A New Fine-Tuning Architecture Based on Bert for Word Relation Extraction

Fanyu Meng, Junlan Feng, Danping Yin, and Min Hu

China Mobile Research Institute, Beijing, China
{mengfanyu,fengjunlan,yindanping,humin}@chinamobile.com

Abstract. We introduce a new attention-based neural architecture to fine-tune Bidirectional Encoder Representations from Transformers (BERT) for semantic and grammatical relationship classificaiton at word level. BERT has been widely accepted as a base to create the state-of-the-art models for sentence-level and token-level natural language processing tasks via a fine tuning process, which typically takes the final hidden states as input for a classification layer. Inspired by the Residual Net, we propose in this paper a new architecture that augments the final hidden states with multi-head attention weights from all Transformer layers for fine-tuning. We explain the rationality of this proposal in theory and compare it with recent models for word-level relation tasks such as dependency tree parsing. The resulting model shows evident improvement comparing to the standard BERT fine-tuning model on the dependency parsing task with the English TreeBank data and the semantic relation extraction task of SemEval-2010Task-8.

Keywords: Relation Extraction · Dependency parsing · Attention.

1 Introducton

Recently a series of great works on language model pre-training have shown to be effective for improving a large suite of downstream NLP tasks spanning from sentence-level tasks to token-level tasks[12, 8, 13, 2].Particularly, the recent release of the BERT models have become the latest milestone in NLP. It is seen as an inflection point for the NLP field. BERT uses masked language models to obtain pre-trained deep bidirectional representations of characters, words and sentences. The state-of-the-art of a wide range of NLP tasks have been advanced via a standard BERT fine-tuning process. In sequence-level and token-level classification tasks, it takes the final hidden states (the last layer output of the multi-head transformer) of the first [CLS] sequence token or each individual token as input for a classification layer over a label set.

In this paper, we are motivated to extend BERT including the BERT architecture and the model itself to a category of NLP tasks: word-level relation classification. It focuses on classifying relationship between two words in a sentence such as syntactic dependency relationship in dependency parsing and semantic relationship between nominal words.

Dependency parsing is defined as a task to provide a simple description of the grammatical structure of a sentence. Dependency parsers are often evaluated on the

Penn Treebank (PTB) and the Chinese Treebank (CTB 5.1) with the unlabeled attachment score (UAS) and the labeled attachment score (LAS) metrics (excluding punctuation)[10]. UAS measures the accuracy that a model can predict if there is a head-modifier relationship between any given pair of tokens in a sentence. LAS measures the accuracy that a model can predict the specific relationship between any given pair of tokens in a sentence. We follow these evaluation metrics in our experiments.

The classification of semantic relationship between pairs of words is defined to classify various semantic relations between words, such as Cause-Effect(CE) ,Component-Whole(CW) relationship between nominal words [7].

In this paper, we focus on improving BERT fine-tuning process for the above tasks. For dependency parsing, we follow the general framework of Graph-based dependency parsers. For semantic relationship extraction, we treat it as a straight forward classification task. The contributions of our paper are summarized follows:

- We propose a new architecture to fine-tune BERT for word-level relation classification tasks. Rather than taking only the final hidden states as input for a feed-forward classifier, we propose to augment the hidden states with Transformer multi-head attention weights for classification. In experiments, this approach evidently improves the accuracy of the dependency tree parsing and the semantic relation extraction comparing to the standard BERT fine-tuning process.
- We construct a probing task to test and visualize the extent to which BERT representations preserve the word relationship as BERT primarily built on Self-Attention Transformer mechanism.

2 Related Work

2.1 Deep Neural Network for Dependency Parsing

Since the earlier work in [11], Graph-based dependency parsing has been typically formulated with the common structure prediction framework.

Given an input sentence, the parsing task is to select the dependency tree with the highest scores, which is decomposed to the sum of local arc scores for each head to dependent arc.

[1] made the first successful attempt to employ modern neural network into dependency parsing. The input to this network is the concatenation of three embedding vectors of involved words including word embedding, POS tag embedding and arc-tag embedding. Embedding vectors in this work are fed into a non-linear multiple layer perceptron (MLP) classifier. Since then, many other researchers have proposed various deep learning architectures to advance the state-of-the-art.

[9] employed a biLSTM (Bidirectional Long Short-Term Memory) network for both transition-based parsing and graph-based parsing. For the graph-based parsing, the biLSTM is considered as the feature function ϕ for a given arc (w_h, w_m) from the head word w_h and the modifier w_m, where LSTM encodes each word separately and then concatenate them as the arc feature.

$$\phi(s, w_h, w_m) = h(s, w_h) \circ h(s, w_m) \tag{1}$$

These features are used as input for a MLP classifier, which is similar to [1] work.

[5] included a graph-based dependency parser in their multitask neural model architecture. For dependency parsing, similarly it relies on LSTM to embed higher level attention features from low-level word embedding features. Instead of concatenating two LSTM encoded word embeddings as the input for the classifier, Hashimoto et al. (2016) applied a linear attention mechanism to combine them as:

$$\phi(s, w_h, w_m) = h(s, w_h) \cdot (W_d h(s, w_m)) \tag{2}$$

[4] proposed a biaffine network for dependency parsing. It follows the previous work to encode words via a BiLSTM network by taking word embeddings and POS tag embeddings as input. The hidden BiLSTM state for each word is fed to two MLP classifiers to respectively classify the word as a head or a modifier. The MLP output vectors are multiplied to derive arc scores for UAS and LAS. This approach obtains 95.7% UAS and 94.2% LAS. CVT + Multi-Task (Clark et al., 2018) advanced the latest state-of-the-art to 96.61% UAS and 95.02% LAS with a multi-task approach.

2.2 Semantic Relation Extraction

Semantic relation classification is a crucial component in numerous real-life NLP tasks. Multilevel convolutional neural network(CNN) and BiLSTM are the most popular model architectures applied in recent years. On top of that, various entity-aware attentions are proposed in recent research to advance the state-of-the-art performance [15].

The fundamental differences between above approaches are the attention mechanisms, MLP attention, bilinear attention, deep biaffine attention, and entity-aware attention on top of a base LSTM/CNN model. The BERT architecture is purely built on self-attention transformers. Hence, we believe it is natural to extend BERT for word-relation tasks. In the next section, we describe our proposal in detail.

3 Our Approach

Fundamentally, BERT's model architecture is a multi-layer bidirectional Transformer encoder[14]. Transformer entirely relies on self-attention to draw global dependencies among tokens within a given sequence. Intuitively, it is natural to believe the learnt attention weights between words are good candidate features for word relation extraction. In this Section, we first explain pair-wise attention, which is a key concept in our approach. Second, we propose our new architecture to fine-tune BERT explicitly using pair-wise attention weights for the classification layer. Third, we employ this procedure to dependency parsing and semantic relation tasks.

3.1 Pair-Wise Attention

A given input sequence $s = t_1, ..., t_N$ is represented as an embedding matrix X with each row corresponding to a word embedding vector x_i for the token t_i. For each x_i, Transformer creates a Query vector q_i, a Key vector k_i by linearly projecting x_i with

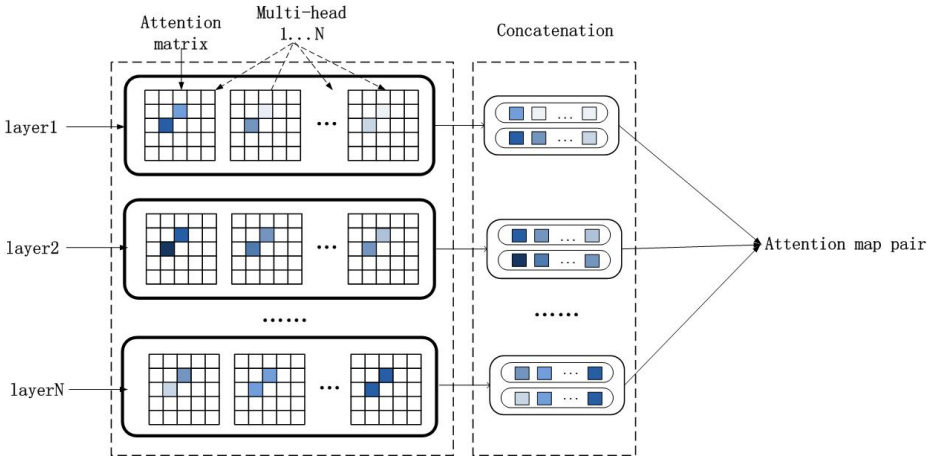

Fig. 1. The process of attention map pair extraction

different, learnt weight matrices W^Q, W^K. We denote the dimentions of q_i,k_i respectively as d_q, d_k. These vectors are packed into matrices Q, K. For a given layer l and a given head h, attention weights are computed as:

$$\begin{aligned} z^{l,h} &= Attention(Q, K) \\ &= softmax(\frac{QK^T}{\sqrt{(d_k)}}) \end{aligned} \tag{3}$$

Each element $z^{lh}_{i,j}$ in $z^{l,h}$ represents the attention of the ith token to the jth token.

Instead of performing a single attention function, [14] found it is beneficial to have multiple attention heads H. Hence, for a BERT model with L layers and H attention heads, there are $L * H$ attention weights $z^{l,h}_{i,j}$ for a token pair (t_i, t_j). We pack these weights in a vector $a^{i,j}$:

$$\begin{aligned} a^{i,j} = (z^{1,1}_{i,j}, ..., z^{l,h}_{i,j}, ..., z^{L,H}_{i,j}) \\ i, j \in (1, ..., N) \end{aligned} \tag{4}$$

In Figure 1, each cell corresponds to an attention weight $z^{l,h}_{i,j}$. The color density represents how much attention they give to each other. It needs to be noted that these weights are not symmetric, $a^{i,j} \neq a^{j,i}$.

We refer to this vector $a^{i,j}$ as the pair-wise attention vector. Figure 1 illustrates the procedure how pair-wise attention is formed. In experiments, we conducted a probing study to analyze and visualize the association between pair-wise attention and dependency relationship.

3.2 Fine-Tuning with Pair-Wise Attention

For word relation extraction tasks, we propose to augment the basic BERT fine tuning architecture with pair-wise attention weights. Figure 2 illustrates our procedure. Inspired by the residual net[6], we add a connection from attention layers to the classifier layer as direct input. Our argument is that these weights directly represent the rich relationship between words and they are not fully represented by the hidden vectors in the output layer. To infer the relationship between a pair of tokens t_i, t_j, we feed the last layer hidden representations $T_i, T_j \in \Re^H$ as well as the pair-wise weight vector $a_{i,j}$. into a classification layer over the relation label set. We denote this input vector as $C \in \Re^{2M+L*H*2}$, where M is the hidden size of BERT, H is the number of self-attention heads, and L is the number of layers.

$$C = T_i \circ T_j \circ a^{i,j} \circ a^{j,i} \tag{5}$$

There are various ways to integrate the four parts. Here we use the simplest concatenation without further complicating the classifier architecture and adding more parameters. The only new parameters added during fine-tuning are for the classification layer $W \in \Re^{K*(2M+L*H*2)}$. K is the number of classifier labels. For a sentence with N tokens, each pair of words is processed independently. During training , the classifier label is given. Pairs with no labels are associated with a label *NONE*. During testing, each pair of words is classified into one of the labels.

As Figure 2 illustrates, this fine-tuning procedure is similar to the standard BERT fine-tuning procedure with exceptions that input to the classifier is beyond simple hidden vectors from the final layer of BERT. In our experiment we use the $BERT_{BASE}$ model with $L = 12, H = 768, A = 12$ as the hyper parameters.

3.3 Fine-Tuning for Dependency Paring

We experiment two existing strategies to apply BERT: feature-based and fine-tuning. The feature-based approach uses the BERT pre-trained model to extract features for the classifier. Only the parameters of the classifier are updated during training. The fine-tuning approach tunes the pre-trained BERT model parameters along with the classifier weights during training. Features in our case include BERT embeddings of the involved two words from the final layer as well as our proposed pair-wise attention vectors. For dependency parsing, we use the PTB corpus as our training and testing data. Accuracies are measured using UAS and LAS as we explained in the Introduction Section.

As with other graph-based models, the predicted dependency tree at training time is the one where each word is a dependent of its highest scoring head. At test time, we employ the Minimum Spanning Tree (MST) to construct a well-formed tree from local classification scores.

Comparing to the previous work, we didn't use other widely used features like POS tag embeddings to tune the performance. To focus on our motivation, we only use BERT pair-wise attention weights and embeddings for the classifier.

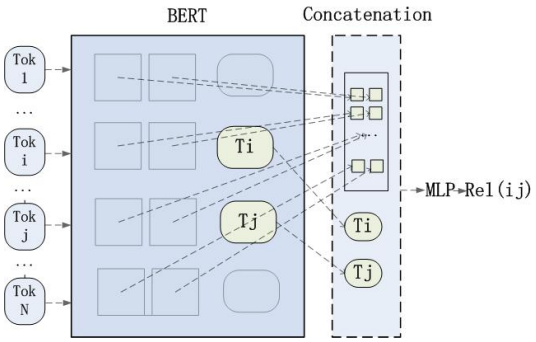

Fig. 2. The process of relation extraction based on Bert

3.4 Fine-Tuning for Semantic Relation Extraction

It is straightforward to apply our proposed model to semantic relation extraction between words. More specifically, we run our experiments on the SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals, one of the most popular relation classification tasks.

Most previous models for relation classification rely on the high-level lexical and syntactic features obtained by NLP tools such as WordNet, dependency parser, POS tagger, and named entity recognizers. We follow the same spirit as the above. We limit ourselves to only BERT features. Experimental results will be given in the Experiments Section.

4 Experiments

We perform three sets of experiments in our study. One is for dependency parsing with different models including our proposed one. Second is our experiments for the semantic relation task. Third, we conduct a probing study to measure and directly visulize the contribution of pair-wise attention weights to dependency tree parsing as well as semantic realtion extraciton.

4.1 Dependency Parsing

For dependency parsing, we use the English Penn Treebank (PTB) data with 42067 sentences for training, 3370 sentences for evaluation and 3761 sentences for testing. We parse all sentences in the dataset with the Stanford parser (v3.6.0)

and split the data into training, develpment and testing in the standard way as [1] configured.

With this default setup, we experiment five different models for dependency parsing.

– Feature-based, pre-trained contextual embeddings(fea:emb): Use BERT to extract fixed contextual word embeddings as features

- Feature-based, pre-trained contextual embeddings, Pair-wise attention(fea:emb+att): Use BERT to extract fixed contextual embeddings and pair-wise attention weights as features
- Feature-based, pre-trained contextual embeddings, Pair-wise attention(fea:ptb:emb+att): The baseline BERT model is further trained with the PTB raw data as a lanague model. The obtained model is then used to extract fixed contextual embeddings and pair-wise attention weights.
- Fine-tuning, contextual embeddings(fine-tune:emb): Fine-tune BERT along with training the classifer. Use contextual embeddings as the only feature
- Fine-tuning, contextual embeddings, Pair-wise attention(fine-tune:emb+att): Fine-tune BERT along with training the classifer using contextual embeddings as well as pair-wise attention weights

Table 1. Accuary of UAS and LAS of on PTB

TASK	**PTB**		**SemiEval-2010**
MODEL	**UAS**	**LAS**	**F1**
fea:emb	62.1	55.5	50.8
fea:emb+att	77.2	68.3	61.1
fea:ptb:emb+att	76.9	67.4	61.1
fine-tune:emb	87.8	86.0	76.5
fine-tune:emb+att	89.3	87.8	**78.4**
fine-tune:emb+att(mst)	**90.9**	**88.9**	$*$

Table 1 presents the dependency classification accuacy with different models. The fine-tuned model with contextual embeddings and pair wise attentions achieve the best accuracy by a large margin. The contribution of pair-wise attention without fine-tuning is 15.1% absolute improvement UAS above the baseline 62.1% and 12.8% improvement on LAS. Fine-tuning dramatically pushes up the performance to 90.9% UAS and 88.9% , which is close to the state-of-the-art performance without additional features[4]. We can conclude that BERT is able to densely represent syntactic information into contextual embeddings as well as transformer attentions. The pair-wise attentions we proposed made a significant contribution. The third model $fea : ptb : emb + att$ tunes the BERT model parameters as a masked language model with the PTB dataset and then use it to extract features. The obtained model leads to accuracy drop nearly 1 point. It is not a successful practice given that the size of PTB dataset is not comparable to BERT original training corpus.

One practical problem we have to face in our experiments is the incompatibility between the WordPiece tokenization BERT uses and PTB tokenization. Such way of tokenization makes the sentence non-grammatical and bings further challenge to dependency parsing. To ease this dilemma, we ignore those words which are not in BERT vocabulary. Here in our experiments, we choose to use our simple approach to ignore the so-called unknown words.

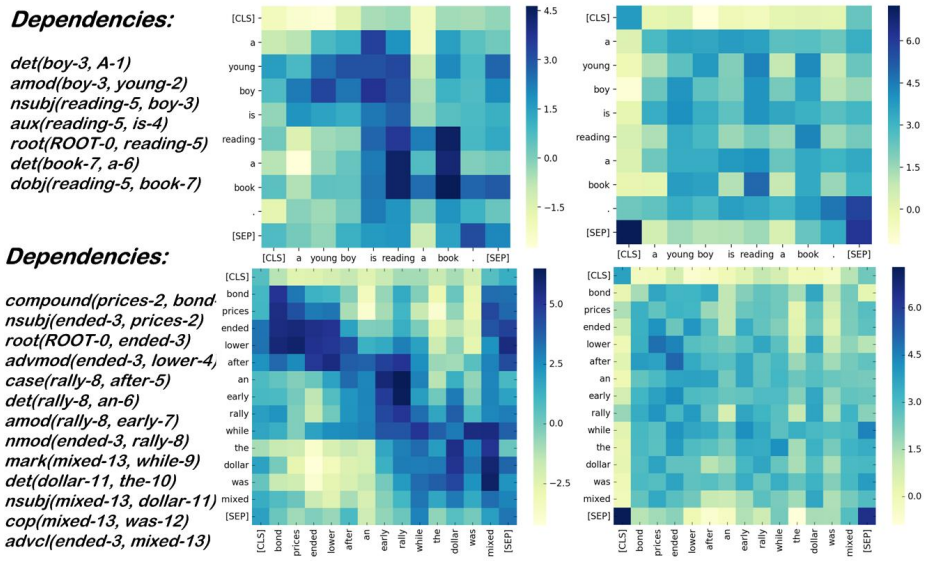

Fig. 3. Attention map of top2 important features of different sentences for dependency relationship extraction

Limited by the BERT vocabulary, the tokenization problem becomes an obstacle for us to take full advantage of the BERT pre-trained model. The best performance we achieve with current experiments is 90.9% UAS with MST, which is nearly 5.7% lower than the latest state-of-the-art(with additional features). In the Discussion section, we share our ongoing work to solve these problems.

4.2 Semantic Realtion Classifcation

We keep the same setup to perform fine-tuning for the task 8 of SemEval-2010. Table 1 provides the results. The fine-tuned model with contextual word embeddings and attention weights achieves the best performance. However, it's still far behind the state-of-the-art model for this task, 9.0% lower. Much richer information including WordNet, Pos Tagging, Entity recognition, etc. is integrated into previous approaches. Without these features, state-of-the-art result has a 6% decrease. Here we limit ourselves to explore a general BERT fine-tuning architecture.

4.3 Probing Study

To further investigate the contribution of pair-wise attention weights, we conduct a probing study. We use a tree-based method to analyze the feature importance. According to our result, the most important 100 features for classification are all pair-wise weights, which support our hypothesis. We visualize several attention matrix to gain direct insights what attention weights represents in terms of word relationship.

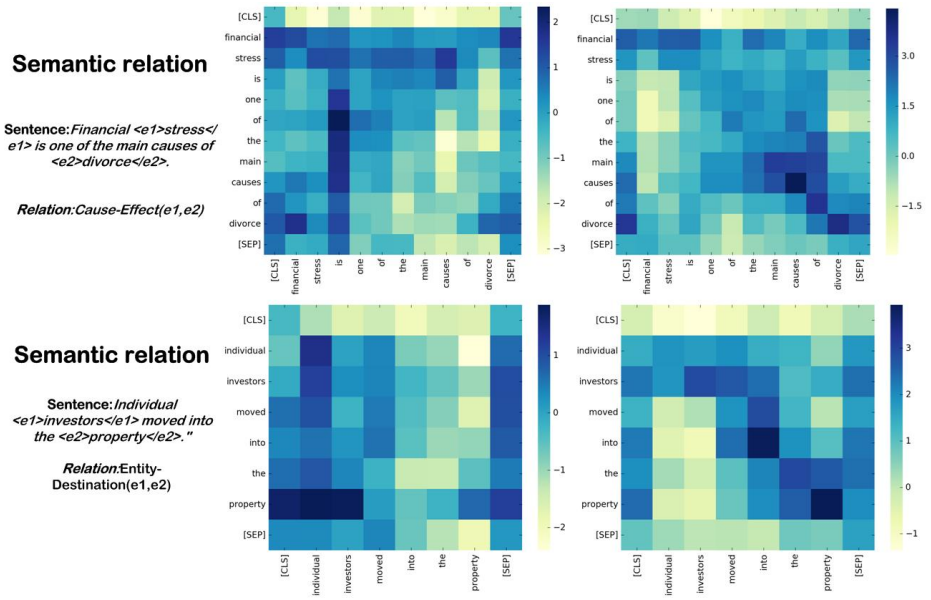

Fig. 4. Attention map of top2 important features of different sentences for nonimal relationship extraction

Figure 3 illustrates the top 2 attention matrices, which are selected for the given sentence according to their contribution to dependency parsing. For instance, the dependency *det(book-7, a-6)* is highlighted with *book* in the x axis and a in the y axis in the first graph. Similar patterns can be observed for difference sentences.

Figure 4 illustrates the top 2 important attention matrices according for the semantic relation task. For both examples, we can observe the first weight matrix graph highlights the connection between entities and the second highlights the type of relation.

5 Conclusion

In this paper, we present a new fine-tuning architecture for word-level relationship extraction. We introduce the pair-wise attention to augment BERT embeddings for dependency parsing and semantic relation extraction. Experimental results prove the effectiveness of our proposal. We also conduct a probing study to visualize how attention weight directly associates with word relation. Work presented in this paper will be open-sources on Github soon once we clean our code.

6 Discussion

In near future, we will strengthen our experiments in this paper in a few aspects. One is to solve the incompatibility between the tokenization the BERT model uses and

the dependency-parsing task, where tokenization causes significant problems. Our way handling it in this paper is coarse. Second, we are in the middle to integrate the biaffine attention mechanisms proposed in [4], which is the state-of-the-art. We believe with these efforts we can possibly advance the state-of-the-art on relation extraction tasks.

In long run, we are interested to combine many tasks into one learning process. The recent success of Transfer Learning [12, 13, 3], evidently helps advance many downstream NLP tasks. In this paper, we extend BERT to fine-tune for relation extraction tasks. While conducting our experiments, we observe the strong reliance among many tasks. Traditional methods traditionally rely on a rich set of features, which often come from other NLP processing tools. The linguistic levels of morphology, syntax, and semantics would benefit each other. Along the past decades, there are many datasets accumulated for various tasks in different contexts. It is ideal if we could jointly learn these tasks and have them benefit each other during training in a mathematically optimal way instead of connecting them as a pipeline.

References

1. Chen, D., Manning, C.: A fast and accurate dependency parser using neural networks. EMNLP pp. 740–750 (01 2014)
2. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. CoRR **abs/1511.01432** (2015)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
4. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. CoRR **abs/1611.01734** (2016)
5. Hashimoto, K., Xiong, C., Tsuruoka, Y., Socher, R.: A joint many-task model: Growing a neural network for multiple NLP tasks. CoRR **abs/1611.01587** (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015)
7. Hendrickx, I., Kim, S., Kozareva, Z., Nakov, P., Pad, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals pp. 33–38 (01 2010)
8. Howard, J., Ruder, S.: Fine-tuned language models for text classification. CoRR **abs/1801.06146** (2018)
9. Kiperwasser, E., Goldberg, Y.: Simple and accurate dependency parsing using bidirectional LSTM feature representations. CoRR **abs/1603.04351** (2016)
10. de Marnee, M.C., Manning, C.: Stanford typed dependencies manual (01 2008)
11. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 523–530. HLT '05, Association for Computational Linguistics, Stroudsburg, PA, USA (2005). https://doi.org/10.3115/1220575.1220641
12. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. CoRR **abs/1802.05365** (2018)
13. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. CoRR **abs/1706.03762** (2017)
15. Wang, L., Cao, Z., de Melo, G., Liu, Z.: Relation classification via multi-level attention cnns. pp. 1298–1307 (01 2016). https://doi.org/10.18653/v1/P16-1123