HIN: Hierarchical Inference Network for Document-Level Relation Extraction

Hengzhu Tang[1,2], Yanan Cao[1], Zhenyu Zhang[1,2], Jiangxia Cao[1,2], Fang Fang[1⋆], Shi Wang[3], and Pengfei Yin[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, China
[3] Institute of Computing Technology, Chinese Academy of Sciences, China
{tanghengzhu,caoyanan,zhangzhenyu1996,caojiangxia,fangfang0703,yinpengfei}@iie.ac.cn
wangshi@ict.ac.cn

Abstract. Document-level RE requires reading, inferring and aggregating over multiple sentences. From our point of view, it is necessary for document-level RE to take advantage of multi-granularity inference information: entity level, sentence level and document level. Thus, how to obtain and aggregate the inference information with different granularity is challenging for document-level RE, which has not been considered by previous work. In this paper, we propose a Hierarchical Inference Network (HIN) to make full use of the abundant information from entity level, sentence level and document level. Translation constraint and bilinear transformation are applied to target entity pair in multiple subspaces to get entity-level inference information. Next, we model the inference between entity-level information and sentence representation to achieve sentence-level inference information. Finally, a hierarchical aggregation approach is adopted to obtain the document-level inference information. In this way, our model can effectively aggregate inference information from these three different granularities. Experimental results show that our method achieves state-of-the-art performance on the large-scale DocRED dataset. We also demonstrate that using BERT representations can further substantially boost the performance.

Keywords: Relation extraction · Hierarchical inference network · Multi granularity

1 Introduction

Relation extraction (RE) aims to detect the semantic relation between entities in plain text, which plays an important role in knowledge base population and natural language understanding. Most previous work focuses on sentence-level RE, i.e., extracting relational facts from a single sentence. In recent years, deep learning models have been widely applied to sentence-level RE and achieved remarkable success [4,16].

⋆ Corresponding Author

Input:
[1] "Nisei" is the ninth episode of the third season of the American science fiction television series The X-Files. [2] It premiered on the Fox network on November 24, 1995. [3] It was directed by David Nutter, and written by ***Chris Carter***, Frank Spotnitz and Howard Gordon. [4] "Nisei" featured guest appearances by Steven Williams, Raymond J. Barry and Stephen McHattie ... [8] The show centers on FBI special agents ***Fox Mulder*** (David Duchovny) and Dana Scully (Gillian Anderson) who work on cases linked to the paranormal, called X-Files …

Subject: ***Chris Carter***
Object: ***Fox Mulder***
Relation: ***creator*** **Supporting Sentences: 1, 3, 8**

Fig. 1. An example from DocRED. Each document in DocRED is annotated with named entity mentions, coreference information, relations, and supporting sentences.

Despite the great success of previous work, sentence-level RE suffers from a serious restriction in practice: a large amount of relational facts are expressed in multiple sentences. Taking Figure 1 as an example, in order to identify the relational fact (*Chris Carter*, *creator*, *Fox Mulder*), one should first identify the fact "Nisei" is an episode of the American science fiction television series from sentence 1, then identify the facts that *Fox Mulder* is a character in "Nisei" and *Chris Carter* is one of the writers of "Nisei" from sentence 8 and 3 respectively. To extract these relational facts, it is necessary to infer and aggregate over multiple sentences. Obviously, most traditional sentence-level RE models often fail to generalize extraction to this situation. To move RE forward from sentence level to document level, many efforts have been made [13,15], but most previous methods used only entity-level information and this is not adequate. Thus, there are still some deep-seated problems unsolved in document-level RE.

To predict the relation between two entities, we argue that the document-level RE model requires taking advantage of multi-granularity inference information: entity level, sentence level and document level. Lets go back to the former example, entity-level inference information is derived from the semantic of all mentions of *Chris Carter* and *Fox Mulder* in the document, sentence-level inference information represents the information related to relational facts in each sentence, document-level inference information aggregates all the necessary information in supporting sentences (sentence 1, 3 and 8) and discards information in noise sentences. Technically, it is clear that document-level RE faces two main challenges: (1) How to obtain the inference information with different granularity; (2) How to aggregate these different granularity inference information and make the final prediction.

In this paper, we propose a new neural architecture, Hierarchical Inference Network (HIN), to tackle above challenges. Specifically, inspired by translation constraint [1], which models a relational fact $r(e_h, e_t)$ with $e_h + r \approx e_t$, we apply this translation constraint to target entity pair. Besides, a bi-affine layer is also used to obtain bilinear representation for the target entity pair. To jointly attend to information from different representation subspaces, we implement the

above two transformations in multiple subspaces in parallel, and acquire entity-level inference information. To obtain the sentence-level inference information, we first apply vanilla attention mechanism to calculate the vector representation for each sentence, which enables our model to pay more attention to the informative words. Then we adopt the semantic matching method which is widely used in natural language inference (NLI) domain to compare the entity-level inference information with each sentence vector. Furthermore, in order to calculate the document-level inference information, we apply a hierarchical BiLSTM and again use attention mechanism to distinguish crucial sentence-level inference information for overall document-level inference representation. Finally, we aggregate inference information of different granularity, the entity-level and document-level inference representations are combined into a fixed-length vector, which is further fed into a classification layer for prediction.

To summarize, we make the following contributions:

1. We propose a Hierarchical Inference Network (HIN) for document-level RE, which is capable of aggregating inference information from entity level to sentence level and then to document level.

2. We conduct thorough evaluation on DocRED dataset. Results show that our model achieves the state-of-the-art performance. We further demonstrate that using BERT representations further substantially boosts the performance.

3. We analyze the effectiveness of our model on different number of supporting sentences and experimental results show that our model performs much better than previous work when the number of supporting sentences is large.

2 Task Description

For document-level RE, the input is a document with annotated entities, as well as multiple occurrences of each entity, i.e., entity mentions, the goal is to identify all the related entity pairs in the document. Following [15], we transform RE into a classification problem. We use upper case letters to represent entities $(E_1, \cdots, E_m)$ and lower case letters to represent mentions $(e_1, \cdots, e_m)$. The RE model is given a relation candidate (E_a, E_b, D) and expected to output the relations between E_a and E_b, where E_a and E_b are entities in the document D.

3 Proposed Approach

Figure 2 gives an illustration of our model. We describe the details of different components in the following sections.

3.1 Input Layer

– **Word Embeddings** In order to capture the meaningful semantic information of words, we map each word into a low-dimensional word embedding vector. The dimension of word embeddings is d_w.

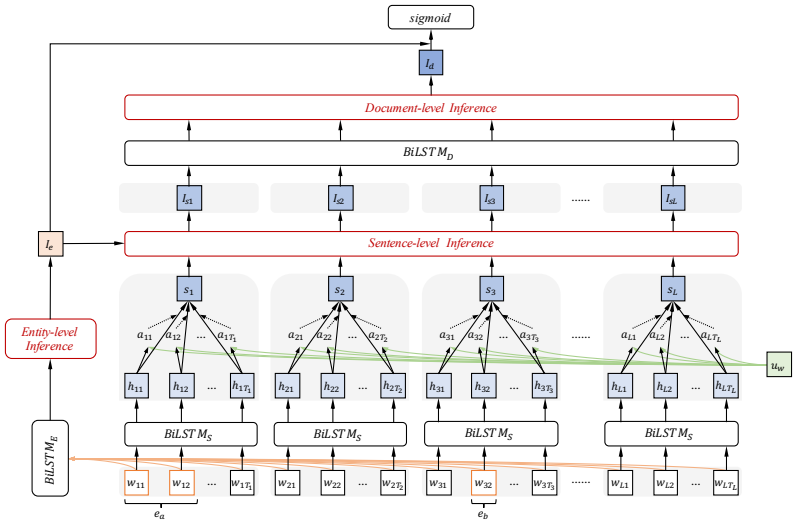

Fig. 2. The overall architecture of the Hierarchical Inference Network (HIN)

– **Entity Type Embeddings** We utilize the entity type information to enrich the representation of the input. The entity type embedding is obtained by mapping the entity type (e.g., PER, LOC, ORG) into a vector. The dimension of entity type embeddings is d_t.
– **Coreference Embeddings** Usually each entity may be mentioned many times in a document. Following previous work, we assign entity mentions corresponding to the same entity with the same entity id, which is determined by the order in which entities appear in the document. Then entity ids are embedded into vectors. The dimension of coreference embeddings is d_c.

We concatenate all three embeddings together for each word w_i, and a document is transformed into a matrix $\mathbf{X} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n]$, where each word vector $\mathbf{w}_i \in \mathbb{R}^{d_w+d_t+d_c}$ and n is the length of the document.

3.2 Entity-Level Inference Module

In this section, we compute the entity-level inference information for target entity pair. To represent each word in its context, we encode the document $\mathbf{X} = \{\mathbf{w}_i\}_{i=1}^{n}$ into a hidden state vector sequence $\{\mathbf{h}_i\}_{i=1}^{n}$ with bi-directional LSTM:

$$\mathbf{h}_i = \text{BiLSTM}_E\left(\mathbf{w}_i\right), i \in [1, n]. \tag{1}$$

where $\mathbf{h}_i \in \mathbb{R}^d$ is a contextualized representation of w_i, summarizing the context information centered around w_i.

Considering that an entity may be mentioned many times in a document and a mention may also contain more than one word, we represent each entity and

mention with the average of the embeddings of different elements. Correspondingly, the mention representation is formed as the average of the words that the mention contains, the entity representation is computed as the average of the mention representations associated with the entity:

$$\mathbf{e}_l = avg_{w_i \in e_l}(\mathbf{h}_i), \quad \mathbf{E}_a = avg_{e_l \in E_a}(\mathbf{e}_l) \tag{2}$$

We claim that it is beneficial to allow the model to jointly attend to information from different representation subspaces, thus, we use different learnable projection matrices to project entities into K subspaces:

$$\mathbf{E}_a^k = \mathbf{W}_k^{(1)}(ReLU(\mathbf{W}_k^{(0)}\,\mathbf{E}_a)) \tag{3}$$

where $\mathbf{E}_a^k \in \mathbb{R}^k$ corresponds to the representation of E_a in the k-th latent space, $\mathbf{W}_k^{(0)} \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_k^{(1)} \in \mathbb{R}^{d\times k}$ are the learnable projection matrices corresponding to the k-th subspace. For each of these projected versions, we perform the entity-level inference in parallel. These are concatenated and once again projected, resulting in the final entity-level inference information.

Inspired by TransE [1] which modelled a triple $r(e_h, e_t)$ with $\mathbf{e}_h + \mathbf{r} \approx \mathbf{e}_t$, we argue that ($\mathbf{E}_b - \mathbf{E}_a$) could represent the relation between E_a and E_b in the document to some extent. In addition, a bilinear representation can be obtained by a bi-affine layer to enhance the expression ability of model. We define the following formula as entity-level inference representation in the k-th latent space:

$$\mathbf{I}_e^k = Concat\left(\mathbf{E}_a^k\,\mathbf{R}^k\,\mathbf{E}_b^k;\, \mathbf{E}_b^k - \mathbf{E}_a^k;\, \mathbf{E}_a^k;\, \mathbf{E}_b^k\right) \tag{4}$$

where $\mathbf{R}^k \in \mathbb{R}^{k\times k\times k}$ is a learned bi-affine tensor, $Concat$ denotes concatenation.

Moreover, we believe that the relative distances between two target entities can help us better judge the relations. Empirically, we use the relative distances between the first mentions of the two entities as the relative distances between two target entities. Finally, all entity-level inference representations in different latent space and the relative distance embeddings are fed into a feed-forward neural network (FFNN) to form the final entity-level inference information:

$$\mathbf{I}_e = G_e\left(\left[\mathbf{I}_e^1;\, ...;\, \mathbf{I}_e^K;\, \mathbf{M}\,(d_{ba}) - \mathbf{M}\,(d_{ab})\right]\right) \tag{5}$$

here G_e is a FFNN with ReLU activation function, $\mathbf{M}$ is an embedding matrix, d_{ab} and d_{ba} are the relative distances between E_a and E_b in the document. $\mathbf{I}_e \in \mathbb{R}^d$ describes relation features between E_a and E_b at entity level.

3.3 Hierarchical Document-Level Inference Module

In this section, we propose a hierarchical inference mechanism, inference information is aggregated from entity level to sentence level and then to document level. In this way, our model can aggregate all useful information of the document.

Sentence-Level Inference Assume that a document contains L sentences, and w_{jt} represent the t-th word in the j-th sentence. Given the j-th sentence S_j, to represent words in its context, the sentence is fed into a BiLSTM encoder:

$$\mathbf{h}_{jt} = \text{BiLSTM}_S\left(\mathbf{w}_{jt}\right), t \in [1, T_j]. \tag{6}$$

Since different words in a sentence are differentially informative, inspired by [14], we introduce the vanilla attention mechanism to enable our model to selectively assign higher weights for the informative words and lower weights for the other words. Then we aggregate the representations of those informative words to form a sentence vector. Specifically,

$$\alpha_{jt} = \mathbf{u}_w^\top \tanh\left(\mathbf{W}_w \mathbf{h}_{jt} + \mathbf{b}_w\right) \tag{7}$$

$$a_{jt} = \frac{\exp\left(\alpha_{jt}\right)}{\sum_t \exp\left(\alpha_{jt}\right)} \tag{8}$$

$$\mathbf{S}_j = \sum_t a_{jt} \mathbf{h}_{jt} \tag{9}$$

where $\mathbf{u}_w, \mathbf{b}_w \in \mathbb{R}^d$ and $\mathbf{W}_w \in \mathbb{R}^{d \times d}$ are learnable parameters. Word hidden state $\mathbf{h}_{jt} \in \mathbb{R}^d$ is first fed through a one-layer MLP, then we obtain weights of words by measuring "which words are more related to the target entities". Finally, we compute the sentence vector $\mathbf{S}_j$ as a weighted sum of the word hidden states.

For obtaining the sentence-level inference information, we adopt a semantic matching method which is used in previous NLI model [2]. Through comparing sentence vector $\mathbf{S}_j$ with entity-level inference representation $\mathbf{I}_e$, we can derive sentence-level inference representation $\mathbf{I}_{sj}$ for the j-th sentence:

$$\mathbf{I}_{sj} = G_s\left(\left[\mathbf{S}_j; \mathbf{I}_e; \mathbf{S}_j - \mathbf{I}_e; \mathbf{S}_j \circ \mathbf{I}_e\right]\right). \tag{10}$$

where G_s is FFNN with ReLU function, a matching trick with elementwise subtraction and multiplication is used for building better matching representations [10]. $\mathbf{I}_{sj}$ represents the inference information derived from the j-th sentence.

Document-Level Inference In order to distinguish crucial sentence-level inference information for overall document-level inference representation, vanilla attention mechanism is again used. We build a BiLSTM followed by the attention network on top of the sentence-level inference vectors ($\mathbf{I}_s$) to aggregate all essential evidence information scattered in different sentences:

$$\mathbf{c}_{sj} = \text{BiLSTM}_D\left(\mathbf{I}_{sj}\right), j \in [1, L] \tag{11}$$

$$\alpha_j = \mathbf{u}_s^\top \tanh\left(\mathbf{W}_s \mathbf{c}_{sj} + \mathbf{b}_s\right) \tag{12}$$

$$a_j = \frac{\exp\left(\alpha_j\right)}{\sum_j \exp\left(\alpha_j\right)} \tag{13}$$

$$\mathbf{I}_d = \sum_t a_j \mathbf{c}_{sj} \tag{14}$$

here $\mathbf{u}_s, \mathbf{b}_s \in \mathbb{R}^d$ and $\mathbf{W}_s \in \mathbb{R}^{d \times d}$ are learnable parameters, $\mathbf{I}_d \in \mathbb{R}^d$ is the document-level inference representation which represents all the inference information that we can obtain from the document.

3.4 Prediction Layer

To better integrate inference information of different granularity, we concatenate entity-level inference representation $\mathbf{I}_e$ and document-level inference representation $\mathbf{I}_d$ together to form the final inference representation. Since there are often multiple relations holding between an entity pair, we use a FFNN with the *sigmoid* function to calculate the probability of each relation:

$$P\left(r|E_a, E_b\right) = sigmoid\left(\mathbf{W}_r \begin{bmatrix} \mathbf{I}_e \\ \mathbf{I}_d \end{bmatrix} + \mathbf{b}_r\right). \tag{15}$$

where $\mathbf{W}_r$, $\mathbf{b}_r$ are the weight matrix and bias for the linear transformation.

A binary label vector $\mathbf{y}$ is set to indicate the set of true relations holding between the entity pair, where 1 means an relation is in the set, and 0 otherwise. In our experiments, we use the binary cross entropy (BCE) as training loss:

$$Loss = -\sum_{r=1}^{l} y_r \log\left(p_r\right) + \left(1 - y_r\right) \log\left(1 - p_r\right). \tag{16}$$

where $y_r \in \{0, 1\}$ is the true value on label r and l is the number of relations.

Given a document, we rank the predicted results by their confidence and traverse this list from top to bottom by F1 score on dev set, the probability value corresponding to the maximum F1 is picked as threshold δ. This threshold is used to control the number of extracted relational facts on test set.

4 Experiments

4.1 Dataset

To evaluate the effectiveness of our model, we use the DocRED dataset [15], which is the largest human-annotated document-level RE dataset constructed from Wikidata and Wikipedia. DocRED contains over 5,053 documents, 40,276 sentences, 132,375 entities and 96 frequent relation types. Entity types in DocRED are annotated. It is also introduced by the author of DocRED that about 40.7% of relational facts can only be extracted from multiple sentences and 61.1% relational instances require a variety of reasoning.

4.2 Comparison Models & Evaluation Metrics

We compare our model against the following document-level RE baselines:

CNN/LSTM/BiLSTM-RE: They first encode a document into a hidden state vector sequence with CNN/LSTM/BiLSTM as encoder, and then predict relations for each entity pair by feeding them into a bilinear function [15].

Table 1. Performance of different models on DocRED (%).

Model	Dev		Test	
	Ign F1	**F1**	**Ign F1**	**F1**
CNN-RE [15]	41.58	43.45	40.33	42.26
LSTM-RE [15]	48.44	50.68	47.71	50.07
BiLSTM-RE [15]	48.87	50.94	48.78	51.06
Context-Aware [12]	48.94	51.09	48.40	50.70
HIN-GloVe	**51.06**	**52.95**	**51.15**	**53.30**
BERT-RE [13]	-	54.16	-	53.20
BERT-Two-Step [13]	-	54.42	-	53.92
HIN-BERT	**54.29**	**56.31**	**53.70**	**55.60**

Context-Aware: It uses an LSTM-based encoder to jointly learn representations for all relations in the context, and then combines other context relations with target relation to make the final prediction [12].

BERT-RE: It uses BERT to encode the document, entities are represented by their average word embedding. A BiLinear layer is applied to predict the relation between entity pairs [13].

BERT-Two-Step: Based on BERT-RE, it models the document-level RE through a two-step process. The first step is to predict whether or not two entities have a relation, the second step is to predict the specific relation [13].

HIN: This is the main model of this paper. Multi-granularity inference information is used to better model complex interactions between entities.

The widely used metric F1 is used in our experiments. Moreover, since some relational facts present in both training and dev/test sets, we also report the F1 excluding those relational facts and denote it as Ign F1.

4.3 Implementation Details

We try two embedding methods in our experiments: 100-dimensional GloVe [11] embeddings and BERT representations [3]. For the BERT representations, the base uncased English model with dimension 768 is used, we map word representations into 100 dimensional vectors by a linear projection layer. Once the word representations are initialized, they are fixed during training. The embedding dimensions of coreference, distance and entity type are all set to be 20. For LSTM encoder, the dimension of the hidden units is 128. The number of latent space is 2. Furthermore, we regularize our network using dropout and the dropout ratio is 0.2. We optimized our model using Adam [5], with learning rate of 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size is set to be 12 and the value of threshold δ is determined by the performance on the dev set.

4.4 Experimental Results and Analyses

Overall Performance Experimental results are shown in Table 1. From the results, we can observe that: (1) Compared with BiLSTM-RE, the state-of-the-

Table 2. Results of ablation study (%).

Setting	Dev	
	Ign F1	F1
HIN-BERT	**54.29**	**56.31**
- Translation mechanism	53.09	55.10
- Bilinear transformation	52.15	54.29
- Multispace	52.44	54.59
- Sentence inference	52.82	55.06
- Hierarchical aggregation	51.36	53.50
- Above all	49.95	52.10

art model without BERT, our HIN-GloVe achieves significant improvements of 2.24% in F1, we claim that it is mainly due to the reasoning mechanism and hierarchical aggregation structure in HIN, which will be further discussed in ablation study. (2) Even though BERT based models provides strong prediction power, HIN-BERT consistently improves over them, which further proves the effectiveness of our hierarchical inference network. (3) Although Context-Aware model combines context relations with the target relation, it can't use the evidence information in document as effectively as HIN. Hence our model also outperforms it by 2.60% in F1. (4) BERT representations further boost the performance of our model, the HIN-BERT approach outperforms all these previous methods, which indicates the importance of prior knowledge.

Ablation Study To study the contribution of each component in HIN-BERT, we run an ablation study on DocRED dev set (see Table 2). From these ablations, we find that: (1) When we remove the translation mechanism and bilinear transformation, F1 score drops by 1.21% and 2.02% respectively, which indicates that these two transformations can enhance the expression ability of HIN at the entity level. (2) Removing the multi-space projection hurts the result by 1.72%, which proves that it is beneficial to allow the model to jointly attend to information from different representation subspaces. (3) F1 drops by 1.25% when we remove the sentence-level inference mechanism, i.e., replacing the sentence-level inference vector with sentence vector. (4) F1 drops by 2.81% when we discard the hierarchical aggregation approach. Instead, we run BiLSTM followed by mean-pooling layer over the whole document to get the document vector. (5) We also observe that F1 drops by 4.21% when we discard the above all factors together. In summary, all components play an important role in our model.

Analysis by the number of supporting sentences As we discussed before, it is challenging for document-level RE to reason from multiple sentences. To further prove the effectiveness of HIN, we analyze the recall on relational facts with different number of supporting sentences here.[4] As shown in Figure 3, we

[4] Since there is no official code for BERT-Two-Step, its results are not counted.

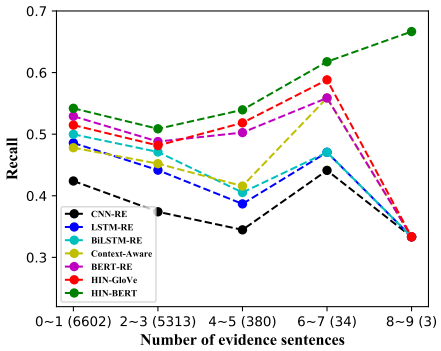

Fig. 3. Recall of models on relational facts with different number of supporting sentences. Numbers in parentheses represent the number of relational facts with different number of supporting sentences in dev set.

find that our model always performs better than other baselines, especially when the number of supporting sentences increases gradually. More specifically, HIN-GloVe even outperforms BERT-RE when the number of supporting sentences exceeds 4, which fully proves the superiority of HIN. Note that when the number of supporting sentences exceeds 7, HIN-GloVe and other baselines behave the same. We think this is because there are very few samples with more than 7 supporting sentences in dev set. We believe when the number of relational facts with more supporting sentences increase our model will achieve better results.

Case Study We compare our model with BERT-RE on some cases from dev set, as shown in Table 3. (1) Example 1 represents the situation that logical reasoning is required. Specifically, in order to identify the relational fact, we have to first identify the fact that *Galaxy S* series is a line of *Samsung* from sentence 0 and 2, then identify the fact *Samsung Galaxy S9* is the latest smartphones in the Galaxy S series from sentence 4. We explain that our model uses a hierarchical aggregation approach to collect inference information from multiple sentences, so that it can better deal with this complex inter-sentence relationship. (2) Example 2 represents the case of coreference reasoning. In this situation, we claim that the attention and reasoning mechanisms in sentence-level inference module can help us to identify that "He" refers to *Robert Kingsbury Huntington* in sentence 3. In the end, our model can identify the right relation while BERT-RE mistakenly assumes that *Los Angeles* is the place where *Robert Kingsbury Huntington* died. (3) Example 3 is a case that needs to combine context information with common-sense knowledge. Through some external common-sense knowledge, we might know that *South America* is a continent and *So Paulo* is a city, which is the useful information to help judge their relation. We think the problem can be solved by adding some external knowledge and we leave it as our future work.

Table 3. The results predicted by BERT-RE and HIN-BERT. The reasoning type of each example is different and the first row for each example is the input document. The ***head***, ***tail***, ***relation*** and **supporting sentences** are colored accordingly.

Logical reasoning	[0] The Galaxy S series is a line of Samsung Electronics, a division of ***Samsung*** [2] Galaxy S line has ... being ***Samsung*** 's flagship smartphones. [4] the latest smartphones in Galaxy S series are the ***Samsung Galaxy S9*** ...
Relation	**Lable**: ***manufacturer*** **BERT-RE**: ***None*** **HIN-BERT**: ***manufacturer***
Coreference reasoning	[0] ***Robert Kingsbury Huntington***, was a naval aircrewman and member of Torpedo Squadron 8. [2] ... ***Huntington*** was shot down during the Battle of Midway ... [3] He was born in ***Los Angeles*** , California ...
Relation	**Lable**: ***birth place*** **BERT-RE**: ***death place*** **HIN-BERT**: ***birth place***
Common-sense reasoning	[0] IBM Research Brazil is one of twelve research laboratories comprising IBM Research , its first in ***South America***. [1] It was established in June 2010 , with locations in ***So Paulo*** and Rio de Janeiro ...
Relation	**Lable**: ***continent*** **BERT-RE**: ***country*** **HIN-BERT**: ***country***

5 Related Work

In recent years, more and more neural models have been applied to RE. Zeng el al. [17] employed a one-dimensional CNN with additional lexical features to encode relations. Miwa et al. [9] used LSTM with tree structures for RE. Zhou el al. [18] showed that combining CNN/RNN with attention mechanism can further improve performance. And the emergence of various optimization algorithms [6,7,8] makes these neural models more effective. Most existing RE work focuses on modeling within a single sentence. However, usually documents provide more information than sentences. Moving research from sentence level to document level is necessary. Recently, there has been increasing interest in document-level RE. Yao et al. [15] proposed a large-scale human-annotated document-level RE dataset, DocRED, and first compute the representations for all entities then predict relations for each entity pair by feeding them into a bilinear function. Wang et al. [13] used BERT to encode the document, it also used bilinear layer to predict the relation between entity pairs, but it modelled the document-level RE through a two-step process. Most previous methods used only entity-level information and this is not adequate. In this paper, we propose to effectively aggregate the inference information of different granularity.

6 Conclusion

In this paper, we proposed a Hierarchical Inference Network (HIN) for document-level RE. It uses a hierarchical inference method to aggregate the inference information of different granularity: entity level, sentence level and document level. We show that our method achieves state-of-the-art performance on the largest

human-annotated DocRED dataset. Experimental analysis shows that both the inference mechanism and hierarchical aggregation approach in our model play an important role. In the future, we plan to incorporate external knowledge to further improve the proposed model.

7 Acknowledgements

This research is supported by the National Key Research and Development Program of China (No.2018YFB1004703).

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS (2013)
2. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced lstm for natural language inference. arXiv preprint arXiv:1609.06038 (2016)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
4. Han, X., Yu, P., Liu, Z., Sun, M., Li, P.: Hierarchical relation extraction with coarse-to-fine grained attention. In: EMNLP (2018)
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Li, Q., Niu, W., Li, G., Cao, Y., Tan, J., Guo, L.: Lingo: linearized grassmannian optimization for nuclear norm minimization. In: CIKM (2015)
7. Li, Q., Wang, Z.: Riemannian submanifold tracking on low-rank algebraic variety. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
8. Li, Q., Wang, Z., Li, G., Cao, Y., Xiong, G., Guo, L.: Learning robust low-rank approximation for crowdsourcing on riemannian manifold. ICCS (2017)
9. Miwa, M., Bansal, M.: End-to-end relation extraction using lstms on sequences and tree structures. arXiv preprint arXiv:1601.00770 (2016)
10. Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., Jin, Z.: Natural language inference by tree-based convolution and heuristic matching. In: ACL (2015)
11. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: EMNLP (2014)
12. Sorokin, D., Gurevych, I.: Context-aware representations for knowledge base relation extraction. In: EMNLP (2017)
13. 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)
14. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: NAACL (2016)
15. 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: ACL (2019)
16. Yu, B., Zhang, Z., Liu, T., Wang, B., Li, S., Li, Q.: Beyond word attention: using segment attention in neural relation extraction. In: IJCAI (2019)
17. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING (2014)
18. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: ACL (2016)