Deep Embedding for Relation Extraction on Insufficient Labelled Data

Haojie Huang
School of Computer Science & Engineering
University of New South Wales
Sydney, Australia
haojie.huang@unsw.edu.au

Raymond Wong
School of Computer Science & Engineering
University of New South Wales
Sydney, Australia
wong@cse.unsw.edu.au

***Abstract*—Many recently proposed relation extraction methods are based on distantly supervised learning. They use data from existing knowledge bases as training data. Although the methods solve the problem of insufficient labelled data and are highly scalable, they suffer from a large amount of incorrectly labelled data. Instead of using these distantly supervised approaches, this paper proposes an alternative relation extraction method. It firstly performs unsupervised learning to train relation embeddings by a neural network. As the relation embeddings encode the semantic information of the original sentences and their entity pairs, these embeddings can be efficiently classified by supervised learning. Since the relation embedding phase is based on unsupervised learning, labelled data is only required in the classification phase. Experiments show that our proposed approach significantly outperforms the state-of-the-art baselines when labelled training data is insufficient.**

***Index Terms*—relation extraction, word embedding, machine learning**

I. Introduction

Constructing a large scale knowledge base like DBpedia [1] and Wikidata [2] involves complicated natural language processing (NLP) tasks. One of the critical tasks is relation extraction (RE), which attempts to extract semantic relationships between entity pairs from a sentence [3].

One crucial problem in the research of relation extraction is the lack of high-quality labelled data. One of the well-known approaches for addressing the problem is distant supervision, which uses relations from existing knowledge bases to helps the extraction process [4], [5]. Although the method is effective for labelling a large amount of data, it introduces wrong labels. One reason for the problem is the existing knowledge bases may not contain the correct relation type for the entity pairs. In another case, the knowledge bases may contain a relation for the entity pairs, but the relation type is wrong. In recent years, relation extraction approaches based on neural networks [6]–[8] achieve state-of-the-art performance. Comparing to traditional statistical approaches, such as graphical models [5], [9], neural relation extraction employs deep learning models such as convolution neural networks (CNN) and recurrent neural networks (RNN) for automatically learning language features within the text. However, these neural network approaches also rely on a large amount of data, and they also suffer from noisy training data.

In this paper, we propose a novel relation extraction framework that consists of two phases. The first phase performs unsupervised learning to learn relation embeddings. More specifically, it employs a bidirectional LSTM (BiLSTM) model with attention mechanism to encodes the word sequence of the sentences. The entity pairs act as the output layer through two fully connected networks from the last hidden layer. We use the last hidden layer as the relation embedding, and the second phase utilises a random forest classifier to learn relation types from these relation embeddings. As the LSTM networks encode semantic information from the original sentences and their entity pairs into relation embeddings, they can be efficiently classified by supervised learning. In addition, the relation embedding phase reduces the amount of language features. Therefore, less labelled training data is required in the classification phase. We conduct experiments to show that our whole framework achieves higher accuracy compared to the state-of-the-art when there are insufficient high-quality labelled data. Our contributions can be summarised as follows.

- The proposed relation extraction framework is beneficial in many real applications, where a large amount of text data is available but with limited resources to label the data.
- The proposed framework can be easily adapted to the existing neural relation extraction methods, and allows these models to be used in our framework.
- The framework outperforms the state-of-the-art models when only a small amount of labelled data is available.

II. Related Work

A. Distantly Supervised Relation Extraction

Early works such as Mintz [4] states supervised learning models for relation extraction require a large amount of labelled training data. It is tedious, time-consuming and error-prone to have such volume of training data labelled manually. To address this problem, distant supervision is proposed to automatically label data by heuristically aligning text to a known knowledge base (KB) such as DBPedia [1].

However, there is a high probability of obtaining wrong labels from noisy knowledge bases. For instance, Riedel et al. discover that a pair of entities in a sentence does not always

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

derive a relation presenting in an external knowledge base [9]. They address the problem by constructing an undirected graphical model with the "expressed-at-least-one" restriction. Hoffmann et al. further propose a joint conditional extraction model for ruling out overlapping relations from Riedel's method [5].

Alternatively, Vashishth et al. propose a graph convolution networks (GCN) based method, called RESIDE, for relation extraction [10]. RESIDE utilise Bi-GRU neural networks and a GCN for syntactic sentence encoding. The method also makes use of side information such as entity types from the knowledge base for improving the accuracy of relation extraction.

Noisy training data also presents as a significant issue in our relation extraction task. In this paper, we perform some experiments based on the NYT dataset published by Riedel et al. [9], and take advantages of the model proposed by Vashishth et al. [10], where the attention mechanism and embedding are applied in the training process.

B. Neural Relation Extraction

Recently, researchers find that deep neural networks can learn underlying features automatically. To utilise neural network methods, Zeng et al. propose a CNN model to capture relevant lexical features on the sentence level automatically [11]. The method is then improved by a piecewise convolutional neural network (PCNN) that deploys a one-sentence-selection strategy [6]. Based on PCNN, Lin et al. introduce an attention mechanism to select informative sentences, where attention is a mechanism for capturing global dependencies between a set of input vectors and their output [7].

Another well-known deep neural network is the recurrent neural network (RNN), which is widely used for training sequential data. One example is a bidirectional long short-term memory (BiLSTM) model by Zhou et al. [12] that outperforms other feature-based models on the SemEval 2010 dataset [13]. Similarly, Yang et al. propose an RNN model integrating with a multi-level attention mechanism, which shows significant improvements using the NYT dataset [14].

Recently, a Transformer model is introduced to compete with RNNs and has gained success in the task of neural machine translation tasks [15]. Researchers also take advantage of the model for relation extraction. For example, Verga et al. modified the Transformer encoder through synthesising convolutions and self-attention [16]. The model is able to extract relations from long bio-medical texts.

In our work, we make use of BiLSTM neural networks, which is similar to the model proposed by Zhou et al. in the relation embedding training phase. Rather than applying a neural network for learning the relation types directly, our model adopts the neural network for training a set of relation embeddings in an unsupervised manner. In this way, it is effective in compressing the vector space. As a result, less labelled training data is required in the relation classification phase.

C. Other Relation Extraction Models

Han et al. incorporate the hierarchical information of relations for distantly supervised relation extraction [17]. They firstly generate a hierarchical structure based on the relation types from an external knowledge graph such as Freebase [18]. In the training step, they apply an attention mechanism on each layer. Open-domain datasets such as NYT usually contain more high-level relations than base-level relations [17]. Hence, their hierarchical structure can perform a coarse-grained selection for entity pairs at top levels, as more training samples enhance its robustness, and perform fine-grained at low levels.

Alternatively, Sun et al. propose a model that jointly learn entities and relations to result in less error propagation and data inefficiency [19]. They treat the entity extraction process as a sequence labelling task, and the relation extraction process as a classification task. The two tasks share the same parameters. Each of them is assigned to a distinct local objective function. The final goal is to minimise the sum of these objective functions with a global loss function for fine tuning.

Compared to our work, none of these works considers dependency parsing in their models, which enhances the performance of the model. Different from Sun's model that based on BiLSTM for encoding the word sequence, we employ an attention layer that aggregates the hidden outputs into a context vector. When considering the attention layer, our model is similar to Han's work. However, we focus on reducing the amount of required labelled training data, where all these models need a full, labelled training set to achieve high accuracy.

III. The Approach

We formulate the task of relation extraction as follows. Given a set of sentences $S = \{s_1, s_2, ..., s_n\}$ where $s_i = [w_1, w_2, ...,]$ denotes a sentence consisting of a series of words w_j. For each sentence s, (e_h, e_t) denotes two entity mentions, where e_h is the head entity, and e_t is the tail entity. The target is to determine $r(e_h, e_t)$, the relation between the two entity mentions e_h and e_t.

A. Architecture

Our proposed framework consists of two phases, as shown in Fig. 1. The first phase is the unsupervised relation embedding phase. It consists of a neural network that learns relation embedding in an unsupervised manner. It takes a sequence of words and the shortest dependency path tokens as input, and the two entity mentions as output. The procedure is considered as unsupervised learning because the target relation type is not required. The last hidden layer of this neural network produces a relation embedding of the entity mentions. The relation embedding is then passed to the second phase for classification.

The second phase performs relation classification. This phase employs a supervised classifier by taking the relation embeddings as input and predicting their corresponding relation types. In the framework, we utilise a random forest

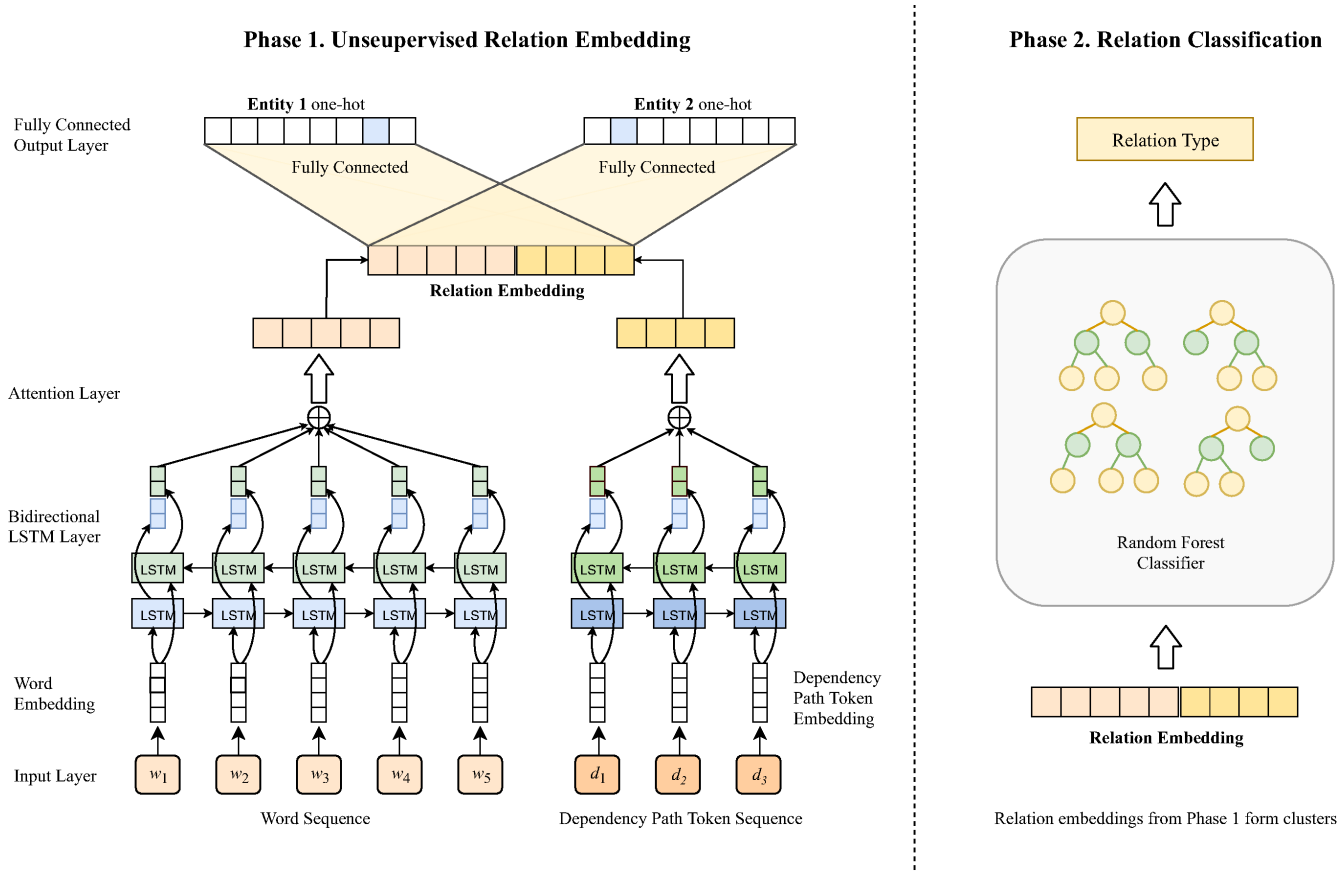

Fig. 1. An overview of our framework. The framework consists of two phases: the unsupervised relation embedding model and relation classification. The relation embeddings constructed in Phase 1 is used as the input of the random forest classifier in Phase 2 to predict the relation types.

for classification. Because the BiLSTM with an attention layer compresses the vector space in the previous phase, the classifier does not require a large number of labelled training samples to achieve reasonable performance compared to other neural relation extraction models.

B. The Unsupervised Relation Embedding Phase

We introduce an attention-based BiLSTM network to construct relation embeddings for each entity pair. The network is built up in 4 layers vertically. The bottom layers under the attention layer are divided into two components, namely the word tokens and dependency path tokens. These two components are identical in structure but different in size. The attention layer creates word embeddings and dependency path embeddings according to these two components. The last hidden layer concatenates these embeddings into relation embedding vectors. Two fully connected neural networks are then attached to the relation embedding, and their output refers to the two entity mentions respectively.

1) Input Layer: There are two types of inputs in the input layer, word tokens and dependency path tokens. A word token w_i is a word id from the input sentence s. Zhou et al. [12] take only word tokens as input in their model for relation classification. However, in a more complicated case, a sentence may contain multiple key entities with different relation types. For example, consider the following sentence from the NYT dataset:

"***Tom's brother*** *lived near* ***Phoenix****, so they went to* ***Arizona*** *last summer to look for a piece of rural property .*"

The relation type $r(Tom's\_brother, Phoenix)$ is different from the relation type $r(Phoenix, Arizona)$. Zhou's model does not consider the difference between entity pairs. Their model will produce the same relation type for both of them, as it considers they have the same features.

To address this problem, we introduce the shortest dependency path token sequence $\{d_1, d_2, ...\}$ in our model. We first perform dependency parsing using Stanford CoreNLP [20], followed by discovering the shortest dependency path p from entity e_h to entity e_t. All the tokens along p form a dependency path sequence, which is $p = \{d_1, d_2, ...\}$. Consider the sentence from the example above, the shortest dependency path from $Tom's\_brother$ to $Phoenix$ can be described as

$$e_h \xrightarrow{nsubj} lived \xrightarrow{nmod:near} e_t$$

where $e_h = Tom's\_brother$ and $e_t = Phoenix$.

Both the word tokens and the shortest dependency path tokens are converted into their vector embeddings before

passed into the BiLSTM layer. More precisely, both word embeddings and dependency path token embeddings are also trained during the training process.

2) Bidirectional LSTM Layer: LSTM is a variant of RNN proposed by Hochreiter et al. to overcome the gradient vanishing problem. In this layer, encoding future and past contexts for the word sequence and the dependency path sequence are beneficial for relation embedding. Therefore, we adopt BiLSTM for leveraging information from both of the LSTM directions. The word sequence and the dependency path sequence are trained separately in two BiLSTM units. In details, each BiLSTM unit contains two LSTM units that encode the input sequence from different directions. Let $\overrightarrow{h_i} \in \mathbb{R}^D$ denote the output of the forward LSTM unit at token w_i; and $\overleftarrow{h_i} \in \mathbb{R}^D$ denote the output of backward LSTM unit at token w_i, where D is the dimensionality of the LSTM units. The hidden output of token w_i is the concatenation of $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$. A formal representation is

$$h_i = [\overrightarrow{h_i} \oplus \overleftarrow{h_i}] \tag{1}$$

where $\oplus$ is the concatenation operation of two vectors. All these hidden outputs are then grouped together to form a matrix $H = \{h_1, h_2, ..., h_T\}$, where T is the maximum sequence length. As it shows in Fig. 1, a matrix H_w for word tokens and a matrix H_p for the shortest dependency path tokens are created separately and passed into the attention layer.

3) Attention Layer: The attention mechanism allows us to attend the LSTM output at different steps rather than considering only the last state. It gains remarkable improvements in tasks such as machine translation [21]. In this section, we adopt an attention layer to encode a relation embedding based on the LSTM output sequence. Because the attention layer in Luong's model is applied to a sequence-to-sequence model, which is different from our task, we modify the attention calculation as follows.

Let $H = \{h_1, h_2, ..., h_T\}$ denote the matrix generated by the LSTM layer that each $h_i \in D$ is a output vector of the LSTM at time step i. A weight vector α_i is calculated by

$$\begin{aligned} v_i &= tanh(h_i) \\ \alpha_i &= softmax(\omega_i \cdot v_i) \end{aligned} \tag{2}$$

where $\omega = \{\omega_0, \omega_1, ..., \omega_T\}$ is a trainable coefficient.

The final output of the attention layer is $c \in \mathbb{R}^T$, which is named a context vector by Luong et al. [21] and is computed in our model as follows:

$$c_i = tanh\Big(\sum_{j=1}^{D} \alpha_i h_{ij}\Big) \tag{3}$$

As shown in Fig. 1, attentions are separately computed for the word tokens and the shortest dependency path tokens, and two context vectors c_w, c_p are generated. The final relation embedding is the concatenation of these two vectors, which is $h = [c_w \oplus c_p]$.

4) Output Layer: Different from all other related works that predict the relation types at the output layer, the output layer in Phase 1 consists of two fully connected neural networks that take the relation embedding vector h as input and the two entities as outputs. We design this structure based on the following experiences.

In open-domain texts, it is noted that the same entity pairs have a high probability of having the same relation type, even if they appear in different sentences. For example, the relation type between "Phoenix" and "Arizona" is likely to be "/location/location/contains" even they appear in different sentences.

In addition, if two entity pairs (e_{h1}, e_{t1}) and (e_{h2}, e_{t2}) have similar input word tokens and dependency path tokens, they are likely to have similar relation types. For example, the following sentences contain similar word tokens and their relation types are both "/person/location/place_of_birth".

1) Mets catcher **Paul_Lo_Duca**$_{e_h}$ was born in **Brooklyn**$_{e_t}$, but he moved to Arizona when he was 2.
2) **Fausto_Vitello**$_{e_h}$ was born in **Buenos_Aires**$_{e_t}$ on Aug. 7, 1946, and came to the United States with his family as a boy.

Because the tokens from the input layer determine the attention output h, and h further determines the entity pairs, it could be trained to encode the hidden semantic information between the entity pair. As a result, h is used as a relation embedding. The training process of Phase 1 is unsupervised as the neural network does not require any relation types labelling.

In the output layer, we pass h through two fully connected neural networks with softmax activation functions, which can be formulated as followings.

$$\begin{aligned} p(e_h|h) &= softmax(W_h h + b_h) \\ p(e_t|h) &= softmax(W_t h + b_t) \\ \hat{e_h} &= \arg\max_{e_h} p(e_h|h) \\ \hat{e_t} &= \arg\max_{e_t} p(e_t|h) \end{aligned} \tag{4}$$

where W_h, W_t, b_h and b_t refers to the weights and biases of the two fully connected networks.

The loss function is a combination of 2 negative log-likelihoods represented as follows.

$$\begin{aligned} J(\theta) &= J_1(\theta) + J_2(\theta) + \lambda\|\theta\|^2 \\ J_1(\theta) &= -\frac{1}{m}\sum_{i=1}^{m} e_{1i} \log(p(e_{1i})) \\ J_2(\theta) &= -\frac{1}{m}\sum_{i=1}^{m} e_{2i} \log(p(e_{2i})) \end{aligned} \tag{5}$$

where $e_{1i}, e_{2i} \in \{0, 1\}^m$ are the ground truth of the entity pairs in one-hot representation and $p(e_{1i}), p(e_{2i}) \in \mathbb{R}^m$ are the estimated probability for the entity pairs across each class by softmax; m is the number of target classes (the total number of possible entities); and λ is the hyperparameter for L2 regularisation to alleviate overfitting.

C. The Relation Classification Phase

In the relation classification phase, we use the relation embedding h from Phase 1 as input, and performs supervised learning to learn the relation types. In our approach, we utilise a random forest for the relation type classification task. Random forest is a type of ensemble learning that preforms prediction based on a number of decision tree classifiers. Each decision tree outputs a class prediction, and the class with the most votes becomes the final prediction. In our approach, we adopt 50 trees as we find the accuracy does not change much when the number of trees is greater than 50. For each tree, we acquire Gini impurity as the impurity measurement for each tree node.

Before training the random forest, we train the unsupervised relation embedding module using all the samples from the training datasets S without their relation types. Then we take a small proportion of sentences $S_l \subset S$ labelled with relation types R_l, pass them through the unsupervised relation embedding phase to get a set of relation embeddings h_i. Each relation embedding h_i has a corresponding relation type $r_i \in R_l$. The random forest takes h_i as input and r_i as output for training.

In the relation classification phase, the size of S_l does not need to be big because the relation embedding compresses semantic information in a less sparse vector space (i.e., h is a much smaller vector compared to the vocabulary size), and it is sufficient with a reasonably good performance. In summary, training samples with similar relation types are closer in high dimensional spaces; hence, less training samples are required to train a classifier.

IV. Experiments

A. Datasets

We evaluate our model on two datasets, namely **SemEval 2010** [13] and New York Time (**NYT**) [9]. **SemEval 2010** is a dataset focusing on multiway classification of semantic relations between pairs of nominals at the sentence level. The dataset contains 19 relation types, and it is divided into a training set with 8,000 samples and a test set with 2,717 samples. Each sample consists of a sentences s, a pair of entities (e_h, e_t) and the relation type r. **NYT** is a large dataset and is widely used in recent works [6], [7], [17]. It is labelled with 53 relation types, including the NA type that indicates no relations between the entity pair. The full dataset is divided into a training set with 522,611 samples and a testing set with 172,448 samples. Similar to SemEval 2010, each sample consists of s, (e_h, e_t) and r.

B. Experiment Settings

Before each experiment, we use the whole dataset without relation types for pretraining each dataset using the unsupervised relation embedding phase. The parameter settings of the unsupervised relation embedding phase are shown in TABLE I.

Then in each experiment, we only use a small portion of the labelled training data to demonstrate the performance of all these approaches with insufficient labelled data. The setups of the two datasets are described as follows.

TABLE I
THE PARAMETER SETTINGS

Parameter	Value
Learning rate	0.0001
Embedding size	150
Max word length	80
Max dep length	10
Word LSTM hidden size	150
Wep LSTM hidden size	100
LSTM dropout	0.6
EMB dropout	0.6
Final dropout	0.65
λ - L2 regularisation	0.00004

For **SemEval 2010**, we compare our results with BiLSTM+ATT [12] and CNN+ATT [22]. Both of the approaches are fully supervised learning. They train the whole training set and achieve $F1 > 0.82$ for the SemEval 2010 dataset. To compare the performance of the models when training data is insufficient, we randomly pick training samples from the original training set to train the models. More specifically, we introduce p as the percentage of samples selected from the original training set, and we choose $p = 1.25\%$, 2.50%, 3.75% and 5.00% in the experiments. To ensure the amount of training data is relatively balanced, we use parameter k to represent the number of training samples for each relation type. For instance in TABLE II, when $p = 2.50\%$, each relation type contains 10 training samples. The size of these sub-training sets is determined according to the proof by Beleites et al. [23]. They found that using about 25 samples per class in the training set is sufficient for determining the learning curves of machine learning models correctly. For more directed comparison, we calculate the precision, recall and F1 score in both macro and micro average schemas according to the evaluation matrix given by official SemEval 2010. All these all the accuracy metric values are measured on the SemEval 2010 testing dataset in (2*9+1)-way, which takes relation directions into account.

For **NYT**, we compare our results with the implementation by Lin et al. [7], namely CNN+ONE, CNN+ATT, PCNN+ONE and PCNN+ATT. CNN stands for convolutional neural network, and PCNN stands for piecewise convolutional neural network. ONE refers to "at-least-one multi-instance learning", and ATT refers to the "sentence-level attention" mechanism. Similar to SemEval 2010, we use p to denote the ratio of training samples selected from the original training dataset. We run our experiments with $p = 5\%$, 10% and 25%, and plot their precision-recall curves. For fair comparisons, we create new training sets by retaining the first p training samples from the original training set. For instance, when $p = 10\%$, only the first 10% training samples from the NYT training set are taken to training the models. We follow the figures plotted by Lin et al., where precision-recall curves are sketched for each model. The NA type is not counted in the precision and recall calculation.

In terms of hardware, we run our experiments on a CentOS Linux 7 virtual machine with CPUs at 2.1 GHz, 46 GB RAM and one TESLA-V100 32GB GPU. Each experiment is executed in a virtual machine independent of others.

C. Experiment Performance Comparison

We report the SemEval 2010 dataset result in TABLE II and the NYT result in Fig. 2. Overall, for both datasets, our model outperforms the existing models when the amount of labelled data is small (e.g., $k = 5$ or 10). From TABLE II and Fig. 2, we can observe the following.

(1) As the number of labelled instances increases, the performance of all models improves. This demonstrates that all the models are able to learn lexical and syntactic information automatically. Moreover, increasing in vocabulary size (i.e., #vocab) results in more diverse input resource that helps a model to learn more features. Hence, all models achieve higher F1 scores.

(2) In the SemEval 2010 dataset, our model performs better than BiLSTM+ATT and CNN+ATT when k is small. The same property holds on the NYT dataset when the ratio is small. This demonstrates that unsupervised relation embeddings help to compress text semantic information into a smaller vector space. Although all competing models have applied word embeddings to their inputs, the vector space is too sparse if only a small amount of labelled data is provided. The relation embeddings help by compressing and condensing the lexical and syntactic features in the vector space, so a typical classifier such as a random forest can effectively learn relation types from a small amount of labelled training data.

(3) In the SemEval 2010 dataset, the CNN+ATT model achieves greater micro F1-score than our model when $p = 5.0\%$. However, its macro F1-score is lower, and this is because the model tends to be biased to one of the relation type. When k is getting smaller, our model outperforms all the others.

(4) Finally, the performance of our model is more stable, as both micro and macro F1-scores are closer to each other. This can also be observed from the NYT experiment results in Fig. 2, in which precisions of the blue curves decrease relatively slower when the recalls increase.

D. Ablation Study

In this section, we discuss the effect of various components of our model by removing layers or components from the model. We run the experiments on the NYT dataset with 10% of labelled data and plot the corresponding precision-recall curves in Fig. 3. In Fig. 3, no_att refers to the model with the attention mechanism removed; $no\_dropout$ refers to the model with the dropout rate = 0; and no_dep refers to the model with the dependency path component removed.

The result shows that the shortest dependency path is one of the most crucial parts of the relation embedding model, since its curve has much lower precision in the entire recall range when compared with the others. This further demonstrates that key tokens which are missed from the word sequences can be enhanced by the shortest dependency path between the entity pairs.

The result also shows that the attention mechanism contributes to the accuracy by combining useful instances and discarding useless ones. Moreover, the dropout technique prevents overfitting of the model and hence improves the accuracy.

E. Case Study

In this section, we select some sample testing cases from the NYT dataset and visualise them in Fig. 4 and Table 5 to demonstrate the effectiveness of our proposed framework. As we train relation embeddings based on sentence s and entity pairs (e_h, e_t), each relation embedding $h_{(s,e_h,e_t)}$ is a high dimensional vector and hardly to be visualised. Hence, we perform T-SNE on the selected $h_{(s,e_h,e_t)}$ samples to reduce the dimensions to 2, and plot them on a 2D plane in Fig. 4. We plot 1,766 instances from the NYT dataset, and colour them according to their true relation type.

Generally, embedding techniques aim to create a vector presentation in lower vector space to represent an item, where semantically similar vectors get closer during the training process. From the scatter plot, we can observe some points form small clusters. To further visualise them, we select four samples with labels s_1, s_2, s_3 and s_4 on the graph, and show their original contents (s, e_h, e_t, r) in Table 5.

One can observe that s_1 and s_2 fall into the same cluster in Fig. 4. From the original sentences, the word embedding of *david_ben-gurion* and *kim_jong-il* carry information that indicates these two entities referring to people. Similarly, the word embedding of *israeli* and *north_korea* are close to other country entities. Additional common features such as the word embeddings of "leader" help the BiLSTM and the attention layer capturing their relationships and push them closer in the vector space. s_3 and s_4 belong to the relation type */people/person/place_lived*. The dependency path embedding contributes more than the word embedding in this case, as both of these examples share a common dependency path $e_h \xrightarrow{nmod:of} e_t$ between the entities e_h and e_t.

V. Conclusions

Most relation extraction models require a large amount of labelled training data, which can be time-consuming and expensive to produce. Recently, neural relation extraction models that learn underlying features automatically have been proposed to address this problem. Compared with neural relation extraction models, our approach learns relation embeddings in an unsupervised manner in the first phase, and then employs a random forest classifier for relation classification in the second phase. The relation embedding phase encodes semantic text information between each sentence and the corresponding entity pair. These relation embeddings allow the random forest classifier to make predictions, and it outperforms the previous works on the SemEval 2010 and NYT dataset when the amount of labelled training data is insufficient. In the relation embedding phase, we show that the dependency path token

TABLE II
MICRO AND MACRO AVERAGED F1 SCORES ON SEMEVAL

Model	p (Ratio)	k	#Vocab	Micro			Macro		
				Precision	Recall	F1	Precision	Recall	F1
BiLSTM + ATT	1.25%	5	870	0.096	0.113	0.104	0.089	0.111	0.087
	2.50%	10	1508	0.136	0.155	0.145	0.149	0.151	0.127
	3.75%	17	2166	0.244	0.289	0.263	0.295	0.285	0.251
	5.00%	22	2751	0.307	0.357	0.330	0.318	0.355	0.328
CNN + ATT	1.25%	5	870	0.295	0.324	0.309	0.303	0.335	0.307
	2.50%	10	1508	0.373	0.438	0.403	0.382	0.429	0.390
	3.75%	17	2166	0.451	0.525	0.485	0.440	0.519	0.466
	5.00%	22	2751	0.476	0.554	**0.512**	0.460	0.555	0.495
Our Model	1.25%	5	870	0.451	0.376	**0.410**	0.384	0.453	**0.416**
	2.50%	10	1508	0.496	0.416	**0.453**	0.424	0.493	**0.456**
	3.75%	17	2166	0.563	0.483	**0.520**	0.457	0.555	**0.501**
	5.00%	22	2751	0.557	0.466	0.508	0.465	0.566	**0.510**

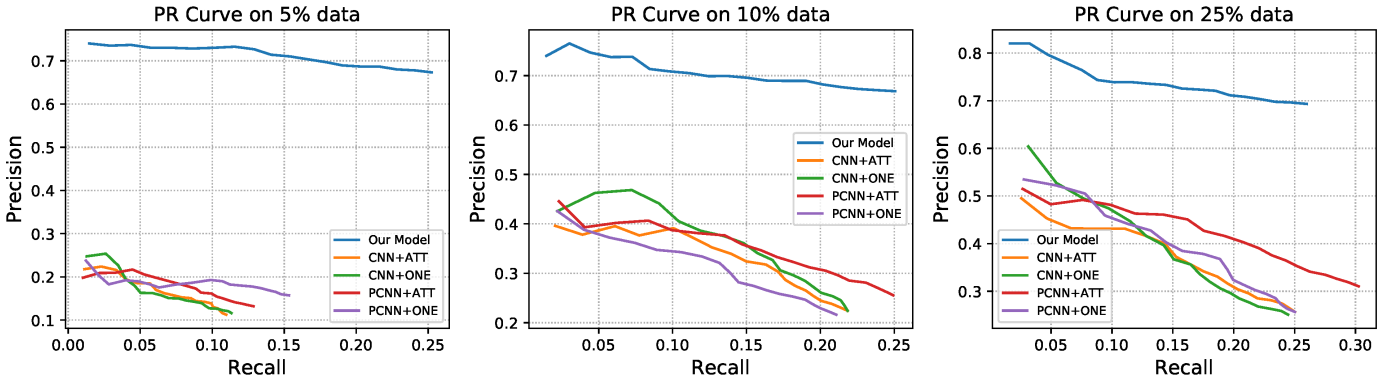

Fig. 2. Precision-recall curves for our proposed model vs various baseline models on the NYT dataset. The original training samples are eliminated in different proportions. Finally 5%, 10% and 25% of the original training set are used to train the models respectively.

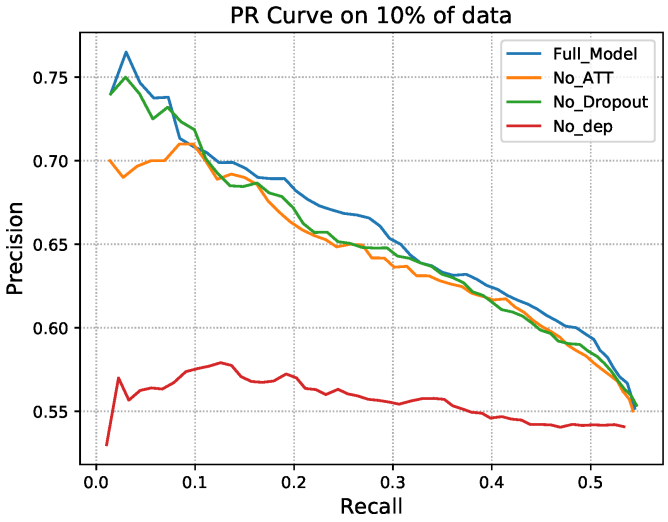

Fig. 3. Performance comparison of different ablated version of our model on the NYT dataset.

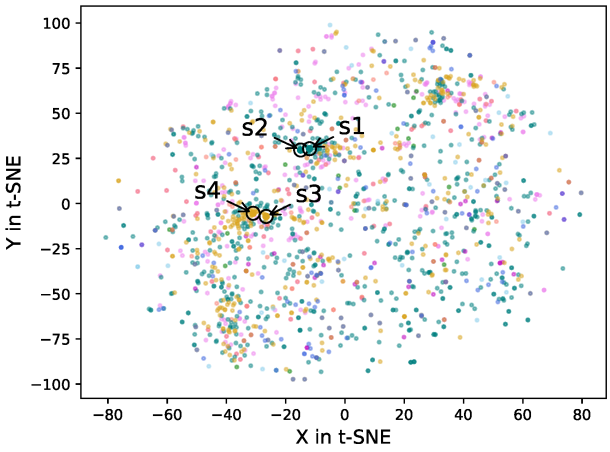

Fig. 4. T-SNE plot of some relation embeddings.

is the most significant feature. We also show that the relation embeddings naturally form small clusters and similar instances can be visualised within these clusters. Our ongoing work is to explore n-ary relations in specialised domain documents. For example, chemistry domain documents may describe relations between multiple reactants and products in chemical reactions. Furthermore, we would like to extend our framework to allow it jointly extract entities and determine their relation types.

ID	Relation	Sentence
s_1	/people /person /nationality	Mr. feldman was sent to meet quietly with **israeli** leaders , particularly **david_ben-gurion** and golda meir , about matters including arms sales , palestinian refugees , and whether israel was building a nuclear weapon.
s_2	/people /person /nationality	Persuading **north_korea** 's leader , **kim_jong-il** , to abandon his nuclear ambitions has seemed an increasingly hopeless exercise since october , when north_korea tested a nuclear device and declared itself a nuclear state.
s_3	/people /person /place_lived	The foreign relations committee approved the resolution by a vote of 12 to 9, with a republican senator , **chuck_hagel** of **nebraska**, joining 11 democrats in supporting it.
s_4	/people /person /place_lived	Senator **mitch_mcconnell** of **kentucky**, the minority leader, this week promised to vigorously fight democratic efforts to bring up a resolution challenging the buildup even as he conceded the political damage the war has inflicted.

Fig. 5. The original text of the four sample sentences marked in the T-SNE plot (Fig. 4). The entity pairs in each sentence is emphasised in bold with their relation types listed.

REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.

[2] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *Proceedings of the 21st international conference on world wide web*. ACM, 2012, pp. 1063–1064.

[3] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1083–1106, 2003.

[4] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "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*. Association for Computational Linguistics, 2009, pp. 1003–1011.

[5] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "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*. Association for Computational Linguistics, 2011, pp. 541–550.

[6] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1753–1762. [Online]. Available: https://www.aclweb.org/anthology/D15-1203

[7] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 2124–2133. [Online]. Available: https://www.aclweb.org/anthology/P16-1200

[8] D. Yang, S. Wang, and Z. Li, "Ensemble neural relation extraction with adaptive boosting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, ser. IJCAI'18. AAAI Press, 2018, p. 4532–4538.

[9] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 148–163.

[10] S. Vashishth, R. Joshi, S. S. Prayaga, C. Bhattacharyya, and P. Talukdar, "Reside: Improving distantly-supervised neural relation extraction using side information," *arXiv preprint arXiv:1812.04361*, 2018.

[11] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2335–2344. [Online]. Available: https://www.aclweb.org/anthology/C14-1220

[12] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2. Association for Computational Linguistics, aug 2016, pp. 207–212.

[13] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. Association for Computational Linguistics, 2009, pp. 94–99.

[14] L. Yang, T. L. J. Ng, C. Mooney, and R. Dong, "Multi-level attention-based neural networks for distant supervised relation extraction," in *AICS*, 2017, pp. 206–218.

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

[16] P. Verga, E. Strubell, and A. McCallum, "Simultaneously self-attending to all mentions for full-abstract biological relation extraction," *arXiv preprint arXiv:1802.10569*, 2018.

[17] X. Han, P. Yu, Z. Liu, M. Sun, and P. Li, "Hierarchical relation extraction with coarse-to-fine grained attention," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2236–2245. [Online]. Available: https://www.aclweb.org/anthology/D18-1247

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

[19] C. Sun, Y. Wu, M. Lan, S. Sun, W. Wang, K.-C. Lee, and K. Wu, "Extracting entities and relations with joint minimum risk training," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2256–2265.

[20] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60.

[21] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1412–1421. [Online]. Available: https://www.aclweb.org/anthology/D15-1166

[22] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 39–48.

[23] C. Beleites, U. Neugebauer, T. Bocklitz, C. Krafft, and J. Popp, "Sample size planning for classification models," *Analytica chimica acta*, vol. 760, pp. 25–33, 2013.