level relation extraction can be formulated as a MRC problem, in which the model is expected to answer: "*which entities in the document have what relations with the target head entity?*". Under such formulation, the input sequence is modified to "`[CLS]+Head Entity+[SEP]+DOCUMENT+[SEP]`", then pre-trained language models is able to output target-specific representations, which have benefits in filtering irrelevant information of the target entity pair as revealed in the experiments. In this paradigm, the times of document modeling and classification are both $n_e$. However, the enumeration of head entities inevitably introduces a number of No-Answer (NA) queries since many entity pairs in the document do not hold pre-defined relations, which act as negative samples and will damage the model performance due to the data imbalance. To be compatible with the unforeseen situation, we add a special candidate `NO-ANSWER` for all instances. In the end, how to solve the no-answer problem becomes the key barrier of applying MRC-style formulation into the document-level relation extraction task.

## 3 NA-aware MRC (NARC)

This section provides NARC in detail. It formulates document-level relation extraction as a machine reading comprehension problem based on a query-context encoder, and solves the no-answer (NA) issue with the answer vector assembler and NA-aware predictor. As shown in Figure 3, we first feed (*head entity, document*) pairs into a pre-trained language model, then the vectorized document tokens pass through a stacked entity graph to derive semantic evidence from the document and enable multi-hop reasoning. Next, the directional attention flow (DAF) is introduced to aggregate the local features for tail entities based on the

mention representation of the pre-trained language model. The results are combined with the entity representation in the entity graph (i.e., global features) to form the final entity vector. For the vectorization of `NO-ANSWER`, each candidate tail entity vector is first decomposed into two components that corresponding to target-specific relevant and irrelevant parts, then all the components of all the candidate tail entities are composed into a no-answer vector. Finally, the no-answer vector is merged into the candidate list as a negative example, and the no-answer score is calculated based on the vector to weigh the prediction. In this way, the model could induce low confidence when there is no valid tail entity due to the dominance of irrelevant components in the no-answer vector.

### 3.1 Query-Context Encoder

Following the MRC-style formulation, the document (*context*) is concatenated with the head entity (*query*) and fed into a pre-trained language model. By introducing such a packed sequence, advanced pre-trained language models such as BERT [6] can encode the document in a query-aware manner owning to the sufficiently deep self-attention architectures. Beyond that, the great success of integrating graph neural networks with pre-trained language models makes it a popular document encoding structure in natural language processing. Here we directly borrow the representative model Entity Graph [5] from multi-hop MRC to achieve global entity features, where mentions of entities are regarded as nodes in the graph while edges encode relations between different mentions (e.g., within- and cross-sentence coreference links or simply co-occurrence in a sentence). Then, the relational graph convolutional networks (R-GCN [20]) are applied to the graph and trained to perform multi-hop relation reasoning[1].

### 3.2 Answer Vector Assembler

It is intuitive that the global entity features obtained from Entity Graph could be treated as the final representations for relation prediction. However, it may fail to effectively exploit the local contextual interaction between mentions. To assemble comprehensive representations vectors for entities and the man-made option `NO-ANSWER`, we propose the entity vector aggregation and NA vector composition modules in this section.

**Entity Vector Aggregation.** In a document, one entity could be mentioned multiple times, and these mentions are the exact elements involved in relation expression and reasoning. To capture such local features, we extract all mention-level representations for each entity from the output of pre-trained language models. Apparently, the importance varies among different tail entity mentions for the target head entity. Thus we introduce directional attention flow (DAF), a variety of BiDAF [21], to measure the difference and compress the mention features into an embedding for each candidate tail entity.

[1] For more details about the construction process of Entity Graph, we recommend readers to reference the original paper [5].

Given the head entity $e_h$ and a candidate tail entity $e_t$, the similarity matrix $\mathbf{S}_{ht} \in \mathbb{R}^{n_m^h \times n_m^t}$ is first calculated by

$$\mathbf{S} = avg_{-1}\mathcal{F}_s([\mathbf{M}_h; \mathbf{M}_t; \mathbf{M}_h \odot \mathbf{M}_t]), \tag{1}$$

where $\mathbf{M}_h \in \mathbb{R}^{n_m^h \times d}$ and $\mathbf{M}_t \in \mathbb{R}^{n_m^t \times d}$ are the mention feature matrixes for these two entities, in which each mention feature is generated by the mean-pooling over corresponding word embeddings. $\mathcal{F}_s$ is a linear transformation, $avg_{-1}$ stands for the average operation in the last dimension. Next, we design the head-to-tail attention matrix $\mathbf{M}_{h2t} \in \mathbb{R}^{n_m^t \times d}$, which signifies the tail mentions that are most related to each mention in the head entity, via

$$\mathbf{M}_{h2t} = dup(softmax(max_{col}(\mathbf{S})))^\top \mathbf{M}_t, \tag{2}$$

where $max_{col}$ is the maximum function applied on across column of a matrix, which transforms $\mathbf{S}_{ht}$ into $\mathbb{R}^{1 \times n_m^t}$. Then the *dup* function duplicates it for $n_m^h$ times into shape $\mathbb{R}^{n_m^h \times n_m^t}$.

The output of DAF is the head mention feature matrix $\mathbf{M}_h$ and head-to-tail attention matrix $\mathbf{M}_{h2t}$. Finally, we utilize mean-pooling to obtain local entity features and concatenate them with the global entity features $\mathbf{e}^G \in \mathbb{R}^d$ generated by the entity graph in query-context encoder:

$$\mathbf{e}_h = [mean(\mathbf{M}_h); \mathbf{e}_h^G], \quad \mathbf{e}_t = [mean(\mathbf{M}_{h2t}); \mathbf{e}_t^G]. \tag{3}$$

**NA Vector Composition.** Different from other candidate answers that point to specific entities, "`NO-ANSWER`" (NA) is a man-made option without corresponding representation. To meet this challenge, we assume that each candidate entity vector could be decomposed into relevant and irrelevant parts with respect to the target head entity and later composited to derive the NA vector based on all candidate tail entities. In other words, every candidate tail entity contributes to the vectorization of `NO-ANSWER`. The key intuition behind this is that `NO-ANSWER` could be regarded as an option similar to the `none-of-the-above` in multiple-choice questions, only after comprehensively considering all other candidate answers can one make such a choice.

Formally, based on the final representation of given head entity $\mathbf{e}_h \in \mathbb{R}^{2d}$, each candidate tail entity vector $\mathbf{e}_t \in \mathbb{R}^{2d}$ is expected to be decomposed into a relevant part $\mathbf{e}_t^+ \in \mathbb{R}^{2d}$ and an irrelevant part $\mathbf{e}_t^- \in \mathbb{R}^{2d}$. Here we adapt the linear decomposition strategy proposed in sentence similarity learning [28] to meet this demand:

$$\mathbf{e}_t^+ = \frac{\mathbf{e}_h^\top \mathbf{e}_t}{\mathbf{e}_t^\top \mathbf{e}_t}\mathbf{e}_t, \quad \mathbf{e}_t^- = \mathbf{e}_t - \mathbf{e}_t^+. \tag{4}$$

The motivation here is that the more similar between $\mathbf{e}_h$ and $\mathbf{e}_t$, the higher the correlation between the head entity and the candidate tail entity, thus the higher proportion of $\mathbf{e}_t$ should be assigned to the similar component. In the composition step, we extract features from both the relevant matrix and the irrelevant matrix for each tail entity as follows:

$$\mathbf{e}_t^n = tanh(\mathbf{W}_{cr}\mathbf{e}_t^+ + \mathbf{W}_{ci}\mathbf{e}_t^- + b_c). \tag{5}$$

where $\mathbf{W}_{cr/ci} \in \mathbb{R}^{2d \times 2d}$ and $b_c \in \mathbb{R}^{2d}$ are trainable weight matrix and bias vector respectively. Afterwards, we apply a max-pooling over all candidate tail entities to obtain the representation of `NO-ANSWER`:

$$\mathbf{n} = max\{\mathbf{e}_t^n\}_{t=1}^{n_e}. \tag{6}$$

### 3.3 NA-Aware Predictor

In the prediction stage, we hope that the model has a preliminary perception about whether there is a valid answer to the query, then give the final relation prediction based on the perception. Moreover, `NO-ANSWER` is regarded as a special candidate entity, which takes $\mathbf{n} \in \mathbb{R}^{2d}$ as representation, thus introducing additional negative examples to guide the model optimization.

Specifically, we pass the NA vector through a linear transformation $\mathcal{F}_n$ followed by a sigmoid function $\delta$ to obtain a score that points to `NO-ANSWER` for the given query: $s_n = \delta(\mathcal{F}_n(\mathbf{n}))$. Next, the NA score is combined with the output logits as an auxiliary weight to achieve the NA-aware prediction:

$$\mathbf{r}_{ht} = \begin{cases} (1 - s_n) \cdot bili(\mathbf{e}_h, \mathbf{e}_t), \text{ if } e_t \in \mathcal{E}, \\ s_n \cdot bili(\mathbf{e}_h, \mathbf{n}), \text{ if } e_t \text{ is } \texttt{NO-ANSWER}. \end{cases} \tag{7}$$

where *bili* denotes the bilinear layer.

**Training and Inference.** Considering that there are multiple relations between an entity pair $(e_h, e_t)$, we take the relation prediction as a multiple binary classification problem, and choose the binary cross-entropy loss between the prediction and ground truth as the optimization objective:

$$\mathcal{L} = -\sum_{i=1}^{n_r} \left( y_{ht}^i \cdot \log(r_{ht}^i) + (1 - y_{ht}^i) \cdot \log(1 - r_{ht}^i) \right) \tag{8}$$

where $r_{ht}^i \in (0, 1)$ is the $i$-th dimension of $\mathbf{r}_{ht}$, indicating the prediction possibility of $i$-th relation, and $y_{ht}^i \in \{0, 1\}$ is the corresponding ground truth label. Specially, $y_{ht}^i$ is always 0 if $e_t$ is `NO-ANSWER`.

Following previous work [31], we determine a thresholds $\theta$ based on the micro F1 on the development set. With the threshold, we classify a triplet $(e_h, r_{ht}^i, e_t)$ as positive result if $r_{ht}^i > \theta$ or negative result otherwise in the test period. It is worth noting that we omit the relational triples whose tail entity is "`NO-ANSWER`" in inference. Finally, We combine the predictions from every sequence generated from the same document and with different queries, in order to obtain all relational facts over the document.

## 4 Experiments

### 4.1 Dataset

We evaluate our model on the public benchmark dataset, DocRED [31]. It is constructed from Wikipedia and Wikidata, covers a broad range of categories with

**Table 1.** Statistics of the DocRED dataset.

| | # Doc. | # Fact | # Pos. Pair | # Neg. Pair | # Rel. |
|---|---|---|---|---|---|
| Train | 3,053 | 34,715 | 38,269 | 1,160,470 | 96 |
| Dev | 1,000 | 11,790 | 12,332 | 384,467 | 96 |
| Test | 1,000 | 12,101 | 12,842 | 379,316 | 96 |

96 relation types, and is the largest human-annotated dataset for general domain document-level relation extraction. Documents in DocRED contain about 9 sentences and 20 entities on average, and more than 40.7% relation facts can only be extracted from multiple sentences. Moreover, 61.1% relation instances require various inference skills such as multi-hop reasoning. We follow the official partition of the dataset (i.e., 3053 documents for training, 1000 for development, and 1000 for test) and show the statistics in Table 1.

### 4.2 Implementation Details

We implement NARC with PyTorch 1.4.0 and *bert-base-uncased* model. The concatenated sequence in the input layer is trimmed to a maximum length of 512. The embedding size of BERT is 768, a linear-transformation layer is utilized to project the BERT embedding into a low-dimensional space with the same size of the hidden state, which is set to 200 (chosen from $[100, 150, 200, 250]$). Besides, the layer number of Entity Graph is set to 2 (chosen from $[1, 2, 3, 4]$), the batch size is set to 10 (chosen from $[5, 8, 10, 12]$), the learning rate is set to $1e^{-5}$ (chosen from $1e^{-4}$ to $1e^{-6}$). We optimize our model with Adam and run it on one 16G Tesla V100 GPU for 50 epochs. All hyper-parameters are tuned on the development set. Evaluation on the test set is done through CodaLab[2]. Following popular choices and previous work, we choose micro F1 and micro Ign F1 as evaluation metrics. Ign F1 denotes F1 excluding relational facts that appear in both the training set and the development or test set.

### 4.3 Baselines

We compare our NARC model with the following two types of baselines.

**Baselines w/o BERT**: On this track, we select 5 representative classic models without BERT. (1-3) CNN/BiLSTM/ContextAware [31]: These models leverage different neural architectures to encode the document, which are all text-based models and official baselines released by the authors of DocRED. (4) AGGCN [8]: It is the state-of-the-art sentence-level relation extraction model, which takes full dependency trees as inputs and constructs latent structure by self-attention. (5) EoG [4]: It constructs an edge-oriented graph and uses an iterative algorithm over the graph edges, which is a recent state-of-the-art model in biomedical domain document-level relation extraction.

[2] https://competitions.codalab.org/competitions/20717

**Table 2.** Main results on DocRED, bold marks highest number among all the models. BERT-MRC indicates the vanilla MRC-style formulation without NA-related module, and $\text{NARC}_{\text{w/o EG}}$ indicates the NARC model without entity graph.

| (year) **Model** | Dev<br>Ign F1 / F1 | Test<br>Ign F1 / F1 |
|---|---|---|
| (2019) CNN [31] | 41,58 / 43.45 | 40.33 / 42.26 |
| (2019) LSTM [31] | 48.44 / 50.68 | 47.71 / 50.07 |
| (2019) BiLSTM [31] | 48.87 / 50.94 | 48.78 / 51.06 |
| (2019) ContextAware [31] | 48.94 / 51.09 | 48.40 / 50.70 |
| (2019) AGGNN [8] | 46.29 / 52.47 | 48.89 / 51.45 |
| (2019) EoG [4] | 45.94 / 52.15 | 49.48 / 51.82 |
| (2019) BERT-RE [26] | 52.04 / 54.18 | 51.44 / 53.60 |
| (2020) BERT-HIN [24] | 54.29 / 56.31 | 53.70 / 55.60 |
| (2020) BERT-Coref [32] | 55.32 / 57.51 | 54.54 / 56.96 |
| (2020) BERT-GLRE [25] | - / - | 55.40 / 57.40 |
| (2020) BERT-LSR [17] | 52.43 / 59.00 | **56.97** / 59.05 |
| (ours) BERT-MRC | 55.49 / 57.59 | 54.86 / 57.13 |
| (ours) $\text{NARC}_{\text{w/o EG}}$ | 56.94 / 59.05 | 55.99 / 58.33 |
| (ours) NARC | **57.73** / **59.84** | 56.71 / **59.17** |

**Baselines w/ BERT**: On this track, we select 5 recent methods that adopt bert-base as the basic encoder. (1) BERT-RE [26]: It is the standard form of using BERT for relation extraction described in Section 2. (2) BERT-HIN [24]: It aggregates inference information from entity, sentence, and document levels with a hierarchical inference network to predict relation. (3) BERT-Coref [32]: It proposes an auxiliary training task to enhance the reasoning ability of BERT by capturing the co-refer relations between noun phrases. (4) BERT-GLRE [25]: It is a graph-based model by encoding the document information in terms of entity global and local features as well as relation features. (5) BERT-LSR [17]: It dynamically constructs a latent document-level graph for information aggregation in the entire document with an iterative refinement strategy.

### 4.4 Performance Comparison

Comparing the performance of different models in Table 2, the first conclusion we draw is that *NARC outperforms all baseline models in almost all the evaluation matrices*, which demonstrates the effectiveness of our NA-aware MRC solution, as well as the motivation of formulating document-level relation extraction as a machine reading comprehension problem. Secondly, *BERT-MRC outperforms BERT-RE by a significant margin*. We consider that the MRC-style model captures the interaction between the head entity and the document based on the deep self-attention structure, which helps to extract establish target-centric representations and extract information from relevant tokens from the document.

**Table 3.** Ablation study on DocRED development set to investigate the influence of different modules in NARC. † indicates that we also remove the NA-Aware Prediction module, because it relies on the NA Vector Composition.

| | Ign F1 | F1 |
|---|---|---|
| NARC | 57.73 | 59.84 |
| – Entity Vector Aggregation | 56.36 | 58.58 |
| – NA-Aware Prediction | 56.86 | 59.01 |
| – NA Vector Composition† | 56.84 | 58.98 |

Thirdly, *$NARC_{\text{w/o EG}}$ improves BERT-MRC by about 1.5% in F1 score.* We attribute the performance gain to the composition of NA vector. As the training set contains queries with and without valid answers, the vectorization process associated with all entities allows the model to automatically learn when to pool relevant and irrelevant portions to construct the NA vector. In optimization, the NA vector is used to increase or decrease the confidence of prediction, and thus makes the model aware of no-answer queries and alleviates its harmful effects. Lastly, *NARC exhibits a remarkable gain compared with $NARC_{\text{w/o EG}}$*, which demonstrates that the graph structure can exploit useful reasoning information among entities to capture rich non-local dependencies.

The effectiveness of each module in NARC is investigated in Table 3. From these ablations, we observe that: (1) The operation of Entity Vector Aggregation is indispensable since the ablation hurts the final result by 1.26% F1. It verifies the effectiveness of integrating the global and local features for an entity, as well as introducing the directional attention flow to take into account the fine-grained interaction between mention pairs. (2) NA-Aware Prediction is also a necessary component that contributes 0.83% gain of F1 to the ultimate performance. This is strong evidence that the NA score associated with all candidate tail entities is capable of providing powerful guidance for the final prediction. (3) When further removing NA Vector Composition, there are only slight fluctuations in performance. In other words, there is no remarkable improvement when only composing the NA vector as negative samples but not using the NA-aware prediction. The principle behind this phenomenon is that merely adding negative samples is not an effective way to boost performance, even if the generated negative samples are incredibly informative.

### 4.5 Performance Analysis

To further analyze the performance of NARC, we split the DocRED development set into several subsets based on different analytical strategies and report the performance of different models in each subset.

**Performance on Various Distances between Entities.** In this part, we examine the model performance in terms of entity distance, which is defined as the relative distances of the first mentions of the two entities in the document. As shown in Figure 4(a), the F1 score suffers a quick and pronounced drop

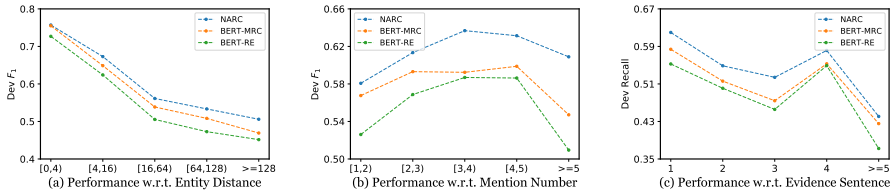


**Fig. 4.** Performance analysis on (a) detecting long-distance relations, (b) aggregating multiple mention information, and (c) reasoning multi-hop relations. We report the *F1* score for the first two analyses, while report *Recall* for the last one.

with the increase of entity distance, which is accordant with human intuition that detecting long-distance relations is still a challenging problem for current relation extraction models. Nevertheless, BERT-MRC consistently outperform BERT-RE by a sizable margin, due to the strong power of our MRC-style in reserving the relevant content of the target entities. Moreover, NARC outperforms all other baselines as the entity distance increases. This is because NARC breaks the limitation of sequence modeling and effectively captures long-distance interactions of semantic information by introducing the graph structure.

**Performance on Various Amounts of Entity Mentions.** To explore the capacity of different models in aggregating information from multiple mentions, we measure the performance in terms of the average mention number for each entity pair and report the F1 score in Figure 4(b). Interestingly, all the models do not achieve their best performance when the mention number is small. We explain that the relevant information carried by a single mention is quite limited, making relations harder to be predicted, especially when the extraction scope is enlarged to the document level, the long-distance between two mentions making relations harder to be predicted. When the number of mentions is large, the performance of BERT-RE and NARC is devastating once again. This is because not every entity mention is involved in the relational facts, and aggregating information indiscriminately may introduce a large amount of noisy context, which will confuse the classifier. On the contrary, the directional attention flow measures the tail entity mentions and selects the most important one for each head entity mention, so that the proposed NARC maintains a relatively high performance when there are many mentions of the entity pair.

**Performance on Various Amounts of Evidence Sentences.** To assess the model's ability in multi-hop reasoning, we plot the preference curve in Figure 4(c) when different amounts of evidence sentences are available for the relational facts. Unlike the previous two statistical features, the evidence sentence number is a semantic feature and can only be counted for positive labels, thus we report the Recall score for evaluation. Again, NARC outperforms all methods. Furthermore, the results indicate that the performance gap between NARC and BERT-RE/MRC reaches the maximum when the number of evidence sentences is 3. It is because a 2-layer entity graph is constructed in NARC. Typically,

**Table 4.** F1 score w.r.t. NA query ratio. Both *0x* and *All* indicate no negative sampling process, the former does not use NA query, while the latter uses all NA queries.

| | 0x | 1x | 2x | 3x | All |
|---|---|---|---|---|---|
| BERT-MRC | 67.32 | 63.33 | 59.74 | 57.66 | 57.59 |
| NARC | 68.06 | 64.41 | 61.29 | 59.70 | 59.84 |

**Table 5.** Computational cost analysis on DocRED dev set. For the test time, we execute 5 independent runs and report the average value for each model.

| | BERT-RE | BERT-MRC | NARC |
|---|---|---|---|
| Para. Num. | 114.3M | 114.3M | 127.4M |
| Test Time | 134.3s | 408.8s | 416.2s |

considering there are three entities (*head entity*, *tail entity*, and a *relay entity*) distributed in three sentences, the reasoning chain *head-relay-tail* could be exactly achieved by two times message propagation. From this viewpoint, it is a natural phenomenon that the gap gradually decreases with the further increase of evidence sentence numbers.

**NA Influence Analysis.** NA query is an unexpected problem that arises after formulating document-level relation extraction as machine reading comprehension, and we assume that it drags down the model performance. In this experiment, we conduct random negative sampling for NA queries based on the number of non-NA queries in each document, and report the results conditioned on different negative ratios on the development set, as summarized in Table 4. We observe that NARC achieves relatively similar performance with BERT-MRC when there is no negative instance (0x). With the increase of the proportion of negative samples, the performance gap increases gradually, demonstrating that NARC is effective in mitigating the negative effect of having too many NA queries.

### 4.6 Computational Cost Analysis

While BERT-RE runs document modeling only once to create general representations and extract all possible relational facts, NARC enumerates the document $n_e$ times to establish representations specific to each target head entity. This means NARC is more time-consuming than BERT-RE in theory ($\mathcal{O}(n_e)$ vs. $\mathcal{O}(1)$), To study the actual computational cost, we run them on the DocRED development set with the same setting and present the results in Table 5. The test time of BERT-MRC and NARC is very close, both about 3 times of BERT-RE. This is an acceptable result, because intuitively speaking, the time overhead of BERT-MRC seems to be 20 times that of BERT-RE (there is an average of 20 entities in a document). We assume the reason is that the inference complexity of BERT-MRC is one order less than that of BERT-RE ($\mathcal{O}(n_e)$ vs. $\mathcal{O}(n_e^2)$).

Through further investigation, we find that BERT-RE needs to enumerate and preprocess all possible entity pairs for each input in the dataloader, which is an extraordinarily time-consuming process, accounting for 85% of the test time. If we only calculate the inference time without considering the data processing, BERT-MRC takes about 0.7s and BERT-RE takes about 1.2s for a batch. Moreover, we may also prune some queries to further accelerate in real application since some types of entities may not become the head entity. Taken altogether, NARC is not as time-consuming as expected. It sacrifices a little efficiency in exchange for a substantial performance improvement.

## 5 Related Work

This work builds on a rich line of recent efforts on relation extraction and machine reading comprehension models.

**Relation Extraction** Relation extraction is always a research hotspot in the field of data mining and natural language processing. Early approaches mainly focus on the sentence-level relation extraction [33–36], which aims at predicting the relation label between two entities in a sentence. This kind of method does not consider interactions across mentions and ignores relations expressed across sentence boundaries. Afterward, many researchers show interest in the cross-sentence relation extraction problem [9, 18, 22], yet they restrict all relation candidates in a continuous sentence span with a fixed length. Meanwhile, there are also some efforts to expand the extraction scope to the entire document in the biomedical domain but only considering a few relations [3, 4, 37]. However, these idealized settings make their solutions not suitable for complex and diversified real-world scenarios. Recently, Yao et al. [31] propose DocRED, a large-scale document-level relation extraction dataset with 96 relations that constructed from Wikipedia and Wikidata. Nowadays, the document-level relation extraction task has attracted a lot of researchers' interest [17, 24, 26, 32].

In the long history of relation extraction, how to fully capture the specific information related to the target entities is an eternal topic. Wang et al. [27] propose diagonal attention to depict the strength of connections between entity and context for sentence-level relation classification. He et al. [10] first utilize intra- and inter-sentence attentions to learn syntax-aware entity embedding, and then combine sentence and entity embedding for distantly supervised relation extraction. Li et al. [15] incorporate an entity-aware embedding module and a selective gate mechanism to integrate task-specific entity information into word embeddings. Beyond that, Jia et al. [12] propose an entity-centric, multi-scale representation learning on a different level for $n$-ary relation extraction. However, due to a large number of entity pairs in documents, there are few works to consider entity-specific text-modeling in document-level relation extraction.

**Machine Reading Comprehension** Machine reading comprehension is a general and extensible task form, and many tasks in natural language processing can

be framed as reading comprehension: Li et al. [13] propose a MRC-based unified framework to handle both flat and nested named entity recognition. Li et al. [14] formulate the entity-relation extraction task as a multi-turn question-answering problem. The most similar task to document-level relation extraction is multi-hop machine reading comprehension [29], which takes (*head entity, relation, ?*), not utterance, as query. The last few years have witnessed significant progress on this task: Typically, De Cao et al. [5] introduce Entity-GCN, which takes entity mentions as nodes and learns to answer questions with graph convolutional networks. On this basis, Cao et al. [2] apply bi-directional attention between graph nodes and queries to learn query-aware representation for reading comprehension. This success inspires us to pay more attention to the interaction between query and document, along with the reasoning process in multi-hop relations.

In the formulation of multi-hop machine reading comprehension, every query could retrieval an accurate answer from its candidate list, which is inconsistent with the scenario of document-level relation extraction. Recently, Rajpurkar et al. [19] release SQuAD 2.0 by augmenting the SQuAD dataset with unanswerable questions, which officially opens the curtain for solving unanswerable questions in span-based machine reading comprehension. Then some approaches for the challenging problem are proposed: Sun et al. [23] present a unified model with a no-answer pointer and answer verifier to predict whether the question is answerable. Hu et al. [11] introduce a read-then-verify system to check whether the extracted answer is legitimate or not. However, considering the technical gap between span-based and multi-hop reading comprehension (i.e., a sequence labeling problem vs. a classification problem), how to deal with numerous no-answer queries is still an open problem after we transform the document-level relation extraction into the paradigm of machine reading comprehension.

## 6 Conclusion

In this paper, we propose a NA-aware MRC model for document-level relation extraction, connecting the relation extraction problem to the well-studied machine reading comprehension field. The proposed approach facilitates the model focusing on the context related to each given head entity in the document, and yields significant improvements compared to the conventional solution. Interesting future work directions include employing other advanced pre-trained language models (e.g., DeFormer[1], Roberta [16]) to further improve the efficiency and performance, as well as adapting the proposed paradigm and model to other knowledge-guided tasks in information extraction (e.g., event extraction).

## Acknowledgments

We would like to thank all reviewers for their insightful comments and suggestions. The work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences (grant No.XDC02040400), and the Youth Innovation Promotion Association of Chinese Academy of Sciences (grant No.2021153).

## References


1. Cao, Q., Trivedi, H., Balasubramanian, A., Balasubramanian, N.: Deformer: Decomposing pre-trained transformers for faster question answering. In: Proc. of ACL. pp. 4487–4497 (2020)
2. Cao, Y., Fang, M., Tao, D.: Bag: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In: Proc. of NAACL. pp. 357–362 (2019)
3. Christopoulou, F., Miwa, M., Ananiadou, S.: A walk-based model on entity graphs for relation extraction. In: Proc. of ACL. pp. 81–88 (2018)
4. Christopoulou, F., Miwa, M., Ananiadou, S.: Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In: Proc. of EMNLP. pp. 4927–4938 (2019)
5. De Cao, N., Aziz, W., Titov, I.: Question answering by reasoning across documents with graph convolutional networks. In: Proc. of NAACL. pp. 2306–2317 (2019)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of NAACL. pp. 4171–4186 (2019)
7. Feng, R., Yuan, J., Zhang, C.: Probing and fine-tuning reading comprehension models for few-shot event extraction. arXiv preprint arXiv:2010.11325 (2020)
8. Guo, Z., Zhang, Y., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: Proc. of ACL. pp. 241–251 (2019)
9. Gupta, P., Rajaram, S., Schütze, H., Runkler, T.: Neural relation extraction within and across sentence boundaries. In: Proc. of AAAI. pp. 6513–6520 (2019)
10. He, Z., Chen, W., Li, Z., Zhang, M., Zhang, W., Zhang, M.: See: Syntax-aware entity embedding for neural relation extraction. In: Proc. of AAAI. pp. 5795–5802 (2018)
11. Hu, M., Wei, F., Peng, Y., Huang, Z., Yang, N., Li, D.: Read+verify: Machine reading comprehension with unanswerable questions. In: Proc. of AAAI. pp. 6529–6537 (2019)
12. Jia, R., Wong, C., Poon, H.: Document-level n-ary relation extraction with multiscale representation learning. In: Proc. of NAACL. pp. 3693–3704 (2019)
13. Li, X., Feng, J., Meng, Y., Han, Q., Wu, F., Li, J.: A unified mrc framework for named entity recognition. In: Proc. of ACL. pp. 5849—-5859 (2019)
14. 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: Proc. of ACL. pp. 1340–1350 (2019)
15. Li, Y., Long, G., Shen, T., Zhou, T., Yao, L., Huo, H., Jiang, J.: Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. In: Proc. of AAAI. pp. 8269–8276 (2020)
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
17. Nan, G., Guo, Z., Sekulić, I., Lu, W.: Reasoning with latent structure refinement for document-level relation extraction. In: Proc. of ACL. pp. 1546–1557 (2020)
18. Peng, N., Poon, H., Quirk, C., Toutanova, K., Yih, W.t.: Cross-sentence n-ary relation extraction with graph lstms. TACL **5**, 101–115 (2017)
19. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad. In: Proc. of ACL. pp. 784–789 (2018)

20. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Proc. of ESWC. pp. 593–607 (2018)
21. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603 (2016)
22. Song, L., Zhang, Y., Wang, Z., Gildea, D.: N-ary relation extraction using graph-state lstm. In: Proc. of EMNLP. pp. 2226–2235 (2018)
23. Sun, F., Li, L., Qiu, X., Liu, Y.: U-net: Machine reading comprehension with unanswerable questions. arXiv preprint arXiv:1810.06638 (2018)
24. Tang, H., Cao, Y., Zhang, Z., Cao, J., Fang, F., Wang, S., Yin, P.: Hin: Hierarchical inference network for document-level relation extraction. In: Proc. of PAKDD. pp. 197–209 (2020)
25. Wang, D., Hu, W., Cao, E., Sun, W.: Global-to-local neural networks for document-level relation extraction. In: Proc. of EMNLP. pp. 3711–3721 (2020)
26. Wang, H., Focke, C., Sylvester, R., Mishra, N., Wang, W.: Fine-tune bert for docred with two-step process. arXiv preprint arXiv:1909.11898 (2019)
27. Wang, L., Cao, Z., De Melo, G., Liu, Z.: Relation classification via multi-level attention cnns. In: Proc. of ACL. pp. 1298–1307 (2016)
28. Wang, Z., Mi, H., Ittycheriah, A.: Sentence similarity learning by lexical decomposition and composition. In: Proc. of COLING. pp. 1340–1349 (2016)
29. Welbl, J., Stenetorp, P., Riedel, S.: Constructing datasets for multi-hop reading comprehension across documents. TACL **6**, 287–302 (2018)
30. Wu, W., Wang, F., Yuan, A., Wu, F., Li, J.: Coreference resolution as query-based span prediction. In: Proc. of ACL. pp. 6953—-6963 (2020)
31. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., Sun, M.: Docred: A large-scale document-level relation extraction dataset. In: Proc. of ACL (2019)
32. Ye, D., Lin, Y., Du, J., Liu, Z., Sun, M., Liu, Z.: Coreferential reasoning learning for language representation. In: Proc. of EMNLP (2020)
33. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proc. of EMNLP. pp. 1753–1762 (2015)
34. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proc. of COLING. pp. 2335–2344 (2014)
35. Zhang, Y., Zhong, V., Chen, D., Angeli, G., Manning, C.D.: Position-aware attention and supervised data improve slot filling. In: Proc. of EMNLP. pp. 35–45 (2017)
36. Zhang, Z., Shu, X., Yu, B., Liu, T., Zhao, J., Li, Q., Guo, L.: Distilling knowledge from well-informed soft labels for neural relation extraction. In: Proc. of AAAI. pp. 9620–9627 (2020)
37. Zhang, Z., Yu, B., Shu, X., Liu, T., Tang, H., Wang, Y., Guo, L.: Document-level relation extraction with dual-tier heterogeneous graph. In: Proc. of COLING. pp. 1630–1641 (2020)