Exploring Graph Neural Networks, Bidirectional LSTMs, and BERT for Enhanced Relation Extraction

University of Manchester

Yuhang Wu Neel More Ye Liu

Abstract

Relation extraction stands as a pivotal task in natural language processing, where adeptly capturing semantic relationships between entities holds paramount importance. Among the crucial features in relation extraction tasks, entity information stands out prominently. However, prevailing neural network models have yet to fully exploit entity information. As joint entity and relation extraction is one of the most important facet in Natural language processing tasks, even so a few existing had focused on considering possible relational information between entities before extracting them, leading the models to not constitute valid triplets. To address this gap, we propose two models. The first model amalgamates multi-head attention mechanisms with Bi-LSTM This amalgamation effectively harnesses contextual information from the input sequence, dynamically directing attention to segments pertinent to the relation extraction task. The second model uses Syntax-Induced pre-training with dependency masking to improve upon Heterogeneous graph neural network for relation extraction . Our experiments on the SemEval-2010 Task 8 benchmark dataset showcase the performance of our model. It enhances the model's generalization capability and training efficiency to a notable degree, facilitating more accurate predictions when handling text sequences with intricate structures.

1 Introduction

In the field of Natural Language Processing (NLP), relation extraction stands as a critical task aimed at identifying and extracting semantic relationships between entities from textual data. This task holds significant importance for various NLP applications, including information retrieval, question answering systems, and knowledge graph construction (Nguyen and Grishman, 2015). The relation classification task is defined as predicting the semantic relationship between two annotated entities in a sentence (Hendrickx et al., 2019).

Sentence	
Financial <e1>*stress*</e1> is one of the main causes of <e2>*divorce*</e2>	
Entity 1: *stress*	**Entity 2:** *divorce*
Relation	
Cause-Effect(e1,e2)	

Table 1: A example of relation extraction

Consider the sentence provided in Sentence Table 1 as an illustration. The entities within the sentence have been labeled accordingly, with *<e1></e1>* enclosing the first entity and *<e2></e2>* enclosing the second entity. In this instance, the relationship between these entities is classified as *Cause-Effect(e1,e2)*.

Accurate relationship classification is crucial for precise sentence interpretation, discourse processing, and higher-level tasks in translation NLP (Hendrickx et al., 2010). Consequently, the task of relation extraction has garnered significant attention over the past decades (Qian et al., 2009; Rink and Harabagiu, 2010). Supervised methods, meticulously crafted from lexical and semantic resources, have achieved remarkable performance (Hendrickx, 2010; Rink and Harabagiu, 2010). However, effectively integrating feature selection and knowledge sources remains a challenging aspect of relation classification.

In previous research, deep neural networks have been widely applied to relation extraction tasks, exhibiting the capability to derive effective features from both lexical and sentence levels (CN Santos, 2015; Zhang, 2015; Zeng et al., 2014; Yu et al., 2014).

Recently due to the introduction of BERT (Bidirectional Encoder Representations from Transformers), which requires a single layer on top of the pre-trained bidirectional representations many state-of-the art models can be built with ease. Allowing us to achieve cutting-edge performances and fostering innovation in the domain of language processing(Jacob et al., 2019).

Building upon prior work, we introduce two novel approaches. The first method is rooted in traditional neural network models, where we amalgamate multi-head attention mechanisms with Bi-LSTM. Furthermore, we incorporate Dropout regularization techniques and Xavier initialization methods. This results in the *Multi-Attention Bi-LSTM model*, which achieves enhanced accuracy without relying on additional knowledge or NLP systems, thereby effectively bolstering the model's generalization capability and training efficiency. In the second model, to learn a better text encoder that has important structural knowledge about the sentence context the model employs the method of training the encoder with masking meanwhile recovering word to word dependency and word connections that are analyzed from the parsers. The parser thus can be considered as weekly supervised but sufficiently provided with the syntax integration (Tian et al., 2022).

The significance of dependency parsers can clearly be observed in the demonstrations of many of the previous works where LSTM combined with short dependency path proves valuable in relation classification. With this knowledge we have tried to embed the encoder with syntax knowledge (Yan Xu et .al, 2015). The remainder of the paper is structured as follows. Section 3 elaborates on the structure of the Multihead Attention BiLSTM model, while Section 4 introduces the architecture of SIP-RIFRE model. Section 5 covers our dataset, experimental setup, as well as experimental results and discussions. Finally, Section 6 provides a summary of our paper.

2 Related Work

Numerous studies have investigated relation classification, employing various methodologies. Early approaches predominantly entailed manual feature engineering using NLP tools or handcrafted kernels (Lee, 2019). For example, Rink and Harabagiu (2010) proposed utilizing a Support Vector Machine (SVM) model alongside external corpus features for relation extraction.

As neural networks advanced, they became prevalent in relation extraction tasks. Zeng et al. (2014) pioneered the use of Convolutional Neural Networks (CNNs) for relation classification, achieving an accuracy of 82.7% by capturing local text features through convolutional operations. Subsequently, dos Santos et al. (2015) enhanced this approach with the CR-CNN model, which leverages ranking methods to capture entity relations. Introducing attention mechanisms, Huang and Shen (2016) devised the Attention CNN model, augmenting the model's focus on salient information and achieving an accuracy of 84.3%. Wang et al. (2016) further advanced this field with the Multi-Attention CNN model, attaining an accuracy of 88.0%.

Additionally, Recurrent Neural Networks (RNNs) have seen widespread adoption in relation extraction. Zhang et al. (2015) proposed Bidirectional Long Short-Term Memory Networks (Bi-LSTM) for relation classification, achieving an accuracy of 82.7%. Xiao and Liu (2016), as well as Zhou et al. (2016), introduced the Hierarchical Attention Bi-LSTM and Attention Bi-LSTM models, respectively, incorporating attention mechanisms and achieving accuracies of 84.3% and 84.0%. More recently, Lee et al. (2019) introduced the Entity Attention Bi-LSTM model, which refines the model's focus on entity information using entity-aware attention mechanisms, yielding an accuracy of 85.2%. Our primary model leverages RNNs, integrating multi-head attention mechanisms and Bi-LSTM, thereby enriching the model's comprehension of information across diverse semantic levels through multi-level attention mechanisms.

The previous studies in the field of Bidirectional encoder representation have showcased the importance of pre-training the model with the same training data but on different directions to achieve a fine-tuning scheme, and hyper-parameters for BERT (Vaswani A et .al, 2017). The studies in Graph networks especially graph attention networks with masking had shown extraordinary performances when incorporated with masking and self-attention mechanisms allowing to assign different weights to different nodes within the same neighborhood, at the same time dealing with a large neighborhood. These graph networks deals with the node and edges representation of data does not seem to require the entire graph structure upfront, thus allows to expand from a small isolated context space gradually. These graph networks does not rely on complex and intensive matrix operations and are hereby more computationally efficient. This study can be seen in the work done by (Petar V et .al, 2018).

3 Multi-Attention Bi-LSTM Model

3.1 Word Embeddings

Let a input sentence is denoted by $S = \{x_1, x_2, \ldots, x_T\}$, where T is the number of words, we transform each word x_T into a vector representations e_i by looking up word embedding matrix $W_{word} \in \mathbb{R}^{d_w|V|}$, where d_w is the dimension of the vector, $|V|$ is a fixed-sized vocabulary, W_{word} is a parameter needs to be learned. We use matrix-vector product to get word embedding e_i, then the word representations $embs = \{e_1, e_2, \ldots, e_T\}$, are obtained, and fed into next layers.

We used the pre-trained word embedding model from GloVe, and to prevent overfitting, a Dropout layer is added after the word embedding layer. This layer randomly drops some elements of the word embedding vectors, reducing the model's reliance on the training data excessively and enhancing its generalization ability.

3.2 Bidirectional LSTM

The Long Short-Term Memory (LSTM) unit was introduced by Hochreiter and Schmidhuber (1997) to address the vanishing gradient problem. Subsequently, numerous LSTM variants have emerged. We employ a variant proposed by Graves et al. (2013), which incorporates a weighted glimpse connection carousel (CEC) from constant error to gates within the same memory block. This mechanism enables the direct generation of gate current cell states using the current cell state (Graves, 2013).

The following is the LSTM formula for a single memory block, where i_t is the *Input Gates*, f_t is the *Forget Gates*, c_t is *Cells*, o_t is *Output Gates*, h_t is *Cell Outputs*, and σ is the activation function:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$

$$c_t = f_t c_{t-1} + \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

$$h_t = o_t \tanh(c_t)$$

In this paper, we use Bi-LSTM, The network comprises two subnetworks dedicated to left-sequence context and right-sequence context, respectively, facilitating forward and backward propagation. We use the following equation to combine the forward and backward pass outputs:

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

3.3 Multi-head Attention

The multi-head attention mechanism is a powerful neural network structure commonly employed for processing sequential data. It enables simultaneous attention over different parts of the input sequence and learns meaningful representations from them (Voita et al., 2019). In our model, we utilize a multi-head self-attention mechanism, structured in figure 1.

We input a sequence $X = \{x_1, x_2, \ldots, x_n\}$ and utilize linear transformations to map X to d-dimensional query (Q), key (K), and value (V) spaces, yielding Q, K, and V. For each attention head i, we compute the attention weights ai for each head and perform a weighted sum over the corresponding sequence values V, yielding the output for each head as follows:

$$\alpha_i = \mathrm{softmax}\left(\frac{QW_i^Q\left(KW_i^K\right)^T}{\sqrt{d_k}}\right)$$

Then we concatenate the outputs of h attention heads together to form the output of the attention mechanism.

In our model, the multi-head attention mechanism takes the output vector matrix H from the LSTM layer as input and computes a weighted sum to obtain the representation r of the sentence pair, the equation as following:

$$r = H\alpha^T$$

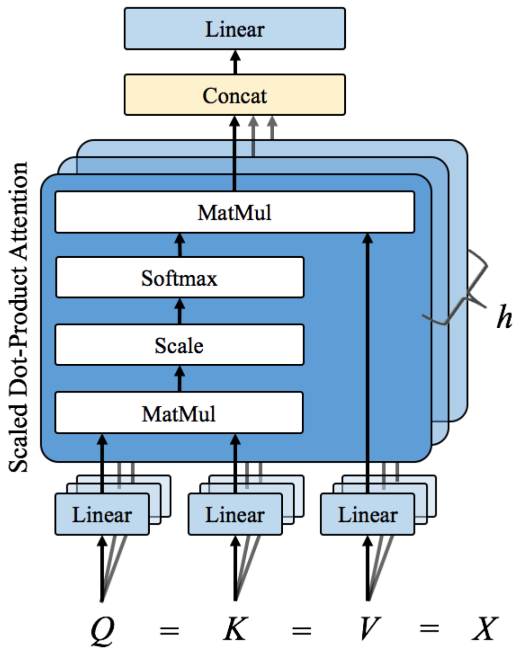

Figure 1: Multi-Head Self Attention (Lee et al., 2019).

Finally, through a tanh activation function, we obtain the final representation $h*$ for classification of the sentence pair.

$$h^* = \tanh(r)$$

This representation preserves crucial information from the sentence pair and can be utilized for subsequent relation classification tasks.

3.4 Classifying and Training

We employ a SoftMax classifier to predict the label $\hat{y}$for sentence S, where the labels are drawn from a discrete set of classes Y. The classifier takes the hidden state as input, and the loss function is the negative log-likelihood of the true class labels $\hat{y}$, and the conditional probability $p(y|S;\theta)$ is as following:

$$p(y|S;\theta) = \mathrm{softmax}(W_O z + b_O)$$

We utilize dropout for regularization, a technique introduced by Hinton et al. (2012). Dropout is applied to the embedding layer, LSTM layer, and the penultimate layer. Additionally, we rescale the weight vectors w to have a L2 norm of $||w|| = s$ after each gradient descent step if $||w|| > s$.

4 Second Model

4.1 Word Embedding

The word embedding used in the SIP-RIFRE model relies on order dependencies. Dependency parser is applied onto the input to obtain the underlying tree structure Tx . The input sentences can be represented $X = \{x_1, x_2, \ldots ,\ x_T\}$

$$\mathrm{Y} * \mathrm{M} = \mathrm{D_M}(\mathrm{D_E}(\mathrm{T_x}))$$

The approach extracts first, second and third order dependency $\mathrm{T_x}$ and represents it in the form of a tuple (x_i , x_j , type). Where there is a relation between x_i, x_j and type is directional. For first order dependency a direct connection from $\mathrm{T_x}$, however for second order dependency a connection is made between x_i and x_j if there exist another word that connects both x_i and x_j with the connections (x_i , x_0) and (x_0 , x_j) in $\mathrm{T_x}$, Three types for their connections namely; ancestor, sister, and descendant are defined according to the position of xi and xj in the dependency tree TX. Similarly for third order other dependencies are defined, ancestor, uncle, nephew, and descendant.

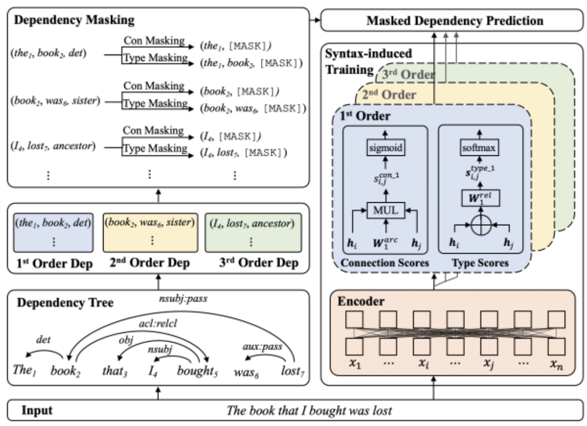

Figure 2: The left part shows the process to extract and mask dependencies (connection and type masking, respectively) in first, second, and third orders, where the word subscript denotes its sentential index. The right part illustrates the process to compute the scores of dependency connections and types in different orders to recover the ones that being masked (Tian, Y et .al, 2022).

5 Experiments

5.1 Dataset

Our model is evaluated on the SemEval-2010 Task 8 dataset, comprising 10 distinct relations: Cause-Effect, Instrument-Agency, Product-Producer, Content-Container, Entity-Origin, Entity-Destination, Component-Whole, Member-Collection, Message-Topic, and Other. The first nine relations are bidirectional, while Other is non-directional, resulting in a total of 19 relations. The dataset consists of 10,717 annotated sentences, with 8,000 samples allocated for training and 2,717 samples for testing. We utilize the official evaluation metric of SemEval-2010 Task 8, which is based on the macro-averaged F1-score.

5.2 Implementation Details

The hyperparameters utilized during the training process for our proposed two models are as follows:

Hyper-Parameters	Value	Description
word_dim	100	Word Embedding size
epoch	50	Number of epoch
batch_size	10	Mini_Batch size
lr	1.0	Learning Rate
dropout	0.3	Word Embedding layer
	0.3	BLSTM layer
	0.5	Entity-aware Attention layer
layers_num	1	Number of LSTM
L2_decay	1e-05	L2 Regularization Coefficient

Table 2: Hperparmeters used for Multi-BiLSTM

Hyper-Parameters	Value	Description
hidden_size	768	Hidden dimension
epoch	50	Number of epoch
vocab_size	10	Vocabulary file
lr	1.0	Learning Rate
intermediate_size	3072	Intermediate size
max_position_em	512	Maximum position embeddings

Table 3: Hyperparamters used in SIP-RIFRE

Architecture	Accuracy
Multi-Attention CNN (Wang et al. 2016)	88.0
Entity Attention Bi-LSTM (Lee et al., 2019)	85.2
Attention Bi-LSTM (Zhou et al., 2016)	84.0
REDN	91
RELA	90.6
SP	91.9
Our models	
SIP-RIFRE	91.3
Mult-Att BiLSTM	83.82

Table 3: Experimental Results

The table above displays various architectures used for relation extraction and the corresponding accuracy scores produced on the SemEval-2010 Task-8 Dataset.

The Multi attention CNNs implementation shown above combines the strengths of CNNs in capturing local features and attention mechanisms in focusing on relevant parts of the text to identify relationships between entities however adding multiple heads contributes to model complexity. CNNs generally acquire noise over the period and are also not very adept al learning long term dependencies contributing to the poor performance compared to other Bidirectional Graph neural network models.

Long Short Term Memory has been used since a long time and had seen a lot of variations in Natural Language Processing. Bidirectional LSTM models also tend to become computationally complex at some point if we try to make it retain most of the information, therefore it is better to use models with learned weights rather than relying on large amounts of data which LSTMs benefit from.

Graph networks have proved to be less computationally extensive and are able to capture relational information just as efficiently as CNNs but require the text to be transformed into a graph structure. However GNNs are sensitive to noisy data thus focuses mostly on structured data.

Introduction of BERT has revolutionized the NLP domain and has still many other applications to be explored. Our BERT based model uses heterogeneous graph structures that encode relations and words in different nodes allowing us to explore the contexts more efficiently, our other model combines Multi attention with Bidirectional LSTMs which helps the model to cover more textual information and explore and learn embedding in all directions.

Conclusion

During our exploration of different models we have come to a few conclusions:

Relation extraction has witnessed significant advancements with various architectures offering unique approaches. Multi-attention CNNs and Bidirectional LSTMs have mainly steadied the field for a while the emergence of BERT has opened new avenues for relation extraction, particularly with it's ability to handle heterogeneous graphs. Overall the choice of model depends on factors like data availability, computation resources, and desired level of interpretability.

References

[1] Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Séaghdha, D.O., Padó, S., Pennacchiotti, M., Romano, L. and Szpakowicz, S., 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. arXiv preprint arXiv:1911.10422.

[2] Lee, J., Seo, S. and Choi, Y.S., 2019. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. Symmetry, 11(6), p.785.

[3] Nguyen, T.H. and Grishman, R., 2015, June. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the 1st workshop on vector space modeling for natural language processing (pp. 39-48).

[4] Qian, L., Zhou, G., Kong, F. and Zhu, Q., 2009, August. Semi-supervised learning for semantic relation classification using stratified sampling strategy. In Proceedings of the 2009 conference on empirical methods in natural language processing (pp. 1437-1445).

[5] Rink, B. and Harabagiu, S., 2010, July. Utd: Classifying semantic relations by combining lexical and semantic resources. In Proceedings of the 5th international workshop on semantic evaluation (pp. 256-259).

[6] Santos, C.N.D., Xiang, B. and Zhou, B., 2015. Classifying relations by ranking with convolutional neural networks. arXiv preprint arXiv:1504.06580.

[7] Yu, M., Gormley, M. and Dredze, M., 2014, December. Factor-based compositional embedding models. In NIPS workshop on learning semantics (pp. 95-101).

[8] Zeng, D., Liu, K., Lai, S., Zhou, G. and Zhao, J., 2014, August. Relation classification via convolutional deep neural network. In Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers (pp. 2335-2344).

[9] Zhang, S., Zheng, D., Hu, X. and Yang, M., 2015, October. Bidirectional long short-term memory networks for relation classification. In Proceedings of the 29th Pacific Asia conference on language, information and computation (pp. 73-78).

[10] Graves, A., 2013. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

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

[12] Lee, J., Seo, S. and Choi, Y.S., 2019. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. Symmetry, 11(6), p.785.

[13] Santos, C.N.D., Xiang, B. and Zhou, B., 2015. Classifying relations by ranking with convolutional neural networks. arXiv preprint arXiv:1504.06580.

[14] Shen, Y. and Huang, X.J., 2016, December. Attention-based convolutional neural network for semantic relation extraction. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 2526-2536).

[15] Voita, E., Talbot, D., Moiseev, F., Sennrich, R. and Titov, I., 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. arXiv preprint arXiv:1905.09418.

[16] Wang, L., Cao, Z., De Melo, G. and Liu, Z., 2016, August. Relation classification via multi-level attention cnns. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1298-1307).

[17] Xiao, M. and Liu, C., 2016, December. Semantic relation classification via hierarchical recurrent neural network with attention. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 1254-1263).

[18] Zeng, D., Liu, K., Lai, S., Zhou, G. and Zhao, J., 2014, August. Relation classification via convolutional deep neural network. In Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers (pp. 2335-2344).

[19] Zhang, S., Zheng, D., Hu, X. and Yang, M., 2015, October. Bidirectional long short-term memory networks for relation classification. In Proceedings of the 29th Pacific Asia conference on language, information and computation (pp. 73-78).
[20] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H. and Xu, B., 2016, August. 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) (pp. 207-212).
[21] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[22] Zhao, K., Xu, H., Cheng, Y., Li, X. and Gao, K., 2021. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. Knowledge-Based Systems, 219, p.106888.
[23] Tian, Y., Song, Y. and Xia, F., 2022, May. Improving relation extraction through syntax-induced pre-training with dependency masking. In Findings of the Association for Computational Linguistics: ACL 2022 (pp. 1875-1886).
[24] Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H. and Jin, Z., 2015, September. 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 (pp. 1785-1794).
[25] Yan, H., Qiu, X. and Huang, X., 2020. A graph-based model for joint chinese word segmentation and dependency parsing. Transactions of the Association for Computational Linguistics, 8, pp.78-92.