Document-Level Relation Extraction evaluation

Zixuan Huang
Southeast University

Zhenxiao Yu
Southeast University

Wei Shi
Southeast University

Cheng Wang
Southeast University

Runze Zhang
Southeast University

ABSTRACT

Information extraction is an important unsolved problem in natural language processing (NLP). This is a problem of extracting entities from text documents and the relationships between them. Relationship extraction is the task of extracting semantic relationships from text. Extracted relationships typically occur between two or more entities of a particular type (for example, people, organizations, places) and belong to many semantic categories (for example, marriage, employment, residence). And there are a large number of unstructured electronic texts including newswire, blog, email correspondence, government documents, chat records and so on. How to annotate and understand this data becomes a problem. Therefore, for the computer, how to identify a text with interesting semantic attributes becomes the premise to make correct annotation, and extracting semantic relations between entities in natural language text is a key step to realize the application of natural language understanding. In this paper, we focus on methods to identify and extract relationships between entities in unstructured text, and analyze and compare the performance of these methods based on DocRED data sets.

KEYWORDS

Relation extraction, Document-level

1 INTRODUCTION

1.1 Background introduction

Information extraction is an important unsolved problem of natural language processing (NLP). It isthe problem of extracting entities and relations among them from text documents. Examples of enti-ties are people, organizations, and locations. Examples of relations are person-affiliationand organization-location. The person-affiliation relation means that a particular personis affiliated with a certain organization. For instance, the sentence "John Smith is the chief scien-tist of the Hardcom Corporation" contains the person-affiliation relation between the person"John Smith" and the organization "Hardcom Corporation". In this paper, we mainly focus on the problem of extracting such relations from natural language text.

1.2 Relation extraction

Relationship extraction is the task of extracting semantic relationships from a text. Extracted relationships usually occur between two or more entities of a certain type (e.g. Person, Organisation, Location) and fall into a number of semantic categories (e.g. married to, employed by, lives in).

There exists a vast amount of unstructured electronic text on the Web, including newswire, blogs,email communications, governmental documents, chat logs, and so on. How could a human be helped to understand all of this data? A popular idea is turn unstructured text into structured by annotating semantic information. However the sheer volume and heterogeneity of data make human annotation impossible. Instead, we would like to have a computer annotate all data with the structure of our interest. Normally, we are interested in relations between entities, such as person, organization, and location. Examples of relations are person-affiliation and organization location. Current state-of-the-art named entities recognizers, can automatically label data with high accuracy. However, the whole relation extraction process is not a trivial task. The computer needs to know how to recognize a piece of text having a semantic property of interest in order to make a correct annotation. Thus, extracting semantic relations between entities in natural language text is a crucial step towards natural language understanding applications. In this paper, we will focus on methods of recognizing relations between entities in unstructured text.

A relation is defined in the form of a tuple t = (e1, e2, ..., en) where the ei are entities in a predefined relation r within document D. Most relation extraction systems focus on extracting binary relations.

1.2.1 Capturing discriminative attributes . Capturing discriminative attributes is a binary classification task where participants were asked to identify whether an attribute could help discriminate between two concepts. Unlike other word similarity prediction tasks, this task focuses on the semantic differences between words.[6]

e.g. red(attribute) can be used to discriminate apple (concept1) from banana (concept2) -> label 1

More examples:

Attribute	concept1	concept2	label
bookcase	fridge	wood	1
bucket	mug	round	0
angle	curve	sharp	1
pelican	turtle	water	0
wire	coil	metal	0

1.2.2 Multi-Way Classification of Semantic Relations Between Pairs of Nominals . SemEval-2010 introduced 'Task 8 - Multi-Way Classification of Semantic Relations Between Pairs of Nominals'. The task is, given a sentence and two tagged nominals, to predict the relation between those nominals and the direction of the relation. The dataset contains nine general semantic relations together with a tenth 'OTHER' relation.

Example:
There were apples, pears and oranges in the bowl. (content-container, pears, bowl)The main evaluation metric used is macro-averaged F1, averaged across the nine proper relationships (i.e. excluding the OTHER relation), taking directionality of the relation into account.

Several papers have used additional data (e.g. pre-trained word embeddings, WordNet) to improve performance. The figures reported here are the highest achieved by the model using any external resources.

1.2.3 features. The semantic relation is determined between two mentions. In addition, we distinguish the argument order of the two mentions (M1 for the first mention and M2 for the second mention), e.g. M1-Parent-Of-M2 vs. M2-Parent-Of-M1. For each pair of mentions3, we compute various lexical, syntactic and semantic features.

1.Words According to their positions, four categories of words are considered:

- the words of both the mentions
- the words between the two mentions
- the words before M1
- the words after M2

For the words of both the mentions, we also differentiate the head word4 of a mention from other words since the head word is generally much more important. The words between the two mentions are classified into three bins: the first word in be-tween, the last word in between and other words in between. Both the words before M1 and after M2 are classified into two bins: the first word next to the mention and the second word next to the men-tion. Since a pronominal mention (especially neu-tral pronoun such as 'it' and 'its') contains little information about the sense of the mention, the co-reference chain is used to decide its sense.

2.Entity Type This feature concerns about the entity type of both the mentions, which can be PERSON, ORGANIZATION, FACILITY, LOCATION and Geo-Political Entity or GPE.

3.Overlap This category of features includes:

- MB: number of other mentions in between
- MB: number of other mentions in between
- $M1 > M2 or M1 < M2$: flag indicating whether M2/M1is included in M1/M2. Normally, the above overlap features are too general to be effective alone. Therefore, they are also combined with other features:

(1) ET12+M1>M2
(2) ET12+M1<M2
(3) HM12+M1>M2
(4) HM12+M1<M2

1.3 Document-level RE

The task of relation extraction (RE) is to iden-tify relational facts between entities from plaintext, which plays an important role in large-scaleknowledge graph construction. Most existing RE work focuses on sentence-level RE, i.e., extract-ing relational facts from a single sentence. Inrecent years, various neural models have beenexplored to encode relational patterns of entitiesfor sentence-level RE, and achieve state-of-the-art performance.while, existing datasetsfor document-level RE either only have a smallnumber of manually-annotated relations and en-tities, or exhibit noisy annotations from distantsupervision, or serve specific domains or ap-proaches. In order to accelerate the research ondocument-level RE, we urgently need a large-scale, manually-annotated, and general-purposedocument-level RE dataset.

DocRED is a large-scalehuman-annotated document-level RE dataset con-structed from Wikipedia and Wikidata. DocRED is constructed with the following threefeatures:

(1) DocRED contains 132,375 enti-ties and 56,354 relational facts annotated on5,053 Wikipedia documents, making it the largesthuman-annotated document-level RE dataset. (2)As at least 40.7 percent of the relational facts in Do-cRED can only be extracted from multiple sen-tences, DocRED requires reading multiple sen-tences in a document to recognize entities and in-ferring their relations by synthesizing all informa-tion of the document. This distinguishes DocREDfrom those sentence-level RE datasets.

(3)It also provide large-scale distantly supervised datato support weakly supervised RE research.

For each annotated article, the annotation result of DocRED contains the structured information that appears in the article, as well as the source sentences of these structured information. The following figure is a typical structured result.

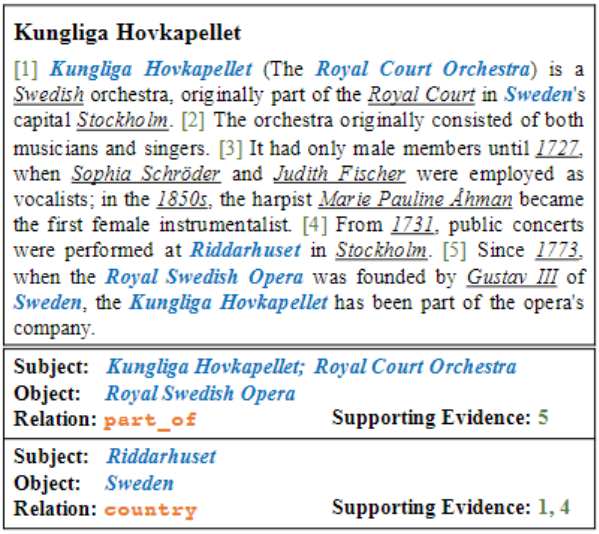

Figure 1: An example from DocRED

It can be seen that the structured results of DocRED are not only generated from the sentence-level (intra-sentence) corpus, but also contain some inter-sentence information. The relationships involve at least two sentences; 40.7 percent of the relationships require information from at least two sentences to be obtained. In addition to the cross-sentence feature, relation extraction in the dataset involves additional inference work, as shown in the following image:

Reasoning Types	%	Examples
Pattern recognition	38.9	[1] ***Me Musical Nephews*** is a 1942 one-reel animated cartoon directed by Seymour Kneitel and animated by Tom Johnson and George Germanetti. [2] Jack Mercer and Jack Ward wrote the script. ... **Relation:** `publication_date` **Supporting Evidence:** 1
Logical reasoning	26.6	[1] "Nisei" is the ninth episode of the third season of the American science fiction television series The X-Files. ... [3] It was directed by David Nutter, and written by Chris Carter, Frank Spotnitz and Howard Gordon. ... [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. ... **Relation:** `creator` **Supporting Evidence:** 1, 3, 8
Coreference reasoning	17.6	[1] ***Dwight Tillery*** is an American politician of the Democratic Party who is active in local politics of Cincinnati, Ohio. ... [3] He also holds a law degree from the University of Michigan Law School. [4] ***Tillery*** served as mayor of Cincinnati from 1991 to 1993. **Relation:** `educated_at` **Supporting Evidence:** 1, 3
Common-sense reasoning	16.6	[1] ***William Busac*** (1020-1076), son of William I, Count of Eu, and his wife Lesceline, ... [4] ***William*** appealed to King Henry I of France, who gave him in marriage Adelaide, the heiress of the county of Soissons. [5] Adelaide was daughter of Renaud I, Count of Soissons, and Grand Master of the Hotel de France. ... [7] ***William*** and Adelaide had four children: ... **Relation:** `spouse` **Supporting Evidence:** 4, 7

Figure 2: types of reasoning required for document-level RE on docred

It can be seen that only 38.9 percent of the relationships in the data set appear in a single sentence, and the proportions of relationships that require logical reasoning, coreference reasoning, and common sense reasoning are 26.6 percent, 17.6 percent, and 16.6 percent, respectively.

Therefore, we can see some features of DocRED:

- Big amount of data
- A large number of entity relationships need to be extracted across statements
- Completing the extraction requires additional inference

2 APPROACH OVERVIEW

2.1 Summary of current ways of Relation Extraction

2.1.1 Rule-based RE. Many instances of relations can be identified through hand-crafted patterns, looking for triples (X, , Y) where X are entities and are words in between. For the "Paris is in France" example, ="is in". This could be extracted with a regular expression.

Figure 3: Named entities in sentence

Figure 4: Part-of-speech tags in sentence

Only looking at keyword matches will also retrieve many false positive. We can mitigate this by filtering on named entities, only retrieving (CITY, is in, COUNTRY). We can also take into account the part-of-speech (POS) tags to remove additional false positive. These are examples of doing word sequence patterns, because the rule specifies a pattern following the order of the text. Unfortunately these type of rules fall apart for longer-range patterns and sequences with greater variety. E.g. "Fred and Mary got married" cannot successfully be handled by a word sequence pattern.

Instead, we can make use of dependency paths in the sentences,

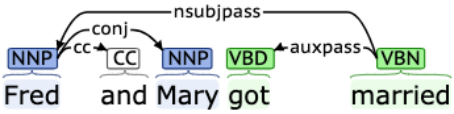

Figure 5: Dependency paths in sentence

knowing which word is having a grammatical dependency on what other word. This can greatly increase the coverage of the rule without extra effort.

We can also transform the sentences before applying the rule. E.g. "The cake was baked by Harry" or "The cake which Harry baked" can be transformed into "Harry baked the cake". Then we are changing the order to work with our "linear rule", while also removing redundant modifying word in between. Pros:

(1) Humans can create pattern which tend to have high precision
(2) Can be tailored to specific domains

Cons:

(1) Human patterns are still often low-recall (too much variety in languages)
(2) A lot of manual work to create all possible rules
(3) Have to create rules for every relation type

2.1.2 Supervised RE. A common way to do Supervised Relation Extraction is to train a stacked binary classifier (or a regular binary classifier) to determine if there is a specific relation between two entities. These classifiers take features about the text as input, thus requiring the text to be annotated by other NLP modules first. Typical features are: context words, part-of-speech tags, dependency path between entities, NER tags, tokens, proximity distance between words, etc. We could train and extract by:

(1) Manually label the text data according to if a sentence is relevant or not for a specific relation type. E.g. for the "CEO" relation: "Apple CEO Steve Jobs said to Bill Gates." is relevant "Bob, Pie Enthusiast, said to Bill Gates." is not relevant
(2) Manually label the relevant sentences as positive/negative if they are expressing the relation. E.g. "Apple CEO Steve Jobs said to Bill Gates.": (Steve Jobs, CEO, Apple) is positive (Bill Gates, CEO, Apple) is negative
(3) Learn a binary classifier to determine if the sentence is relevant for the relation type
(4) Learn a binary classifier on the relevant sentences to determine if the sentence expresses the relation or not
(5) Use the classifiers to detect relations in new text data

Some choose to not train a "relevance classifier", and instead let a single binary classifier determine both things in one go.

Pros:

(1) High quality supervision (ensuring that the relations that are extracted are relevant)
(2) We have explicit negative examples

Cons:

(1) Expensive to label examples
(2) Expensive/difficult to add new relations (need to train a new classifier)

(3) Does not generalize well to new domains
(4) Is only feasible for a small set of relation types

2.1.3 using Pretrained model. According to the performance of the docred dataset's model score ranking list, many of them use pre-trained models, such as bert and glove. Why Pre-training?

Substantial work has shown that pre-trained models (PTMs), on the large corpus can learn universal language representations, which are beneficial for downstream NLP tasks and can avoid training a new model from scratch. With the development of computational power, the emergence of the deep models, and the constant enhancement of training skills, the architecture of PTMs has been advanced from shallow to deep. The first-generation PTMs aim to learn good word embeddings. Since these models themselves are no longer needed by downstream tasks they are usually very shallow for computational efficiencies, such as Skip-Gram and GloVe[2].Although these pre-trained embeddings can capture semantic meanings of words, they are context-free and fail to capture higher-level concepts in context, such as polysemous disambiguation, syntactic structures, semantic roles, anaphora. The second-generation PTMs focus on learning contextual word embeddings, such as CoVe, ELMo , OpenAI GPT and BERT. These learned encoders are still needed to represent words in context by downstream tasks.

With the development of deep learning, the number of model parameters has increased rapidly. The much larger dataset is needed to fully train model parameters and prevent overfitting. However, building large-scale labeled datasets is a great challenge for most NLP tasks due to the extremely expensive annotation costs, especially for syntax and semantically related tasks[1]. In contrast, large-scale unlabeled corpora are relatively easy to construct. To leverage the huge unlabeled text data, we can first learn a good representation from them and then use these representations for other tasks. Recent studies have demonstrated significant performance gains on many NLP tasks with the help of the representation extracted from the PTMs on the large unannotated corpora. The advantages of pre-training can be summarized as follows:

(1) Pre-training on the huge text corpus can learn universal language representations and help with the downstream tasks.
(2) Pre-training provides a better model initialization, which usually leads to a better generalization performance and speeds up convergence on the target task.
(3) Pre-training can be regarded as a kind of regularization to avoid overfitting on small data.

2.2 BERT

BERT (Bidirectional Encoder Representation from Transformers), is a pre-trained language representation model. It emphasizes that the traditional one-way language model or the method of shallowly splicing two one-way language models for pre-training is no longer used as before, but a new masked language model (MLM) is used to generate deep Bidirectional Linguistic Representation. When the BERT paper was published, it was mentioned that new state-of-the-art results were obtained in 11 NLP (Natural Language Processing) tasks, which was astounding.

This model has the following main advantages:

(1) Bidirectional Transformers are pre-trained with MLM to generate deep bidirectional language representations.
(2) After pre-training, it is only necessary to add an additional output layer for fine-tune to achieve state-of-the-art performance in a variety of downstream tasks. No task-specific structural modifications to BERT are required in this process.

2.3 RoBERTa

From the perspective of the model, RoBERTa (a Robustly Optimized BERT Pretraining Approach) has basically no innovation, mainly made some adjustments on the basis of BERT [5]:

(1) The training time is longer, the batch size is larger, and the training data is more;
(2) The next predict loss is removed ;
(3) The training sequence is longer;
(4) The Masking mechanism is dynamically adjusted.
(5) Byte level BPE RoBERTa is trained with dynamic masking, FULL - SENTENCES without NSP loss, large mini-batches and a larger byte-level BPE.

2.4 Masked Language Modeling (MLM)

Intuitively, it is reasonable to believe that a deep bidirectional model is strictly more powerful than either a left-to-right model or the shallow concatenation of a left-to-right and a right-to-left model [9]. Unfortunately, standard conditional language models can only be trained left-to-right or right-to-left, since bidirectional conditioning would allow each word to indirectly "see itself", and the model could trivially predict the target word in a multi-layered context.

In order to train a deep bidirectional representation, Devlin et al. [**?**] simply mask some percentage of the input tokens at random, and then predict those masked tokens. They refer to this procedure as a "masked LM" (MLM), although it is often referred to as a Cloze task in the literature (Taylor, 1953). In this case, the final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary, as in a standard LM. In all of our experiments, we mask 15% of all WordPiece tokens in each sequence at random. In contrast to denoising auto-encoders (Vincent et al., 2008), masked words are predicted rather than reconstructing the entire input.

Although this allows Devlin et al. to obtain a bidirectional pre-trained model, a downside is that they are creating a mismatch between pre-training and fine-tuning, since the [MASK] token does not appear during fine-tuning. To mitigate this, they do not always replace "masked" words with the actual [MASK] token. The training data generator chooses 15% of the token positions at random for prediction. If the i-th token is chosen, the i-th token is replaced with:

(1) the [MASK] token 80% of the time;
(2) a random token 10% of the time;
(3) the unchanged i-th token 10% of the time. Then, T_i will be used to predict the original token with cross entropy loss. We compare variations of this procedure in Appendix C.2.

2.5 Next Sentence Prediction (NSP)

NSP is a binary classification loss for predicting whether two segments follow each other in the original text. Positive examples are created by taking consecutive sentences from the text corpus. Negative examples are created by pairing segments from different documents. Positive and negative examples are sampled with equal probability. The NSP objective was designed to improve performance on downstream tasks, such as Natural Language Inference (Bowman et al., 2015), which require reasoning about the relationships between pairs of sentences [5].

3 DETAIL OF ALGORITHM

3.1 ATLOP-RoBERTa-large

For document-level REs, a document contains multiple entity pairs, and the relationships between them need to be classified at the same time [Statement-level REs only contain one pair of entity pairs.For document-level REs, an entity pair can appear multiple times in documents associated with different relations [for sentence-level REs, only one relation can appear per entity pair] -> multi-label problem Document-level relation extraction (RE) poses new challengescompared to its sentence-level counterpart. One document-commonly contains multiple entity pairs, and one entity pairoccurs multiple times in the document associated with multiplepossible relations. In the paper "Document-Level Relation Extraction with Adaptive Thresholding and LocalizedContext Pooling", the authors propose two novel tech-niques, adaptive thresholding and localized context pooling, tosolve the multi-label and multi-entity problems. The adaptivethresholding replaces the global threshold for multi-label clas-sification in the prior work with a learnable entities-dependentthreshold. The localized context pooling directly transfers at-tention from pre-trained language models to locate relevantcontext that is useful to decide the relation. At present, the mainstream method for document relationship extraction is to use graph-based methods, but many BERT-based works can also get good results, and in the experimental part of graph-based models, it is also proved that BERT and BERT-like The huge improvement of the pre-trained model makes people wonder if it is necessary to introduce GNN? The author found that if only BERT is used, the rep of the entity is the same for different entity pairs. This is a big problem. Can we solve this problem without introducing graph? thesis method localized context pooling Problem solved: solved the problem of using the same entity embedding for allentity pairs Approach: Augment the entity embedding with additional context related to the current entity pair. Instead of training a new context attention layer from scratch, directly transfer the attention heads from the pre-trained language model to entity-level attention adaptive thresholding Problem Solving: Problem 1's Multiple Entity Pair Problem and Problem 2 Entity Pair with Multiple Relationships Method: Replaced with the previously learned global threshold for multi-label classification, which is a learnable threshold for dependent entities.

3.1.1 Adaptive Thresholding. The RE classifier outputs the probability P(rjes; eo) within the range [0; 1], which needs thresholding to be converted to relation labels. As the threshold neither has a closed-form solution nor is differentiable, a common practice for deciding threshold is enumerating several values in the range (0; 1) and picking the one that maximizes the evaluation metrics (F1 score for RE). However, the model may have different confidence for different entity pairs or classes in which one global threshold does not suffice. The number of relations varies (multi-label problem) and the models may not be globally calibrated so that the same probability does not mean the same for all entity pairs. This problem motivates us to replace the global threshold with a learnable, adaptive one, which can reduce decision errors during inference. For the convenience of explanation, we split the labels of entity pair T = (es; eo) into two subsets: positive classes PT and negative classes NT , which are defined as follows: • positive classes PT R are the relations that exist between the entities in T. If T does not express any relation, PT is empty[3].

3.1.2 localized context pooling. The logsumexp pooling accumulates the embedding of all mentions for an entity across the whole document and generates one embedding for this entity. The entity embedding from this document-level global pooling is then used in the classification of all entity pairs. However, for an entity pair, some context of the entities may not be relevant. For example, in Figure 1, the second mention of John Stanistreet and its context are irrelevant to the entity pair (John Stanistreet, Bendigo). Therefore, it is better to have a localized representation that only attends to the relevant context in the document that is useful to decide the relation for this entity pair.

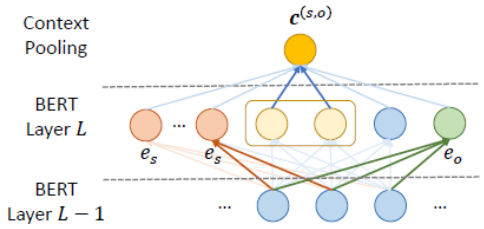

Figure 6: Illustration of localized context pooling

3.2 LSR-BERT-Base

LSR model consists of three components: node constructor, dynamic reasoner, and classifier. The node constructor first encodes each sentence of an input document and outputs contextual representations. Representations that correspond to mentions and tokens on the shortest dependency path in a sentence are extracted as nodes. The dynamic reasoner is then applied to induce a document-level structure based on the extracted nodes. Representations of nodes are updated based on information propagation on the latent structure, which is iteratively refined. Final representations of nodes are used to calculate classification scores by the classifier.

A context encoder is applied to get the contextualized representations of sentences. The representations of mentions and words in the meta dependency path are extracted as mention nodes and MDP nodes. An average pooling is used to construct the entity node from the mention nodes. For example,in the figure, the entity node Lutsenko is constructed by averaging representations of its mentions Lutsenko and He. The dynamic reasoner has two modules,

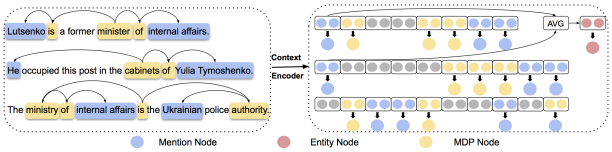

Figure 7: Overview of the Node Constructor

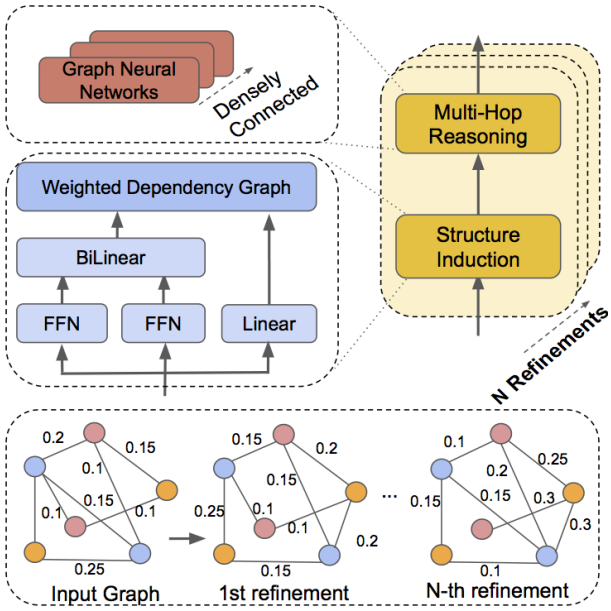

Figure 8: Overview of the Dynamic Reasoner

structure induction and multi-hop reasoning as shown in Figure. The structure induction module is used to learn a latent structure of a document-level graph. The multi-hop reasoning module is used to perform inference on the induced latent structure, where representations of each node will be updated based on the information aggregation scheme. We stack N blocks in order to iteratively refine the latent document-level graph for better reasoning. In other words, each block consists of two sub-modules: structure induction and multi-hop reasoning. The first module takes the nodes constructed by the Node Constructor as inputs. Representations of nodes are fed into two feed-forward networks before the bilinear transformation. The latent document-level structure is computed by the MatrixTree Theorem. The second module takes the structure as input and updates representations of nodes by using the densely connected graph convolutional networks. We stack N blocks which correspond to N times of refinement. Each iteration outputs the latent structure for inference.

Unlike existing models that use co-reference links or heuristics to construct a document-level graph for reasoning, LSR model treats the graph as a latent variable and induces it in an end-to-end fashion. The structure induction module is built based on the structured attention.

In the Multi-Hop Reasoning, we use dense connections tothe GCNs in order to capture more structural information on a large document-level graph. With the help of dense connections, we are able to train a deeper model, allowing richer local and non-local

Table 1: The formulation of entity structure.

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

information to be captured for learning a better graph representation. The computations on each graph convolution layer is similar to Equation

$$\mathbf{u}_i^l = \sigma\left(\sum_{j=1}^{n} \mathbf{A}_{ij}\mathbf{W}^l\mathbf{u}_j^{l-1} + \mathbf{b}^l\right) \tag{1}$$

where given a graph G with n nodes, which can be represented with an n × n adjacency matrix A induced by the previous structure induction module, the convolution computation for the node i at the l-th layer, which takes the representation $\mathbf{u}_i^{l-1}$ from previous layer as input and outputs the updated representations $\mathbf{u}_i^l$, $\mathbf{w}^l$ and $\mathbf{b}^l$ are the weight matrix and bias vector for the l-th layer, respectively. σ is the ReLU activation function. $\mathbf{u}_i^0 \in \mathbb{R}^d$ is the initial contextual representation of the i-th node constructed by the node constructor.

3.3 SSAN-RoBERTa-base

The SSAN-RoBERTa-base is proposed by Xu et al. (2021) [8]. They formulate an entity structure under a unified framework, where they define various mention dependencies that cover the interactions in between. They then propose SSAN (Structured Self-Attention Network), which is an improved version of standard self-attention mechanism. The SSAN inherits the architecture of Transformer[7],and its attention flow is guided by the special entity structure. They also set a transformation module to combine the standard self-attention mechanism with its entity structure. As the pretrained language model, they use Transformer-based pretrained language models including BERT and RoBERTa.

3.3.1 Entity structure. They construct an entity structure to describe the distribution of entity instances over texts and the dependencies among them. In the specific scenario of document-level texts, the following two structures are considered.

- 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, so True or False distinguishes intra-sentential interactions which depend on local context from intersentential 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 consider together, while False implies a pair of distinctive entities that are only 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.

The four types of dependencies just consider the relationship between entity mentions. For there are also tokens not instances of entity, and a dependency may exist when considering a entity

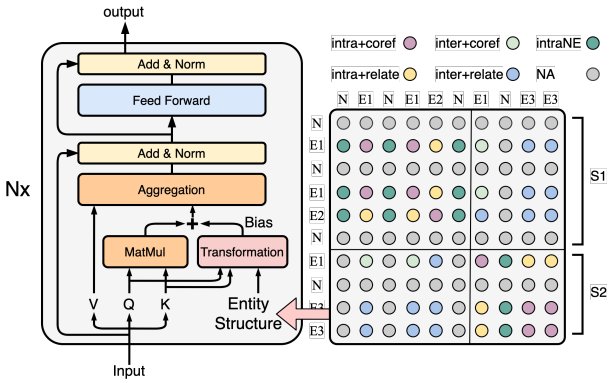

Figure 9: The overall architecture of SSAN.Left illustrates structured self-attention as its basic building block. Right explains the entity structure formulation.

mention and a non-entity token pair intra-sentential, this kind of dependencies is denote as intraNE. Other kinds of dependencies, we assume there are no dependencies between them according to our common sense of language, this kind of dependencies is denote 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. Among them, we assume that there are no dependencies in a NA token pair, and just consider the remaining 5 types. The entity structure is showed in right side of 9.This 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 ,different colors are used to represent different dependencies. The document is transformed into the token-level and all the words are placed in form of tokens and is marked by its entity id if it is an entity.

3.3.2 SSAN. . SSAN (Structured Self-Attention Network), its base on the architecture of Transformer (Vaswani et al. 2017[7]) encoder, which is a stack of identical building blocks, wrapped up with feedforward network, residual connection, and layer normalization. As its core component, this model combine the normal self-attention mechanism with two alternative transformation modules, which is designed to accept the entity structure as the prior knowledge. 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 ∈{1, 2, ..., n} and s_{ij} ∈ { intra+coref, inter+coref, intra+relate, inter+relate, intraNE, NA } . is a discrete variable denotes the dependency from x_i to x_j. If mention instance consists of multiple subwords (E3 in Figure 9, 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 the standard self-attention in layer l , the attention score is produced by:

$$e^l_{ij} = \frac{q^l_i {k^l_j}^T}{\sqrt{d}} \tag{2}$$

where,e_{ij} is the naive attention score,q_i k_j is the i-th and j-th token's query vector and key vector, where $q_i = x_i W^Q_l$ and $k_i = x_i W^K_l$, W^Q_l and W^K_l are trainable tensors, x^l_i is the input of l layer, d is the size of key vector.

Parallel to it, they employ an additional module to model the structural dependency conditioned on their contextualized query / key representations. The module is called transformation, its input is the query and key of the words pair and s_{ij}, its output $\tilde{e}_{ij}$ is the additional bias of the SSAN, into attentive bias, then impose it upon the naive attention score:

$$\tilde{e}_{ij} = e_{ij} + \frac{transformation(q_i, k_j, s_{ij})}{\sqrt{d}} \tag{3}$$

where the $\tilde{e}_{ij}$ is the regulated attention score for considering token i and token j. As a consequence, the model benefits from the guidance of structural dependencies. After we obtain the regulated attention scores , a softmax operation is applied, and the value vectors are aggregated accordingly:

$$z^{l+1}_i = \sum_n^{j=1} \frac{\exp \tilde{e}^l_{ij}}{\sum_{k=1}^n \exp \tilde{e}^l_{ik}} v^l_j \tag{4}$$

where v^l_j is the value vector of the input, it is calculated by $v^l_i = x^l_i W^V_l$, W^K_l is a trainable tensor, z^{l+1} is the updated contextual representation of x^l_i.

3.3.3 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[4]. As a result, for each input structure S composed of s_ij , it has a structured model composed of corresponding layer parameters. In this model, it defines 6 kinds of dependenciesintra+coref, inter+coref, intra+relate, inter+relate, intraNE, NA. Exclude the NA, which means no meaningful dependencies between the pair, there are 5 parts trainable model corresponding to the 5 types dependencies. As for the specific design of these neural layers, it propose two alternatives: Biaffine Transformation and Decomposed Linear Transformation:

$$bias^l_{ij} = Biaffine(s_{ij}, q^l_i, k^l_j) = q^l_i A_{l,s_{ij}} {k^l_j}^T + b_{l,s_{ij}} \tag{5}$$

$$bias^l_{ij} = Decomp(s_{ij}, q^l_i, k^l_j) = q^l_i Q_{l,s_{ij}} + K_{l,s_{ij}} {k^l_j}^T + b_{l,s_{ij}} \tag{6}$$

where $A_{l,s_{ij}}$, $Q_{l,s_{ij}}$, $K_{l,s_{ij}}$ are trainable neural layer, in each layer exist 5 such neural layers corresponding to all different types dependencies expect the NA, $bias^l_{ij}$ is the output of this module.

The two different transformation module share the same designing aim and idea, so they are alternative.

3.4 CorefRoBERTa-large

Language representation models such as BERT could effectively capture contextual semantic information from plain text, and have been proved to achieve promising results in lots of downstream NLP tasks with appropriate fine-tuning. However, most existing language representation models cannot explicitly handle coreference, which is essential to the coherent understanding of the whole discourse.

To address this issue, Deming Ye et al. (2020) [9] present CorefBERT, a novel language representation model that can capture the coreferential relations in context. The experimental results show that, compared with existing baseline models, CorefBERT can

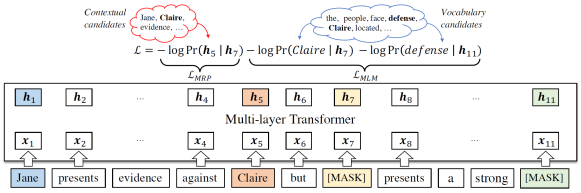

Figure 10: An illustration of CorefBERT's training process

achieve significant improvements consistently on various downstream NLP tasks that require coreferential reasoning, while maintaining comparable performance to previous models on other common NLP tasks.

CorefBERT is a language representation model, which aims to better capture the coreference information of the text. As illustrated in Figure 10[9], CorefBERT adopts the deep bidirectional Transformer architecture and utilizes two training tasks:

(1) Mention Reference Prediction (MRP) is a novel training task which is proposed to enhance coreferential reasoning ability. MRP utilizes the mention reference masking strategy to mask one of the repeated mentions and then employs a copybased training objective to predict the masked tokens by copying from other tokens in the sequence.

(2) Masked Language Modeling (MLM) is proposed from vanilla BERT, aiming to learn the general language understanding. MLM is regarded as a kind of cloze tasks and aims to predict the missing tokens according to its final contextual representation. Except for MLM, Next Sentence Prediction (NSP) is also commonly used in BERT, but some previous works have revealed that NSP is not as helpful as expected.

Formally, given a sequence of tokens $X = (x_1, x_2, \cdots, x_n)$, we first represent each token by aggregating the corresponding token and position embeddings, and then feeds the input representations into deep bidirectional Transformer to obtain the contextual representations, which is used to compute the loss for pre-training tasks. The overall loss of CorefBERT is composed of two training losses: the mention reference prediction loss L_{MRP} and the masked language modeling loss L_{MLM}, which can be formulated as:

$$L = L_{MRP} + L_{MLM}.$$

3.5 DocuNet-RoBERTa-large

The authors propose a document U-shaped network (DocuNet) model (to capture the local contextual information and global interdependencies between triples of document-level REs) to express document-level REs as semantic segmentation. Specifically, an encoder module is used to capture the contextual information of entities, and a U-shaped segmentation module is referenced to capture the global interdependencies among triplets on image-style feature maps.[10] Specifically, the DocuNet model is divided into three modules:

(1) Encoder Module.

We treat triple extraction as a sequence-to-sequence task to better model cross dependencies between entities and relations. We define input text and output triples as source and target sequences. The source sequence consists only of the tokens of the input sentence, e.g. "[CLS] The United States President Trump was raised in the borough of Queens ...[SEP]". We concatenate triples of each entity/relationship separated by special tokens "<e>" and "</e>" as the target sequence.

(2) U-shaped Segmentation Module.

There are local semantic dependencies among triples, and CNN in semantic segmentation can facilitate the local information exchange between entity pairs in the receptive field. Document-level RE also needs global information to infer the relationship between triples. Downsampling and upsampling in the semantic segmentation module can expand the receptive field of the current entity pair to the embedding , and can enhance the global implicit reasoning: We take the entity-level relationship $F \in R^{N^* N^* D}$ matrix as a D-channel image, and we formulate document-level relationship prediction as a pixel-level mask, where N is the maximum number of entities counted from all dataset samples.

(3) Classification Module.

Given the feature representations of entity pairs and the entity-level relation matrix Y, we map them to a hidden representation z using a feed-forward neural network. Then, we obtain the probability representation of the relationship prediction between entity pairs through the bilinear function.

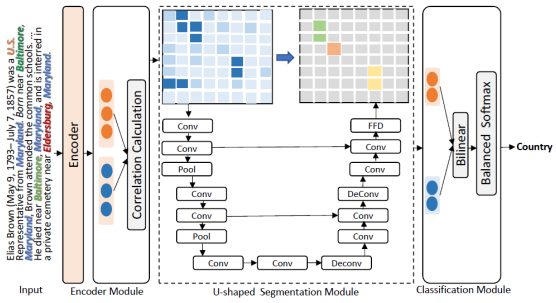

Figure 11: Architecture of our Document U-shaped Network (DocuNet)

This paper innovatively proposes the DocuNet model, which for the first time analogizes the task of document-level relation extraction to the task of semantic segmentation in computer vision. The DocuNet model utilizes an encoder module to capture the contextual information of entities, and adopts a U-shaped segmentation module to capture the global interdependencies among triplets on image-style feature maps, and captures local and global information by predicting entity-level relationship matrices to enhance document-level relation extraction. Experimental results show that our method can achieve SOTA performance on three benchmark datasets DocRED, CDR and GDA.[1]

4 EXPERIMENTS

4.1 dataset

We evaluate next all models on the same public documentlevel relation extraction dataset. The dataset statistics are shown in Table 1. DocRED (Yao et al. 2019) is a large-scale crowdsourced dataset for document-level RE. It is constructed from Wikipedia articles. DocRED consists of 3053 documents for training. For entity pairs that express relation(s), about 7percent of them have more than one relation label.

Statistics	DocRED
# Train	3053
# Dev	1000
# Test	1000
# Relations	97
Avg.# entities per Doc.	19.5

Figure 12: Statistics of the datasets of docred

4.2 Evaluation

4.2.1 Evaluation metric. according to the paper < DocRED: A Large-Scale Document-Level Relation Extraction Dataset>, Two widely used metrics F1 and AUC are used in the dataset docred. However,some relational facts present in both the trainingand dev/test sets, thus a model may memorize theirrelations during training and achieve a better per-formance on the dev/test set in an undesirable way,introducing evaluation bias. However, the over-lap in relational facts between the training anddev/test sets is inevitable, since many common re-lational facts are likely to be shared in different documents. Therefore, we also report the F1 scores excluding those relational facts shared by the training and dev/test sets, denoted as Ign F1.

F1-score: The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

$$\begin{aligned} F_1 &= \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \\ &= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \\ &= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})} \end{aligned} \tag{7}$$

why no Accuracy : Accuracy is the most common and basic evaluation metric. However, in the case of binary classification and unbalanced positive and negative examples, especially when we are more interested in minority class, the accuracy evaluation is basically of no reference value. What fraud detection (fraud detection), cancer detection, are in line with this situation. Take a chestnut: In the test set, there are 100 samples, 99 negative examples, and only 1 positive example. If my model predicts a negative example for any sample indiscriminately, then the accuracy of my model is correct number/total number = 99/100 = 99 percent. This is called accuracy paradox. And in the DocRED dataset, most entity pairs have no relationship, resulting in label imbalance, i.e. most entity pairs belong to N/a relationship.

4.2.2 Evaluation code. The test set of docred is handed over to the codalab website for evaluation, so there is no need to write the code for calculating the metric of the test set. Considering that the relevant papers all use the data of the validation set, we wrote a code to calculate the F1 score and Ign F1 score of the validation set. Code screenshots are as follows.

4.3 Fine-tune Bert for DocRED with Two-step Process

In this section, we will explain how to use the two-steps training process to further improve the performance compared to simply using bert.

4.3.1 Two-step Training Process. In the DocRED dataset, there is no relation formost entity pairs, which causes a large label imbalance, i.e., most entity pairs belong to the N/Arelation. To alleviate this problem, we use a two-step training process.In the first step, we only identify whether or notthere exists a relation between a given entity pair,i.e., we simplify the problem to a binary classi-fication problem. We use BERT for this step asmentioned above, where all of the annotated datais used to train the model. Sub-sampling is appliedto balance relational and N/A pairs in each batch.In the second step, we learn a model to iden-tify the specific relationship between a given pairof entities. The model structure is the same BERTmodel as in the first step. The difference lies inthe training data and labels: We only use these re-lational facts (i.e., entity pairs with relations) totrain the model, so that the model can learn to dis-tinguish between these different relations. Empir-ically we found that the second step is relativelyeasy, as we achieve about 90 percent accuracy. The bot-tleneck of the problem lies in the first step, whichis to distinguish whether there is a relation or not.

4.3.2 Implementation Details. We use BERT-base in our experiments. The learn-ing rate is set to 105. The embedding sizeof BERT model is 768. A transformation layeris used to project the BERT embedding into alow-dimensional space of size 128. In the low-dimension space, a BiLinear layer is applied topredict the relation for a given entity pair.In the first step, we set the relation label for allrelational instances to be 1, while the label for allN/A relations to be 0. We randomly sample N/Arelations at a ratio 3 : 1 within a batch. In thesecond step, we train a new model using only re-lational instances, and the specific relation label iskept in this step.

4.3.3 Results. We compare the BERT model with several baselines presented in includinga CNN, LSTM , bidirectional LSTM(BiLSTM) and Context-Awaremodels. The first three models differ from the BERT model in the encoder, i.e., they use CNN,LSTM, and BiLSTM as encoder respectively.The main results are present.

Table 2: Comparison of the BERT model with other baselines. We report F1 score on the Dev and Test set.

Model	Dev	Test
CNN	43.45	42.26
LSTM	50.68	50.07
BiLSTM	50.94	51.06
Context-Aware	51.09	50.70
BERT	54.16	53.20
BERT-Two-Step	54.42	53.92

4.4 Five chosen sota model

Below are the experimental results of the five models selected by our group. We conduct comprehensive and comparable experiments on DocRED dataset. We report both F1 and Ign F1.Ign F1 is computed

by excluding relational facts that already appeared in the training set.Next are our results.

Table 3: Results of chosen models on docred dataset

Model	Dev	Test
Model	Ign F1 / F1	Ign F1 / F1
LSR+BERT	52.43 / 59.00	56.97 / 59.05
CorefRoberta Large	57.84 / 59.93	57.68 / 59.91
SSAN-RoBERTa-base	58.51 / 60.57	58.82 / 61.13
RoBERTa-ATLOPLARGE	61.23 / 63.12	61.39 / 63.40
DocuNet-RoBERTalarge	62.21 / 64.12	62.39 /64.55

Results First, compared to the baseline results and bert two-step Process above, the performance of these models is significantly improved. The worst LSR is also about four percentage points higher than the best F1 score above. The above table lists the test indicators of the models selected by our group on the docred data in ascending order. It can be seen that the DocuNet-RoBERTalarge model has the best effect, compared with the latter in F1. The score and Ign F1score improved by almost a percentage point. Another point is that we can't judge that the effect of the LSR model is not good, because the LSR model is based on the bert base pre-training model, and the other four are based on the Roberta large pre-training model. In terms of the ability of the pre-training model, Roberta large has already opened a big gap compared to bert base. The data on this point can be verified from roberta's paper.Therefore, we mainly compare the capabilities of the remaining four models. Under the combination of the same pre-trained model, the difference of the obtained model effect is convincing. We can find that the differences between ATLOP, DocuNet and Coref, SSAN models are significant. Coref and SSAN, ATLOP and Docunet are all about one percentage point behind, while ATLOP and ssan do lead by about three percentage points in each indicator. This is not a small improvement. Therefore, we can think that the model design ideas of ATLOP and Docunet are better.

4.5 Ablation Study - Compared to other Ptm baselines

Since training CorefBERT from scratch would be time-consuming, we initialize the parameters of CorefBERT with BERT released by Google3, which is also used as our baselines on downstream tasks. Similar to previous language representation models, the author adopt English Wikipeida4 as training corpus, which contains about 3,000M tokens. CorefBERT was trained with contiguous sequences of up to 512 tokens, and randomly shorten the input sequences with 10percent probability in training.

To verify the effectiveness of the method, i test the model on the downstream tasks of relation extraction by adding an linear layer and an Bilinear layer.

Baseline considering they are all pretrained models which are used widely, we compare the corefbert with BERT Base,BERT Large,RoBERTa Base,RoBERTa Large.Their results are as follows.

Results Table above shows the performance on DocRED.we can see the BERT BASE model is a good baseline model.And it's obvious that the Larger version pretrained model behaves better than Base model whose F1 score higher almost one point than base version.We can find that effect of Roberta is better than bert with the same size. We know large version pretrained model owns higher transformer layer and more training epochs, which leads to more powerful abilities.CorefBERTBASE outperforms BERTBASE model by 0:7 percent F1. Coref BERT LARGE beats BERT LARGE by 0:5 percent F1. we can draw a conclusion that considering coreference information of text is helpful despite the improvement not clear.

Table 4: Results on DocRED measured by micro ignore F1 (IgnF1) and micro F1. IgnF1 metrics ignores the relational facts shared by the training and dev/test sets.

Model	Dev	Test
Model	Ign F1 / F1	Ign F1 / F1
BERT Base Baseline	56.29 / 58.60	55.08 / 57.54
BERT Large Baseline	56.51 / 58.70	56.01 / 58.31
RoBERTa Base Baseline	57.47 / 59.52	57.27 / 59.48
RoBERTa Large Baseline	58.45 / 60.58	58.43 / 60.54
CorefBERT BASE	55.32/ 57.51	54.54 /56.96
CorefBERT Large	56.73 / 58.88	56.48 / 58.70

Table 5: Performance of different based on bert base on docred

Model	Dev	Test
Model	Ign F1 / F1	Ign F1 / F1
GloVe+LSR	48.82 / 55.17	52.15 / 54.18
BERT +LSR	52.43 / 59.00	56.97 / 59.05
CorefBERT	55.32 / 57.51	54.54 / 56.96
SSAN+bert	58.29 / 60.22	57.72 / 59.75
BERTATLOP	59.22 /61.09	59.31 /61.30
DocuNetBERT	59.86/ 61.83	59.93 /61.86

Results It's obvious that from the perspective of ptm, bert is much better than Glove. We haven't find the statistics of Roberta+LSR or bert large+LSR, and it's difficult for us to use enough calculation resources like expensive Gpus to trained a model like Roberta+LSR.But ,from tables above, we can easily infer that LSR+roberta will behave better.So it's not fair to compare bertLSR with other four models which are based on the roberta.

To be fair,we use the same Pretrained model to compare their effects. We can find the following two points.

(1) Since both use the bert base pre-training model, after returning to the same starting line, the gap between the LSR model and coref has become smaller, from the previous difference of about five percentage points to about three percentage points, which shows that the strong The pre-trained model does have a direct improvement in downstream effects.
(2) The ability gap between the models has become larger because of the change from bert to Roberta. In Table 2, the index value difference between CorefRoberta Large and SSAN-RoBERTa-base is about 1 percentage point, and after changing to bert, the difference between CorefBERT and SSANbert comes to about 3 percentage points. This once again proves the power of pre-trained models.

5 RELATEWORKS

Pre-training language representation models aim to capture language information from the text, which facilitate various downstream NLP applications. Early works focus on learning static word embeddings from the unlabeled corpus, which have the limitation that they cannot handle the polysemy well. Recent years, contextual language representation models pre-trained on large-scale unlabeled corpora have attracted intensive attention and efforts from both academia and industry. SALSTM and ULMFiT pre-trains language models on unlabeled text and perform task-specific fine-tuning. ELMo further employs a bidirectional LSTM-based language model to extract context-aware word embeddings.

Moreover, OpenAI GPT and BERT learn pre-trained language representation with Transformer architecture, achieving state-of-the-art results on various NLP tasks. Beyond them, various improvements on pre-training language representation have been proposed more recently, including

(1) Designing new pre-trainning tasks or objectives such as Span- BERT with span-based learning, XLNet considering masked positions dependency with auto-regressive loss, MASS and BART with sequence-to-sequence pre-training, ELECTRA learning from replaced token detection with generative adversarial networks and InfoWord with contrastive learning;
(2) Integrating external knowledge such as factual knowledge in knowledge graphs;
(3) Exploring multilingual learning or multimodal learning. Though existing language representation models have achieved a great success, their coreferential reasoning capability are still far less than that of human beings.

Formerly, researchers have made efforts to explore feature-based unsupervised coreference resolution methods. After that, Word-LM uncovers that it is natural to resolve pronouns in the sentence according to the probability of language models. Moreover, WikiCREM builds sentence-level unsupervised coreference resolution dataset for learning coreference discriminator. However, these methods cannot be directly transferred to language representation models since their task-specific design could weaken the model's performance on other NLP tasks. To address this issue, CorefRoBERTa-large introduces a mention reference prediction objective, complementary to masked language modeling, which could make the obtained coreferential reasoning ability compatible with more downstream tasks.

6 CONCLUSION

6.1 Acknowledgments

Compared with designing different training tasks for different tasks, the improvement of the ability of the pre-training model cannot be ignored for the downstream tasks. And this, since bert was born, has gradually become the consensus of everyone, and the results of this article also confirm this.A good representation should express general-purpose priors that are not task-specific but would be likely to be useful for a learning machine to solve AI-tasks and capture the implicit linguistic rules and common sense knowledge hiding in text data, such as lexical meanings, syntactic structures, semantic roles, and even pragmatics.

6.2 Document Level Relation Extraction Task

The current models are mainly divided into two categories

For sequence modeling, for example, directly use an LSTM to generate a hidden state, and then judge. Of course, some people use bidirectional LSTM, Attention-LSTM (context-aware) for modeling. The main bias introduced by this serialized modeling method is the lack of rich connection information in the document. Still in the above example, Xiao Mao and him intuitively have information that is directly related. But the sequence model, must first see "chengdian go to school", which becomes even more unreasonable in a document with a few hundred words.
For graph models, graph models can be mainly divided into two categories.

(1) Heterogeneous network graph. This kind of graph distinguishes the type of network according to different edges. In fact, it feels a bit like a method of representation learning, especially the RGCN introduced by GCNN, which is similar to the first half of SACN. The main representative work is EOG GCNN (I feel that HIN is also a bit like that, and the hierarchy is that the relationship between the levels is not clearly stated)

(2) Homogeneous network latent structure diagram
Although the types of nodes are different and the types of edges are different, we treat all nodes as homogeneous relationships, and use attention or other methods to automatically distinguish.

6.3 Summary of good methods for document-level relation extraction

6.3.1 The modeling layer contains the Mention layer. This is easy to understand. In the past few years, the paper attention is all your need has proved that the attention coefficient is strong enough, and the pre-training model of bert that uses the attention structure also proves the importance of attention.
In the sentence-level relationship extraction task, there are very few different expressions of an entity in the same sentence, so the task is usually directly modeled on the process of token->entity->entity pair->relation. In the document-level relationship extraction task, "knowledge graph" may appear first, and then "graph" appears in the following sentence. Both Mentions refer to the entity "knowledge graph".

In the specific implementation, since the embedding layer is usually at the token level, we assume that an article contains n tokens $D = \{t_1, t_2, \cdots, t_n\}$, one of which is a $m_i = \text{pooling}\left(t_j, \cdots, t_{j+k}\right)$, and the pooling method can be mean pooling or max pooling, etc., And different mention embeddings pointing to the same entity can also get entity embedding through pooling. Attention makes the embedding of each token dynamic rather than static, which is very useful for solving the problem of document-level relation extraction, where the same word has different meanings in different positions. Extract global information.

6.3.2 Extract global information. We can usually refer to the intra-sentence feature information as local features, and the inter-sentence, chapter-level feature information as global information.

The extraction of local information is basically equivalent to the encoding model of sentence-level relation extraction, such as Word2Vec/ GloVe+Bi-LSTM, BERT, etc. A feature sequence with the same length as the token sequence can be obtained.

For the extraction of global information, I divide it into three categories: methods based on hierarchical networks, methods based on global graphs, and methods based on BERT hard coding.

(1) Hierarchical Network-Based Methods. The method based on hierarchical network attempts to achieve hierarchical feature extraction from token level -> sentence level -> document level through different levels of networks, and concatenate features at different levels to classify the relationship between entity pairs.
The typical representative is HIN (Hierarchical Inference Network) , which uses BiLSTM at different levels to extract feature sequences at each level, and weights the overall features with the attention mechanism, so as to obtain local + global feature information to characterize entities .
(2) Global graph-based approach. The method based on the global graph is the most mainstream method in recent years. On the one hand, GNN is very popular, and on the other hand, GNN has unique advantages in mining structural features.
The standard process of global graph-based methods can be summarized into three parts: encoding and composition, GNN iteration (Multi-hop inference), and entity-to-relationship classification. The following figure is a flow chart of a typical global graph-based method.
(3) BERT hard coding. The methods based on BERT hard-coding try to model the global confidence through the original Transformer or a transformed Transformer that introduces external information.
For example, the article only introduced entity category and global entity labels before and after entity Mention on the basis of the original BERT, and used BiLinear to output the relationship category information between entity pairs to implement the One-pass model, while article The process of entity filtering will be added before BiLinear output relation to reduce the amount of computation.
The article "Denoising RE from Document-Level" designed three sub-tasks of Mention-Entity Matching, Relation Detection and Relation Fact Alignment to pre-train the BERT-based document-level RE model.
SSAN hard-codes whether Mention is in the same sentence and points to the same entity as Entity Structure information. The article believes that this is the dependency information of the mention level (the implementation is token level), and uses BiAffine's The method is introduced into the Score calculation part of the Transformer Encoder, and it is added as a bias to the self attention score of the token embedding, and finally the relationship between the entity pairs is output by the bilinear+multi-label sigmoid method.

6.3.3 Pretrained model. A powerful pre-training model often improves the effect of downstream tasks immediately. There are enough papers to prove this conclusion, so I won't repeat it.

7 APENDIX

7.1 Types of Named Entities

In this paper, we adapt the existing types of namedentities used in Tjong Kim Sang and De Meul-der (2003) to better serve DocRED. These typesinclude "Person (PER)", "Organization (ORG)","Location (LOC)", "Time (TIME)", "Number(NUM)", and "other types (MISC)". The types ofnamed entities in DocRED and their covered con-tents are shown below.

Types	Content
PER	People, including fictional
ORG	Companies, universities, institutions, political or religious groups, etc.
LOC	Geographically defined locations, including mountains, waters, etc. Politically defined locations, including countries, cities, states, streets, etc. Facilities, including buildings, museums, stadiums, hospitals, factories, airports, etc.
TIME	Absolute or relative dates or periods.
NUM	Percents, money, quantities
MISC	Products, including vehicles, weapons, etc. Events, including elections, battles, sporting events, etc. Laws, cases, languages, etc

Figure 13: Types of named entities in DocRED

7.2 Hyper-parameters Setting

In this section, we provide more details of our ex-periments.

Table 6: Hyper-parameters in training

Model	Batch size	Epoch
RoBERTa-ATLOP	4	30
Coref-BERT	4	32
SSAN-RoBERTa	4	40

7.3 Number of Parameters and Average Runtime

CorefBERT's architecture is a multi-layer bidirectional Transformer (Vaswani et al., 2017). Tables 13 shows the parameter number of Coref- BERTs with different model size. Compared to BERT (Devlin et al., 2019), CorefBERT add a few parameters for computing the copy-based objective. Hence, CorefBERT keeps similar number of parameters as BERT with the same size.

REFERENCES

[1] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain Adaptation via Pseudo In-Domain Data Selection. In *Empirical Methods in Natural Language Processing*.
[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations*.
[3] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Empirical Methods in Natural Language Processing*.
[4] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv: Learning* (2019).

[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. (2019).
[6] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. *arXiv: Computation and Language* (2018).
[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
[8] Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity Structure Within and Throughout: Modeling Mention Dependencies for Document-Level Relation Extraction. *arXiv preprint arXiv:2102.10249* (2021).
[9] D. Ye, Y. Lin, J. Du, Z. Liu, and Z. Liu. 2020. Coreferential Reasoning Learning for Language Representation. (2020).
[10] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research* 3 (2003), 1083–1106.