Entity Structure Within and Throughout: Modeling Mention Dependencies for Document-Level Relation Extraction

Benfeng Xu[1]*, Quan Wang[2], Yajuan Lyu[2], Yong Zhu[2], Zhendong Mao[1]†

[1] School of Information Science and Technology, University of Science and Technology of China, Hefei, China
[2] Baidu Inc., Beijing, China
benfeng@mail.ustc.edu.cn, {wangquan05, lvyajuan, zhuyong}@baidu.com, zdmao@ustc.edu.cn

Abstract

Entities, as the essential elements in relation extraction tasks, exhibit certain structure. In this work, we formulate such structure as distinctive dependencies between mention pairs. We then propose SSAN, which incorporates these structural dependencies within the standard self-attention mechanism and throughout the overall encoding stage. Specifically, we design two alternative transformation modules inside each self-attention building block to produce attentive biases so as to adaptively regularize its attention flow. Our experiments demonstrate the usefulness of the proposed entity structure and the effectiveness of SSAN. It significantly outperforms competitive baselines, achieving new state-of-the-art results on three popular document-level relation extraction datasets. We further provide ablation and visualization to show how the entity structure guides the model for better relation extraction. Our code is publicly available.[1] [2]

1 Introduction

Relation extraction aims at discovering relational facts from raw texts as structured knowledge. It is of great importance to many real-world applications such as knowledge base construction, question answering, and biomedical text analysis. Although early studies mainly limited this problem under an intra-sentence and single entity pair setting, many recent works have made efforts to extend it into document-level texts (Li et al. 2016a; Yao et al. 2019), making it a more practical but also more challenging task.

Document-level texts entail a large quantity of entities defined over multiple mentions, which naturally exhibit meaningful dependencies in between. Figure 1 gives an example from the recently proposed document-level relation extraction dataset DocRED (Yao et al. 2019), which illustrates several mention dependencies: 1) *Coming Down Again* and *the Rolling Stones* that both reside in the 1st sentence are closely related, so we can identify **R1: Performer** (blue link) based on their local context; 2) *Coming Down Again* from the 1st sentence, *It* from the 2nd sentence, and *The song* from the 5th sentence refer to the same entity (red link), so it is necessary to consider and reason with them together; 3) *the Rolling Stones* from the 1st sentence and *Mick Jagger* from the 2nd sentence, though not display direct connections, can be associated via two coreferential mentions: *Coming Down Again* and *it*, which is essential to predict the target relation **R2: Member of** (green link) between the two entities. Similar dependency also exists between *the Rolling Stones* and *Nicky Hopkins*, which helps identify **R3: Member of** between them. Intuitively, such dependencies indicate rich interactions among entity mentions, and thereby provide informative priors for relation extraction.

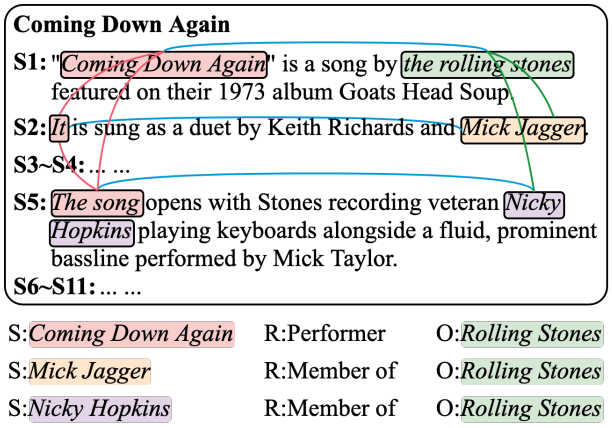

Figure 1: An example excerpted from DocRED. Different mention dependencies are distinguished by colored edges, with the target relations listed in below.

Many previous works have tried to exploit such entity structure, in particular the coreference dependency. For example, it is a commonly used trick to simply encode coreferential information as extra features, and integrate them into the initial input word embeddings. Verga, Strubell, and McCallum (2018) propose an adapted version of multi-instance learning to aggregate the predictions from coreferential mentions. Others also directly apply average pooling to the representations of coreferential mentions (Yao et al. 2019). In summary, these heuristic techniques only use entity dependencies as complementary evidence in the pre- or

*Work done while the first author was an intern at Baidu Inc..
†Corresponding author.

[1]https://github.com/PaddlePaddle/Research/tree/master/KG/AAAI2021_SSAN
[2]https://github.com/BenfengXu/SSAN

post- processing stage, and thus bear limited modeling ability. Besides, most of them fail to include other meaningful dependencies in addition to coreference.

More recently, graph-based methods have shown great advantage in modeling entity structure (Sahu et al. 2019; Christopoulou, Miwa, and Ananiadou 2019; Nan et al. 2020). Typically, these methods rely on a general-purpose encoder, usually LSTM, to first obtain contextual representations of an input document. Then they introduce entity structure by constructing a delicately designed graph, where entity representations are updated accordingly through propagation. This kind of approach, however, isolates the context reasoning stage and structure reasoning stage due to the heterogeneity between the encoding network and graph network, which means the contextual representations cannot benefit from structure guidance in the first place.

Instead, we argue that structural dependencies should be incorporated within the encoding network and throughout the overall system. To this end, we first formulate the aforementioned entity structure under a unified framework, where we define various mention dependencies that cover the interactions in between. We then propose **SSAN** (Structured Self-Attention Network), which is equipped with a novel extension of self-attention mechanism (Vaswani et al. 2017), to effectively model these dependencies within its building blocks and through all network layers bottom-to-up. Note that although this paper only focus on entity structure for document-level relation extraction, the method developed here is readily applicable to all kinds of Transformer-based pretrained language models to incorporate any structural dependencies.

To demonstrate the effectiveness of the proposed approach, we conduct comprehensive experiments on DocRED (Yao et al. 2019), a recently proposed entity-rich document-level relation extraction dataset, as well as two biomedical domain datasets, namely CDR (Li et al. 2016a) and GDA (Wu et al. 2019). On all three datasets, we observe consistent and substantial improvements over competitive baselines, and establish the new state-of-the-art. Our contribution can be summarized as follows:

- We summarize various kinds of mention dependencies exhibited in document-level texts into a unified framework. By explicitly incorporating such structure within and throughout the encoding network, we are able to perform context reasoning and structure reasoning simultaneously and interactively, which brings substantially improved performance on relation extraction tasks.
- We propose SSAN that extends the standard self-attention mechanism with structural guidance.
- We achieve new state-of-the-art results on three document-level relation extraction datasets.

2 Approach

This section elaborates on our approach. We first formalize entity structure in section 2.1, then detail the proposed SSAN model in section 2.2 and section 2.3, and finally introduce its application to document-level relation extraction in section 2.4.

2.1 Entity Structure

Entity structure describes the distribution of entity instances over texts and the dependencies among them. In the specific scenario of document-level texts, we consider the following two structures.

- Co-occurrence structure: Whether or not two mentions reside in the same sentence.
- Coreference structure: Whether or not two mentions refer to the same entity.

Both structures can be described as *True* or *False*. For **co-occurrence structure**, we segment documents into sentences, and take them as minimum units that exhibit mention interactions. So *True* or *False* distinguishes intra-sentential interactions which depend on local context from inter-sentential ones that require cross sentence reasoning. We denote them as *intra* and *inter* respectively. For **coreference structure**, *True* indicates that two mentions refer to the same entity and thus should be investigated and reasoned with together, while *False* implies a pair of distinctive entities that are possibly related under certain predicates. We denote them as *coref* and *relate* respectively. In summary, these two structures are mutually orthogonal, resulting in four distinctive and undirected dependencies, as shown in table 1.

		Coreference	
		True	False
Co-occurence	True	*intra+coref*	*intra+relate*
	False	*inter+coref*	*inter+relate*

Table 1: The formulation of entity structure.

Besides the dependencies between entity mentions, we further consider another type of dependency between entity mentions and its intra-sentential *non-entity* (NE) words. We denote it as *intraNE*. For other inter-sentential non-entity words, we assume there is no crucial dependency, and categorize it as *NA*. The overall structure is thus formulated into an entity-centric adjacency matrix with all its elements from a finite dependency set: {*intra+coref*, *inter+coref*, *intra+relate*, *inter+relate*, *intraNE*, *NA*} (see figure 2).

2.2 SSAN

SSAN inherits the architecture of Transformer (Vaswani et al. 2017) encoder, which is a stack of identical building blocks, wrapped up with feedforward network, residual connection, and layer normalization. As its core component, we propose structured self-attention mechanism with two alternative transformation modules.

Given an input token sequence $x = (x_1, x_2, ..., x_n)$, following the above formulation, we introduce $S = \{s_{ij}\}$ to represent its structure, where $i, j \in \{1, 2, ..., n\}$ and $s_{ij} \in$ {*intra+coref*, *inter+coref*, *intra+relate*, *inter+relate*, *intraNE*, *NA*} is a discrete variable denotes the dependency from x_i to x_j. Note that here we extend dependency from mention-level to token-level for practical implementation. If

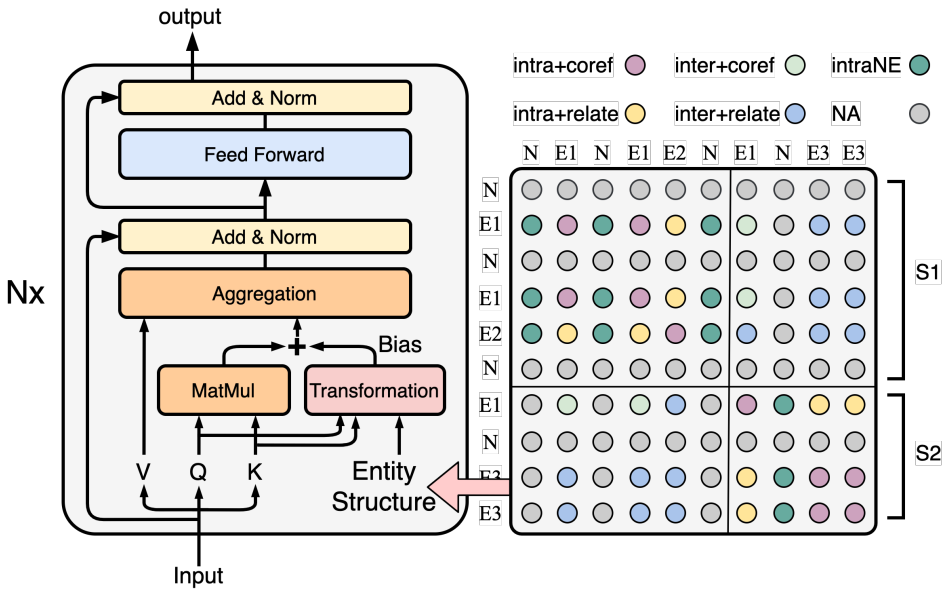

Figure 2: The overall architecture of SSAN. Left illustrates structured self-attention as its basic building block. Right explains our entity structure formulation. This minimum example consists of two sentences: $S1$, $S2$, and three entities: $E1$, $E2$ and $E3$. N denotes non-entity tokens. Element in row i and column j represents the dependency from query token x_i to key token x_j, we distinguish dependencies using different colors.

mention instance consists of multiple subwords ($E3$ in figure 2, $S2$), we assign dependencies for each token accordingly. Within each mention, subword pairs should conform with *intra+coref* and thus are assigned as such.

In each layer l, the input representation $\boldsymbol{x}_i^l \in \mathbb{R}^{d_{in}}$ is first projected into query / key / value vector respectively:

$$\boldsymbol{q}_i^l = \boldsymbol{x}_i^l \boldsymbol{W}_l^Q, \boldsymbol{k}_i^l = \boldsymbol{x}_i^l \boldsymbol{W}_l^K, \boldsymbol{v}_i^l = \boldsymbol{x}_i^l \boldsymbol{W}_l^V \quad (1)$$

where $\boldsymbol{W}_l^Q, \boldsymbol{W}_l^K, \boldsymbol{W}_l^V \in \mathbb{R}^{d_{in} \times d_{out}}$. Based on these inputs and entity structure S, we compute unstructured attention score and structured attentive bias, and then aggregate them together to guide the final self-attention flow.

The unstructured attention score is produced by query-key product as in standard self-attention:

$$e_{ij}^l = \frac{\boldsymbol{q}_i^l {\boldsymbol{k}_j^l}^T}{\sqrt{d}} \quad (2)$$

Parallel to it, we employ an additional module to model the structural dependency conditioned on their contextualized query / key representations. We parameterize it as transformations which project s_{ij} along with query vector $\boldsymbol{q}_i^l$ and key vector $\boldsymbol{k}_j^l$ into attentive bias, then impose it upon e_{ij}^l:

$$\tilde{e}_{ij}^l = e_{ij}^l + \frac{transformation(\boldsymbol{q}_i^l, \boldsymbol{k}_j^l, s_{ij})}{\sqrt{d}} \quad (3)$$

The proposed transformation module regulates the attention flow from x_i to x_j. As a consequence, the model benefits from the guidance of structural dependencies.

After we obtain the regulated attention scores $\tilde{e}_{ij}^l$, a softmax operation is applied, and the value vectors are aggregated accordingly:

$$\boldsymbol{z}_i^{l+1} = \sum_{j=1}^{n} \frac{exp\ \tilde{e}_{ij}^l}{\sum_{k=1}^{n} exp\ \tilde{e}_{ik}^l} \boldsymbol{v}_j^l \quad (4)$$

here $\boldsymbol{z}_i^{l+1} \in \mathbb{R}^{d_{out}}$ is the updated contextual representation of $\boldsymbol{x}_i^l$. Figure 2 gives the overview of SSAN. In the next section, we describe the transformation module.

2.3 Transformation Module

To incorporate the discrete structure s_{ij} into an end-to-end trainable deep model, we instantiate each s_{ij} as neural layers with specific parameters, train and apply them in a compositional fashion. As a result, for each input structure S composed of s_{ij}, we have a structured model composed of corresponding layer parameters. As for the specific design of these neural layers, we propose two alternatives: Biaffine Transformation and Decomposed Linear Transformation:

$$\begin{aligned} bias_{ij}^l &= Biaffine(s_{ij}, \boldsymbol{q}_i^l, \boldsymbol{k}_j^l) \\ &or \\ &= Decomp(s_{ij}, \boldsymbol{q}_i^l, \boldsymbol{k}_j^l) \end{aligned} \quad (5)$$

Biaffine Transformation Biaffine Transformation computes the bias as:

$$bias_{ij}^l = \boldsymbol{q}_i^l \boldsymbol{A}_{l,s_{ij}} {\boldsymbol{k}_j^l}^T + b_{l,s_{ij}} \quad (6)$$

here we parameterize dependency s_{ij} as trainable neural layer $\boldsymbol{A}_{l,s_{ij}} \in \mathbb{R}^{d_{out} \times 1 \times d_{out}}$, which attends to the query and key vector simultaneously and directionally, and projects them into a single-dimensional bias. As for the second term $b_{l,s_{ij}}$, we directly model prior bias for each dependency independent to its context.

Decomposed Linear Transformation Inspired by how Dai et al. (2019) decompose the word embedding and position embedding in Transformer, we propose to introduce bias upon query and key vectors respectively, the bias is thus decomposed as:

$$bias^l_{ij} = \boldsymbol{q}^l_i \boldsymbol{K}^T_{l,s_{ij}} + \boldsymbol{Q}_{l,s_{ij}} \boldsymbol{k}^{l\,T}_j + b_{l,s_{ij}} \tag{7}$$

where $\boldsymbol{K}_{l,s_{ij}}$,$\boldsymbol{Q}_{l,s_{ij}} \in \mathbb{R}^d$ are also trainable neural layers. Intuitively, these three terms respectively represent: 1) bias conditioned on query token representation, 2) bias conditioned on key token representation, and 3) prior bias.

So the overall computation of structured self-attention is:

$$\begin{aligned} \tilde{e}^l_{ij} =& \frac{\boldsymbol{q}^l_i \boldsymbol{k}^{l\,T}_j + transformation(\boldsymbol{q}^l_i, \boldsymbol{k}^l_j, s_{ij})}{\sqrt{d}} \\ =& \frac{\boldsymbol{q}^l_i \boldsymbol{k}^{l\,T}_j + \boldsymbol{q}^l_i \boldsymbol{A}_{l,s_{ij}} \boldsymbol{k}^{l\,T}_j + b_{l,s_{ij}}}{\sqrt{d}} \\ or& \\ =& \frac{\boldsymbol{q}^l_i \boldsymbol{k}^{l\,T}_j + \boldsymbol{q}^l_i \boldsymbol{K}^T_{l,s_{ij}} + \boldsymbol{Q}_{l,s_{ij}} \boldsymbol{k}^{l\,T}_j + b_{l,s_{ij}}}{\sqrt{d}} \end{aligned} \tag{8}$$

As these transformation layers model structural dependencies adaptively according to context, we do not share them across different layers or different attention heads.

Previously, Shaw, Uszkoreit, and Vaswani (2018) have proposed to model relative position information of input token pair within the Transformer. They first map the relative distance into embedding, then add them with key vectors before computing the attention score. Technically, such design can be seen as a simplified version of our Decomposed Linear Transformation, with query conditioned bias only.

2.4 SSAN for Relation Extraction

The proposed SSAN model takes document text as input, and builds its contextual representations under the guidance of entity structure within and throughout the overall encoding stage. In this work, we simply use it for relation extraction with minimum design. After the encoding stage, we construct a fixed dimensional representation for each target entity via average pooling, which we denote as $\boldsymbol{e}_i \in \mathbb{R}^{d_e}$. Then, for each entity pair, we compute the probability of relation r from the pre-specified relation schema as:

$$P_r(\boldsymbol{e}_s, \boldsymbol{e}_o) = sigmoid(\boldsymbol{e}_s \boldsymbol{W}_r \boldsymbol{e}_o) \tag{9}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{d_e \times d_e}$. The model is trained using cross entropy loss:

$$L = \sum_{<s,o>} \sum_r CrossEntropy(P_r(\boldsymbol{e}_s, \boldsymbol{e}_o), \overline{y}_r(\boldsymbol{e}_s, \boldsymbol{e}_o)) \tag{10}$$

and $\overline{y}$ is the target label. Given N entities and a relation schema of size M, equation 9 should be computed $N \times N \times M$ times to give all predictions.

3 Experimental Setup

3.1 Datasets

We evaluate the proposed approach on three popular document-level relation extraction datasets, namely DocRED (Yao et al. 2019), CDR (Li et al. 2016a) and GDA (Wu et al. 2019), all involving challenging relational reasoning over multiple entities across multiple sentences. We summarize their information in Appendix A.

DocRED DocRED is a large scale dataset constructed from Wikipedia and Wikidata. It provides comprehensive human annotations including entity mentions, entity types, relational facts, and the corresponding supporting evidence. There are 97 target relations in total and approximately 26 entities on average in each document. The data scale is 3053 documents for training, 1000 for development set, and 1000 for test. Besides, DocRED also collects distantly supervised data for alternative research. It utilizes a finetuned BERT model to identify entities and link them to Wikidata. Then the relation labels are obtained via distant supervision, producing 101873 document instances at scale.

CDR The Chemical-Disease Reactions dataset is a biomedical dataset constructed using PubMed abstracts. It contains 1500 human-annotated documents in total that are equally split into training, development, and test sets. CDR is a binary classification task that aims at identifying induced relation from chemical entity to disease entity, which is of significant importance to biomedical research.

GDA Like CDR, the Gene-Disease Associations dataset is also a binary relation classification task that identify Gene and Disease concepts interactions, but with a much more massive scale constructed by distant supervision using MEDLINE abstracts. It consists of 29192 documents as the training set and 1000 as the test set.

3.2 Pretrained Transformers

We initialize SSAN with different pretrained language models including BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019) and SciBERT (Beltagy, Lo, and Cohan 2019).

BERT BERT is one of the first works that find the success of Transformer in pretraining language models on large scale corpora. Specifically, it is pretrained using Masked Language Model and Next Sentence Prediction on BooksCorpus and Wikipedia. BERT is pretrained under two configurations, Base and Large, respectively contains 12 and 24 self-attention layers. It can be easily finetuned on various downstream tasks, producing competitive baselines.

RoBERTa RoBERTa is an optimized version of BERT, which removes the Next Sentence Prediction task and adopts way larger text corpora as well as more training steps. It is currently one of the superior pretrained language models that outperforms BERT in various downstream NLP tasks.

SciBERT SciBERT adopts the same model architecture as BERT, but is trained on scientific text instead. It demonstrates considerable advantage in a series of scientific domain tasks. In this paper, we provide SciBERT-initialized SSAN on the two biomedical domain datasets.

Model	Dev Ign F1 / F1	Test Ign F1 / F1
ContexAware (2019)	48.94 / 51.09	48.40 / 50.70
EoG*(2019)	45.94 / 52.15	49.48 / 51.82
BERT Two-Phase (2019a)	- / 54.42	- / 53.92
GloVe+LSR (2020)	48.82 / 55.17	52.15 / 54.18
HINBERT (2020)	54.29 / 56.31	53.70 / 55.60
CorefBERT Base (2020)	55.32 / 57.51	54.54 / 56.96
CorefBERT Large (2020)	56.73 / 58.88	56.48 / 58.70
BERT+LSR (2020)	52.43 / 59.00	56.97 / 59.05
CorefRoBERTa (2020)	57.84 / 59.93	57.68 / 59.91
BERT Base Baseline	56.29 / 58.60	55.08 / 57.54
$\text{SSAN}_{\text{Decomp}}$	56.68 / 58.95	**56.06 / 58.41**
$\text{SSAN}_{\text{Biaffine}}$	**57.03 / 59.19**	55.84 / 58.16
BERT Large Baseline	58.11 / 60.18	57.91 / 60.03
$\text{SSAN}_{\text{Decomp}}$	58.42 / 60.36	57.97 / 60.01
$\text{SSAN}_{\text{Biaffine}}$	**59.12 / 61.09**	**58.76 / 60.81**
RoBERTa Base Baseline	57.47 / 59.52	57.27 / 59.48
$\text{SSAN}_{\text{Decomp}}$	58.29 / 60.22	**57.72** / 59.75
$\text{SSAN}_{\text{Biaffine}}$	**58.83 / 60.89**	57.71 / **59.94**
RoBERTa Large Baseline	58.45 / 60.58	58.43 / 60.54
$\text{SSAN}_{\text{Decomp}}$	59.54 / 61.50	59.11 / 61.24
$\text{SSAN}_{\text{Biaffine}}$	**60.25 / 62.08**	**59.47 / 61.42**
+ Adaptation	**63.76 / 65.69**	**63.78 / 65.92**

distantly supervised data from DocRED

Table 2: Results on DocRED. Subscript $_{\text{Decomp}}$ and $_{\text{Biaffine}}$ refer to Decomposed Linear Transformation and Biaffine Transformation. Test results are obtained by submitting to official Codalab. Result with * is from Nan et al. (2020).

3.3 Implementation Detail

On each dataset, we give comprehensive results of SSAN initialized with different pretrained language models along with their corresponding baselines for fair comparisons. The parameters in newly introduced transformation modules are learned from scratch. All results are obtained using grid search for hyper-parameters (see appendix B for detail) on the development set, then the best model is selected to produce results on the test set. On DocRED, following the official baseline implementation (Yao et al. 2019), we utilize naive features including entity type and entity coreference, which is added to the input word embedding. We also concatenate entity relative distance embedding of each entity pair before the final classification. We preprocess CDR and GDA dataset following Christopoulou, Miwa, and Ananiadou (2019). On CDR, after the best hyper-parameter is set, we merge the training set and dev set to train the final model, on GDA, we split 20% of the training set for development.

4 Experiments and Results

4.1 DocRED Results

We conduct comprehensive and comparable experiments on DocRED dataset. We report both F1 and Ign F1 according to Yao et al. (2019). Ign F1 is computed by excluding relational facts that already appeared in the training set.

As shown in table 2, SSAN with both *Biaffine* and *Decomp* transformation can consistently outperform their baselines with considerable margin. In most of the results, *Biaffine* brings more considerable performance gain compared to *Decomp*, which demonstrates that the former is of greater

Model	Dev F1	Test F1	Intra- / Inter-Test F1
(Gu et al. 2017)	-	61.3	57.2 / 11.7
BRAN(2018)	-	62.1	- / -
CNN+CNNchar(2018)	-	62.3	- / -
GCNN(2019)	57.2	58.6	- / -
EoG (2019)	63.6	63.6	68.2 / 50.9
LSR (2020)	-	61.2	66.2 / 50.3
LSR w/o MDP (2020)	-	64.8	68.9 / 53.1
BERT (2020)	-	60.5	- / -
SciBERT (2020)	-	64.0	- / -
methods using external resources			
(Peng, Wei, and Lu 2016)	-	63.1	- / -
(Li et al. 2016b)	-	67.7	58.9 / -
(Panyam et al. 2018)	-	60.3	65.1 / 45.7
(Zheng et al. 2018)	-	61.5	- / -
BERT Base Baseline	61.7	61.4	69.3 / 44.9
$\text{SSAN}_{\text{Decomp}}$	63.0	61.2	68.6 / **45.1**
$\text{SSAN}_{\text{Biaffine}}$	**64.7**	**62.7**	**70.4** / 44.7
BERT Large Baseline	65.3	63.6	70.8 / 49.0
$\text{SSAN}_{\text{Decomp}}$	64.9	64.5	71.2 / 50.2
$\text{SSAN}_{\text{Biaffine}}$	**65.8**	**65.3**	**71.4 / 52.0**
SciBERT Baseline	68.2	65.8	71.9 / 53.3
$\text{SSAN}_{\text{Decomp}}$	67.9	67.0	72.6 / 55.8
$\text{SSAN}_{\text{Biaffine}}$	**68.4**	**68.7**	**74.5 / 56.2**

Table 3: Results on CDR dev set and test set.

Model	Dev F1	Test F1	Intra- / Inter-Test F1
EoG (2019)	78.7	81.5	85.2 / 49.3
LSR (2020)	-	79.6	83.1 / 49.6
LSR w/o MDP (2020)	-	82.2	85.4 / 51.1
BERT Base Baseline	79.8	81.2	84.7 / 60.3
$\text{SSAN}_{\text{Decomp}}$	81.5	**83.4**	**86.7 / 62.3**
$\text{SSAN}_{\text{Biaffine}}$	**81.6**	82.1	86.1 / 56.8
BERT Large Baseline	80.4	81.6	84.9 / 61.5
$\text{SSAN}_{\text{Decomp}}$	82.0	83.8	86.6 / **65.0**
$\text{SSAN}_{\text{Biaffine}}$	**82.2**	**83.9**	**86.9** / 63.9
SciBERT Baseline	81.4	83.6	**87.2** / 61.8
$\text{SSAN}_{\text{Decomp}}$	82.5	83.2	87.0 / 60.0
$\text{SSAN}_{\text{Biaffine}}$	**82.8**	**83.7**	86.6 / **65.3**

Table 4: Results on GDA dev set and test set.

Dependency	Ign F1	F1
$\text{SSAN}_{\text{Biaffine}}$ (RoBERTa Large)	60.25	62.08
– *intra+coref*	59.59	61.57
– *intra+relate*	59.92	61.91
– *inter+coref*	59.87	61.74
– *inter+relate*	59.92	61.84
– *intraNE*	59.96	61.97
– all	58.45	60.58

Table 5: Ablation for entity structure formulation on DocRED dev set. Results when each dependency is excluded, and "-all" degenerates to RoBERTa Large baseline.

Bias Term	Ign F1	F1
RoBERTa Large baseline (w/o bias)	58.45	60.58
$+b_{s_{ij}}$	58.62	60.59
$+\boldsymbol{Q}_{s_{ij}}\boldsymbol{k}_j^T$	58.79	60.65
$+\boldsymbol{q}_i\boldsymbol{K}_{s_{ij}}^T$	59.26	61.31
$+\boldsymbol{q}_i\boldsymbol{K}_{s_{ij}}^T + \boldsymbol{Q}_{s_{ij}}\boldsymbol{k}_j^T + b_{s_{ij}}$	59.54	61.50
$+\boldsymbol{q}_i\boldsymbol{A}_{s_{ij}}\boldsymbol{k}_j^T$	59.83	61.75
$+\boldsymbol{q}_i\boldsymbol{A}_{s_{ij}}\boldsymbol{k}_j^T + b_{s_{ij}}$	60.25	62.08

Table 6: Ablation for bias terms of two transformation modules on DocRED dev set. Refer to equation 6 and equation 7 for specifics, we have removed the layer index l because the ablation is implemented across all layers.

ability to model structural dependencies.

We compare our model with previous works that either do not consider entity structure or do not explicitly model them within and throughout encoders. Specifically, ContexAware (Yao et al. 2019), BERT Two-Phase (Wang et al. 2019a) and HINBERT (Tang et al. 2020) do not consider the structural dependencies among entities. EOG (Christopoulou, Miwa, and Ananiadou 2019) and LSR (Nan et al. 2020) utilize graph methods to perform structure reasoning, but only after the BiLSTM or BERT encoder. CorefBERT and CorefRoBERTa (Ye et al. 2020) further pretrain BERT and RoBERTa with a coreference prediction task to enable implicit reasoning of coreference structure. Results in table 2 shows that SSAN performs better than these methods. Our best model, $\text{SSAN}_{\text{Biaffine}}$ built upon RoBERTa Large, is **+2.41 / +1.79 Ign F1** better on dev / test set than CorefRoBERTa Large (Ye et al. 2020), and **+1.80 / +1.04 Ign F1** better than our baseline. In general, these results demonstrate both the usefulness of entity structure and the effectiveness of SSAN.

Although SSAN is well compatible with pretrained Transformer models, there still exists a distribution gap between parameters in newly introduced transformation layers and those already pretrained ones, thus impedes the improvements of SSAN to a certain extent. In order to alleviate such distribution deviation, we also utilize the distantly supervised data from DocRED, which shares identical format with the trainset, to first pretrain SSAN before finetuning on the annotated training set for better adaptation. Here we choose our best model, $\text{SSAN}_{\text{Biaffine}}$ built upon RoBERTa Large, and denote it as **+Adaptation** in table 2 (see appendix B for hyperparameters setting). The resulting performance are greatly improved, achieving **63.78 Ign F1** and **65.92 F1** on test set as well as the 1st position on the leaderboard[3] at the time of submission.

distantly supervised data from DocRED

4.2 CDR and GDA Results

On CDR and GDA datasets, besides BERT, we also adopts SciBERT for its superiority when dealing with biomedical domain texts. On CDR test set (see Table 3), SSAN obtains **+1.3 F1/+1.7 F1** gain based on BERT Base/Large and **+2.9 F1** gain based on SciBERT, which significantly outperform the baselines and all existing works. On GDA (see Table 4), similar improvements can also be observed. These results demonstrate the strong applicability and generality of our approach.

4.3 Ablation Study

We perform ablation studies of the proposed approach on DocRED. Again, we consider $\text{SSAN}_{\text{Biaffine}}$ built upon RoBERTa Large. Table 5 gives the results of SSAN when each structural dependency is excluded. It is clear that all five dependencies contribute to the final improvements. We can arrive at the conclusion that the proposed entity structure formulation is indeed helpful priors for document-level relation extraction. We can also see that *intra+coref* effects the most among all dependencies.

We also look into the design of two transformation modules by testing each bias term respectively. As shown in table 6, all bias terms can improve the result over baseline, including the prior bias $+b_{s_{ij}}$ that is only individual values. Among all bias terms, biaffine bias $+\boldsymbol{q}_i\boldsymbol{A}_{s_{ij}}\boldsymbol{k}_j^T$ is the most effective, brings **+1.38 Ign F1** improvements solely. For Decomposed Linear Transformation, key conditioned bias $+\boldsymbol{Q}_{s_{ij}}\boldsymbol{k}_j^T$ produces better results than query conditioned bias $+\boldsymbol{q}_i\boldsymbol{K}_{s_{ij}}^T$, which implies that the key vectors might be associated with more entity structure information.

4.4 Visualization of Attentive Biases

As a key feature of SSAN is to formulate entity structure priors into attentive biases, it would be instructive to explore how such attentive biases regulate the propagation of self-attention bottom-to-up. To this purpose, we collect all attentive biases produced by $\text{SSAN}_{\text{Biaffine}}$ (built upon RoBERTa Large) for DocRED dev instances, categorized according to dependency types, and averaged across all attention heads and all instances. Figure 3 (a) is the resultant heatmap, where each cell indicates the value of averaged bias at each layer (horizontal axis) for each entity dependency type (vertical axis). We can observe meaningful patterns: 1) Along the horizontal axis, the bias is relatively small at bottom layers, where the self-attention score will be mainly decided by unstructured semantic contexts. It then grows gradually and reaches the maximum at the top-most layers, where the

[3] https://competitions.codalab.org/competitions/20717#results

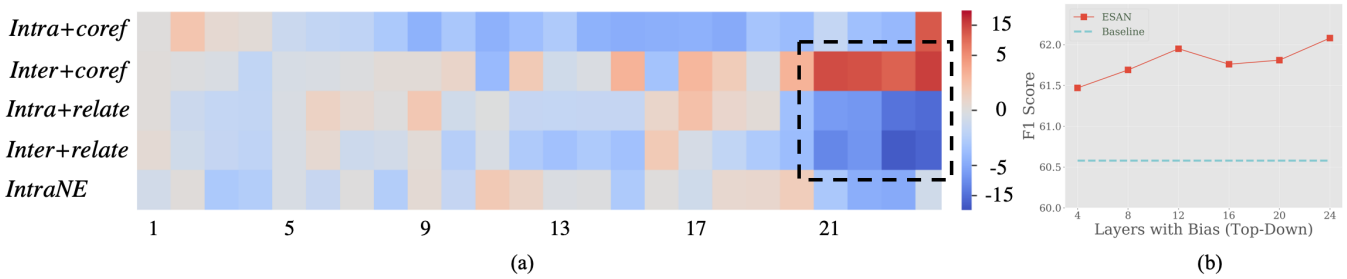

Figure 3: (a): Visualization on the learned attentive bias from different layers and different mention dependencies. Results are averaged over the entire dev set and different attention heads. (b): Ablation on number of layers to impose attentive biases.

self-attention score will be greatly regulated by the structural priors. 2) Along the vertical axis, at the top-most layers (inside the dotted bounding box), bias from *inter+coref* is significantly positive. This conforms with human intuition that coreferential mention pairs might act as a bridge for cross-sentence reasoning, thus should enable more information passing. While biases from *intra+relate* and *inter+relate* appear in contrast.

Based on the discussion, we further investigate the effect of different layers to impose attentive biases. As shown in Figure 3 (b), with only the top 4 layers (1/6 of the total layers) integrated with entity structure, SSAN can keep +0.89 F1 gain, which confirms that these top-most layers with larger biases indeed impact more significantly. In the meantime, with more layers included, the performance still improves, and reaches the best of +1.50 F1 with all 24 layers equipped with structured self-attention.

5 Related Work

Document-level RE Recent years have seen growing interests for relation extraction beyond single sentence (Quirk and Poon 2017; Peng et al. 2017a). Among the most influential works, many have proposed to introduce intra-sentential and inter-sentential syntactic dependencies (Peng et al. 2017b; Song et al. 2018; Gupta et al. 2019). More recently, document-level relation extraction tasks have been proposed (Li et al. 2016a; Yao et al. 2019), where the goal is to identify relations of multiple entity pairs from the entire document text, and rich entity interactions are thereby involved. In order to model these interactions, many graph based methods are proposed (Sahu et al. 2019; Christopoulou, Miwa, and Ananiadou 2019; Nan et al. 2020). However, these graph networks are built upon their contextual encoder, which is different from our approach that model entity interactions within and throughout the system.

Entity Structure Entity structure has been shown to be useful in many NLP tasks. In early works, Barzilay and Lapata (2008) propose an entity-grid representation for discourse analysis, where the document is summarized into a set of entity transition sequences that record distributional, syntactic, and referential information. Ji et al. (2017) introduce a set of symbolic variables and state vectors to encode the mentions and their coreference relationships for language modeling task. Dhingra et al. (2018) propose Coref-GRU, which incorporates mention coreference information for reading comprehension tasks. In general, many works have utilized entity structure in various formulation for different tasks.

For document-level relation extraction, entity structure also is essential prior. For example, Verga, Strubell, and McCallum (2018) propose to merge predictions from coreferential mentions. Nan et al. (2020) propose to model entity interactions via latent structure reasoning. And Christopoulou, Miwa, and Ananiadou (2019) construct a graph of mention nodes, entity nodes, and sentence nodes, then connect them using mention-mention coreference, mention-sentence residency etc., such design provides much more comprehensive entity structure information. Based on the graph, they further utilize an edge-oriented method to iteratively refine the relation representation between target entity pairs, which is quite different from our approach.

Structured Networks Neural networks that incorporate structural priors have been extensively explored. In previous works, many have investigated how to infuse the tree-like syntax structure into the classical LSTM encoder (Kim et al. 2017; Shen et al. 2019; Peng et al. 2017b). For Transformer encoder, it is also a challenging and thriving research direction. Shaw, Uszkoreit, and Vaswani (2018) propose to incorporate relative position information of input tokens in the form of attentive bias, which inspired part of this work. Wang et al. (2019b) further extend this method to relation extraction task, where the relative position is adjusted into entity-centric form.

6 Conclusion and Future Work

In this work, we formalize entity structure for document-level relation extraction. Based on it, we propose SSAN to effectively incorporate such structural priors, which performs both contextual reasoning and structure reasoning of entities simultaneously and interactively. The resulting performance on three datasets demonstrates the usefulness of entity structure and the effectiveness of the SSAN model.

For future works, we give two promising directions: 1) apply SSAN to more tasks such as reading comprehension, where the structure of entities or syntax is useful prior information. 2) extend the entity structure formulation to include more meaningful dependencies, such as more complex interactions based on discourse structure.

Dataset		Train	Dev	Test	Entities / Doc	Mentions / Doc	Mention / Sent	Relation
DocRED	Annotated	3053	1000	1000	19.5	26.2	3.58	96
	Distant	101873	-	-	19.3	25.1	3.43	96
CDR		500	500	500	6.8	19.2	2.48	1
GDA		29192	-	1000	4.8	18.5	2.28	1

Table 7: Summary of DocRED, CDR and GDA datasets. For column *Mention / Sent*, we exclude sentences that do not contain any entity mention.

Dataset	DocRED			CDR	GDA	
Model	Base	Large	Distant Pretrain	-	Base	Large
learning rate	$5e-5$	$3e-5$	$2e-5$	$5e-5$	$5e-5$	$3e-5$
epoch	$\{40, 60, 80, 100\}$		10	$\{10, 20, 30, 40\}$	$\{2, 4, 6\}$	
batch size	4			4	$\{4, 8\}$	

Table 8: Hyper-parameters Setting.

Acknowledgments

We thank all anonymous reviewers for their valuable comments. This work is supported by the National Key Research and Development Project of China (No.2018YFB1004300, No.2018AAA0101900), and the National Natural Science Foundation of China (No.61876223, No.U19A2057).

Appendix

A Datasets

Table 7 details statistics of entities along with other related information of three selected datasets. We can see that all three datasets entail more than two dozen mentions per document on average, with each sentence contains approximately three mentions on average. These statistics further demonstrate the complexity of entity structure in document-level relation extraction tasks.

B Hyper-parameters Setting

Table 8 details our hyper-parameters setting. All experiment results are obtained using grid search on the development set. All comparable results share the same search scope.

References

Barzilay, R.; and Lapata, M. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics* 34(1): 1–34.

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1371. URL https://www.aclweb.org/anthology/D19-1371.

Christopoulou, F.; Miwa, M.; and Ananiadou, S. 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4925–4936. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1498. URL https://www.aclweb.org/anthology/D19-1498.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1285. URL https://www.aclweb.org/anthology/P19-1285.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Dhingra, B.; Jin, Q.; Yang, Z.; Cohen, W.; and Salakhutdinov, R. 2018. Neural Models for Reasoning over Multiple Mentions Using Coreference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 42–48. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-2007. URL https://www.aclweb.org/anthology/N18-2007.

Gu, J.; Sun, F.; Qian, L.; and Zhou, G. 2017. Chemical-induced disease relation extraction via convolutional neural network. *Database* 2017.

Gupta, P.; Rajaram, S.; Schütze, H.; and Runkler, T. 2019. Neural relation extraction within and across sentence boundaries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6513–6520.

Ji, Y.; Tan, C.; Martschat, S.; Choi, Y.; and Smith, N. A. 2017. Dynamic Entity Representations in Neural Language Models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1830–1839. Copenhagen, Denmark: Association for Computational Linguistics. doi:10.18653/v1/D17-1195. URL https://www.aclweb.org/anthology/D17-1195.

Kim, Y.; Denton, C.; Hoang, L.; and Rush, A. M. 2017. Structured Attention Networks. In *5th International Conference on*

Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net. URL https://openreview.net/forum?id=HkE0Nvqlg.

Li, J.; Sun, Y.; Johnson, R. J.; Sciaky, D.; Wei, C.-H.; Leaman, R.; Davis, A. P.; Mattingly, C. J.; Wiegers, T. C.; and Lu, Z. 2016a. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database* 2016.

Li, Z.; Yang, Z.; Lin, H.; Wang, J.; Gui, Y.; Zhang, Y.; and Wang, L. 2016b. CIDExtractor: A chemical-induced disease relation extraction system for biomedical literature. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 994–1001. IEEE.

Liu, X.; Fan, J.; and Dong, S. 2020. Document-Level Biomedical Relation Extraction Leveraging Pretrained Self-Attention Structure and Entity Replacement: Algorithm and Pretreatment Method Validation Study. *JMIR Medical Informatics* 8(5): e17644.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Nan, G.; Guo, Z.; Sekulic, I.; and Lu, W. 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. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.141. URL https://www.aclweb.org/anthology/2020.acl-main.141.

Nguyen, D. Q.; and Verspoor, K. 2018. Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings. In *Proceedings of the BioNLP 2018 workshop*, 129–136. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/W18-2314. URL https://www.aclweb.org/anthology/W18-2314.

Panyam, N. C.; Verspoor, K.; Cohn, T.; and Ramamohanarao, K. 2018. Exploiting graph kernels for high performance biomedical relation extraction. *Journal of biomedical semantics* 9(1): 1–11.

Peng, N.; Poon, H.; Quirk, C.; Toutanova, K.; and tau Yih, W. 2017a. Cross-Sentence N-ary Relation Extraction with Graph LSTMs. *Transactions of the Association for Computational Linguistics* 5(0): 101–115. ISSN 2307-387X. URL https://transacl.org/ojs/index.php/tacl/article/view/1028.

Peng, N.; Poon, H.; Quirk, C.; Toutanova, K.; and Yih, W.-t. 2017b. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics* 5: 101–115.

Peng, Y.; Wei, C.-H.; and Lu, Z. 2016. Improving chemical disease relation extraction with rich features and weakly labeled data. *Journal of cheminformatics* 8(1): 53.

Quirk, C.; and Poon, H. 2017. Distant Supervision for Relation Extraction beyond the Sentence Boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 1171–1182. Valencia, Spain: Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1110.

Sahu, S. K.; Christopoulou, F.; Miwa, M.; and Ananiadou, S. 2019. Inter-sentence Relation Extraction with Document-level Graph Convolutional Neural Network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4309–4316. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1423. URL https://www.aclweb.org/anthology/P19-1423.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 464–468. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-2074. URL https://www.aclweb.org/anthology/N18-2074.

Shen, Y.; Tan, S.; Sordoni, A.; and Courville, A. 2019. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=B1l6qiR5F7.

Song, L.; Zhang, Y.; Wang, Z.; and Gildea, D. 2018. N-ary Relation Extraction using Graph-State LSTM. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2226–2235. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/D18-1246. URL https://www.aclweb.org/anthology/D18-1246.

Tang, H.; Cao, Y.; Zhang, Z.; Cao, J.; Fang, F.; Wang, S.; and Yin, P. 2020. HIN: Hierarchical Inference Network for Document-Level Relation Extraction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 197–209. Springer.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Verga, P.; Strubell, E.; and McCallum, A. 2018. Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 872–884. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-1080. URL https://www.aclweb.org/anthology/N18-1080.

Wang, H.; Focke, C.; Sylvester, R.; Mishra, N.; and Wang, W. 2019a. Fine-tune Bert for Docred with two-step process. *arXiv preprint arXiv:1909.11898* .

Wang, H.; Tan, M.; Yu, M.; Chang, S.; Wang, D.; Xu, K.; Guo, X.; and Potdar, S. 2019b. Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1371–1377. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1132. URL https://www.aclweb.org/anthology/P19-1132.

Wu, Y.; Luo, R.; Leung, H. C.; Ting, H.-F.; and Lam, T.-W. 2019. Renet: A deep learning approach for extracting gene-disease associations from literature. In *International Conference on Research in Computational Molecular Biology*, 272–284. Springer.

Yao, Y.; Ye, D.; Li, P.; Han, X.; Lin, Y.; Liu, Z.; Liu, Z.; Huang, L.; Zhou, J.; and Sun, M. 2019. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 764–777. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1074. URL https://www.aclweb.org/anthology/P19-1074.

Ye, D.; Lin, Y.; Du, J.; Liu, Z.; Sun, M.; and Liu, Z. 2020. Coreferential Reasoning Learning for Language Representation. *arXiv preprint arXiv:2004.06870* .

Zheng, W.; Lin, H.; Li, Z.; Liu, X.; Li, Z.; Xu, B.; Zhang, Y.; Yang, Z.; and Wang, J. 2018. An effective neural model extracting document level chemical-induced disease relations from biomedical literature. *Journal of biomedical informatics* 83: 1–9.