Multi-level Attention-Based Neural Networks for Distant Supervised Relation Extraction

Linyi Yang

A thesis submitted in part fulfillment of the
degree of MSc. Computer Science by Negotiated Learning
with the supervision of Dr. Ruihai Dong

School of Computer Science
University College Dublin

August 2017

ACKNOWLEGEMENTS

First and foremost, I would like to thank my supervisor, Dr. Ruihai Dong for the guidance and continuous support to accomplish the entire research project. I would like to thank him for the positive encouragement and motivation.

In addition, my heartfelt thank you extends to other colleagues, Mr. James Ng, Mr. Sixun Ouyang and Mr. Jinghui Lu that taught me how to start my work and help me fix some bugs in my code. Besides, I would like to express many thanks to the Ms. Zihui Li who manages the high-performance computer my code running on.

Last but not least, I would want to finalise my acknowledgment by thanking Prof. Barry Smyth, Ms. Olivia McCrum and UCD Insight Centre's manager, Mr. Oliver Daniels for providing such a cheerful working environment and unwavering supports when in need.

DECLARATION

"I hereby certify that this dissertation is entirely my own work. Neither the work nor parts thereof have been published elsewhere in either paper or electronic form unless indicated otherwise through referencing".

Signed: _____

Date: _____

Linyi Yang

ABSTRACT

Distant supervised relation extraction is a widely applicable approach to identify relational facts from text. However, distant supervision inevitably brings many incorrect labelling data, which would hurt the performance of relation extraction. To alleviate these issues, we propose a multi-level attention-based neural networks for distant supervised relation extraction. In this model, we first adopt gated recurrent unit to represent the semantic information. Then, we introduce a customized multi-level attention mechanism, which is expected to reduce the weight of those noisy words and sentences. Experimental results demonstrate that our model achieves significantly improvement on relation extraction tasks compared to traditional feature-based models and existing neural network based methods.

Contents

1 Introduction

In recent years, relation extraction which aims at extracting relational data from natural language text has attracted increasing research interests. It plays a key role in many natural language processing (NLP) tasks, including question answering, web search, and knowledge-based construction. Several approaches have been applied to the task of relation extraction, including supervised methods, semi-supervised methods, and unsupervised methods. According to (Zeng et al., 2014), compared with semi-supervised methods and unsupervised methods, supervised learning methods can be used to extract more effective features, and bring better performance. However, traditional supervised approaches are unlikely to apply to the extracting large amount of relations found on the Web as its require a large amount of hand-labelling training data, which is very time consuming.

Due to the limitations of the traditional supervised approaches, (Mintz et al., 2009) proposed distant supervision to automatically generate training data via aligning the New York Times news text with the large-scale knowledge base Freebase (Bollacker et al., 2008), which contains more than 7300 relationships and more than 900 million entities. They assume that if two entities have a relationship in a known knowledge base, then all sentences that contain these entity pairs will express this relationship in some way. For example, (*Ireland*, capital, *Dublin*) is a relational triple fact stored in Freebase. All sentences with synonyms for both entities, *Ireland* and *Dublin*, are considered to be an expression of the fact that (*Ireland*, capital, *Dublin*) holds. And all these sentences will be regarded as positive instances for relation contain.

Although distant supervision is an effective strategy for automatically labelling training data and a sound solution to leverage the availability of big data on the web, it suffers from wrong labelling problem as the assumption is too strong. For example, the sentence “modern Ireland also has Dublin, whose budding metropolitan area is home to about 1.5 million people of Ireland 's population of close to 4 million.” does not express the relation capital between two entities, but will still be regarded as a positive instance. The multi-instance learning introduced by (Riedel et al., 2010; Hoffmann et al., 2010; Surdeanu et al., 2012) can alleviate the wrong labelling problem but still far from satisfactory. Since these feature-based methods highly rely on the natural language processing toolkits, such as part-of-speech annotations and syntactic parsing. And the output of pre-existing NLP systems often leads to error propagation which will hurt the performance of proposed models.

To solve this problem, many scholars (Socher et al., 2012; Zeng et al., 2015; Lin et al., 2016) attempt to apply deep learning techniques instead of feature-based methods to relation extraction task, and our work will also focus on that. In the work of Zeng et al. (2015), multi-instance learning is integrated into a deep neural network model, which assumes that if two entities participate in a relation, at least one sentence that mentions these two entities might express that relation. Their integrated models are trained by selecting the most likely instance for each entity pair, and it is apparent that the method will lose a large amount of information in those neglected instances. In the work of Lin et al. (2016), they propose a sentence-level attention-based convolutional neural network (CNN) which aims to make full use of sentences by allocating different weights to different instances in terms of their contribution in expressing the semantic relation information. Their proposed method achieves better results compared to Zeng

et al. (2015). To make further use of the sentiment information of words in the text, we propose a customized attention networks integrating word-level attention with sentence-level attention-based neural networks.

Inspired by (Lin et al., 2016) and (Zhou et al., 2016), we propose a novel bidirectional Gated Recurrent Unit (BiGRU) network integrated with a multi-level attention mechanism to further improve the performance of distant supervised relation extraction. Our proposal is an extension of Lin et al., (2016) which only consider the influence of sentence-level attention. To further alleviate the wrong labelling problem, we build a multi-level attention mechanism in addition to sentence-level attention mechanism, which is expected to dynamically reduce the weights of both noisy words and sentences. Thus, it is expected to exhibit superior performance compared with the model proposed by Lin et al., (2016). We evaluate our model on a widely used dataset developed by Riedel et al., (2010) and another real-word dataset collected in Wikidata which was established on 2012 by Wikimedia community. Experimental results show that our model achieves significant improvement compared to the state-of-art methods.

The contribution of our work can be summarized as follows:

- We propose a BiGRU-based architecture for distant supervised relation extraction to automatically extract features without manual intervention. To the best of our knowledge, we are the first to use BiGRU-based recurrent neural networks with a customized attention mechanism for distant supervised relation extraction.
- As compared to existing attention-based neural networks, our model can acquire more informative instance embeddings for relation extraction by our multi-level attention mechanism.

- In the experiments, we show that our multi-level attention mechanism is beneficial on two different datasets in the task of relation extraction.

We investigate the literature review in chapter 2. The dataset we adopt is described in chapter 3. In chapter 4, we introduce our BiGRU-based model with multi-level attention mechanism in details. We evaluate the effects our model and compare the performance of our method to several state-of-the-art methods in chapter 5. Finally, we give a simple conclusion in chapter 6.

2 Literature Review

Relation extraction (RE) generally relates to the extraction of relational facts, or world knowledge from the Web (Yates, 2009). It is one of the most important subtasks in information extraction. The existing relationship extraction approaches can be divided into supervised learning methods, e.g. (Chan and Roth, 2010), semi-supervised learning methods, e.g. (Sun et al., 2011), and unsupervised learning methods, e.g. (Banko et al., 2007).

In the purely unsupervised relation extraction approaches, strings of words between entities are extracted, clustered and simplified to produce relation-strings (Banko et al. 2007). Such approaches can scale to very large amounts of data and extracting very large number of relations. However, the extracted relation instances can be very noisy and may be difficult to align with existing relations in a specific knowledge base.

Supervised approaches are the most commonly used methods for relation extraction. In the supervised approaches, sentences in a corpus are first hand labelled by domain experts to produce labelled examples of specific relations. The identified examples are then used to induce rules for identifying additional instances of relations. In other words, relation extraction is considered to be a multi-class classification problem. While supervised methods can achieve high precision, labelling training data requires enormous amount of effort from domain experts. Pervious works on supervised RE have mostly employed kernel methods, e.g. (Nguyen et al., 2011; Moncecchi et al., 2010). However, such approaches typically focus on a small number of relation types and are unlikely to scale to the thousands of relations found in text on the web.

The main drawback of these traditional supervised systems is that it acquires a large amount of hand-labelling training data, which is time consuming. For this reason, distant supervision was first proposed by Mintz et al. (2009) to make up the deficiencies. Distant supervision is an alternative learning paradigm which assumes that if two entities have a relationship in a known knowledge base, then all sentences that contain these two entities will express this relationship (Mintz et al., 2009; Hoffmann et al., 2010; Surdeanu et al., 2012). For example, (Apple, founder, Steve Jobs) is a relational fact in a knowledge base. Distant supervision will regard all sentences that contain these two entities as active instances for relation "founder". As a form of weak supervision, distant supervision exploits relation repositories including Freebase (Bollacker et al., 2008), Yago (Suchanek et al., 2007), and DBPedia (Auer et al., 2007) to define a set of relation types and identify the text in a corpus which associate with the relations to produce the training data. Over the recent years, although distant supervision has emerged as a popular choice for training relation extractors and shows promising results in the task of relation extraction, it inevitably accompanies with the wrong labelling problem. For example, the sentence "Steven Jobs passed away the day before Apple unveiled iPhone 4s in late 2011." does not express the relation "founder" but is still selected as a training instance. Hence, (Riedel et al., 2010; Hoffmann et al., 2010; Surdeanu et al., 2012) applied multi-instance learning to alleviate the wrong labelling problem. These conventional methods inherit the knowledge discovered by the Natural Language Processing (NLP) toolkits for the pre-processing tasks. Hence, their performance was intensely affected by the quality of supervised NLP toolkits. However, the output of pre-existing NLP systems often leads to error propagation or accumulation.

To address the problem described above, neural relation extraction (NRE) was introduced. It aims to extract relations from plain text with neural network models and achieves state-of-the-art for relation extraction tasks. One recent work was proposed by Zhang and Wang (2015), which utilizes bidirectional RNN to learn patterns of relations from raw text data. A RNN is trained by unfolding and back propagation through time. However, back propagation through time involving taking the product of many gradients which can lead to vanishing (component gradients less than 1) or exploding (greater than 1) gradients. As a consequence, the range of context in bidirectional RNN is limited. For instance, consider trying to predict the last word in the text "I grew up in *China*… I speak *Chinese*." Recent information suggests that the next word is probably the name of a language, but if we want to narrow down which language, we need the context of *China* from further back. It is entirely possible for the gap between the relevant information and the point where it is needed to become very large. As that gap grows, the vanishing (exploding) gradient problem prevents RNNs to remember long range information. To address this problem, Long Short-Term Memory (LSTM), a special kind of RNN, capable of learning long-term dependencies is introduced by Hochreiter and Schmidhuber (1997). It is refined and popularized by many people in following work (Cho et al., 2014; Yao et al., 2015). Furthermore, there are a variety of LSTM architectures. There have been many attempts to simplify the LSTM architecture. A major variation on the LSTM is called Gated Recurrent Unit (GRU) introduced by Cho, et al. (2014), which has attracted a lot of interests of researchers recently. The GRU network allow units to merge the cell state and hidden state. It also combines the forget and input gates into a single update gate. As the cell state used in the LSTM network, the update gate controls how much information from the previous hidden state will pass to the current hidden state. It is much simpler to compute and implement

compared with the standard LSTM. There are two main drawbacks of LSTMs. First, it compressed a lot of information into a finite sized vector. The use of a finite sized vector is a bottleneck in improving the performance. Second, although the LSTMs including GRU are resistant to exploding and vanishing gradients problems that simple RNNs have and can avoid some of the worst, mathematically both phenomena are still possible. Therefore, Bahdanau et al. (2015) first proposed to use attention mechanism for translating from matrix-encoded sentences. The attention mechanism means that we will attend to different words in the source sentence at each time step. The weighting of the input columns at each time step is called attention. The attention based models have also been applied to a wide range of tasks such as image caption generation (Xu et al., 2015), speech recognition (Chorowski et al. 2015), and neural relation extraction (Lin et al., 2016).

For relation extraction tasks, some recent works (Soher et al., 2012; Zeng et al., 2014) utilize deep neural networks in relation classification without handcrafted features. Zeng et al. (2014) employ a convolutional neural network (CNN) to extract lexical and sentence level features for relation classification which achieves better results compared with traditional distant supervision methods. Although neural network based methods provide an effective way of reducing the number of handcrafted features, these approaches which build classifier based on sentence-lever annotated data, cannot be applied to large-scale knowledge bases due to the lack of training data. Therefore, a novel model dubbed Piecewise Convolutional Neural Networks with multi-instance learning was proposed by Zeng et al. (2015). This method assumes that at least one sentence that mentions two entities will express their relation, and only selects the most likely sentence for each entity pair in training and prediction. However, in practice, this

model will lose amount of information containing in neglected sentences. To tackle this problem, Lin et al. (2016) proposed a sentence-level attention-based convolutional neural network for distant supervised relation extraction which is the first to propose adopting attention-based model in distant supervised relation extraction. In practice, it brings better performance compared to Zeng et al., (2015) on the same dataset. Although the method achieves significant improvement by utilizing all informative sentences, it ignores the semantic information on word level. Besides, a recent work was proposed by Zhou et al. (2016), which employs neural attention mechanism with Bidirectional Long Short-Term Memory Networks (BLSTM) to capture the most important semantic information in a sentence. In practice, they conduct experiments on the SemEval-2010 Task 8 dataset and achieve an F1-score of 84.0%. Their model achieves significant and consistent improvements on relation classification tasks as compared with baseline. However, similar to the work of Lin et al. (2016), their selective attention mechanism only focuses on sentence-level. Hence, we propose a BiGRU-based with multi-level attention mechanism. To the best of our knowledge, this is the first effort to adopt multi-level attention based model in distant supervised relation extraction.

3 Data

It is crucial to build a dataset of large corpus aligning with a popular relation repository. The documents sampled from the news article sources consisted of significantly higher-quality writing than the documents sampled from the message board sources. Generally, we noted that the news articles had many desirable properties. Therefore, we present relation classification performance on two separate news datasets.

First, to compare our model with several state-of-art approaches, we will conduct experiment on a widely used dataset which is generated by Riedel et al. (2010). Then, we used the second dataset developed by Lin et al. (2017) to evaluate the expansibility of our model. Entity mentions for both datasets are recognized using the Stanford open-source toolkit called entity tagger (Finkel et al. 2005). Pre-Trained Word Vectors are learned from New York Times Annotated Corpus (LDC Data LDC2008T19), which should be obtained from LDC (https://catalog.ldc.upenn.edu/LDC2008T19).

3.1 Riedel Dataset

In recent years, many large-scale knowledge bases (KBs) have been developed to store structured knowledge about the real world, such as Freebase, Wikidata, and DBpedia. KBs are playing a significant role in many AI and NLP applications such as information retrieval and question answering. The facts in KBs are typically organized in the form of triplets. Following the literature (Hoffmann et al., 2011; Surdeanu et al., 2012; Zeng et al., 2015; Lin et al., 2016), we use the relation extraction dataset introduced in (Riedel

et al., 2010). In the work of Riedel et al. (2010), they developed a dataset which was generated by aligning Freebase relations with the New York Times corpus. Entity mentions are found using the Stanford library called entity tagger developed by Finkel et al. (2005). Freebase is an online collection of structured data harvested from many sources, including individual, user-submitted wiki contributions. In this dataset, all relations are extracted from a December 2009 snapshot of Freebase. Four categories of Freebase relations are used: “people”, “business”, “person”, and “location”. These types of relations are chosen because they appear frequently in the newswire corpus. Inspired by Riedel et al. (2010), the Freebase relation instances are divided into two parts, one for training and one for testing. For the choice of text corpus, the New York Times corpus was used. The sentences from the years 2005-2006 of the NYT corpus are aligned with Freebase relations to generate the training instances. And the test instances are generated by aligning the sentences from 2007 to 2010 with Freebase relations.

There are 53 possible relationships within this dataset including a special relation NA which represents that there is no relation between two entities. A total of 522,611 sentences, 281,270 entity pairs and 18,252 relational facts are stored in the training data, while the testing data includes 172,448 sentences, 96,678 entity pairs, and 1,950 relational facts. The details of the relations are shown in the following table 3.1.

Table 3.1 Part of relations stored in the Freebase

Relation	Example
NA	*(British, Dublin)*
/location/neighborhood/neighborhood_of	*(Bushwick, Brooklyn)*
/business/company/founders	*(Pixar, Steve Jobs)*
/people/person/place_of_birth	*(Bill Gates, Seattle)*
/people/deceased_person/place_of_death	*(Julius Caesar, Rome)*
/people/person/religion	*(Hamid Karzai, Islam)*
/business/company/place_founded	*(BBC, London)*
/business/person/company	*(Miuccia Prada, Prada)*
/business/business_location/parent_company	*(Milan, Prada)*
/location/country/capital	*(Ireland, Dublin)*

3.2 Wiki Dataset

After empirically comparing our work with the previous work, we would continue to test whether our model can achieve state-of-art performance on another dataset. For this purpose, inspired by Lin et al. (2017), we generate our second dataset by aligning English Wikipedia articles with the Wikidata relations. Specifically, for each entity pair, we generate relation candidates and distantly label source sentences using our Wikidata knowledge base. The relational facts of Wikidata in this dataset are divided into two parts for training and testing respectively. There are 176 relations including a special relation NA indicating there is no relation between entities. There are 1,022,239 sentences, 47,638 relational facts stored in the training data, while the testing data includes 162,018 sentences, and 4,326 relational facts.

4 Methodology

Distant supervised relation extraction problem is considered as a multi-instance problem. In this chapter, we present a novel neural network architecture that incorporate multi-level attention mechanism into a bidirectional gated recurrent unit network to fulfil this task. Figure 4.1 shows our neural network architecture for distant supervised relation extraction which demonstrates the pipeline that handles one instance of a bag.

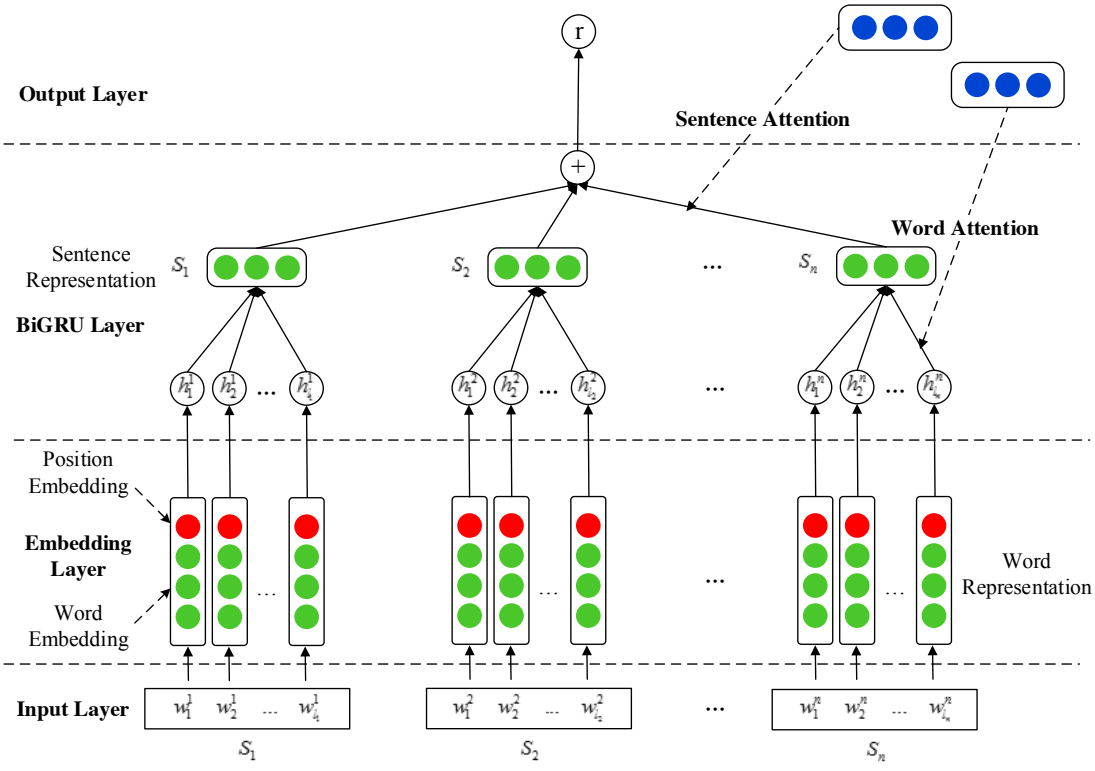

Figure 4.1 Bidirectional LSTM model with multi-level attention

As shown in Figure 4.1, the model proposed in this paper contains six components:

(1) Input layer: original sentences input to this model;

(2) Embedding layer: each word is mapped into a 50-dimension vector;

(3) BiGRU layer: Using a neural network to get features automatically;

(4) Word attention: produce a weight vector on word level, and merge the word-level features into a sentence-level representation;

(5) Sentence attention: allocate different weights to different sentences in terms of their contribution in expressing the semantic relation information;

(6) Output layer: extract relation with the relation vector weighted by sentence-level attention.

These components will be introduced in detail in this chapter.

4.1 Vector Representations

4.1.1 Word Embeddings

In order to bring the matter of natural language understanding into machine learning, the first step is to find an approach to mathematically model words. The most intuitive and commonly used word representation so far is One-hot Representation in the task of NLP. For example, the word “Apple” can be represented as [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 …], and the word “Banana” can be represented as [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 …]. However, one-hot representation has two main shortcomings. First, the dimension of a vector increases accordingly when the number of words in the sentence increases. Second, any two words represented by One-hot Representation are isolated and cannot express the information between words at the semantic level. Hence, in the task of deep learning, we usually adopt distributed representation method which was first used by Hinton et al. (1986) instead of one-hot representation.

Distributed presentation is introduced as we need richer representations expressing semantic similarity. It can help learning algorithms to perform better in NLP tasks by

grouping similar words in a latent space. Such representation is usually called world embedding which is used widely for feature learning in NLP. There are serval language modelling algorithms to achieve the goal of word embedding, including word2vec introduced by Mikolov et al. (2013) and GloVe developed by Stanford. Specifically, given a sentence x consisting of m words x = $\{w_1, w_2, \ldots, w_{m-1}, w_m\}$, every word w_i is represented by a valued vector. In the work of Mikolov et al. (2013), they employed two novel model architectures for computing continuous vector representations of words from very large data sets, one is called Continuous Bag-of-Words Model (CBOW), another is called Continuous Skip-gram Model. The input of the CBOW model could be w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, the preceding and following words of the current word. The output of the neural network will be w_i. Hence, CBOW model could predict a centre word from the surrounding context. In contrast, we can take Skip-gram model as predicting surrounding context words given a centre word. The input of the Skip-gram model is w_i, while the output could be w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}. According to Mikolov et al. (2013) , Skip-gram model works well with small amount of the training data, represents well even rare words or phrases, whereas CBOW is several times faster to train than the skip-gram, slightly better accuracy for the frequent words. In this work, we use a pretrained word vector dataset originally released by Lin et al. (2016). This dataset is learned from New York Times Annotated Corpus (LDC Data LDC2008T19) using Skip-gram model and the number of dimension is set to 50.

4.1.2 Position Embeddings

Position embeddings are first proposed by (Colobert et al. 2011) for semantic role labelling. The main idea behind the use of word position embedding in relation extraction task is to give some reference to the convolutional layer of how close a word

is to the target nouns, based on the assumption that closer words have more impact than distant words. For instance, in the sentence "$[Steve\ Jobs]_{entity1}$ was the co-founder, chairman, and chief executive officer of $[Apple\ Inc]_{entity2}$.", the relative distances of *co-founder* to *Steve Jobs* and *Apple Inc* are respectively 3 and -9. Each relative distance is further mapped to a randomly initialized hyperparameter d_{pf} dimensional vector. Supposing pf_3 and pf_{-9} are the corresponding vectors of 3 and -9, the word position embeddings feature of *co-founder i*s given by concatenating these two vectors $[pf_3, pf_{-9}]$.

The experimental result reported in (dos Santos et al., 2015) suggests that the use of word position embeddings is informative. This effect of word position embeddings is first reported by (Zeng et al., 2014). In practice, the position embedding is further used as a vector concatenated with the word embedding.

4.2 Bidirectional Recurrent Neural Networks

A recurrent neural network (RNN) can be thought of as multiple copies of the same network, each passing a message to a successor. The unfolded recurrent neural network is depicted in Fig. 4.2. For instance, if the input is a sentence with 10 words, it will compute five times, and the RNN rolls into a 10-layer neural network, with each word in a layer.

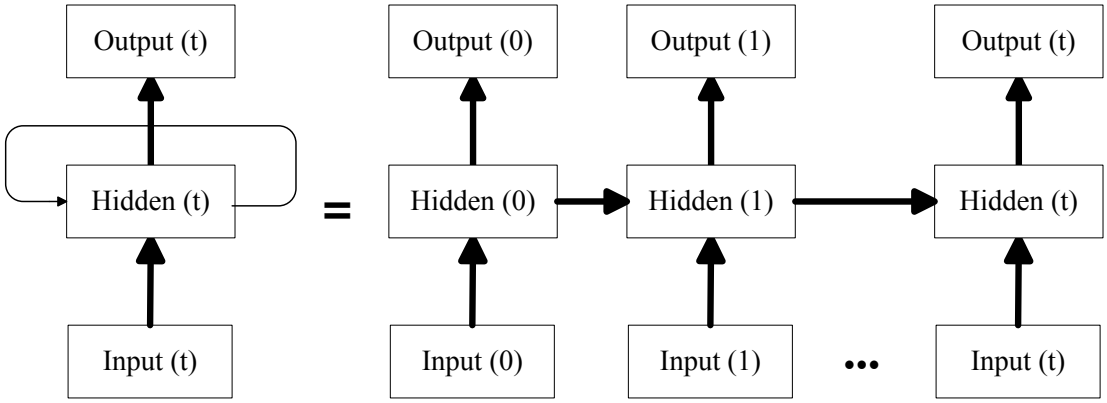

Figure 4.2: An unfolded recurrent neural network

4.2.1 Bidirectional Long Short-Term Memory Networks

We can train an RNN by unfolding and back-propagating through time, summing the derivates for each weight as we go through the sequence. However, in an RNN, the gradient blows up or decays exponentially over time due to the back propagation through time. Therefore, Long Short-Term Memory network (LSTM), a special kind of RNN, is first proposed by Hochreiter and Schmidhuber (1997) to overcome the gradients vanishing or exploding problem. The key to LSTM is having cell state cooperating with the adaptive gating mechanism. The cell state works like a horizontal line running through the top of the diagram. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by a new unit type called gates. Gates are a way to optionally let information through in LSTM units, which includes input gate, forget gate, and output gate. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation. The final LSTM units are illustrated in Fig. 4.3.

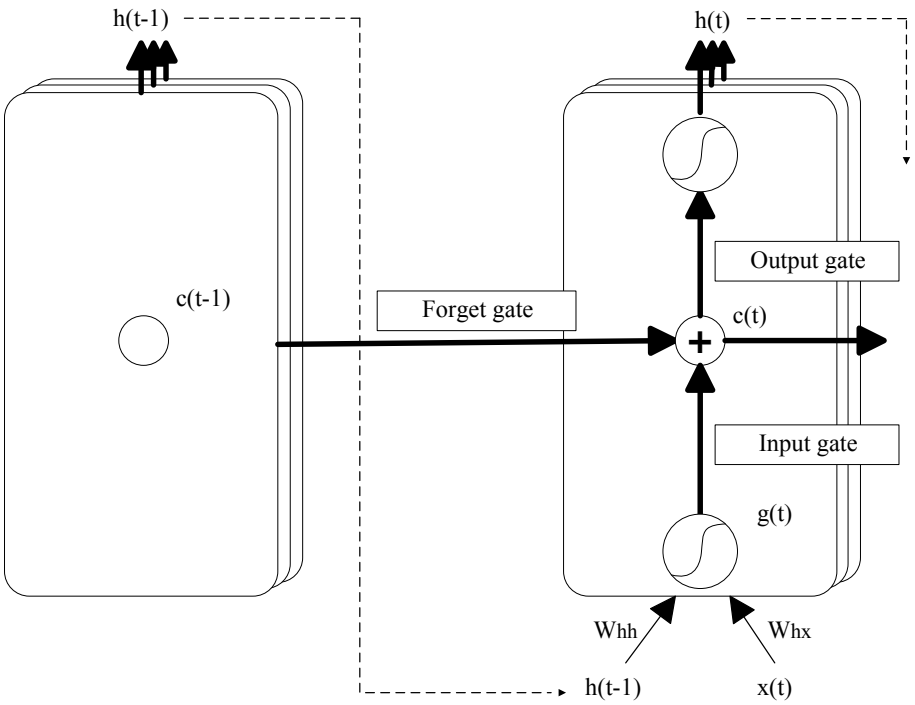

Figure 4.3: The repeating module in an LSTM

The first step in LSTM is to decide what information will be thrown away from the cell state. This decision is made by a sigmoid layer called the "forget gate". It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state c(t). A value 1 represents "completely keep this" while a 0 represents "completely forget the past". The next step is to update the cell state. First. the "input gate" controls which values will be updated in the internal state c(t). Next, a hyperbolic tangent layer creates a vector of new candidate values, g(t), that could be added to the state vector of new candidate values. Input and forget gates together allow the network to control what information is stored and overwritten at each step. c(t) combines previous state c(t-1) and the current LSTM input g(t). Finally, we need to control how much of each unit's activation is output by the hidden state. First, a sigmoid layer which decides what parts of the cell state we're going to output will be applied. Then, the cell state will be put through hyperbolic tangent (pushing the values between -1 and 1) and multiply it by the output of the sigmoid gate. Just as these following equations demonstrate:

$$\mathrm{I(t)} = \sigma(W_{ix}x(t) + W_{ih}h(t-1) + b_i) \quad (1)$$

$$\mathrm{F(t)} = \sigma(W_{fx}x(t) + W_{fh}h(t-1) + b_f) \quad (2)$$

$$\mathrm{O(t)} = \sigma(W_{ox}x(t) + W_{oh}h(t-1) + b_o) \quad (3)$$

$$\mathrm{g(t)} = \tanh(W_{cx}x(t) + W_{ch}h(t-1) + b_c)) \quad (4)$$

$$\mathrm{c(t)} = \mathrm{I(t)g(t)} + \mathrm{F(t)c(t\text{-}1)} \quad (5)$$

$$\mathrm{h(t)} = \mathrm{O(t)}\tanh(c(t)) \quad (6)$$

where σ represents the sigmoid function, tanh represents the hyperbolic tangent, $W_{ix\ldots ch}$ terms denote weight matrix, the b terms denote bias vectors, $x(t)$ and $h(t-1)$ both are the input vector and pervious hidden state respectively, $h(t)$ is the output vector, c(t) is the cell state vector, and I(t), F(t), and O(t) are the input gate, forget gate, and output gate respectively. With initial values $\mathrm{c}(0) = 0$ and $\mathrm{h}(0) = 0$.

Bidirectional Long Short-Term Memory network (BLSTM) networks are based on the idea that the output at time *t* may not only depend on the past information, but also the future information. In practice, it contains two sub-networks in the hidden layer, which are forward and backward pass respectively. The forward LSTM is used to exploit the information from the past, while the backward LSTM is used to capture the future information. Combining the forward and backward by element-wise sum, the output is shown in the following equation:

$$\mathrm{h(t)} = \left[\overrightarrow{h(t)} \oplus \overleftarrow{h(t)}\right] \quad (7)$$

where $\oplus$ is the element-wise sum of the vectors. BLSTM networks therefore usually perform better than unidirectional LSTM networks. For this reason, we use BLSTM in this work.

4.2.2 Bidirectional Gated Recurrent Unit Networks

The network described above is a standard LSTM. There is a popular LSTM variant, peephole connections, proposed by Gers and Schmidhuber (2000). In this model, the gate layers will look at the cell state. Besides, a more dramatic variant on the LSTM is the Gated Recurrent Unit (GRU) network, which was first introduced by Cho, et al. (2014). Let us describe how the GRU works. Taking how the *j*-th hidden unit is computed as an example. First, it merges the cell state and hidden state then generates the *reset* gate r_j , which is computed by:

$$r_j = \sigma\big([W_r x]_j + [U_r h(t-1)]_j\big) \tag{8}$$

where σ represents the sigmoid function, $[.]_j$ is the *j*-th element of a vector, x and $h(t-1)$ are the input vector and pervious hidden state respectively, and W_r and U_r denote weight matrices.

Second, it combines the forget and input gates into a single *update* gate. The *update* gate z_j is computed by:

$$z_j = \sigma\big([W_z x]_j + [U_z h(t-1)]_j\big) \tag{9}$$

Finally, the actual activation of the proposed unit h_i is computed by:

$$h_j(t) = z_j h_j(t-1) + (1 - z_j)\widetilde{h_j}(t) \tag{10}$$

where

$$\widetilde{h_j}(t) = tanh\left([Wx]_j + \big[U\big(r \odot h(t-1)\big)\big]_j\right) \tag{11}$$

In the GRU network, when the reset gate is close to 0, the hidden state is prompted to ignore the pervious hidden state and reset with the current input at the same time. Then, like the cell state used in the LSTM network, the update gate controls how much

information from the previous hidden state will pass to the current hidden state. It is much simpler to compute and implement compared with the normal LSTM.

Finally, we combine the states produced by the LSTM layer from left to right and negative direction together into the output of the j^{th} word.

$$h_j(t) = \left[\overrightarrow{h_j(t)} \oplus \overleftarrow{h_j(t)}\right] \tag{12}$$

4.3 Attention

Attention-based neural networks are first introduced by Bahdanau (2015) for sequence to sequence learning in machine translation. In this section, we adopt the attention mechanism for distant supervised relation extraction tasks. Our attention mechanism aims to use word-level attention to obtain the representations of the sentences, then employ sentence-level attention to reduce the influence of false-negative sentences existing in entity pairs.

4.3.1 Word-level Attention

After utilizing the bidirectional GRU (BGRU) network to exploit information both from the past and future, we obtain the hidden state of recurrent networks. However, not all words contribute equally to the semantic relation information of an entity pair. Therefore, instead of feeding the hidden layer of each LSTM unit to the sentence-level attention layer directly, we introduce a word-level attention mechanism to select the informative words which really express the relation between entity pairs. The word-level attention would dynamically pay attention to the words in sentences that are more significant for semantic relation information.

In practice, suppose we are given an instance s containing n words, and every word including entity identifiers is mapped to a real-valued vector by word embeddings. Then word embeddings are passed to GRU units respectively to get hidden states [h (1), h (2), …, h (T)], where *T* is the length of the given sentence. Let H be a matrix consisting these hidden states produced by GRU. Then, we form the representation r of the sentence by a weighted sum of these informative words based on how important they are at current time step. Inspired by Zhou et al., (2016), we can obtain the representation of sentences through the following equations:

$$\mathrm{M} = \tanh(H) \tag{13}$$

$$\alpha = softmax(w^T M) \tag{14}$$

$$\mathrm{r} = H\alpha^T \tag{15}$$

where $H \in \mathbb{R}^{d^w \times T}$, d^w is the dimension of the word vectors, T is the length of the given sentence, w is a learning parameter of appropriate dimension and w^T is a transpose. The weights of the input columns at each time-step is called attention α. Moreover, the dimensions of w, α, r are d^w, T, d^w respectively.

Finally, we obtain the sentence representations for the subsequent sentence-level attention layer from:

$$s_i = \tanh(r) \tag{16}$$

4.3.2 Sentence-level Attention

After getting all the representations of sentences corresponding to each entity pair respectively, we use them to acquire the representation of the set S which contains n sentences for entity pair (entity1, entity2), S = $\{s_1, s_2, \ldots, s_i\}$. Then, the set S is represented with a real-vector *S* when predicting the relation *r*. Inspired by Lin et al.,

(2016), the representation of S is computed as a weighted sum of these sentence vectors $\{s_1, s_2, \ldots, s_i\}$:

$$S = \sum_{i=1}^{i} \alpha_i s_i \tag{17}$$

where α_i is the weight of each sentence vector s_i.

According to Lin et al. (2016), the semantic information of set S would rely on the representations of all the sentences, each of which contains information that whether the entity pair expresses the relation. However, due to the existing false-positive sentences in distant supervision for relation extraction, if we assume that each sentence contributes equally, the wrong labelling sentences will bring in massive amount of noise during training which would degrade the performance of our model. Therefore, we adopt a sentence-level attention to minimize the influence of the noisy sentences. Hence, α_i is calculated as:

$$\alpha_i = \frac{\exp(\Phi(s_i, r))}{\sum_k \exp(\Phi(s_k, r))} \tag{18}$$

where $\Phi(\cdot)$ is a query-base function which scores how well the input sentence s_i and the relation *r* matches. $\Phi(\cdot)$ is defined as:

$$\Phi(s_i, r) = s_i A r \tag{19}$$

where A denotes a weight matrix, and *r* is the representation of relation *r*. The sentence-level attention mechanism first measures the relevance between the instance embedding and the relation *r*. Then, it would allocate more weight to true-positive instances and less weight to wrong labelling instances to reduce the influence of noisy sentences.

4.4 Output

The output layer determines the relation label of an input sentence set. In practice, we calculate the conditional probability through a softmax function as:

$$\mathrm{p(r|S)} = \frac{\exp(\mathrm{o_r})}{\sum_{\mathrm{k=1}}^{\mathrm{n_r}} \exp(\mathrm{o_k})} \tag{20}$$

where n_r denotes the number of relations and o is the output of our model, which is defined as:

$$\mathrm{o = RS + b} \tag{21}$$

where R is the representation matrix of relations and $b \in \mathbb{R}^{n_r}$ is a bias vector.

Inspired by Zeng et al., (2015) and Lin et al., (2016), we employ a loss function using cross-entropy at the entity-pair level. Then loss function is defined as:

$$J(\theta) = \sum_{i=1}^{N} \log p(r_i|S_i;\ \theta) \tag{22}$$

where N denotes the number of sentence sets for each entity pair and θ indicates all parameters of this model. For optimization problem, we adopt the Adaptive Moment Estimation (Adam) update rule to learn parameters by minimizing the loss function. For learning, we randomly pick a certain number of instances as a mini-batch from the training set, and iterate this step until converge.

Furthermore, in order to prevent overfitting, we apply dropout (Srivastava et al., 2014) on the output layer. The strategy of dropout aims to achieve better performance during testing phase by randomly dropping out neural units during training phase. Then, the output of our model is rewritten based on equation (21) as follows:

$$\mathrm{o = R(S \circ h) + b} \tag{23}$$

Finally, the scaled set vector $\widehat{r_i}$ is used to predict relations during the test phase.

5 Experiments

In this chapter, we describe our experiments to evaluate our relation extraction system. We evaluate our model on two different knowledge bases, including Riedel's dataset and Wiki dataset. The generation of these two datasets is described in detail in chapter 3. Our experiments aim to illustrate that our deep neural networks with sentence-level attention integrated with word-level attention can alleviate the wrong labelling problem benefiting from taking advantage of all informative words for relation extraction. Empirical experiments use a customised GRU library written in Python and an open-source platform Tensorflow.

We first specify our settings such as values of hyper-parameters and describe the methods that we use for our evaluations. Next, we compare the performance of our model on a widely used dataset with several state-of-the-art methods, including traditional featured-based methods and neural network approaches. And we show that our approach, BiGRU+2ATT, can consistently and effectively improve the previous best performing model, PCNN+ATT. Finally, we evaluate our model on a Wiki dataset to demonstrate its consistency.

5.1 Evaluation metrics

Like Mintz et al., (2009), we adopt held-out evaluation to assess our model in distant supervised relation extraction. The held-out evaluation compares the relation facts between entity pairs discovered from the test articles with those in knowledge base. However, the new relation instances that are not in knowledge base also could be discovered by the testing systems. We just assume that the testing systems have similar

performance in relation facts inside and outside knowledge base so that we can provide an approximate measure of precision without manually evaluation. Here we report both the precision/recall curves and Precision @ N (P@N).

5.2 Experiments settings

5.2.1 Word Embeddings Setup

The word embeddings used in this work are initialized by means of unsupervised pretraining. Similar to previous work (Lin, et al., 2016), we use the Skip-gram neural network architecture available in the word2vec tool developed by Mikolov et al. (2013). For both datasets, we adopt the same NYT corpus to train word embeddings with word2vec. We first drop the words which appear less than 100 times in the corpus and keep the rest as our vocabulary set. Then, we generate the word embedding in 50 dimensions. Finally, we concatenate the words of an entity when it has multiple words.

5.2.2 Parameter Settings

For the Riedel dataset, we keep the same value and size of parameter with the baseline (Lin et al., 2016) in order to highlight the increase of performance comes from method rather than the increase of parameter size. Specifically, we select learning rate λ for Adam optimizer as 0.001, the sliding window sides l as 3, the batch size as 160, and sentence embedding size n as 230.

For the Wiki dataset, we tune the hyperparameters using three-fold validation on the training corpus following previous work. In the training phrase, the batch size is fixed to 160. After selecting parameters among possible values, the optimal parameter values

of parameters were obtained. We illustrate hyperparameters used in the experiments on two different datasets respectively in Table 5.1.

Table 5.1: Parameter settings

Dataset	Freebase	Wiki
Window size l	3	3
Sentence embedding size	230	230
Word dimension	50	50
Position dimension	5	5
Batch size	**50**	**16**
Dropout probability	0.5	0.5

5.3 Comparison with Previous Approaches

5.3.1 Baselines

To evaluate the proposed method, we compare our approach against three representative feature-based methods and four neural network approaches to show that our model can improve the performance of the previous best-performing model.

Feature-Based Methods:

- **Mintz** (Mintz et al., 2009): This is an original model for distant supervised relation extraction based on the idea that if two entities have a relationship in a known knowledge base, then all sentences that contain these two entities will express this relationship.
- **MultiR** (Hoffmann et al., 2011): This is a DS for a relation extraction model based on multi-instance learning which handles overlapping relations. MultiR transforms relation extraction into a multi-instance problem, but learns using a perceptron algorithm and uses a “at least-one” assumption.

- **MIML** (Surdeanu et al., 2012): It jointly models the latent assignment of labels to instances and dependencies between labels assigned to the same entity pair.

Neural Network Methods:

- **CNN/PCNN** (Zeng et al., 2015): A convolutional neural network (CNN) is used to embed contextual sentences for relation classification. PCNN is an improved CNN model integrated with multi-instance learning. In PCNN, piecewise max pooling is used to handle the three pieces of a contextual sentence (split by the two entities) separately. Both methods select only one instance in each instance set to train and test.
- **CNN+ATT/PCNN+ATT** (Lin et al., 2016): This work proposes a convolution neural network model based on sentence-level attention mechanism and adopts the model developed by Zeng et al., (2015) as the benchmark system. It takes advantage of all informative sentences to strengthen the previous model. PCNN+ATT is the best-performing model on the Riedel dataset so far.

5.3.2 Results Analysis and Evaluation

Following pervious works, we use held-out evaluation to assess whether our model is consistently effective. This evaluation provides an approximate measure of precision and recall without manually evaluation.

Comparison with Feature-based Methods:

We implement three feature-based methods with the source codes released by the authors and the results are shown in Figure 5.1. It demonstrates the precision and recall curves for the three baseline models developed respectively by Mintz, Hoffmann, and Surdeanu and our proposed model.

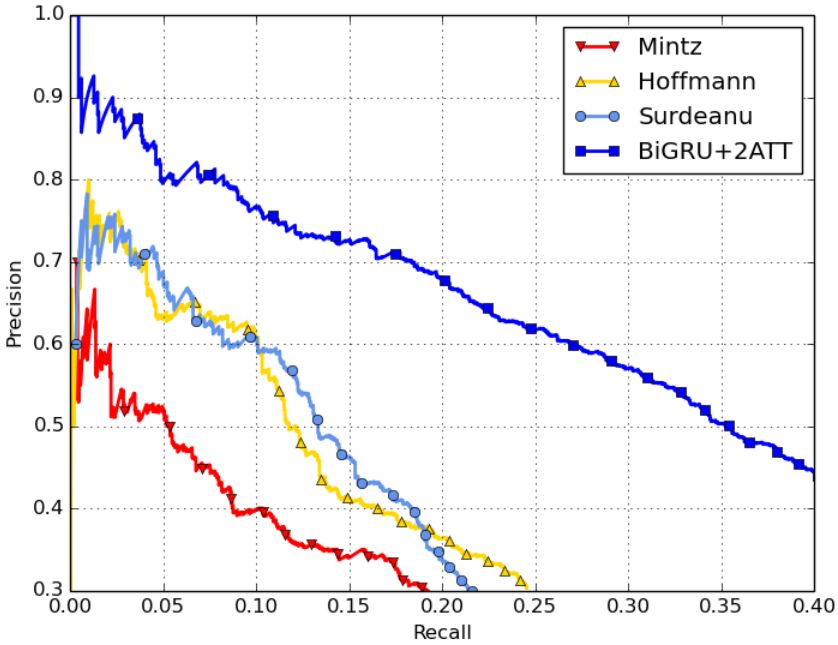

Figure 5.1: Performance comparison of precision and recall curve for the Riedel dataset of the proposed model and three feature-based baseline methods.

We can note that our model significantly and consistently outperforms the three feature-based methods over the entire range of recall. Besides, it is worth noting that when the recall is greater than 0.1, the performance of feature-based methods drops quickly while our model still has a reasonable precision. It illustrates that the artificially designed feature not only cannot express the semantic meaning of the sentences, but also cannot alleviate the problem of inevitable error brought by NLP tools. In contrast, our model which learns the semantic representations of sentences automatically can express each sentence well and alleviate the problem of wrong labelling.

Comparison with Neural Network Methods:

We implement the previous best-performing model PCNN+ATT proposed in (Lin e al., 2016) by ourselves which achieves comparable results as the authors reported.

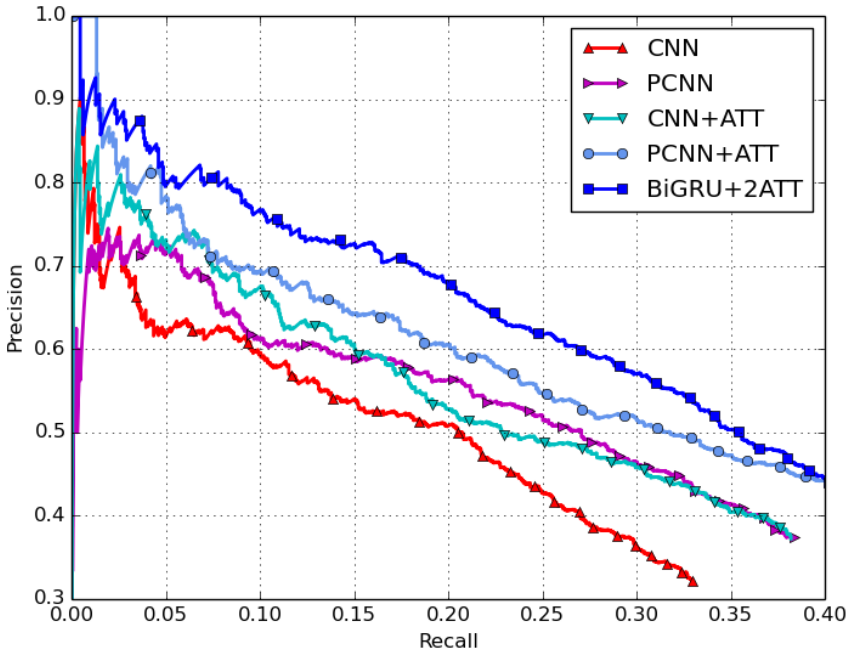

Figure 5.2: Performance comparison of precision and recall curve for the Riedel dataset of the proposed model and state-of-the-art neural network methods.

To demonstrate the effects of our customized attention mechanism, we empirically compare different methods. The CNN model is proposed by Zeng et al. (2014) and the PCNN model is also proposed by Zeng et al. (2015). Besides, we select the CNN and PCNN model with selective attention over instances proposed by Lin et al. (2016) as our baselines which achieve the best performance so far on the Riedel dataset. The results are illustrated in Figure 5.2 which shows that our model achieves the best performance among the four neural network approaches.

From Figure 5.2, we observe that for both CNN and PCNN, the models with sentence-level attention mechanism brings better results as compared to original CNN and PCNN respectively. The reason is that the selective attention can effectively filter out meaningless sentences and reduce the negative effects of false-positive instances

through degrading their weight. By contrast, both CNN and PCNN models only select the most confident instance in the instance set to train and test. Moreover, compared with both CNN+ATT and PCNN+ATT, BiGRU+2ATT obtains better performance over the nearly entire range of recall. It indicates that the proposed LSTM-GRU integrated with word-level attention is beneficial. The reason is that the word-level attention would dynamically focus on the more informative words within sentences for the given relation.

5.3.3 Case study

The selected three examples of our customized attention mechanism from the testing set are shown in the Table 5.2. For each sample, we display our attention weights of sentences, and we highlight the entity pairs with bold face.

From Table 5.2, we see that: The first bag of sentences is composed of two sentences which are related to the triple *Founders (Robert L. Johnson, Black Entertainment Television)* which is stored in the Freebase. The sentence with low attention weight does not express the relation *Founders* clearly. While the next sentence, with high attention weight, demonstrates directly that *Robert L. Johnson* found *Black Entertainment Television*. The second example is related to the triple *Founders (Muhammad Yunus, Grameen Bank)*. In this example, the relation fact also contains two sentences. The first sentence with low attention weight expresses the relation *Founders* implicitly, while the high one expresses directly what position *Muhammad Yunus* holds in the *Grameen Bank***.** The last example is related to the triple *Contains (Ireland, Cork)*. The result demonstrates that the two sentences have the equal attention weight expressing the relation that Cork is located in the Ireland.

Table 5.2: Three examples of sentence-level attention in NYT corpus

Relation	Founders
Low 0.0674	Sports sunday new act for a media mogul **Robert L. Johnson** sold **Black Entertainment Television** in 2000.
High 0.9326	For mrs. clinton, the strategy for reaching black voters at this early stage of the campaign … followed by phone calls to reinforce her candidacy from her husband and supporters like **Robert L. Johnson**, who founded **Black Entertainment Television**.
Relation	Founders
Low 0.0063	**Muhammad Yunus**, who won the Nobel peace prize last year, demonstrated with **Grameen Bank** the power of microfinancing.
High 0.9937	On sunday, though, there was a significant shift of the tectonic plates of Bangladeshi politics as **Muhammad Yunus**, the founder of a microfinance empire known as the **Grameen Bank** and the winner of the 2006 Nobel peace prize …
Relation	Contains
Equal 0.5	ConocoPhillips, the third-largest American oil company, began producing some diesel from soybean oil last year at a plant in **Cork**, **Ireland.**
Equal 0.5	Zingerman's is unique in that it has a continental reach in the united states, said peter foynes, curator of the butter museum in **Cork**, **Ireland** …

5.4 Performance on Wiki Dataset

In order to test the generality of our model, BiGRU + 2ATT, we execute our model on another dataset aligning wiki articles with relations in the Wikidata. In practice, we evaluate the performance of our model under different conditions to prove the robustness of our model.

In the original testing data set, there are many entity pairs that correspond to only one sentence. Like Lin et al. (2016), since our multi-level attention mechanism works on the entity pairs containing various sentences, we select the entity pairs which have more than one sentence to compare the performance on the BiGRU with sentence-level attention (BiGRU+ATT) and BiGRU with word-level attention integrated with sentence-level attention (BiGRU+2ATT). Then, we evaluate our models in different settings:

- **One**: For each entity pair within the testing set, we randomly select one sentence and use this sentence for relation extraction.
- **Two**: For each entity pair within the testing set, we randomly select two sentences and then use these two sentences to predict relation.
- **All**: Using all sentences of each entity pair to test.

Note that, we use all the training instances during training.

Table 5.3: P@N for the top 100, top 200, and top 300 extracted relation instances

Test Settings	One			Two			All		
P@N (%)	100	200	300	100	200	300	100	200	300
BiGRU+ATT	**83.0**	72.0	66.0	79.0	77.0	71.0	85.0	80.0	76.7
BiGRU+2ATT	**83.0**	**74.0**	**67.0**	**86.0**	**81.5**	**76.3**	**88.0**	**82.5**	**77.7**

Table 5.3 presents the P@N for the top 100, top 200, and top 300 extracted instances. For BiGRU, the multi-level attention method achieves the best performance in most of test settings. The results show that our multi-level attention mechanism can consistently improve the performance compared to only using neural network with sentence-level attention.

6 Conclusion

In this dissertation, we develop BiGRU with multi-level attention, which automatically realizes learning features from data and makes full use of all informative words and sentences. We adopt word-level attention integrated with sentence-level attention to achieve better instance representation for the distant supervised relation extraction task. In practice, we evaluate our model on two different datasets to present the effect of multi-level attention mechanism. Experimental results show that our model outperforms not only state-of-the-art feature based methods also neural network methods.

7 References

[1] Zeng, D., Liu, K., Lai, S., Zhou, G. and Zhao, J. (2014). T Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING* pp. 2335-2344.

[2] Mintz, M., Bills, S., Snow, R. and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*: Volume 2-Volume 2 pp. 1003-1011.

[3] Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* pp. 1247-1250.

[4] Riedel, S., Yao, L. and McCallum, A. (2010). Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases*, pp.148-163.

[5] Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L. and Weld, D.S. (2011). Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* pp. 541-550.

[6] Surdeanu, M., Tibshirani, J., Nallapati, R. and Manning, C.D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* pp. 455-465.

[7] Socher, R., Huval, B., Manning, C.D. and Ng, A.Y. (2012). Semantic

compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* pp. 1201-1211.

[8] Zeng, D., Liu, K., Chen, Y. and Zhao, J. (2015). Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *Proceedings of EMNLP* pp. 1753-1762.

[9] Lin, Y., Shen, S., Liu, Z., Luan, H. and Sun, M. (2016). Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* pp. 2124-2133.

[10] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H. and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* pp. 207-212.

[11] Yates, A. (2009). Extracting world knowledge from the web. *Computer*, 42(6).

[12] Chan, Y.S. and Roth, D. (2010). Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics* pp. 152-160.

[13] Sun, A., Grishman, R. and Sekine, S. (2011). Semi-supervised relation extraction with large-scale word clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* pp. 521-529.

[14] Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M. and Etzioni, O. (2007). Open Information Extraction from the Web. In *Proceedings of IJCAI* pp. 2670-2676.

[15] Nguyen, T.V.T. and Moschitti, A. (2011). End-to-end relation extraction using

distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* pp. 277-282.

[16] Moncecchi, G., Minel, J.L. and Wonsever, D. (2010). A survey of kernel methods for relation extraction. In *Workshop on NLP and Web-based Technologies*.

[17] Suchanek, F.M., Kasneci, G. and Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web* pp. 697-706.

[18] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *The semantic web* pp.722-735.

[19] Zhang, D. and Wang, D. (2015). Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006.*

[20] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9(8)* pp.1735-1780.

[21] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[22] Yao, K., Cohn, T., Vylomova, K., Duh, K. and Dyer, C. (2015). Depth-gated LSTM. *arXiv preprint arXiv:1508.03790*.

[23] Bahdanau, D., Cho, K. and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[24] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of International Conference on Machine Learning* pp. 2048-2057.

[25] Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K. and Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems* pp. 577-585.

[26] Lin, Y., Liu, Z. and Sun, M. (2017) Neural Relation Extraction with Multi-lingual Attention. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* pp. 34-43.

[27] Finkel, J.R., Grenager, T. and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics* pp. 363-370.

[28] Hinton, G.E. (1986). Learning distributed representations of concepts. In Proceedings of the eighth annual conference of the cognitive science society Vol. 1, pp. 12.

[29] Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

[30] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12: pp. 2493-2537.

[31] Santos, C.N.D., Xiang, B. and Zhou, B. (2011). Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*.

[32] Gers, F.A. and Schmidhuber, J. (2000). Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference*. Volume 3, pp. 189-194.

[33] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal*

of machine learning research, pp.1929-1958.