The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

Classification of Semantic Relations between Pairs of Nominals Using Transfer Learning

Linrui Zhang, Dan Moldovan
The University of Texas at Dallas
800 West Campbell Road ; MS EC31, Richardson, TX 75080 U.S.A
linrui.zhang@utdallas.edu, moldovan@hlt.utdallas.edu

Abstract

The representation of semantic meaning of sentences using neural network has recently gained popularity, due to the fact that there is no need to specifically extract lexical syntactic and semantic features. A major problem with this approach is that it requires large human annotated corpora. In order to reduce human annotation effort, in recent years, researchers made several attempts to find universal sentence representation methods, aiming to obtain general-purpose sentence embeddings that could be widely adopted to a wide range of NLP tasks without training directly from the specific datasets. InferSent, a supervised universal sentence representation model proposed by Facebook research, implements 8 popular neural network sentence encoding structures trained on natural language inference datasets, and apply to 12 different NLP tasks. However, the relation classification task was not one of these. In this paper, we re-train these 8 sentence encoding structures and use them as the starting points on relation classification task. Experiments using SemEval-2010 datasets show that our models could achieve comparable results to the state-of-the-art relation classification systems.

Introduction

The mission of SemEval-2010 task 8 (Hendrickx et al. 2009) is to predict semantic relations between pairs of nominals. Formally, given a sentence S with the annotated pairs of nominals e_1 and e_2, the task is to identify the semantic relation between e_1 and e_2 from a pre-defined relation set. For instance, the following sentence contains an example of the CAUSE-EEFFECT (PRESSURE, BURST) relation between the nominals **burst** and **pressure**.

The **burst** has been caused by water hammer **pressure**.

Recent work on relation classification has focused on the use of deep neural networks. Most of these models follow a standard architecture: (1) a sentence embedding layer (sentence encoder) that generates sentence embeddings which represents the semantic meaning of the original text, and (2) a classification layer that generates a probabilistic distribution of the potential relations from the sentence embeddings. In recent years, several neural network-based sentence embedding structures were proposed, such as Recursive Neural Networks (Socher et al. 2012), Convolutional Neural Networks (Zeng et al. 2014) (Santos, Xiang, and Zhou 2015), Long Short Term Memory Networks (LSTM) (Xu et al. 2015), Bidirectional LSTM (Zhang and Wang 2015), and Bidirectional LSTM with attention (Zhou et al. 2016). However, all these structures are trained in a supervised manner directly on the training data of SemEval-2010 dataset. Relying on SemEval-2010 training datasets hinders new applications. An obvious answer to this problem is to use unsupervised architectures to learn the sentence representation. Recent works include Paragraph Vectors (Le and Mikolov 2014), Skip-thoughts (Kiros et al. 2015) and FastSent (Hill, Cho, and Korhonen 2016). However, these models have not reach satisfactory performance in practice. Another possible solution is to use transfer learning.

Transfer learning is a machine learning strategy where a model trained on a task is re-used as the starting point for a new task, so the new task will take the advantage of the pre-learned knowledge from the previous task. The idea of transfer learning has been proven successful in computer vision (Taigman et al. 2014), (Antol et al. 2015). On the contrary, it has attracted less attention in the NLP community. Facebook AI research released InferSent (Conneau et al. 2017) that performs supervised learning of universal sentence representation from Natural Language Inference data. It implemented 8 popular sentence encoding structures such as GRU, LSTM, BiLSTM (with mean or max pooling), self-attentive network and Hierarchical ConvNet, and trained them on the Stanford Natural Language Inference Datasets (SNLI) (Bowman et al. 2015) which contains 570K human-generated English sentence pairs. After the sentence encoding structures are trained to learn the representation of sentences, they have been applied to 12 different NLP tasks such as Semantic Textual Similarity, Paraphrase Detection and Caption-Image Retrieval. The results show that transfer learning-based models outperform the models using unsupervised sentence representation methods like SkipThought and are very comparable to state-of-the-art models that directly trained from the training data of each of these 12 specific tasks.

Even though Infersent has been transferred to 12 different NLP tasks, the evaluation on Relation Classification task was not covered in their paper. The primary contribution of our paper is to fill in this gap and extend their research to

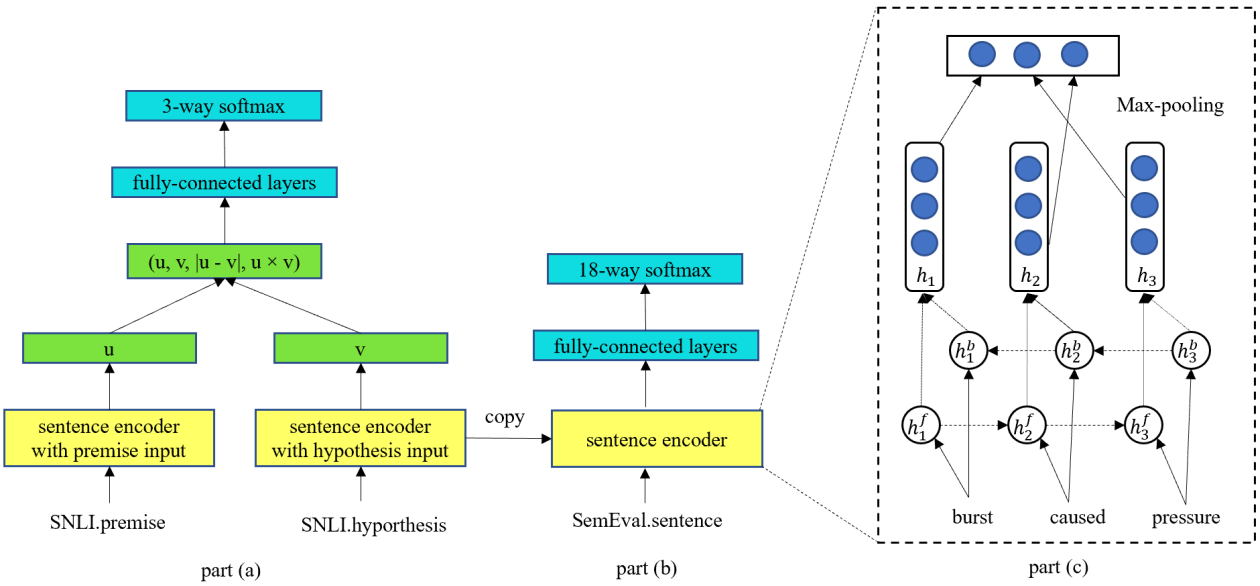

Figure 1: The main structure of our system. A BLSTM with max-pooling network is used as an example of sentence encoder defined in InferSent.

the area of relation classification. To the best of our knowledge, this is the first attempt to solve Relation Classification problem using transfer learning.

Approach

In this section, we will introduce our approach consisting of five steps: (1) data processing, (2) sentence encoder trained on Natural Language Inference task (original task), (3) neural network model built for Relation Classification task (transferred task), (4) sentence encoder architecture and (5) parameter setting. Figure 1 illustrates the structure of our system.

Data Processing

Based on previous literature (Xu et al. 2015), the shortest dependency path between two nominals is mostly sufficient to represent the semantic relation between them. Instead of feeding in the whole sentence, we utilize the words in the shortest dependency path between two nominals as the raw input to the sentence embedding layer. Figures 2 and 3 illustrate the dependency graph of the sentence "The burst has been caused by water hammer pressure", and the shortest dependency path between nominals **burst** and **pressure**. In this example, the words sequence "burst caused pressure" will substitute the whole sentence as the system input.

Sentence Encoder Trained on Natural Language Inference Task

The task of Natural Language Inference, also known as Textual Entailment, is to detect the relationships between two sentences, a premise sentence and a hypothesis sentence. The relation between the sentence pair is one of three categories: entailment (hypothesis cannot be false when premise

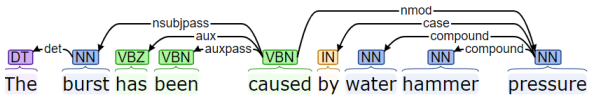

Figure 2: The dependency graph of sentence " The burst has been caused by water hammer pressure". *nsubjpass*, *nmod*, *case* etc. are the dependency relations between tokens

$$\textbf{burst} \xleftarrow{nsubjpass} caused \xrightarrow{nmod} \textbf{pressure}$$

Figure 3: Figure 3: The shortest dependency path between **burst** and **pressure**. in the sentence. " The burst has been caused by water hammer pressure."

is true), contradiction (hypothesis is false whenever premise is true) and neutral (the truth of hypothesis could not be determined on the basis of premise).

Figure 1 Part (a) shows how the pre-defined sentence encoders in InferSent are trained on Natural Language Inference Task. The sentence pairs (premise and hypothesis) are first passed through a sentence encoder to generate sentence embeddings (u and v), u is the sentence embedding for the premise sentence and v is the sentence embedding for hypothesis sentence. A connection layer will later concatenate u, v, the element-wise absolute difference of u and v, and the element-wise multiplication of u and v, and generate a concatenated vector $(u, v, |u - v|, u * v)$ that represents the relationships between the original sentence pairs. The concatenated vector $(u, v, |u - v|, u * v)$ is fed into a 3-classifier (Entailment, Neutral and Contradiction) to perform classification and generate the probabilistic distribution of the three

candidate relations.

After the system is trained, parameters in the sentence encoder are saved to be used in transfer task.

Sentence Encoder on Relation Classification Task

Figure 1 Part (b) shows how the pre-trained sentence encoders are transferred to Relation Classification task. First, we will set up the sentence encoder by re-loading the parameters trained from the previous task. The words in the shortest dependency path between nominals will be fed into the sentence encoder and generate the embeddings that could represent the semantic relations between nominals. An 18-classifier is built on top of the embedding layer to generate the probabilistic distribution of the 18 candidate relations defined in SemEval-2010 relation set.

Sentence Encoder Architectures

In this section, we will briefly introduce the structures of encoders defined in InferSent.

InferSent pre-defined 8 different sentence encoding structures. (1) InferSentEncoder (2) BLSTMprojEncoder (3) BGRUlastEncoder (4) InnerAttentionMILAEncoder (5) InnerAttentionYANGEncoder (6) InnerAttentionNAACLEncoder (7) ConvNetEncoder and (8) LSTMEncoder. Except from encoder (7) ConvNetEncoder with the structure of Hierarchical Convolutional Architecture, the rest are all sequential encoders based on (Bi-)GRU/(Bi-)LSTM that could model the sequential information and long-distance patterns in the text. In the following section, InferSentEncoder (1) will be used as an example to introduce the math behind these sequential encoder architectures.

The architecture of InferSentEncoder is a Bi-directional LSTM neural network with max-pooling layer. The structure of the model is shown in Figure 1 part (c). We will first illustrate the structure of LSTM neural network, then extend it to bi-directional model, and at last introduce the max-pooling strategy. The mathematical formulation of LSTM units is as follows:

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \tag{1}$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \tag{2}$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \tag{3}$$

$$\tilde{c}_t = tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \tag{4}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{5}$$

$$h_t = o_t \circ tanh(c_t) \tag{6}$$

It contains five components: an input gate i_t, a forget gate f_t, an output gate o_t, a new memory cell $\tilde{c}_t$, and a final memory cell c_t. Three adaptive gates i_t, f_t, o_t and new memory cell $\tilde{c}_t$ are computed based on the previous state h_{t-1} , current input x_t, and bias term b. The final memory cell c_t is a combination of previous cell content c_{t-1} and new memory cell $\tilde{c}_t$ weighted by the forget gate f_t and input gate i_t. The final output of the LSTM hidden state h_t is computed with the output gate o_t and final memory cell c_t.

A Bi-LSTM could be viewed as a network that maintains two hidden LSTM layers together, one for the forward propagation $\overrightarrow{h}_t$ and another for the backward propagation $\overleftarrow{h}_t$ at each time-step t. The final prediction $\hat{y}_t$ is generated through the combination of the score results produced by both hidden layers $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$. The mathematical representation of a simplified Bi-LSTM is shown as follows:

$$\overrightarrow{h}_t = f(\overrightarrow{W}x_t + \overrightarrow{V}\overrightarrow{h}_{t-1} + \overrightarrow{b}) \tag{7}$$

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t-1} + \overleftarrow{b}) \tag{8}$$

$$\hat{y}_t = g(Uh_t + c) = g(U[\overrightarrow{h}_{t-1}; \overleftarrow{h}_{t-1}] + c) \tag{9}$$

The idea behind max pooling is that the local features (the output of hidden state at each time step t) are not strong enough to represent the entire sentence but could represent the local patterns well. The final representation of the sentence could be achieved by merging the representations of each strongest local patterns together.

Besides the max pooling, attention mechanism is another strategy to learn which local pattern is important for the final representation. The attention mechanism is calculated in 3 steps. First, we feed the hidden state h_t through a one-layer perceptron to get u_t which could be viewed as a hidden representation of h_t. We later multiply u_t with a context vector u_w and normalized the results through a softmax function to get the weight a_t of each hidden state h_t. The context vector could be seen as a high-level vector to select informative hidden state and will be jointly learned during the training process. The final sentence representation is computed as a sum over of the hidden state h_t and its weights a_t. The mathematic representation is as follows:

$$u_t = tanh(Wh_t + b) \tag{10}$$

$$a_t = \frac{exp(u_t^T u_w)}{\sum_t exp(u_t^T u_w)} \tag{11}$$

$$S = \sum_t a_t h_t \tag{12}$$

Sentence encoders (4) to (6) all use different kinds of attention mechanisms. InnerAttentionMILAEncoder is inspired from (Lin et al. 2017), InnerAttentionYANGEncoder is based on (Liu et al. 2016), and InnerAttentionNAACLEncoder is built on top of (Yang et al. 2016).

For more details about encoder structures, please refer to the aforementioned citations and its implementation on GitHub [1].

Parameter Setting

Natural Language Inference task (original task) and Relation Classification task (transferred task) are both classification problems. We use cross-entropy loss between the system output and the gold annotated output as the training function. Glove.840B.300d (Pennington, Socher, and Manning 2014) is used to initialize the word embeddings.

In the training procedure of Natural Language Inference task, we only change the number of hidden units in sentence

[1] https://github.com/facebookresearch/InferSent

Relationship	Sentences with labeled nominals
Entity-Destination(e_1, e_2)	$[People]_{e_1}$ have been moving back into $[downtown]_{e_2}$.
Instrument-Agency(e_2, e_1)	Even commercial $[networks]_{e_1}$ have moved into $[high-definition\ broadcast]_{e_2}$.
Message-Topic(e_2, e_1)	Bob Parks made a similar $[offer]_{e_1}$ in a $[phonecall]_{e_2}$ made earlier this week.
Cause-Effect(e_2, e_1)	He had chest pains and $[headaches]_{e_1}$ from $[mold]_{e_2}$ in the bedrooms.
Entity-Origin(e_1, e_2)	The $[staff]_{e_1}$ was removed from his $[position]_{e_2}$
Product-Producer(e_2, e_1)	The $[bacterial\ aerosol]_{e_1}$ was generated from an up-draft $[nebulizer]_{e_2}$.
Component-Whole(e_2, e_1)	I have recently developed creases in my $[ear]_{e_1}$ $[lobes]_{e_2}$
Content-Container(e_1, e_2)	The $[key]_{e_1}$ was in a $[chest]_{e_2}$.
Cause-Effect(e_2, e_1)	In the same way, a $[society]_{e_1}$ is built up of many $[individuals]_{e_1}$.

Table 1: Examples of relations and Sentences with labeled nominals in SemEval-2010

embedding layer (sentence encoder) and keep the rest of the parameters as default settings in InferSent. The gradient descent optimizer is SGD with learning rate of 0.1.

In the Relation Classification task, the number of hidden nodes in the embedding layer are the same as in the original task. The number of hidden nodes in the classification layer is associated with the number of nodes and the structure of the embedding layer. For example, if the number of nodes in embedding layer is n, and the structure of the embedding later is LSTM or Bi-LSTM, then the number of nodes in the classifier will be n or $2n$. ADAM is used as the gradient descent optimizer with learning rate of 0.01.

Experiment and Results

Dataset

We use the SemEval-2010 Task 8 dataset to perform our experiments. This dataset contains 10,717 instances (8000 for training and 2717 for testing) annotated with 9 ordered relationships (with two directions) and an undirected *Other* relation, resulting in 19 relation classes. Table 1 shows examples of relations and sentences with labeled nominals in SemEval-2010 Task 8 dataset.

Experimental Setup

We build our experiment in 4 steps:

1. Extract the Shortest Dependency Path between Nominals using Stanford CoreNLP (Manning et al. 2014).
2. Re-train 8 sentence encoders within InferSent on SNLI dataset and save these well-trained encoders.
3. Reload the encoders for relation classification task and build an 18-way classifier (exclude *Other* class) as output layer.
4. Evaluate the new relation classification model on SemEval-2010 Task 8 testing dataset.

We use the official evaluation metric of SemEval-2010 which is based on macro-average F1 score that takes the directionality of 9 relations into consideration and ignore the *Other* class.

Empirical Results

The results obtained are shown in Table 2. We compare our results with other state-of-the-art relation classification systems using deep learning approach trained directly on SemEval-2010 training dataset. Table 2 shows the system name and F1 score of 5 recent state-of-the-art systems and our 8 universal sentence encoding systems evaluated on SemEval-2010 Task 8 testing dataset. Since we do not enhance our sentence embeddings with linguistic features such as POS, WordNet, NER, for a fair comparison, in Table 2, we list the results of recent systems obtained without using linguistic features as well. Their results optimized with linguistic features are in parentheses.

Models	F1
Recent Model (directly trained from training data)	
RNN (Socher et al. 2012)	74.8 (77.6)
MV-RNN (Socher et al. 2012)	79.1 (82.4)
CNN (Zeng et al. 2014)	69.7 (82.7)
SDP-LSTM (Xu et al. 2015)	82.4 (83.7)
BLSTM (Zhang et al. 2015)	82.7 (84.3)
Att-LSTM (Zhou et al. 2016)	84.0
Our model (transfer learning model)	
InferSent	80.2
BGRUlast	80.7
BLSTMproj	76.4
InnerAttentionMILA	74.6
InnerAttentionYang	77.5
InnerAttentionNAACL	76.9
ConvNet	77.9
LSTM	80.8

Table 2: Comparison of relation classification systems. We did not list the variation of word embeddings in the table. (Socher et al. 2012) (Zeng et al. 2014) use 50-d word embedding. (Zhang et al. 2015) (Zhou et al. 2016) use 100-d word embedding. (Xu et al. 2015) use 200-d word embedding.

Results Analysis

From the results, we have several observations:

First, our model could achieve comparable results compared with current state-of-the-art models, especially compared with the models that solely take word embeddings as features without using human annotated linguistic features.

Some of our architectures can even outperform the (Socher et al. 2012) RNN model that includes linguistic features. This shows the effectiveness of our approach. However, the overall results indicate that the universal sentence encoder still performs below than the supervised encoder (used in the current state-of-the-art models) that was directly trained for the specific task. For example, except ConvNetEncoder, the rest of the encoders all use (Bi-)GRU/(Bi-)LSTM structure, and none of them could beat the results of (Xu et al. 2015), (Zhang et al. 2015) and (Zhou et al. 2016) that use similar structures. As observed in (Conneau et al. 2017), using transfer learning on a new task can be viewed as an unsupervised approach.

Second, among the transfer learning models, encoders with simple structure like LSTMEncoder could outperform the encoders with complicated structure like InnerAttentionMILAEncoder (BiLSTM with Attention). This is because complicated structures could overfit on the original SNLI dataset, hence they could not generalize well when being transferred to SemEval dataset. We set up an experiment to prove this. We initialized the number of hidden nodes in InnerAttentionMILA to 25, 50 and 100 (more hidden nodes imply stronger learning ability), and the model received an increasing 0.80, 0.81 and 0.82 F1 score on SNLI testing data. However, when transferred to SemEval 2010 testing data, we obtained a decreasing F1 score of 74.6, 73.9 and 73.0. We can overcome this issue by an early stop of training process of the encoders before they are over-specified for the original tasks.

At last, in this paper, we just implemented a shallow experiment that does not consider linguistic features and fine-tuning the weights (we use most of the default parameters in InferSent). The next step is to improve the results by combining extra features and fine-tuning the weights to see if it can overperform supervised encoders directly trained for the task. We would also like to compare our encoder with other universal sentence encoding methods such as Paragraph Vectors, FastSent and SkipThought.

Related Work

Over the years, various methods have been proposed for universal sentence representation.

(Arora, Liang, and Ma 2016) proposed a simple but tough-to-beat baseline for sentence embeddings. They represent the sentence simply as a weighted average of word vectors and modified a bit with PCA, but their model improves the performance by about 10% to 30% in textual similarity tasks compared with sophisticated supervised methods such as RNN and LSTM.

(Le and Mikolov 2014) represented Paragraph Vectors that represents each document by a dense vector which is trained to predict words in the document. It overcome the weaknesses of bag-of-word models that loss the ordering of the words and could capture the semantic of the words.

(Kiros et al 2015) proposed Skip-Thought Vectors. They train an encoder-decoder model based on the continuity of text (S_i) from books and to predict the sentences around them (S_{i-1} and S_{i+1}). In this case, sentence that share semantic and syntactic properties will be clustered together.

(Hill et al., 2016) developed FastSent, a simple log-bilinear model that predicts adjacent sentences based on a Bag-of-word representation of sentences in context, similar to Skip-Though, but it needs much lower computational expense. They also point out that the task on which sentence embeddings are trained significantly impacts their quality.

Conclusion

In this paper, we extend (Conneau et al 2018)'s research work on transfer learning to a new problem, relation classification between pairs of nominals. Our model re-trained 8 popular sentence encoding architectures defined in InferSent with natural language inference data (SNLI) and transferred the encoder to relation classification task and evaluated their performance on SemEval 2010 dataset. Results have shown that our transferred universal sentence encoders obtain comparable results with the supervised encoders trained directly for the relation classification problem.

References

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Arora, S.; Liang, Y.; and Ma, T. 2016. A simple but tough-to-beat baseline for sentence embeddings.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Hendrickx, I.; Kim, S. N.; Kozareva, Z.; Nakov, P.; Ó Séaghdha, D.; Padó, S.; Pennacchiotti, M.; Romano, L.; and Szpakowicz, S. 2009. 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*, 94–99. Association for Computational Linguistics.

Hill, F.; Cho, K.; and Korhonen, A. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, 3294–3302.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 1188–1196.

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Liu, Y.; Sun, C.; Lin, L.; and Wang, X. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*.

Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 55–60.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Santos, C. N. d.; Xiang, B.; and Zhou, B. 2015. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 1201–1211. Association for Computational Linguistics.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Xu, Y.; Mou, L.; Li, G.; Chen, Y.; Peng, H.; and Jin, Z. 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*, 1785–1794.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2335–2344.

Zhang, D., and Wang, D. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Zhang, S.; Zheng, D.; Hu, X.; and Yang, M. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, 73–78.

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