Unsupervised Pre-training With Sequence Reconstruction Loss for Deep Relation Extraction Models

Zhuang Li[1], **Lizhen Qu**[1,2], **Qiongkai Xu**[1,2], **Mark Johnson**[3]
[1] DATA61, Australia
[2] The Australian National University
[3] Macquarie University
lizhuang144@gmail.com,
{Lizhen.Qu,Qiongkai.Xu}@data61.csiro.au
mark.johnson@mq.edu.au

Abstract

Relation extraction models based on deep learning have been attracting a lot of attention recently. Little research is carried out to reduce their need of labeled training data. In this work, we propose an unsupervised pre-training method based on the sequence-to-sequence model for deep relation extraction models. The pre-trained models need only half or even less training data to achieve equivalent performance as the same models without pre-training.

1 Introduction

Relation extraction (RE) is the task of detecting and categorizing semantic relations between named entities mentioned in a text corpus. This is important for a wide variety of practical applications. For example, tourism planning bodies are interested in mining social media such as tweets to identifying which restaurants tourists eat in and which hotels those same tourists stay in.

RE has been intensively studied for several years (Chan and Roth, 2011; Chan and Roth, 2010). Recently, RE models based on deep neural networks (DNN) have achieved better performance than conventional RE models that rely on handcrafted features (Xu et al., 2015). However, these DNN models require a large amount of annotated training data, which is difficult and expensive to obtain. The data problem is not completely solved by relying on methods such as large external knowledge bases and distant supervision because i) models employing only large knowledge bases often still perform poorly on RE (Angeli et al., 2014); ii) the external knowledge bases are incomplete; and iii) many important applications lack the relevant domain specific knowledge bases. This paper asks the question: can we use unlabeled data to help training DNN RE models? Although unsupervised pre-training is known to be effective for training deep neural networks, it remains unclear how to apply it to the recently proposed DNN RE models. The main advantage of deep models (compared to the shallow counterparts) is that they automatically learn distributed representations of the relevant components of the model (e.g., words, entities, relations, etc.). If we can encode rich syntactic-semantic patterns of relation expressions into the automatically learned, low-dimensional representations, and require these representations to be similar if they play a similar role using only unlabeled data, it should be possible for a DNN RE system to achieve a high level of generalization from only small amount of labeled data.

In a relational expression, the named entities and words around it provide useful context information. For example, in the sentence *"By 1982 the BL Cars Ltd division renamed itself Austin Rover Group shortly before the launch of the Maestro." renamed itself* is used much less often than an expression such as *was founded in* to indicate the relation *org:foundedIn*. Thus it is likely that *was founded in* will be found in the training set, even if *renamed itself* does not appear in the training set. Despite this, the co-occurrence of *1982* and *Austin Rover Group*, as well as keywords such as *by*, form a context that is similar to that of *Austin Rover Group was founded in 1982*. If such shared contextual information can require the similarity of the representations of these expressions, a classifier can easily infer that *renamed itself* is likely to indicate *org:foundedIn*. Inspired by observations such as these, we seek methods that exploit context information composed of words and named entities to learn representations of expressions, such that semantically similar expressions tend to have similar representations.

In this paper, we propose a pre-training method that generalizes well-known sequence-to-

Zhuang Li, Lizhen Qu, Qiongkai Xu and Mark Johnson. 2016. Unsupervised Pre-training With Seq2Seq Reconstruction Loss for Deep Relation Extraction Models. In *Proceedings of Australasian Language Technology Association Workshop*, pages 54–64.

sequence model (Dai and Le, 2015) for deep RE models. This approach formalizes unsupervised pre-training as minimizing reconstruction errors of input sequences. For a given DNN RE model, our approach first pre-trains it on a large, unlabeled, domain-general corpus, and then fine-tunes it on target corpora. Our experiments show that, especially when the size of the labeled training data is small, the deep relation extraction models pre-trained with our unsupervised pre-training method using half or even a quarter of the labeled data are able to achieve similar performance as the models without pre-training. Our unsupervised approach does not need domain-specific corpora for pre-training; in fact, they work well with 13,000 sentences randomly sampled from Wikipedia.

2 Related Work

Recent advance of relation extraction demonstrates the power of deep learning by showing that the deep models significantly outperform the conventional approaches (Jiang and Zhai, 2007; Chan and Roth, 2010; Chan and Roth, 2011) on the ACE relation extraction datasets. Except for the FCM model (Yu et al., 2015), at the core of almost all deep RE models are variants of convolutional neural networks (CNN) (Zeng et al., 2014; Nguyen and Grishman, 2015; Wang et al., 2016; Miwa and Bansal, 2016), recurrent neural networks (RNN) (Zhang et al., 2015; Socher et al., 2012; Ebrahimi and Dou, 2015; Lin et al., 2016), or both of them (Liu et al., 2015; Cai et al., 2016).

Several RE systems (Chen et al., 2006a; GuoDong et al., 2009; Li et al., 2010; LongHua and Qiaoming, 2008; Chen et al., 2006b; Kim and Lee, 2012) are built upon the semi-supervised learning algorithm *label propagation* to exploit the use of unlabeled data. This family of algorithms start with building a similarity graph between each pair of relation mentions, and propagate relation labels from labeled ones to unlabeled ones. However, deep RE models require substantial change in order to use these algorithms, while our methods just need to replace the training criterion during pre-training, which is easy-to-implement by using a standard deep learning toolkit. It is also too expensive to involve all unlabeled data in both training and prediction processes for each target dataset. In contrast, our pre-training algorithms are performed only once on a general corpus and the resulted models are fine-tuned only on target corpora.

There is also ample of work exploring the idea of distant supervision for knowledge base completion (Riedel et al., 2013; Weston et al., 2013; Yang et al., 2014; Bordes et al., 2013) in order to avoid the use of manually labeled data. Although some of these models include a relation extraction component (Surdeanu et al., 2012; Angeli et al., 2014; Toutanova et al., 2015), the outputs of their systems are whether a relation holds between entities rather than entity mentions. In contrast, we aim to classify relation mentions no matter if a target relation exists in a knowledge base or not.

There have also been other efforts towards minimizing the use of labeled data. In (Sun, 2009), they proposed a bootstrapping approach to extract textual patterns for training a SVM-based relation extraction system. In (Chan and Roth, 2011), they show that supervised models equipped with syntactico-semantic features are capable of classifying relation mentions with a few labeled data. However, both work are customized for supervised models with handcrafted features and relations between nominals. In other lines of research, active learning (Fu and Grishman, 2013; Sun and Grishman, 2012) and domain adaptation (Nguyen and Grishman, 2014) pursued to select high quality training examples for training relation extraction models. Jiang (2009) leverages the knowledge of known relations to predict new relations in a weakly supervised setting. These approaches have different problem settings than ours, which focus on the use of unlabeled data.

Since 2006, various pre-training techniques are proposed to make the training of deep neural networks practical (Hinton and Salakhutdinov, 2006; Dahl et al., 2010; Bengio, 2009). They are not universally applicable for all problems and most of them focus on computer vision problems. To the best of our knowledge, we are the first to explore the use of pre-training for deep RE models.

3 Relation Extraction Models

Suppose we are given a relation mention, which is a pair of named entity mentions (m_h, m_t) together with its relation expression in a sentence S. Each mention m is disambiguated into an entity e. Let $x \in \mathcal{X}$ denote a relation mention, where $\mathcal{X}$ is the space of all relation mentions, RE models assign a binary relation $y \in \mathcal{Y}$ to x, where $\mathcal{Y}$ is a finite set

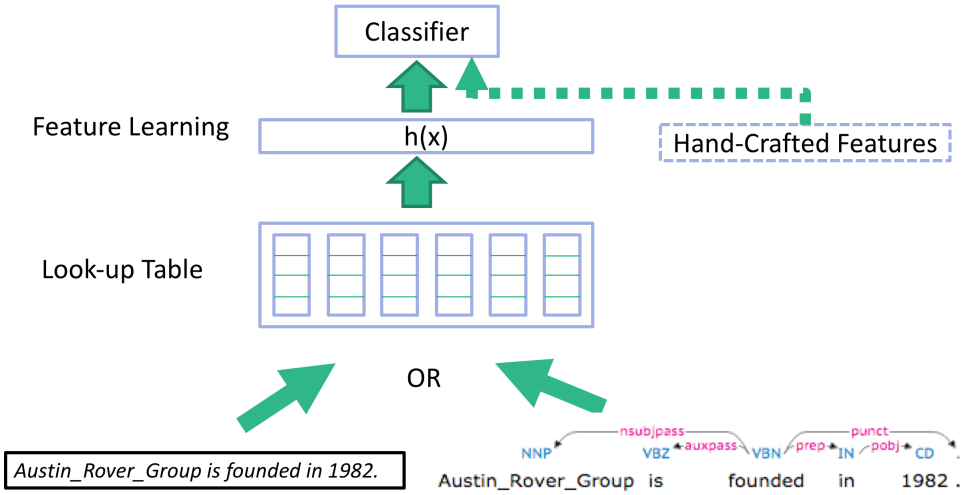


Figure 1: General Architecture for deep RE models.

of all possible relations. As a result, an RE model is a function $g : \mathcal{X} \rightarrow \mathcal{Y}$.

Given a training set $(x_1, y_1), ..., (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, we can directly learn an RE model by minimizing a supervised loss function $L_s : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. In absence of sufficient supervised training data, we resort to a two-stage approach. In the first stage, we pre-train RE models on a dataset annotated with named entity mentions and their corresponding entities by minimizing an unsupervised loss $L_u : \mathcal{X} \rightarrow \mathbb{R}$. In the second stage, we fine-tune the pre-trained models on the labeled dataset by applying the supervised loss L_s. In our experiments, L_s is the cross-entropy loss, as a result of applying multi-class logistic regression (LR) in the supervised setting.

The deep RE models proposed recently are variants of Long Short Term Memory (LSTM) (Graves and Schmidhuber, 2005) and Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012). As representative examples we consider three recent RE models: i) bidirectional LSTM that takes words around entity mentions as input (Zhang et al., 2015), coined BiLSTM; ii) LSTM taking shortest paths in dependency trees as input, coined DepTreeLSTM; iii) CNN taking words sequences and position embeddings as input (dos Santos et al., 2015), coined PCNN.

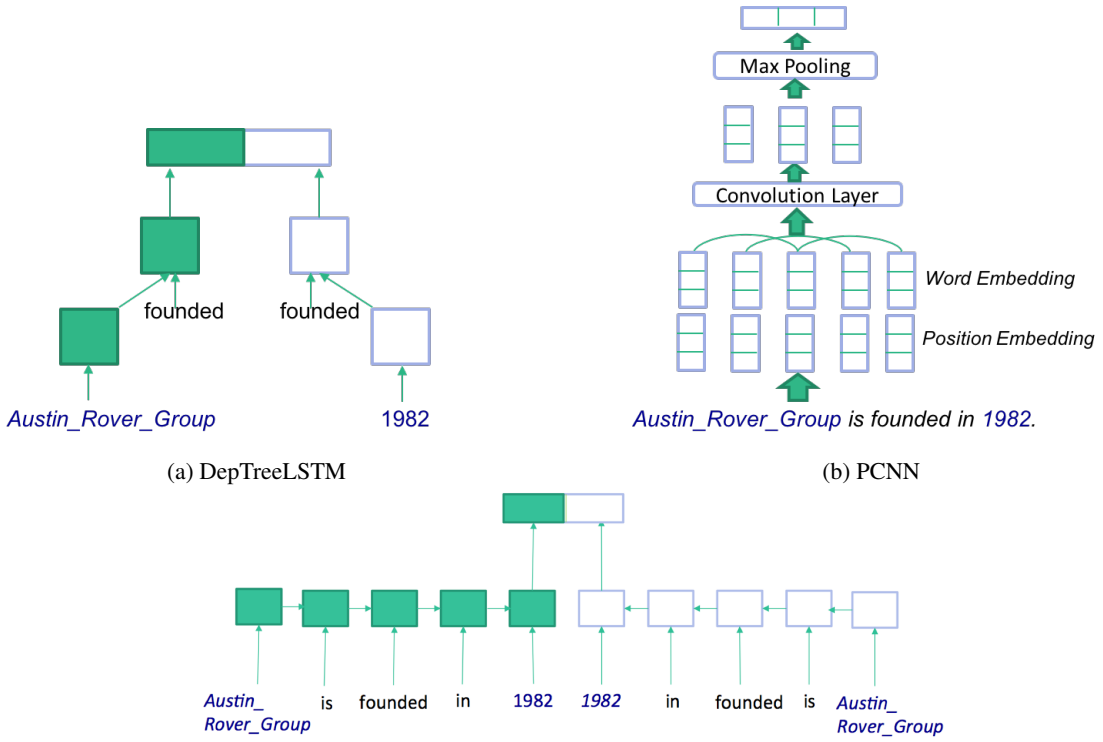
All three RE models consist of four components. As illustrated in Figure 1, as input they take either word sequences between two entity mentions or the shortest dependency path between two entity mentions. A look up table maps each input word into its word embedding. Herein we denote the word embedding of a word i by $\mathbf{e}_i \in \mathbf{R}^M$, where M is the dimension of word embeddings. All word embeddings are initialized with the ones pre-trained on a large domain-general corpus (Qu et al., 2015). As suggested in (Qu et al., 2015), we do not update these word embeddings during training to avoid overfitting. In the next step, a feature learning component projects the embeddings into a hidden representation $\mathbf{h}$. If it is in a supervised setting, both $\mathbf{h}$ and handcrafted features are taken as the input of a multi-class LR classifier for categorizing target relations. In case of unsupervised pre-training, $\mathbf{h}$ is fed into a classifier for a designated unsupervised predictive task.

The RE models based on BiLSTM and TreeLSTM are extensions of LSTM. LSTM is a recurrent neural network capable of capturing long dependencies (Graves and Schmidhuber, 2005). At the t-th time step, the LSTM layer takes the form:

$$\mathbf{u}_t, \mathbf{c}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{u}_{t-1}, \mathbf{c}_{t-1}) \quad (1)$$

where $\mathbf{x}_t$ is the input to LSTM at time step t, and $\mathbf{u}_t$ and $\mathbf{c}_t$ are the hidden states and memory states of LSTM at time step t, respectively.

BiLSTM reads an input word sequence in both directions with two separate LSTM layers. As illustrated in Figure 2c, one LSTM reads the word sequence between two entity mentions in forwards direction, while the other with shared parameters reads the same sequence in the reverse direction. As a result, they generate two hidden representations $\overrightarrow{\mathbf{h}}$ and $\overleftarrow{\mathbf{h}}$, which are further concatenated to form the input vector $\mathbf{h}$ of the classifier.



(a) DepTreeLSTM

(b) PCNN

(c) BiLSTM

Figure 2: Deep relation extraction models.

DepTreeLSTM takes as input the shortest path between two entity mention in a syntactic dependency tree. The shortest path consists of two subpaths, which starts from an entity mention and ends at their lowest common ancestor. Since both subpaths are word sequences, as shown in Figure 2a, the feature learning component is composed of two LSTM layers with shared parameters to read the two subpaths respectively. The resulted two representations are concatenated as the input of the classifier. This model can be viewed either as the model proposed in (Ebrahimi and Dou, 2015) by replacing the recursive neural networks with LSTM, or as simplifying the model proposed in (Xu et al., 2015) by removing the max-pooling layer. The max-pooling layer leads to degraded performance in our preliminary experiments.

PCNN implements the model in (dos Santos et al., 2015), which takes as input the word sequence between two entity mentions. It starts with mapping each input word to its word embedding. Each word embedding is further concatenated with its position embedding, which encodes relative distance of the word w.r.t. each entity mention. To cope with input word sequences of varying length, the embedding sequences smaller than the pre-specified maximal length are padded with the embedding of the padding token. Then a convolutional layer and a max pooling layer are applied in sequel to generate the input $\mathbf{h}$ for the classifier.

For all three models, we augment them with the handcrafted features used in the top conventional RE systems that do not rely on deep learning techniques. They lead to improved results according to our preliminary experiments. In particular, we include lexical, collocation, and dependency features proposed in (Chan and Roth, 2010). The other features used in (Chan and Roth, 2010) are dropped because the relevant information is not available in our target datasets. In addition, we implemented the POS features and the base phrase chunk features introduced in (Chan and Roth, 2011).

4 Unsupervised Pre-training

Inspired by the semi-supervised sequence-to-sequence model (Dai and Le, 2015), our unsupervised pre-training methods tackle the learning of deep RE models in two steps. First, we learn entity embeddings by using a stepwise training strategy. Second, we train the feature learning components $h(x)$ of deep RE models by using *sequence recon-*

struction loss.

4.1 Learning Entity Embeddings

Entities often provide vital information for relation extraction (Chan and Roth, 2010). Qu et al. (2015) show that the extraction of entity mentions benefits significantly from distributional similarity, thus we learn entity embeddings by using the Skip-gram model (Mikolov et al., 2013). An entity mention such as *Austin Rover Group* often spans more than one word, while the Skip-gram model works on sequences of tokens. Therefore we retokenize text by mapping each entity mention into a single token, and replace them with the IDs of the referred entities.

The domain specific RE corpora are often small. The retokenization of documents further leads to a substantial number of infrequent entity tokens. We can only obtain embeddings of poor quality for these tokens if we train them from scratch (Collobert et al., 2011). To circumvent the problems, we employ a stepwise strategy. First, we initialize all word embeddings with the pre-trained ones on a large corpus (Qu et al., 2015). Second, we initialize each entity embedding by averaging the embeddings of all the words ever occurred in its mentions, following (Socher et al., 2013). Third, we update only entity embeddings by using the Skip-Gram model. This allows us to update them with an aggressive learning rate since we expect a large change of these embeddings. And we keep the pre-trained word embeddings intact to preserve the knowledge of distributional similarity learned from a large general corpus, as suggested in (Qu et al., 2015). After training with the Skip-gram model, we also do not update these entity embeddings while training with the deep RE models because updating these embeddings was not shown to be useful in our preliminary experiments.

4.2 Sequence Reconstruction Loss

Given pre-trained word and entity embeddings, the randomly initialized deep RE models still suffer from poor performance if the target training datasets are too small compared to their vast number of model parameters. Inspired by Autoencoders (Vincent et al., 2010), our key idea is to obtain high quality representations by reconstructing the corresponding inputs. During the process of reconstruction, if two expressions share similar context, we expect that they end up with having similar representations.

We draw inspiration from the semi-supervised sequence-to-sequence (seq2seq) model (Dai and Le, 2015) for pre-training deep RE models. Its underlying seq2seq (Sutskever et al., 2014) model consists of an LSTM encoder and an LSTM decoder. The encoder reads a sequence of words and map them into a hidden representation. Then the decoder takes the representation as input and predicts the most likely sequence of words. The training objective is to minimize the discrepancy between the predicted sequence and the input sequence.

All of the three deep RE models presented in Sec 3 take as input word sequences, and generate a hidden representation $\mathbf{h}$ for the classifier. Our key idea of generalizing the semi-supervised seq2seq model is to reuse the feature learning component $h(x)$ as the encoder and reconstruct the input sequence in each direction by using an LSTM decoder. The change of encoder is particularly interesting for *PCNN*, which adopts a different type of model than the decoder.

Given an entity mention pair, the input of both *PCNN* and *BiLSTM* is the word sequence between both mentions and the mentions themselves. *PCNN* applies CNN to read the input sequence in both forwards and backwards directions, and results in two hidden representations $\overrightarrow{\mathbf{h}}$ and $\overleftarrow{\mathbf{h}}$ respectively. Its LSTM decoder reads each representation and reconstructs the input sequence in the corresponding direction, respectively. In the same manner, *BiLSTM* applies the two LSTM layers to read and reconstruct input sequences in both directions. Although *DepTreeLSTM* takes input from dependency trees, it follows the same way as the other two models by reconstructing two word sequences in their respective reading direction. Herein, each sequence is read from the entity mention to their lowest common ancestor.

The LSTM decoder consists of an LSTM in the form of Equation (1) and a softmax classifier. At time step t, the LSTM layer reads the previous hidden state $\mathbf{u}_{t-1}$ and the predicted word x_{t-1} at time step $t-1$, followed by generating the current hidden state $\mathbf{u}_t$. The current hidden state $\mathbf{u}_t$ is fed into the softmax classifier to predict the word x_t, where the softmax classifier is defined as:

$$P(x = j|\mathbf{u}_t) = \frac{\exp(\mathbf{e}_j^{\mathrm{T}}\mathbf{u}_t)}{\sum_{k=1}^{|\mathcal{V}|}\exp(\mathbf{e}_k^{\mathrm{T}}\mathbf{u}_t)}$$

where $\mathcal{V}$ denotes the vocabulary. When $t = 1$, the LSTM initializes the initial state as $\mathbf{u}$ and $\mathbf{c}_0 = \mathbf{0}$.

For the sake of computational efficiency, we minimize the reconstruction loss by approximating the cross-entropy loss of the softmax function by using the negative sampling technique in (Mikolov et al., 2013). As a result, at the t-th time step during decoding, we minimize

$$-\log\sigma(\mathbf{e}_{x_t}^{\mathrm{T}}\mathbf{u}_t) - \sum_{j=1}^{k}\mathbb{E}_{x_j\sim P_n(x)}\log\sigma(-\mathbf{e}_{x_j}^{\mathrm{T}}\mathbf{u}_t)$$

where x_t is the corresponding word observed in the input sequence, σ denotes the sigmoid function, and $P_n(x)$ is the noise distribution for drawing k negative samples. In our experiments, we employ uniform distribution as the noise distribution. Then the loss function L_u is the sum of the above loss over all words in input sequences.

5 Experimental Setup

5.1 Datasets and Evaluation Protocol

We use the Stanford Relation Extraction corpus (`StanfordRE`) (Angeli et al., 2014) as the target corpus for evaluation. Each entity mention is associated with a canonical name. We map each canonical name to an entity ID in two ways. If the canonical name can be found in Freebase, we replace the mention with its Freebase machine ID. Otherwise we replace the mention with the ID based on its canonical name. In addition, we filter out the relation mentions with a annotator agreement lower than 80% as well as the ones labeled as *no relation*, because they are the source of label noise based on our manual inspection. This is beyond the scope of this work. As a result, we obtain 9150 relation mentions and 40 relations in total.

Among all relation mentions in the `StanfordRE` corpus, we hold out 20% relation mentions for testing, 10% for development, and the remaining for training. In order to test the impact of the volume of training data for fine-tuning, we split the training portion of the corpus into 10 partitions based on a log scale, and created 10 successively larger training sets $S_1, S_2, ..., S_{10}$ by merging these partitions from smallest to largest. As a result, S_{i+1} is twice the size of the S_i and S_{10} is the full training set.

For pre-training, we use the FIGER corpus, which is a sample of Wikipedia annotated with millions of entity mentions (Desmet and Hoste, 2014). Because each entity mention is also linked to a canonical name, we convert each mention to an entity ID in the same way as for `StanfordRE`.

To investigate the influence of size of pre-training corpora, we create three corpora for pre-training:

(i) `StanfordWiki`: to verify if the relation mentions from the FIGER corpus, whose entity mention pairs also occur in target corpora, are most relevant during pre-training, we collect all sentences, in which there are at least one entity pair occurring also in a sentence from the `StanfordRE` corpus. Then we merge them with the `StanfordRE` to build a corpus, which contains 133,793 relation mentions in total.

(ii) `WikiRandom`: we randomly sample five non-overlapped subsets from the FIGER corpus, each of them contains similar number of relation mentions as `StanfordWiki`.

(iii) `WikiWhole`: we collect all sentences in the whole FIGER corpus, which contain at least two entity mentions. As a result, we get 1,004,831 sentences and 3,886,998 relation mentions.

In this paper, we mainly present the pre-training results of all models on `WikiRandom`, because i) they are similar to those on `StanfordWiki` and `WikiWhole`; ii) random sentence samples are easy to acquire. For the experiments on `WikiRandom`, we perform one run on each of the five random samples, report averaged micro-F1 scores over all five runs as well as their standard deviations.

5.2 Baselines

We compared pre-trained deep RE models with their randomly initialized counterparts, which differ in their input features.

Handcrafted: an LR classifier with the same handcrafted features as the deep RE models.

Avg_embed: deep RE models with handcrafted features, pre-trained word embeddings, and entity embeddings generated by averaging the embeddings of the words occurred in mentions. The model parameters of the feature learning component and the LR classifier are randomly initialized.

Random_stepwise: deep RE models with handcrafted features, pre-trained word embeddings, and entity embeddings trained by our stepwise training strategy. Their model parameters are randomly initialized in the same way as *avg_embed*.

We compare both LSTM based RE models in two different settings of pre-training: i) the LSTM in the decoder does not share parameters with the

LSTM in the feature learning component; ii) both LSTM layers share parameters.

Given small training datasets, the performance of neural network models often depend on randomly initialized parameters, thus we perform five runs with different random initialization and report the averaged micro-F1 score.

5.3 Implementation Details

In our experiments, we reuse the 200-dimensional pre-trained word embeddings based on the Skip-gram model from our prior work (Qu et al., 2015). The corresponding negative samples is 10 and the size of local context window is 5. During stepwise training, all entity embeddings are fine-tuned with a learning rate 0.001 for 50 epochs within a local context window of size 5, the number of negative samples is set to 10. For both LSTM variants, we implemented LSTM in the same way as in (Vinyals et al., 2014), the dimension of hidden units is fixed to 200. For *PCNN*, the dimension of each position embedding is 70, as in (dos Santos et al., 2015), the size of the context window is 3, and the output of the convolutional layer consists of 200 hidden units. During pre-training, the number of negative samples is set to 10. In both pre-training and fine-tuning, we adopt AdaGrad (Duchi et al., 2011) and L2 regularizer for optimization. We tune all hyperparameters on the development set. As a result, the initial learning rates ϵ of AdaGrad is 0.1 for both LSTM variants and 0.05 for *PCNN* during pre-training, and it is fixed to 0.05 during supervised training. For all models, the hyperparameter of L2 regularization is fixed to $10E^{-6}$.

6 Results and Discussions

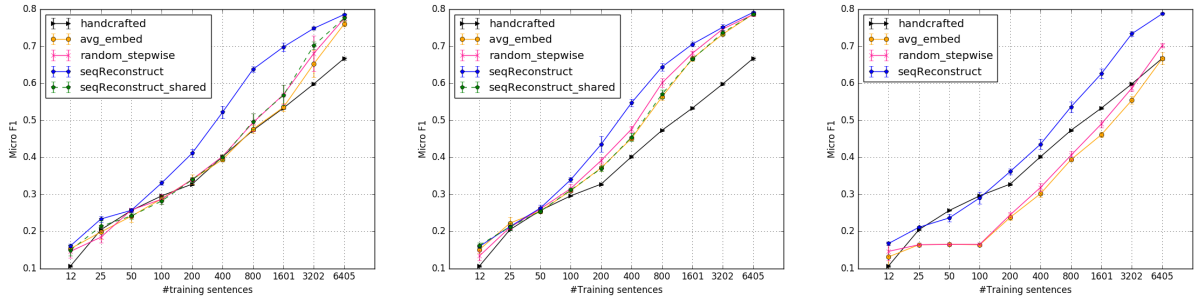
As illustrated in Figure 3, all deep RE models pre-trained with the best method outperform the baselines with a wide margin unless the full train set is used. And the performance of these pre-trained models has small variance across all five random training samples. Among all these models, pre-trained *DepTreeLSTM* is the best performing model on `StanfordRE` on average. The pre-trained *BiLSTM* achieves the largest improvement w.r.t. its randomly initialized counterpart with the entity embeddings computed by averaging word embeddings. It needs merely 800 sentences to achieve similar performance as the randomly initialized one trained on 3200 sentences.

Both LSTM based models show that it is better off not sharing the parameters of LSTM between encoders and decoders. Otherwise they achieve only similar performance as the best baselines. We also observe that the gap between both pre-trained LSTM variants and their competitors narrows as the size of the in-domain training data grows more than 1000 sentences. For them, pre-training is only useful when training data is small.

In contrast, the pre-trained *PCNN* follows a different trend by achieving the highest improvement over the random initialized one when there are more than 3200 target relation mentions for training. Without pre-training, *PCNN* performs even worse than the baseline with handcrafted features unless the full training set is used. We conjecture, the opposite trend is caused by the high variance introduced by max-pooling and the learning of position embeddings. This model is indeed more difficult to train than the other two models, because it obtains the highest variance among all three models when parameters are randomly initialized. Despite this, the pre-training provides significantly better initialization of model parameters and leads to small variance across all pre-training samples.

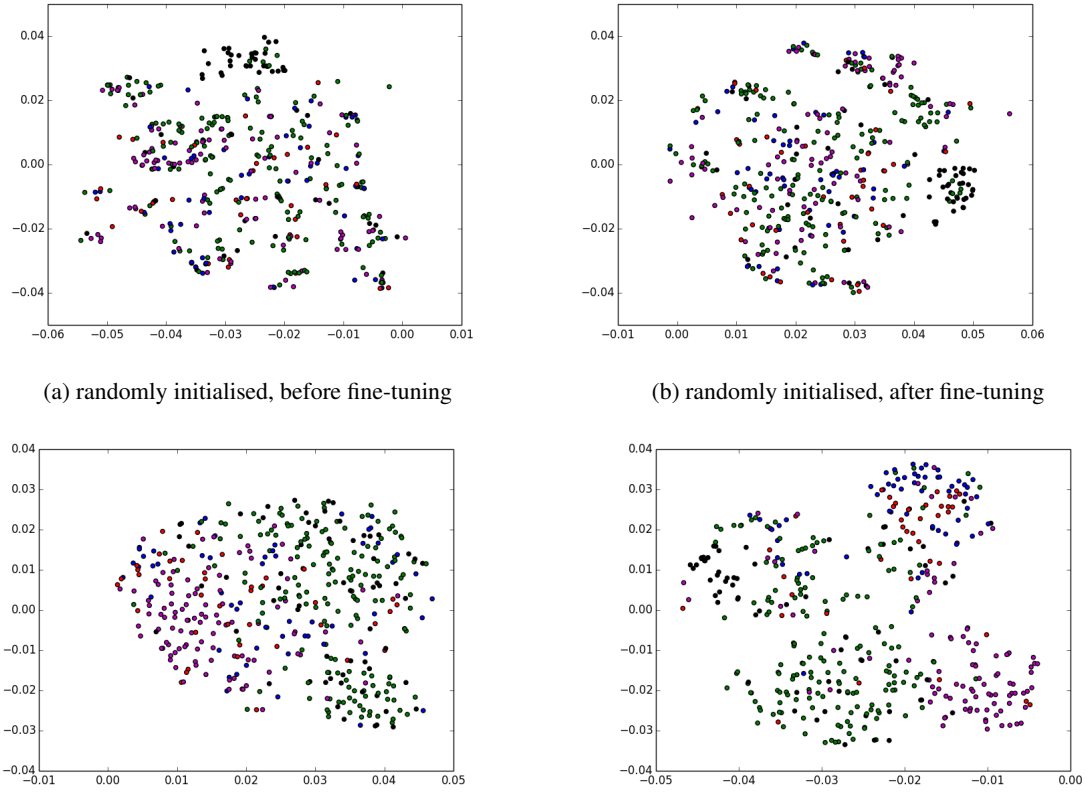
The stepwise training strategy is helpful for improving entity embeddings regardless the type of models, as shown in Figure 3. However, it is also not the main power booster during pre-training as the largest improvement is achieved always by unsupervised training losses. In case of *BiLSTM*, the improvement over the averaged word embeddings becomes clear when more than 800 training instances are used.

In order to gain a deeper understanding of the effect of pre-training, we compare the representations generated by the pre-trained models with the ones without pre-training. We compare them also at the begin and the end of fine-tuning respectively. In particular, we apply T-SNE (Maaten and Hinton, 2008) to visualize the expression representations generated by the feature learning component $h(x)$ of *PCNN*. As the Figure 4 illustrates, compare to the randomly initialized *PCNN*, at the begin of fine-tuning, we are more likely to find the representations closed to each other with the one pre-trained with the sequence reconstruction loss, if the corresponding expressions express the same relation. It is an evidence of our high-level intuition: our unsupervised pre-training losses are able to build similar representations for similar relation



(a) BiLSTM (b) DepTreeLSTM (c) PCNN

Figure 3: Comparison between baselines, stepwise training of entity embeddings, and the pre-trained models. The error bars indicate standard deviation computed on all five experiments.



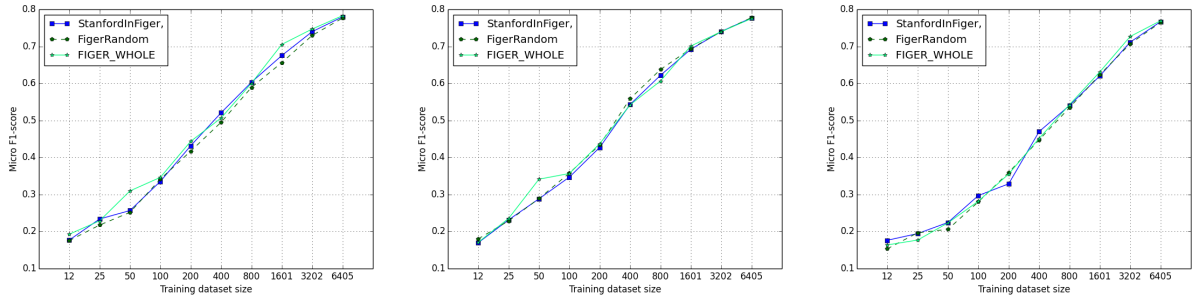
(a) randomly initialised, before fine-tuning (b) randomly initialised, after fine-tuning

(c) sequence reconstruction, before fine-tuning (d) sequence reconstruction, after fine-tuning

Figure 4: Visualization of the relation expressions of the top 5 most frequent relations sampled from the development set. The representations of these expressions are generated by using $h(x)$ of *PCNN* and further visualized by T-SNE. The top two figures are generated by randomly initialized *PCNN*, while the bottom ones are generated by *PCNN* pre-trained with *SeqReconstruct*. Different relations are marked with different colors.

expressions. After fine-tuning, the expressions of the same relation form more compact clusters by the pre-trained model than by the randomly initialized one. This explains the performance improvement achieved by the pre-trained *PCNN*.

The size and sampling strategies of unlabeled data have little influence on pre-training. Figure 5 shows that all models achieve similar results on random samples as on `WikiRandom`. Using the whole FIGER corpus leads to a marginal improvement up to 3% F1 score. This suggests that a few thousand randomly selected sentences are sufficient for achieving the pre-training effect with this sequence reconstruction loss.



(a) BiLSTM (b) DepTreeLSTM (c) PCNN

Figure 5: Impact of the size of training data.

7 Conclusion

In the absence of large amount of manually labeled training data, we propose the sequence reconstruction loss as a generalization of semi-supervised seq2seq model for pre-training deep RE models. The pre-trained models achieve competitive performance as their counterparts without pre-training while employing merely half or even a quarter of the training data.

Acknowledgments

This research was supported by NICTA, funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

References

Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D. Manning. 2014. Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, October 25-29, Doha, Qatar*, pages 1556–1567.

Yoshua Bengio. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 152–160.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 551–560. Association for Computational Linguistics.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006a. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 129–136. Association for Computational Linguistics.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006b. Semi-supervised relation extraction with label propagation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 25–28. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

George Dahl, Abdel-rahman Mohamed, Geoffrey E Hinton, et al. 2010. Phone recognition with the mean-covariance restricted boltzmann machine. In *Advances in neural information processing systems*, pages 469–477.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3061–3069.

Bart Desmet and Véronique Hoste. 2014. Fine-grained dutch named entity recognition. *Language Resources and Evaluation*, 48(2):307–343.

Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China*, pages 626–634.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Javid Ebrahimi and Dejing Dou. 2015. Chain based RNN for relation classification. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1244–1249.

Lisheng Fu and Ralph Grishman. 2013. An efficient active learning framework for new relation types. In *IJCNLP*, pages 692–698.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.

Zhou GuoDong, Qian LongHua, and Zhu QiaoMing. 2009. Label propagation via bootstrapped support vectors for semantic relation extraction between named entities. *Computer Speech & Language*, 23(4):464–478.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *HLT-NAACL*, pages 113–120.

Jing Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. 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*, pages 1012–1020. Association for Computational Linguistics.

Seokhwan Kim and Gary Geunbae Lee. 2012. A graph-based cross-lingual projection approach for weakly supervised relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 48–53. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Haibo Li, Yutaka Matsuo, and Mitsuru Ishizuka. 2010. Semantic relation extraction based on semi-supervised learning. In *Asia Information Retrieval Symposium*, pages 270–279. Springer.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 285–290.

ZHOU GuoDong LI JunHui QIAN LongHua and ZHU Qiaoming. 2008. Semi-supervised learning for relation extraction. In *Third International Joint Conference on Natural Language Processing*, page 32.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *ACL (2)*, pages 68–74.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 39–48.

Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, Nathan Schneider, and Timothy Baldwin. 2015. Big data small data, in domain out-of domain, known word unknown word: The impact of word representations on sequence labelling tasks. In *Proceedings of the 19th Conference on Computational Natural Language Learning (CoNLL 2015)*, pages 83–93.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 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, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1201–1211.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 926–934.

Ang Sun and Ralph Grishman. 2012. Active learning for relation type extension with local and global data views. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1105–1112. ACM.

Ang Sun. 2009. A two-stage bootstrapping algorithm for relation extraction. In *Recent Advances in Natural Language Processing, RANLP 2009, 14-16 September, 2009, Borovets, Bulgaria*, pages 76–82.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 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*, pages 455–465. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1785–1794.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Mo Yu, Matthew R. Gormley, and Mark Dredze. 2015. Combining word embeddings and feature embeddings for fine-grained relation extraction. In *The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA*, pages 1374–1379.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2335–2344.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, PACLIC 29, Shanghai, China, October 30 - November 1, 2015*.