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

Zhenyu Zhang
Institute of Information Engineering, Chinese Academy of Sciences
zhangzhenyu1996@iie.ac.cn

Bowen Yu
Institute of Information Engineering, Chinese Academy of Sciences
yubowen@iie.ac.cn

Tingwen Liu*
Institute of Information Engineering, Chinese Academy of Sciences
liutingwen@iie.ac.cn

## ABSTRACT

Document-level relation extraction aims to identify relational facts 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 remove irrelevant information while keep relevant content for the target entity pair. In this paper, we tackle the problem by formalizing document-level relation extraction as a machine reading comprehension task. The final paradigm, H-RT, takes each (head) entity in a document as query and all (tail) entities as candidate answers, and performs relation classification for each candidate tail entity. Inevitably, many queries have no answer after such task formulation since there are quite a number of entity pairs without semantic relation in a document. To this end, we add an artificial answer NO-ANSWER (NA) and propose NARC, a NA-aware machine Reading Comprehension model, based on H-RT. The input sequence formulated as the concatenation of head entity and document is fed into a query-context encoder to obtain comprehensive target-aware representations for each entity. Then, a query-specific NA vector is dynamically generated based on the decomposition and composition of all candidate tail entity representations. Finally, a NA score is calculated to weight the prediction results. Experimental results on DocRED with extensive analysis demonstrate the effectiveness of the NARC model and H-RT paradigm.

## CCS CONCEPTS

• **Natural language processing → Information extraction**;

## KEYWORDS

Relation extraction, Document-level NLP task, Machine reading comprehension, No-answer query

**ACM Reference Format:**
Zhenyu Zhang, Bowen Yu, and Tingwen Liu. 2021. NA-Aware Machine Reading Comprehension for Document-Level Relation Extraction. In *WWW '21: 30th The Web Conference, April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 11 pages. https://doi.org/xx.xx

*Corresponding Author


*WWW '21, April 19–23, 2021, Ljubljana, Slovenia*

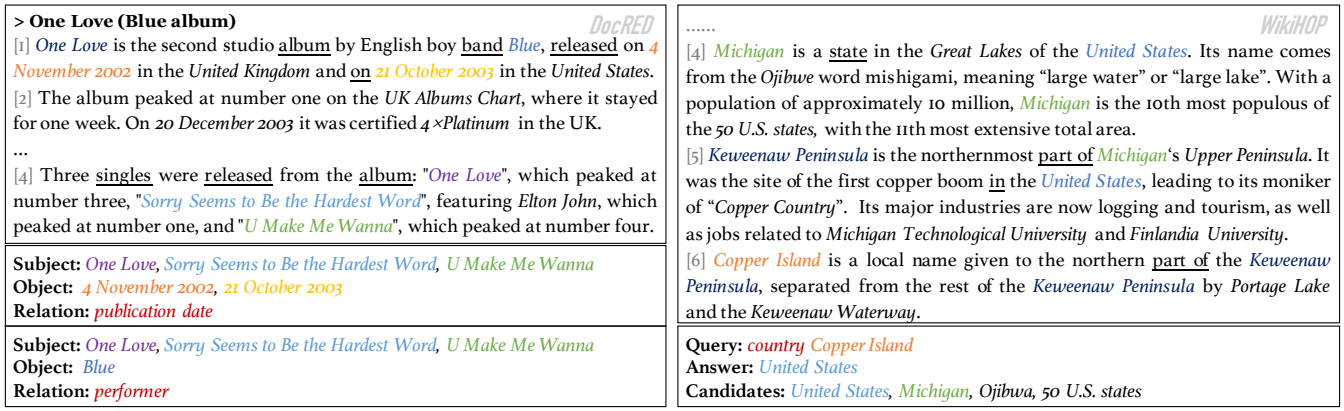
ACM ISBN XX…$XX.XX
https://doi.org/xx.xx

## 1 INTRODUCTION

Reading text to identify and extract relational facts in the form of (*head entity, relation, tail entity*) between pairs of entities is one of the most fundamental tasks in information extraction (IE) and natural language processing (NLP). For quite some time, researchers mainly focus on extracting relational facts among two target entities in a sentence, i.e., sentence-level relation extraction [7, 35, 37]. However, such an ideal setting makes it powerless to handle the 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 [32], in which each document is annotated with a set of named entities and relations, and over 40.7% of the relational facts in the dataset require reading and multi-hop reasoning over multiple sentences. In Figure 1(a), 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 Wanna` is a 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 that devoted to the problem of document-level relation extraction (DRE). Some classic and popular techniques in the well-studied sentence-level relation extraction (e.g., attention mechanism, graph network networks, and pre-trained language models) are introduced and make remarkable improvements [18, 26, 33]. Specifically, most of them take the document as a long word sequence and generate a target-agnostic representation for each token with various encoder, then gather entity representations based on the offsets of mentions and perform relation classification for each entity pair. Despite the great success, we argue that such a general representation learning formulation is suboptimal for extracting relation between specific target entities, since some target-irrelevant words could introduce noise and cause confusion to the prediction.

Inspired by the current trend of formalizing NLP problems as query answering tasks [14, 15, 31], we introduce a new framework that is capable of realizing target-specific document modeling. Instead of treating DRE as a simple entity pair classification problem, we propose to formulate it as a machine reading comprehension (MRC) task. Taking Figure 1(b) as an example, multi-hop MRC [30], the knowledge-driven formulation in MRC, takes (*relation, head entity*) as query and *tail entity* as answer. To comprehensively exploit the interaction between query and context, a popular MRC-style paradigm is to concatenate the query and context as input to pre-trained language models (PLM). By introducing the packed sequence, advanced PLMs such as BERT can encode the document in a




(a) An example from the document-level relation extraction dataset, *DocRED*.

(b) An example from the multi-hop machine reading composition dataset, *WikiHOP*.

**Figure 1: The representative paradigms of DRE and multi-hop MRC. Word spans with the same color indicate the same named entity, and the key clues for relation inference are underlined.**

query-aware manner owning to the sufficiently deep self-attention architectures. Therefore, for a document in DRE, it is straightforward to make it in line with the format of multi-hop MRC, by pairing all (head) entities and pre-defined relations into queries and considering all (tail) entities as candidate answers. Intuitively, we name the paradigm as HR-T, i.e., given the *head entity* and *relation*, the model answers the *tail entity* by reading the document.

However, one concern and barrier in such task formulation is the troublesome *no-answer (NA) problem*. Considering a document with $n_e$ entities and a pre-defined relation set with size $n_r$, HR-T requires $n_e \times n_r$ enumerations and encodings for a complete extraction, in which numerous queries have no correct answer 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 queries pointing to `NO-ANSWER` (NA query) far exceeds that of other queries with correct answers (non-NA query), causing extremely imbalanced data distribution and unsatisfactory extraction efficiency. To mitigate these negative effects of having too many NA queries, we simplify the query generation process in HR-T and present a new paradigm H-RT, in which the relation is discarded and only the head entity is used as query. Our preliminary study on DocRED suggests that this simplification reduces the NA/non-NA query ratio from 1:76 to 1:2. Nevertheless, the NA problem has not been fundamentally finished off, there are still a considerable number of NA queries. Thus the remaining challenge is: how to guide H-RT to effectively distinguish NA and non-NA queries.

To address this challenge, we propose an **NA**-aware machine **R**eading **C**omprehension (NARC) model based on the H-RT paradigm. The NARC model has two major modules: a query-context encoder and an answer vector assembler. The former aims at learning neural representations for entities, and the latter generating proper vector for each candidate answer (including `NO-ANSWER`). Specifically, the input sequence is formulated as `"[CLS] + Head Entity + [SEP] + DOCUMENT + [SEP]"` and then fed into the query-context encoder, which is made up of a PLM followed by a stacked text-entity graph (TEG). The PLM serves the target-aware context encoding, while the TEG is constructed to aggregate evidence scattered around multiple sentences and perform multi-hop reasoning. Afterward, the answer vector assembler first integrates features from PLM and TEG as the final representation of each entity, then a decomposition-composition strategy is proposed for the vectorization of the man-made candidate `NO-ANSWER`, where each vector of candidate tail entity is first decomposed into relevant and irrelevant components with respect to the head entity, and then composed to the query-specific NA vector. Finally, this vector is projected into a NA score to weight the prediction.

The **contributions** of our work are summarized as follows: (1) We propose a MRC-style paradigm H-RT for DRE, which encourages the model to construct target-specific document representation for each head entity. To our best knowledge, it is the first attempt to reformalize DRE as a MRC problem. (2) We develop a NA-aware MRC model to handle the NA problem after task formulation, which is capable of dynamically generating NA vector and further guide the model to perceive NA and non-NA queries. (3) We conduct extensive experiments on the largest public dataset, DocRED. Experimental results and detailed analyses validate the effectiveness of the proposed approach.

## 2 RELATED WORK

Our 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 natural language processing. Early approaches mainly focus on the sentence-level relation extraction [16, 34–37], 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 [8, 19, 24, 38], 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 by

only considering a few relations [3, 4, 13]. However, these idealized settings make their solutions not suitable for the complex and diversified real-world scenarios. Recently, Yao et al. [32] propose DocRED, a large-scale document-level relation extraction dataset with 96 relations. It is constructed from Wikipedia and Wikidata, and is oriented to the most primitive and general text data. Nowadays, the document-level relation extraction task has attracted a lot of researchers' interest [18, 26, 27, 33].

In the long history of relation extraction, how to fully capture the specific information related to the target entities is an eternal topic. For example, Wang et al. [28] propose diagonal attention to depict the strength of connections between entity and context for sentence-level relation classification. He et al. [9] 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. Soares et al. [23] augment the original sentence with four reserved tokens to mark the begin and end of two entities in BERT. Beyond that, Jia et al. [11] propose an entity-centric, multi-scale representation learning on different level for *n*-ary relation extraction. However, due to the large number of entity pairs in documents, there is few work 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. [14] propose a MRC-based unified framework to handle both flat and nested named entity recognition. Li et al. [15] 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 [30], 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 experimental setting 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. [20] 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. [25] present a unified model with no-answer pointer and answer verifier to predict whether the question is answerable. Hu et al. [10] 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.

# 3 TASK FORMULATION

In this section, we first briefly recall some basic concepts and classic baselines for DRE, and then describe the transformation from DRE to MRC. Finally, we propose the H-RT paradigm based on the integration consideration of training time and performance.

## 3.1 Preliminaries

Given an annotated document $\mathcal{D} = \{w_i\}_{i=1}^{n_w}$ and its corresponding 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 DRE is to predict all the intra- and inter-sentence 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.

Typically, previous work [32] first takes the input document as a long word sequence $[x_1, x_2, ..., x_{n_w}]$, and then encodes them into hidden states $[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{n_w}]$ with CNN/LSTM/BiLSTM:

$$[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{n_w}] = Encoder([x_1, x_2, ..., x_{n_w}]). \tag{1}$$

The relation prediction in DRE is treated as a multi-label entity pair classification problem. Specially, for each entity pair $(e_i, e_j)$, the previous work gathers entity representations from the hidden state sequence and utilizes a bilinear layer with sigmoid function to compute the probability for each relation type:

$$\mathbf{r}_{ht} = \delta(\mathbf{e}_h \mathbf{W}_o \mathbf{e}_t + b_o), \quad e_h \in \mathcal{E}, e_t \in \mathcal{E}, \tag{2}$$

where $\mathbf{e}_h \in \mathbb{R}^d$ and $\mathbf{e}_t \in \mathbb{R}^d$ are the hidden states of current head entity and tail entity, $\mathbf{W}_o \in \mathbb{R}^{d \times n_r \times d}$ and $b_o \in \mathbb{R}^{n_r}$ are trainable parameters in the bilinear layer, $d$ is the dimension of hidden states, $\delta$ is the *sigmoid* function, each dimension of $\mathbf{r}_{ht} \in \mathbb{R}^{n_r}$ denotes the probability of a relation between these two target entities.

Benefit from the development of pre-trained language models (PLM), Wang et al. [27] propose to utilize BERT [6] for document modeling. To align with the PLM framework, the input sequence is packed to "`[CLS]` + `DOCUMENT` + `[SEP]`", where "`[CLS]`" and "`[SEP]`" are two special tokens in BERT, "`DOCUMENT`" refers the document tokens $[x_1, x_2, ..., x_{n_w}]$. The whole model structure is represented in Figure 2(a). Obviously, this kind of practice only encodes the document once, but the obtained token representations are target-agnostic and the final relation classification needs to enumerate all possible entity pairs by $n_e \times n_e$ times.

## 3.2 DRE as MRC

Generally, DRE focuses on extracting triplet facts (*head entity, relation, tail entity*) in the document. From another viewpoint, it can also be regarded as a latent MRC problem: given a document with annotated entities, the model is expected to answer all relations for each entity pair by reading the document. Coincidentally, there is a similar knowledge-intensive paradigm in multi-hop MRC: given a query (*head entity, relation, ?*), some supporting text, and a set of candidate answers (i.e., *all entities* in the document), the goal is to identify the correct *tail entities* as answers. In this way, DRE could

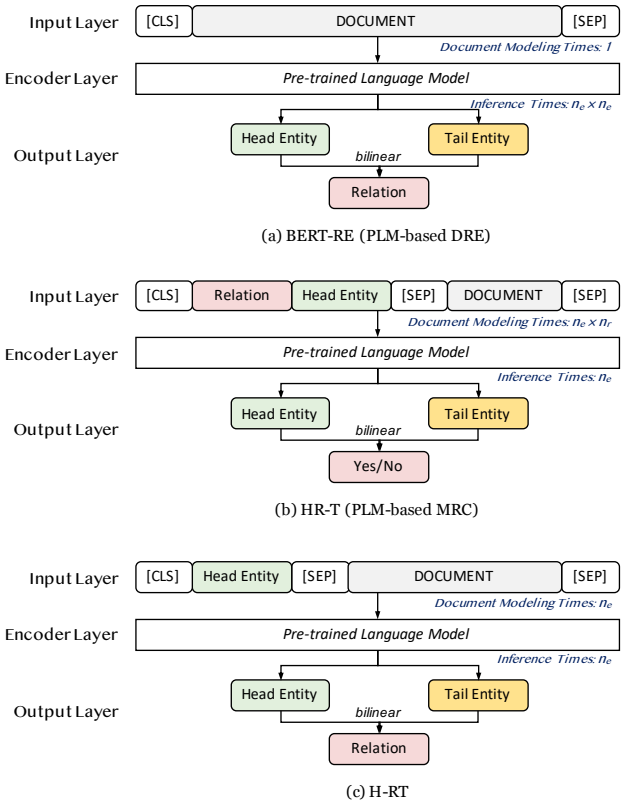



**Figure 2: Diagrams of three paradigms for DRE with PLM. (a) The most traditional model. (b) The MRC-style model after task formulation. (c) The proposed H-RT paradigm.**

be naturally transformed into the MRC paradigm by enumerating *head entities* and *relations* as a query.

*3.2.1 Data Transformation.* To formulate the DRE dataset as a set of (*query, answer, context*) triples, we generate queries by freely combining all entities and relations for each document, then the candidate answer list consists of all entities in the document. Considering a pre-defined relation set with $n_r$ relations, there will be $n_r \times n_e$ queries for a document. In other words, a document in DRE dataset will be expanded to $n_r \times n_e$ instances in the new MRC-style dataset. In the standard MRC formulation, an entity is a correct answer if it is a proper *tail entity* for the query (*head entity, relation, ?*). However, the correct answer number varies from 0 to $n_e$ in the new dataset, as there may be no (or more than one) candidate tail entity that can form a valid relational fact with the given *head entity* and *relation*. To be compatible with the unforeseen situation, we add a special candidate "NO-ANSWER" for all instances. Statistics on DocRED [32] show that more than 98.7% queries point to the NO-ANSWER after data transformation.

*3.2.2 PLM for MRC.* Figure 2(b) shows a simple schematic diagram of using PLM to solve the MRC-style tasks. Each training input is organized by concatenating the query together with document tokens to form: "[CLS] + Relation + Head Entity + [SEP] + DOCUMENT + [SEP]", which is then fed into PLM. Conventionally, we use tokens in the first mention of the head entity to generate query. Similar to BERT-RE, we extract out the head entity embedding and a set of tail entity embeddings from the PLM output, and then feed them into a bilinear layer with sigmoid activation function to predict the probability of current relation $r$ between the head entity $e_h$ and $t$-th tail entity $e_t$, as follows

$$y_t = \delta(\mathbf{e}_h \mathbf{W}_o^{'} \mathbf{e}_t + b_o^{'}), \quad e_t \in \mathcal{E}, \tag{3}$$

where $\mathbf{W}_o^{'} \in \mathbb{R}^{d \times 1 \times d}$ and $b_o^{'} \in \mathbb{R}$ are trainable parameters. It is the largest divergence between PLM-based MRC and PLM-based DRE in output layer, since the relation $r$ is unique and specified in query for MRC. Therefore, if $(e_h, e_t)$ holds the relation $r$, $y_t$ is expected to be 1, otherwise to be 0.

For PLM-based MRC, the key point is to take query and document as a sequence and compute the self-attention over both query and document. However, due to the concatenation in the input layer, this approach has to repeatedly encode a document with each of the generated queries. The number of document modeling times is as high as $n_r \times n_e$ since there are $n_r \times n_e$ queries for a document. More than that, for a (*query, document*) pair, it still needs to traverse all the candidate entities to find correct tail entities in the output layer, causing it to be quite time-consuming and low-efficiency.

## 3.3 H-RT Paradigm

In general domain, the relations involved in a document are only a small part of the pre-defined relation set (about 6% in DocRED). It means that the enumeration of head entities and relations in HR-T introduces a large number of *no-answer (NA) queries*, which is pointless to the final extraction. Meanwhile, the extreme imbalance of training samples will damage the model performance, in a great measure. Furthermore, in the output layer of BERT-RE, a bilinear layer is employed to predict all relations, while in HR-T, the bilinear layer only serves to predict the relation designated in the query. That is to say, the enumeration of relations in the query does not reduce the complexity of inference.

Following the valuable guidance of MRC, we propose a reformed paradigm named H-RT to reduce the dispensable enumeration and alleviate the adverse impact caused by excessive NA queries. (see Figure 2(c)). It is an ingenious trade-off between PLM-based MRC and PLM-based DRE. Specifically, the input layer of H-RT follows the basic idea of MRC, while only enumerate each head entity as query, and the output layer remains the same as PLM-based DRE. This paradigm can be understood as a hope that the model is able to answer a pseudo question by reading supporting context: "*which entities in the document have what semantic relations with the given head entity?*". Formally, the input sequence is modified to "[CLS] + Head Entity + [SEP] + DOCUMENT + [SEP]", then PLM receives the combined sequence and outputs a context representation matrix, from which we can compute the representations for entities and predict tail entities as well as corresponding relations:

$$\mathbf{r}_{ht} = \delta(\mathbf{e}_h \mathbf{W}_o \mathbf{e}_t + b_o), \quad e_t \in \mathcal{E}, \tag{4}$$

where $\mathbf{r}_{ht} \in \mathbb{R}^{n_r}$ denotes the probabilities of all relations between the target head entity and $t$-th tail entity. Intuitively, the paradigm allows the fine-tuning process of internal representation from PLM pay more attention to the contexts related to the head entity.

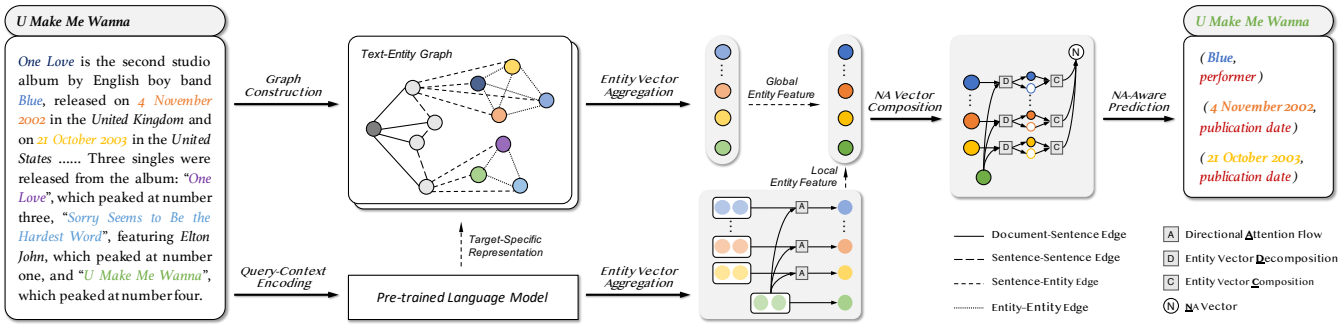



**Figure 3: Overview of the NARC model. The model receives a document and a target *head entity* at a time, and outputs all the related (*tail entity, relation*) pairs. Here we take *U Make Me Wanna* (symbols with green color) as an example. For a good visualization, we only exhibit some representative entities and omit the full-connected edges in the figure.**

There are also some other options that could benefit from the powerful target-aware modeling capability of PLM. A more radical idea is directly taking entity pairs as queries, but such a method would scale quadratically with the number of entities in a document and also result in unbearable training time. For the reformation of HR-T (i.e., PLM-based MRC), another possible paradigm is R-HT. Theoretically, in this paradigm, the number of document modeling times is $n_r$ and the number of inference times is $n_e \times n_e$. Obviously, it still faces similar problem with HR-T: the increase of document modeling times not only has no meaning to the decrease of inference times, but also brings about troublesome no-answer queries. First and last, although the document modeling times of H-RT is slightly higher than that of PLM-based DRE, it strikes an adequate balance between the performance and training time.

# 4 NA-AWARE MRC (NARC)

This section lays out NARC, a **NA**-aware M**RC** model follows the H-RT paradigm. We first expound our key motivation, then give an overview of the model and introduce the details of each component. Lastly, we show the process of model learning and inference.

**Motivation.** H-RT is an improved version of HR-T, which only takes the *head entity* instead of (*head entity, relation*) pair as query. The simple modification efficiently reduces the magnitude of document modeling and the ratio of no answer queries drops to 65.5% in DocRED. Despite all this, the NA problem is only partially alleviated, but not fundamentally resolved. Enlightened by the successful efforts in unanswerable span-based MRC [10, 25], we hypothesis that making the model aware that whether there is no answer for the target head entity facilitates accurate relation prediction. With this in mind, it is intuitive to explore "NA-aware MRC" for the new paradigm of DRE. Different from other candidate answers that point to specific entities, `NO-ANSWER` is a man-made option without corresponding representation. To meet this challenge, we compose the NA vector based on the features of all candidate tail entities, because `NO-ANSWER` has a special meaning that all the tail entities are incompatible with the head entity. Further, all candidate entity vectors, together with the generated NA vector, are utilized to train the NA-aware MRC model for DRE.

## 4.1 Model Overview

The NARC model consists of a query-context encoder (QCE), an answer vector assembler (AVA), and a NA-aware predictor (NAP). Among them, QCE and AVA are the main modules in NARC, which serve the vectorization of document and candidate answers respectively. As shown in Figure 3, we follow the H-RT paradigm to feed (*head entity, document*) pairs into PLM, then the vectorized document tokens pass through a stacked text-entity graph to derive semantic evidence from the document and enable multi-hop reasoning (Section 4.2). Next, the directional attention flow is introduced to aggregate the local features for tail entities based on the mention representation of PLM. The results are combined with the entity representation (i.e., global features) in TEG 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 NA vector (Section 4.3). Finally, the NA vector is merged into the candidate list, and the NA score is calculated based on the vector to weight the prediction (Section 4.4). In this way, the model could induce low confidence when there is no answer present due to the dominance of irrelevant components in the NA vector.

## 4.2 Query-Context Encoder

The great success of integrating GNNs with PLM in multi-hop MRC inspired us to construct a graph for multi-hop reasoning in DRE. Following the H-RT paradigm, the document (*context*) is concatenated with the head entity (*query*) and fed into PLM to achieve target-aware representation for each token. Then a Text-Entity Graph (TEG) is proposed following the design philosophy of Entity Graph [5]. It extends the Entity Graph with different granularities of text nodes to enhance the interaction between text and entities. In the rest of this section, we will introduce the construction process and message passing strategy of TEG.

*4.2.1 Construction of TEG.* Based on the inherent structure of the document, we construct TEG with three kinds of nodes: *document, sentence* and *entity*[1]. The document and sentence nodes

[1]In the entity node construction, we do not make a distinction between the head entity (*query*) and tail entity (*candidate answer*).

represent different granularity levels of text, while the entity node is built to enable multi-hop reasoning. We expect our TEG could pick up semantic clues that related to the relation prediction from the document and help locate the correct answer entity nodes with multi-hop relational reasoning. Mean pooling over corresponding words generates vectors for each sentence, and entity, which can be employed to initialize their node representations. Specially, the document node is initialized as the feature of `[CLS]` token which contains target-specific global information.

We define the following types of edges between pairs of nodes to make powerful connection between entity and text in TEG.

- an edge between two entity nodes if they co-occur within the same sentence.
- an edge between two sentence nodes if they are adjacent in the document.
- an edge between an entity node and a sentence node if the mention appear in the sentence.
- an edge between a sentence node and the document node.
- all entity nodes connect with each other.

*4.2.2 Reasoning over TEG.* Typically, graph-based models follow a layer-wise propagation manner that all the nodes update simultaneously in each layer. In this study, we utilize R-GCN [21] to propagates information over TEG, which can handle high-relational data characteristics and make full use of different edge types. Formally, at $l$-th layer, given the hidden state $\mathbf{h}_i^l \in \mathbb{R}^d$ of node $i$ and its neighbors $\mathcal{N}_i$ with the corresponding edge types $\mathcal{T}$, R-GCN propagates message across different neighboring nodes and generates transformed representation in the next layer for node $i$ via

$$\mathbf{h}_i^{l+1} = \sigma(\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{N}_i^t}\frac{1}{|\mathcal{N}_i^t|}\mathbf{W}_t^l\mathbf{h}_j^l + \mathbf{W}_s^l\mathbf{h}_i^l), \tag{5}$$

where $\mathbf{W}_t^l \in \mathbb{R}^{d\times d}$ refers a type-specific weight matrix, $\mathbf{W}_s^l \in \mathbb{R}^{d\times d}$ is a general matrix for self-connection.

Furthermore, it has been shown that GNNs usually suffer from the over-smoothing problem if the number of layers is large [12], making different nodes have similar representations and lose the distinction among nodes. To tackle this problem, we add a gating mechanism to control the extent of propagating update message to the next layer, in which the update message $\mathbf{u}_i^l$ can be obtained via Equation 5 without non-linear activation function $\sigma$. The gate-level is computed by $\mathbf{u}_i^l$ and $\mathbf{h}_i^l$ with a linear transformation $\mathcal{F}_g$, and the final representation is defined as a gated combination of previous features and a non-linear transformation of update message:

$$g_i^l = \delta(\mathcal{F}_g([\mathbf{u}_i^l;\mathbf{h}_i^l])), \tag{6}$$

$$\mathbf{h}_i^{l+1} = g_i^l \odot tanh(\mathbf{u}_i^l) + (1-g_i^l)\odot\mathbf{h}_i^l, \tag{7}$$

where $\odot$ stands for element-wise multiplication. Roughly speaking, it controls the amount of message from neighborhoods used in aggregation, and if all necessary information to handle the problems is present at a layer which is not the last, the model is expected to learn to stop using neighboring information.

After $L$ times message passing, all document, sentence, and entity nodes achieve their final representations, we refer to the final entity node vectors $\mathbf{E}^G \in \mathbb{R}^{n_e\times d}$ as global entity features, which encode the semantic information throughout the whole document.

## 4.3 Answer Vector Assembler

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

*4.3.1 Entity Vector Aggregation.* In a document, the same entity could be mentioned multiple times, and the 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 PLM. 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 [22], to measure the difference and compress the mention features into an embedding for each candidate tail entity.

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

$$\mathbf{S}_{ht} = avg_{-1}\mathcal{F}_s([\mathbf{M}_h;\mathbf{M}_t;\mathbf{M}_h\odot\mathbf{M}_t]), \tag{8}$$

in which $\mathbf{M}_h \in \mathbb{R}^{n_m^h\times d}$ 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 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

$$\mathbf{M}_{h2t} = dup(softmax(max_{col}(\mathbf{S}_{ht})))^\top\mathbf{M}_t, \tag{9}$$

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 for entity vector aggregation:

$$\mathbf{e}_h^L = mean(\mathbf{M}_h),\ \ \mathbf{e}_t^L = mean(\mathbf{M}_{h2t}), \tag{10}$$

$$\mathbf{e}_h = [\mathbf{e}_h^L;\mathbf{e}_h^G],\ \ \mathbf{e}_t = [\mathbf{e}_t^L;\mathbf{e}_t^G]. \tag{11}$$

*4.3.2 NA Vector Composition.* To assign an informative vector for the special candidate "`NO-ANSWER`", 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 NA vector, the relevance and irrelevance between the given head entity and each candidate tail entity provide guidance on whether there is a valid answer to the query. The key intuition behind is that `NO-ANSWER` can 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 [29] to meet this demand:

$$\mathbf{e}_t^+ = \frac{\mathbf{e}_h^\top \mathbf{e}_t}{\mathbf{e}_t^\top \mathbf{e}_t}\mathbf{e}_t, \tag{12}$$

$$\mathbf{e}_t^- = \mathbf{e}_t - \mathbf{e}_t^+. \tag{13}$$

The motivation for the linear decomposition 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 component matrix and the irrelevant component matrix for each candidate tail entity, which are then fed to a feed-forward linear layer as follows

$$\mathbf{e}_t^n = tanh(\mathbf{W}_{cr}\mathbf{e}_t^+ + \mathbf{W}_{ci}\mathbf{e}_t^- + b_c), \tag{14}$$

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. Then we apply a max-pooling over all candidate tail entities to obtain the representation of NO-ANSWER:

$$\mathbf{n} = max\{\mathbf{e}_t^n\}_{t=1}^{n_e}. \tag{15}$$

Note that the decomposition and composition of lexical semantics over sentences has been successfully used in the past for sentence similarity learning [28], in which each token is decomposed into similar and dissimilar components based on its high-level semantic features. In this study, we find that the MRC-style paradigm can also be generalized under the basic idea of vector decomposition and composition, because the candidate tail entity is partially related to the head entity, and based on the level of relevance, it either increases or decreases the chance of finding a valid answer. The most difference is that we decompose each candidate tail entity vector with respect to the representation of head entity, rather than high-level features of itself, and then composed them to generate a NA vector for each (*query, context*) pair.

## 4.4 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 a final relation prediction based on the perception. Moreover, "NO-ANSWER (NA)" is regarded as a special candidate entity, which takes the NA vector as representation, thus introducing additional negative examples to guide the model optimization.

Firstly, we pass the NA vector through a linear transformation $\mathcal{F}_o$ to obtain a score that points to no answer for the given query. Next, the NA score is combined with the relational logits in Equation 4 as an auxiliary weight to achieve a NA-aware prediction:

$$s_n = \delta(\mathcal{F}_n(\mathbf{n})), \tag{16}$$

$$\mathbf{r}_{ht} = \begin{cases} \delta((1-s_n)\odot(\mathbf{e}_h\mathbf{W}_o\mathbf{e}_t + b_o)), & \text{if} \quad e_t \in \mathcal{E}, \\ \delta(s_n \odot (\mathbf{e}_h\mathbf{W}_o\mathbf{n} + b_o)), & \text{if} \quad e_t = \text{NA}. \end{cases} \tag{17}$$

*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:

$$\mathcal{L} = -\sum_{i=1}^{n_r}\left(y_{ht}^i \cdot \log(r_{ht}^i) + (1-y_{ht}^i)\cdot\log(1-r_{ht}^i)\right) \tag{18}$$

**Table 1: Statistics for the DocRED dataset. *# Enum.* refers the enumeration number in the H-RT paradigm.**

| | # Doc. | # Enum. | # Fact | # Pos. Pair | # Neg. Pair | # Rel. |
|---|---|---|---|---|---|---|
| Train | 3,053 | 59,493 | 34,715 | 38,269 | 1,160,470 | 96 |
| Dev | 1,000 | 19,578 | 11,790 | 12,332 | 384,467 | 96 |
| Test | 1,000 | 19,539 | 12,101 | 12,842 | 379,316 | 96 |

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 $e_t$ = NO-ANSWER.

Following previous work [32], 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 query, in order to obtain all relational facts over the document.

# 5 EXPERIMENTS

In this section, we first introduce the dataset and implementation details. Then, we compare NARC and H-RT with some representative baselines to illustrate the effectiveness. Finally, we present extensive experiments with discussion and analysis to investigate the performance and efficiency of our proposed approach.

## 5.1 Experiment Setup

*5.1.1 Dataset.* We evaluate our model on the public benchmark dataset, DocRED [32]. It is constructed from Wikipedia and Wikidata, covers a broad range of categories with 96 relation types, and is the largest human-annotated dataset for general domain DRE. 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. For more details about the dataset, we recommend readers to reference the original paper [32].

*5.1.2 Implementation Details.* We implement NARC with PyTorch 1.4.0 and BERT-base model. The concatenated sequence in the input layer is trimmed to a maximum length of 512. The embedding size of BERT is 768, and a linear-transformation layer is utilized to project the BERT embedding into a low-dimensional space with the same size of hidden state, which is set to 200 (chosen from [100, 150, 200, 250]). Besides, the layer number of TEG is set to 2 (chosen from [1, 2, 3, 4]), the batch size is set to 10 (chosen from [5, 8, 10, 12]), and the learning rate is set to $1e^{-5}$ (chosen form $1e^{-4}$ to $1e^{-6}$). We optimize our model with Adam and run it on a single 16G Tesla V100 GPU for 50 epochs (which takes about one day). All hyper-parameters are tuned based on the development set.

*5.1.3 Evaluation Metric.* Following popular choices and previous work, we choose micro F1 and micro Ign F1 as the evaluation

**Table 2: Results on the DocRED dataset, bold marks highest number among all models. † marks *PLM-based DRE*, the most important baseline, and the results are computed based on our re-implementation with the official repository. For other baselines, we directly utilize the results of Intra F1 and Inter F1 reported by Nan et al. [18].**

| Model | Dev | | | | Test | |
|---|---|---|---|---|---|---|
| | Intra F1 | Inter F1 | Ign F1 | F1 | Ign F1 | F1 |
| CNN [32] | 51.87 | 37.58 | 41,58 | 43.45 | 40.33 | 42.26 |
| LSTM [32] | 56.57 | 41.47 | 48.44 | 50.68 | 47.71 | 50.07 |
| BiLSTM [32] | 57.05 | 43.49 | 48.87 | 50.94 | 48.78 | 51.06 |
| ContextAware [32] | 56.74 | 42.26 | 48.94 | 51.09 | 48.40 | 50.70 |
| AGGNN [7] | 58.76 | 45.45 | 46.29 | 52.47 | 48.89 | 51.45 |
| EoG [3] | 58.90 | 44.60 | 45.94 | 52.15 | 49.48 | 51.82 |
| BERT-RE [27]† | 60.89 | 46.55 | 52.04 | 54.18 | 51.44 | 53.60 |
| BERT-TwoPhase [27] | 61.80 | 47.28 | - | 54.42 | - | 53.92 |
| BERT-HIN [26] | - | - | 54.29 | 56.31 | 53.70 | 55.60 |
| BERT-Coref [33] | - | - | 55.32 | 57.51 | 54.54 | 56.96 |
| BERT-LSR [18] | 65.26 | 52.05 | 52.43 | 59.00 | **56.97** | 59.05 |
| H-RT | 64.22 | 50.02 | 55.49 | 57.59 | 54.86 | 57.13 |
| NARC | **66.04** | **52.28** | **57.43** | **59.56** | 56.71 | **59.17** |

metrics. Ign F1 denotes F1 excluding relational facts that appear in both the training set and the development or test set. F1 scores for intra- and inter- sentence entity pairs are also reported[2].

## 5.2 Performance Comparison

Recall that we propose NARC based on a new MRC-style paradigm for DRE, and the corresponding baseline is undoubtedly the PLM-based DRE model. To further demonstrate the competitiveness of our proposed model, we also compare them with a series of current state-of-the-art methods and show the main results in Table 2.

*5.2.1 Baselines.* We compare H-RT and NARC with the following two types of baselines, in which H-RT is the new MRC-style paradigm described in Section 3.3 and NARC is our ultimate model.

**Baselines w/o BERT**: On this track, we select six representative baseline models without using BERT.

- CNN/LSTM/BiLSTM/ContextAware [32]: These models leverage different neural architectures to encode the document, in which ContextAware utilizes an attention mechanism to absorb the context relational information. They are all text-based models and official baselines released by the authors.
- AGGCN [7]: 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.
- EoG [4]: It constructs an edge-oriented graph and utilizes an iterative algorithm over the graph edges, which is a recent state-of-the-art model in biomedical domain DRE.

**Baselines w/ BERT**: On this track, we select five recent methods on DocRED that adopt *bert-base-uncased* as base encoder.

- BERT-RE [27]: It is the standard form of using BERT for DRE (i.e., PLM-based DRE), which is described in Figure 2(a).
- BERT-TwoPhase [27]: It is a pipeline model with two-steps, which first predicts whether or not two entities have a relation, and then predicts the specific relation.
- BERT-HIN [26]: It aggregates inference information from entity-level, sentence-level, and document-level with a hierarchical inference network to predict target relation.
- BERT-Coref [33]: It comes up with an auxiliary training task to enhance the reasoning ability of BERT by capturing the co-refer relations between noun phrases.
- BERT-LSR [18]: It dynamically constructs a latent document-level graph for information aggregation in the entire document with an iterative refinement strategy.

*5.2.2 Main Results.* 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 model and H-RT paradigm, as well as the motivation of formulating the DRE task as a MRC problem. Besides, all the baseline models without BERT achieve poor performance than BERT-based models, indicating that BERT contains some useful information such as commonsense knowledge to solve this task.

Secondly, H-RT outperforms BERT-RE by a significant margin, even with some extra fine-tuning strategies (BERT-TwoPhase) or auxiliary training tasks (BERT-Coref). We highlight the improvement benefits from two scientific contributions: (1) The query provides external prior evidence. Due to the document length, there are many entities pairs with kinds of relations in a document. With the H-RT paradigm, the query provides a guide for the model to filter information irrelevant to the target head entity. (2) The MRC-style model with PLM captures the interaction between the head entity and the document based on the self-attention structure, which helps to establish target-centric representations and extract information from relevant tokens from the document.

[2]It is an inter-sentence relation if all the mentions of the head and tail entities do not appear in the same sentence.

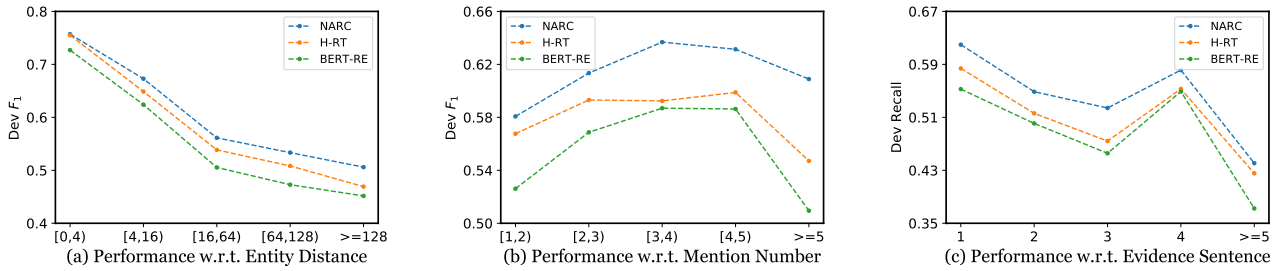



**Figure 4: Performance analysis on (a) detecting long-distance relations, (b) aggregating multiple mention information, and (c) reasoning multi-hop relations. We report the *F1* score for the first two analyses, while report *Recall* for the last one.**

Thirdly, NARC improves by a relative margin of ~2% against H-RT. From the result, we attribute the performance gain to two design choices: (1) The construction of TEG. The challenges of DRE require a model to comprehend multiple sentences and infer the relations among entities by synthesizing relevant information from the entire document, which is beyond the ability of a sequential model. TEG exploits useful structural information among entities and document, and is in the position of capturing richer non-local dependencies and distilling semantic evidence from the context. (2) The composition of `NO-ANSWER` vector. As the training set contains queries with and without valid answers, the vectorization process that 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 NA queries and alleviate its harmful effects.

Lastly, moving on to the intra- and inter-sentence performance, we observe a balanced performance gain when comparing H-RT with BERT-RE, suggesting that H-RT extracts all the semantic evidence related to the query from the document, no matter where the evidence is. In contrast, the difference in F1 score between NARC and H-RT in the inter-sentence setting tends to be larger than that in the intra-sentence setting, suggesting the graph structure included in NARC can better handle long-distance relations.

## 5.3 Performance Analysis

In order to further analyze the performance of our proposed model, we split the DocRED development set into several subsets based on different strategies and report the performance of different models in each subset (see Figure 4).

*5.3.1 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 with the increase of entity distance, which is accordant with human intuition that detecting long-distance relations is still a challenging task for current relation extraction models. Nevertheless, H-RT and NARC consistently outperform BERT-RE by a sizable margin, due to the strong power of BERT in processing MRC-style input. As an exception, H-RT and NARC perform similarly on relations with distance less than 4, since it is an easy task to predict short-distance relations. However, NARC gradually 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 in the entire document.

*5.3.2 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 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, 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 mention number is large, the performance of BERT-RE and H-RT is devastating once again. This is because not every entity mention is involved in the relation, and aggregating information indiscriminately introduces a large amount of noisy context. On the contrary, DAF measures the tail entity mentions and selects the most important one for each head entity mention, so that NARC maintains a relatively high performance when there are many mentions of the entity pair.

*5.3.3 Performance on Various Amounts of Evidence Sentences.* To assess the model's ability in multi-hop reasoning, we plot the preferences 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 H-RT reaches the maximum when the number of evidence sentences is 3. It is because a two-layer TEG is constructed in NARC. Typically, 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 number.

*5.3.4 Performance on Various Amounts of NA Queries.* NA query is an unexpected problem that arises after formulating DRE as MRC,

**Table 3: Performance analysis w.r.t. NA queries. 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 |
|---|---|---|---|---|---|
| HR-T | 76.37 | 72.05 | 69.40 | 66.97 | 53.88 |
| H-RT | 66.81 | 62.13 | 59.64 | 57.66 | 57.59 |
| NARC | 68.06 | 63.41 | 61.39 | 59.73 | 59.56 |

**Table 4: Ablation study on DocRED dev set to investigate the influence of different modules. We remove these modules in order until the simplest model (H-RT) is achieved.**

| | Intra F1 | Inter F1 | Ign F1 | F1 |
|---|---|---|---|---|
| NARC | 66.04 | 52.28 | 57.43 | 59.56 |
| – NAP | 65.33 | 51.90 | 56.86 | 59.01 |
| – NVC | 65.25 | 51.94 | 56.84 | 58.98 |
| – EVA | 64.83 | 51.57 | 56.36 | 58.58 |
| – TEG (H-RT) | 64.22 | 50.02 | 55.49 | 57.59 |

and we assume that it drags down the model performance. In this experiment, we first conduct random negative sampling for NA queries based on the number of non-NA queries in each document, and then employ the sampled data to train and evaluate models. Table 3 summarizes the results of HR-T, H-RT, and NARC. We observe that HR-T achieves the best performance under the same sampling rate, because the query is more informative and can guide the model to achieve more targeted document modeling with the *relation*. Note that the number of NA queries in HR-T is ∼76 times that of non-NA queries, so the performance is close to collapse when using all the data. Another phenomenon is that the improvement of NARC relative to H-RT shows a positive correlation with NA query number. This justifies the first impression of dealing with the NA problem from two aspects: simplifying the query form to reduce the proportion of NA query and entrusting model the ability to distinguish NA and non-NA queries.

## 5.4 Model Ablation Study

To investigate the effectiveness of each module in NARC, we conducted an ablation study in the development set of DocRED. Table 4 reports the comparison results. From these ablations, we observe that: (1) NA-Aware Prediction (NAP) is a necessary component that contributes 0.55% 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. (2) When removing NA Vector Composition (NVC), 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 score for prediction. The principle behind this phenomenon is that merely adding negative samples is not an effective way to boost performance, and extreme negative samples impose an additional burden on the model, even if the generated negative samples are incredibly informative. (3) The operation of Entity Vector Aggregation (EVA) is indispensable since the ablation hurts the final result by 0.40% F1. It verifies the effectiveness of integrating mention-level representations with entity-level representations. The integration combines global and local features for an entity, and the DAF takes into account the fine-grained interaction between mention pairs. All of these contribute to accurate relation prediction, which is reflected in the improvement of inter and intra-sentence performance. (4) Without the Text-Entity Graph (TEG), the performance suffers a sharp deterioration of performance, especially on the inter-sentence pairs. This answers why the combination of graph-based structures and sequential encoders has become a trend in document-level tasks. We also try to remove the text nodes (i.e., the sentence node and document node) in TEG to build a pure entity graph, and the final result of NARC drops to 58.83 in terms of F1. Therefore, we claim that a more advanced structure (such as LSR) will bring further improvement which may be left as future work.

**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 | H-RT | NARC |
|---|---|---|---|
| Para. Num. | 111.7M | 111.7M | 117.4M |
| Test Time | 134.3s | 408.8s | 416.2s |

## 5.5 Computational Cost Analysis

While BERT-RE runs document modeling only once to create general representations and extract all possible relational facts, H-RT enumerates the document $n_e$ times to establish representations specific to each target head entity. This means H-RT is more time-consuming than BERT-RE in theory ($O(n_e)$ vs. $O(1)$), To study the actual computational cost, we run them on the DocRED dev set with the same setting and present the results in Table 5. The test time of H-RT and NARC is very close, both about 3 times of BERT-RE. This is an acceptable result, because intuitively speaking, the time overhead of H-RT 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 H-RT is one order less than that of BERT-RE ($O(n_e)$ vs. $O(n_e^2)$). 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, H-RT takes about 0.7s and BERT-RE takes about 1.2s for a batch. Moreover, we may also prune some query to further accelerate in real application since some types of entities may not become the head entity. Taken altogether, our H-RT paradigm is not as time-consuming as expected. It sacrifices a little efficiency in exchange for a substantial performance improvement.

# 6 CONCLUSION

In this paper, we propose a novel MRC-style paradigm, H-RT, and a NA-aware MRC model for document-level relation extraction, connecting the relation extraction problem to the well-studied machine reading comprehension field. Specifically, we regard the head entity as query while the tail entity as answer, and predict relations

in the output layer. It facilitates the model focusing on the context related to the target head entity in the document. Furthermore, the final representations of all candidate tail entities are decomposed and composed with respect to the head entity for the vectorization of NO-ANWER, which makes the model aware of NA query and alleviates the NA problem caused by task formulation. Interesting future work directions include employing other advanced PLMs (e.g., DeFormer [1], Roberta [17]) to improve the efficiency and performance further, as well as adapting the proposed paradigm and model to other knowledge-guided tasks in information extraction (e.g., event extraction).

## REFERENCES


[1] Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2020. DeFormer: Decomposing Pre-trained Transformers for Faster Question Answering. In *Proceedings ofthe 58th Annual Meeting ofthe Association for Computational Linguistics*. 4487–4497.

[2] Yu Cao, Meng Fang, and Dacheng Tao. 2019. BAG: Bi-directional Attention Entity Graph Convolutional Network for Multi-hop Reasoning Question Answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 357–362.

[3] Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2018. A Walk-based Model on Entity Graphs for Relation Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 81–88.

[4] Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the Dots: Document-level Neural Relation Extraction with Edge-oriented Graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 4927–4938.

[5] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question Answering by Reasoning Across Documents with Graph Convolutional Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2306–2317.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.

[7] Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 241–251.

[8] Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, and Thomas Runkler. 2019. Neural relation extraction within and across sentence boundaries. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. 6513–6520.

[9] Zhengqiu He, Wenliang Chen, Zhenghua Li, Meishan Zhang, Wei Zhang, and Min Zhang. 2018. SEE: Syntax-Aware Entity Embedding for Neural Relation Extraction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 5795–5802.

[10] Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. 2019. Read+Verify: Machine Reading Comprehension with Unanswerable Questions. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. 6529–6537.

[11] Robin Jia, Cliff Wong, and Hoifung" Poon. 2019. Document-Level N-ary Relation Extraction with Multiscale Representation Learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 3693–3704.

[12] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907* (2016).

[13] Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database* 2016 (2016).

[14] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A Unified MRC Framework for Named Entity Recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5849—-5859.

[15] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1340–1350.

[16] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 2124–2133.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[18] Guoshun Nan, Zhijiang Guo, Ivan Sekulić, and Wei Lu. 2020. Reasoning with Latent Structure Refinement for Document-Level Relation Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1546–1557.

[19] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-Sentence N-ary Relation Extraction with Graph LSTMs. *Transactions of the Association for Computational Linguistics* 5 (2017), 101–115.

[20] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 784–789.

[21] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *Proceedings of the 2018 European Semantic Web Conference*. 593–607.

[22] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).

[23] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2895–2905.

[24] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary Relation Extraction using Graph-State LSTM. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2226–2235.

[25] Fu Sun, Linyang Li, Xipeng Qiu, and Yang Liu. 2018. U-net: Machine Reading Comprehension with Unanswerable Questions. *arXiv preprint arXiv:1810.06638* (2018).

[26] Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. 2020. HIN: Hierarchical Inference Network for Document-Level Relation Extraction. In *Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 197–209.

[27] Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. 2019. Fine-tune Bert for Docred with two-step process. *arXiv preprint arXiv:1909.11898* (2019).

[28] Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. 2016. Relation Classification via Multi-Level Attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 1298–1307.

[29] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence Similarity Learning by Lexical Decomposition and Composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*. 1340–1349.

[30] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing Datasets for Multi-hop Reading Comprehension Across Documents. *Transactions of the Association for Computational Linguistics* 6 (2018), 287–302.

[31] Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. Coreference Resolution as Query-based Span Prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6953—-6963.

[32] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

[33] Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

[34] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1753–1762.

[35] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*. 2335–2344.

[36] Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2205–2215.

[37] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 35–45.

[38] Zhenyu Zhang, Xiaobo Shu, Bowen Yu, Tingwen Liu, Jiapeng Zhao, Quangang Li, and Li Guo. 2020. Distilling Knowledge from Well-informed Soft Labels for Neural Relation Extraction. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. 9620–9627.