Relation Classification via Multi-Level Attention CNNs

Linlin Wang[1*], **Zhu Cao**[1*], **Gerard de Melo**[2], **Zhiyuan Liu**[3†]
[1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
[2] Department of Computer Science, Rutgers University, Piscataway, NJ, USA
[3]State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing, China
{ll-wang13,cao-z13}@mails.tsinghua.edu.cn, gdm@demelo.org

Abstract

Relation classification is a crucial ingredient in numerous information extraction systems seeking to mine structured facts from text. We propose a novel convolutional neural network architecture for this task, relying on two levels of attention in order to better discern patterns in heterogeneous contexts. This architecture enables end-to-end learning from task-specific labeled data, forgoing the need for external knowledge such as explicit dependency structures. Experiments show that our model outperforms previous state-of-the-art methods, including those relying on much richer forms of prior knowledge.

1 Introduction

Relation classification is the task of identifying the semantic relation holding between two nominal entities in text. It is a crucial component in natural language processing systems that need to mine explicit facts from text, e.g. for various information extraction applications as well as for question answering and knowledge base completion (Tandon et al., 2011; Chen et al., 2015). For instance, given the example input

"Fizzy [drinks] and meat cause heart disease and [diabetes]."

with annotated target entity mentions e_1 = "drinks" and e_2 = "diabetes", the goal would be to automatically recognize that this sentence expresses a cause-effect relationship between e_1 and e_2, for which we use the notation Cause-Effect(e_1,e_2). Accurate relation classification facilitates precise sentence interpretations, discourse processing, and higher-level NLP tasks (Hendrickx et al., 2010). Thus, relation classification has attracted considerable attention from researchers over the course of the past decades (Zhang, 2004; Qian et al., 2009; Rink and Harabagiu, 2010).

In the example given above, the verb corresponds quite closely to the desired target relation. However, in the wild, we encounter a multitude of different ways of expressing the same kind of relationship. This challenging variability can be lexical, syntactic, or even pragmatic in nature. An effective solution needs to be able to account for useful semantic and syntactic features not only for the meanings of the target entities at the lexical level, but also for their immediate context and for the overall sentence structure.

Thus, it is not surprising that numerous feature- and kernel-based approaches have been proposed, many of which rely on a full-fledged NLP stack, including POS tagging, morphological analysis, dependency parsing, and occasionally semantic analysis, as well as on knowledge resources to capture lexical and semantic features (Kambhatla, 2004; Zhou et al., 2005; Suchanek et al., 2006; Qian et al., 2008; Mooney and Bunescu, 2005; Bunescu and Mooney, 2005). In recent years, we have seen a move towards deep architectures that are capable of learning relevant representations and features without extensive manual feature engineering or use of external resources. A number of convolutional neural network (CNN), recurrent neural network (RNN), and other neural architectures have been proposed for relation classification (Zeng et al., 2014; dos Santos et al., 2015; Xu et al., 2015b). Still, these models often fail to identify critical cues, and many of them still require an external dependency parser.

We propose a novel CNN architecture that addresses some of the shortcomings of previous approaches. Our key contributions are as follows:

1. Our CNN architecture relies on a novel multi-

[*] Equal contribution.
[†] Corresponding author. Email: liuzy@tsinghua.edu.cn

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1298–1307,
Berlin, Germany, August 7-12, 2016.

level attention mechanism to capture both entity-specific attention (primary attention at the input level, with respect to the target entities) and relation-specific pooling attention (secondary attention with respect to the target relations). This allows it to detect more subtle cues despite the heterogeneous structure of the input sentences, enabling it to automatically learn which parts are relevant for a given classification.

2. We introduce a novel pair-wise margin-based objective function that proves superior to standard loss functions.
3. We obtain the new state-of-the-art results for relation classification with an F1 score of 88.0% on the SemEval 2010 Task 8 dataset, outperforming methods relying on significantly richer prior knowledge.

2 Related Work

Apart from a few unsupervised clustering methods (Hasegawa et al., 2004; Chen et al., 2005), the majority of work on relation classification has been supervised, typically cast as a standard multi-class or multi-label classification task. Traditional feature-based methods rely on a set of features computed from the output of an explicit linguistic preprocessing step (Kambhatla, 2004; Zhou et al., 2005; Boschee et al., 2005; Suchanek et al., 2006; Chan and Roth, 2010; Nguyen and Grishman, 2014), while kernel-based methods make use of convolution tree kernels (Qian et al., 2008), subsequence kernels (Mooney and Bunescu, 2005), or dependency tree kernels (Bunescu and Mooney, 2005). These methods thus all depend either on carefully handcrafted features, often chosen on a trial-and-error basis, or on elaborately designed kernels, which in turn are often derived from other pre-trained NLP tools or lexical and semantic resources. Although such approaches can benefit from the external NLP tools to discover the discrete structure of a sentence, syntactic parsing is error-prone and relying on its success may also impede performance (Bach and Badaskar, 2007). Further downsides include their limited lexical generalization abilities for unseen words and their lack of robustness when applied to new domains, genres, or languages.

In recent years, deep neural networks have shown promising results. The Recursive Matrix-Vector Model (MV-RNN) by Socher et al. (2012) sought to capture the compositional aspects of the sentence semantics by exploiting syntactic trees. Zeng et al. (2014) proposed a deep convolutional neural network with softmax classification, extracting lexical and sentence level features. However, these approaches still depend on additional features from lexical resources and NLP toolkits. Yu et al. (2014) proposed the Factor-based Compositional Embedding Model, which uses syntactic dependency trees together with sentence-level embeddings. In addition to dos Santos et al. (2015), who proposed the Ranking CNN (CR-CNN) model with a class embedding matrix, Miwa and Bansal (2016) similarly observed that LSTM-based RNNs are outperformed by models using CNNs, due to limited linguistic structure captured in the network architecture. Some more elaborate variants have been proposed to address this, including bidirectional LSTMs (Zhang et al., 2015), deep recurrent neural networks (Xu et al., 2016), and bidirectional tree-structured LSTM-RNNs (Miwa and Bansal, 2016). Several recent works also reintroduce a dependency tree-based design, e.g., RNNs operating on syntactic trees (Hashimoto et al., 2013), shortest dependency path-based CNNs (Xu et al., 2015a), and the SDP-LSTM model (Xu et al., 2015b). Finally, Nguyen and Grishman (2015) train both CNNs and RNNs and variously aggregate their outputs using voting, stacking, or log-linear modeling (Nguyen and Grishman, 2015). Although these recent models achieve solid results, ideally, we would want a simple yet effective architecture that does not require dependency parsing or training multiple models. Our experiments in Section 4 demonstrate that we can indeed achieve this, while also obtaining substantial improvements in terms of the obtained F1 scores.

3 The Proposed Model

Given a sentence S with a labeled pair of entity mentions e_1 and e_2 (as in our example from Section 1), relation classification is the task of identifying the semantic relation holding between e_1 and e_2 among a set of candidate relation types (Hendrickx et al., 2010). Since the only input is a raw sentence with two marked mentions, it is non-trivial to obtain all the lexical, semantic and syntactic cues necessary to make an accurate prediction.

To this end, we propose a novel multi-level attention-based convolution neural network model. A schematic overview of our architecture is given

Notation	Definition	Notation	Definition
$\mathbf{w}_i^{\mathrm{M}}$	Final word emb.	$\mathbf{z}_i$	Context emb.
W_f	Conv. weight	B_f	Conv. bias
$\mathbf{w}^{\mathrm{O}}$	Network output	W^{L}	Relation emb.
A^j	Input att.	A^{p}	Pooling att.
G	Correlation matrix		

Table 1: Overview of main notation.

in Figure 1. The input sentence is first encoded using word vector representations, exploiting the context and a positional encoding to better capture the word order. A primary attention mechanism, based on diagonal matrices is used to capture the relevance of words with respect to the target entities. To the resulting output matrix, one then applies a convolution operation in order to capture contextual information such as relevant n-grams, followed by max-pooling. A secondary attention pooling layer is used to determine the most useful convolved features for relation classification from the output based on an attention pooling matrix. The remainder of this section will provide further details about this architecture. Table 1 provides an overview of the notation we will use for this. The final output is given by a new objective function, described below.

3.1 Classification Objective

We begin with top-down design considerations for the relation classification architecture. For a given sentence S, our network will ultimately output some $\mathbf{w}^{\mathrm{O}}$. For every output relation $y \in \mathcal{Y}$, we assume there is a corresponding output embedding W_y^{L}, which will automatically be learnt by the network (dos Santos et al., 2015).

We propose a novel distance function $\delta_\theta(S)$ that measures the proximity of the predicted network output $\mathbf{w}^{\mathrm{O}}$ to a candidate relation y as follows.

$$\delta_\theta(S, y) = \left\| \frac{\mathbf{w}^{\mathrm{O}}}{|\mathbf{w}^{\mathrm{O}}|} - W_y^{\mathrm{L}} \right\| \quad (1)$$

using the L_2 norm (note that W_y^{L} are already normalized). Based on this distance function, we design a margin-based pairwise loss function $\mathcal{L}$ as

$$\begin{aligned}\mathcal{L} &= \left[\delta_\theta(S, y) + (1 - \delta_\theta(S, \hat{y}^-))\right] + \beta\|\theta\|^2 \\ &= \left[1 + \left\| \frac{\mathbf{w}^{\mathrm{O}}}{|\mathbf{w}^{\mathrm{O}}|} - W_y^{L} \right\| - \left\| \frac{\mathbf{w}^{\mathrm{O}}}{|\mathbf{w}^{\mathrm{O}}|} - W_{\hat{y}^-}^{\mathrm{L}} \right\|\right] \\ &\quad + \beta\|\theta\|^2, \end{aligned} \quad (2)$$

where 1 is the margin, β is a parameter, $\delta_\theta(S, y)$ is the distance between the predicted label embedding W^{L} and the ground truth label y and $\delta_\theta(S, \hat{y}^-)$ refers to the distance between $\mathbf{w}^{\mathrm{O}}$ and a selected incorrect relation label $\hat{y}^-$. The latter is chosen as the one with the highest score among all incorrect classes (Weston et al., 2011; dos Santos et al., 2015), i.e.

$$\hat{y}^- = \underset{y' \in \mathcal{Y}, y' \neq y}{\operatorname{argmax}} \ \delta(S, y'). \quad (3)$$

This margin-based objective has the advantage of a strong interpretability and effectiveness compared with empirical loss functions such as the ranking loss function in the CR-CNN approach by dos Santos et al. (2015). Based on a distance function motived by word analogies (Mikolov et al., 2013b), we minimize the gap between predicted outputs and ground-truth labels, while maximizing the distance with the selected incorrect class. By minimizing this pairwise loss function iteratively (see Section 3.5), $\delta_\theta(S, y)$ are encouraged to decrease, while $\delta_\theta(S, \hat{y}^-)$ increase.

3.2 Input Representation

Given a sentence $S = (w_1, w_2, ..., w_n)$ with marked entity mentions e_1(=w_p) and e_2(=w_t), $(p, t \in [1, n],\ p \neq t)$, we first transform every word into a real-valued vector to provide lexical-semantic features. Given a word embedding matrix W_V of dimensionality $d_{\mathrm{w}} \times |V|$, where V is the input vocabulary and d_{w} is the word vector dimensionality (a hyper-parameter), we map every w_i to a column vector $\mathbf{w}_i^{\mathrm{d}} \in \mathbb{R}^{d_{\mathrm{w}}}$.

To additionally capture information about the relationship to the target entities, we incorporate word position embeddings (WPE) to reflect the relative distances between the i-th word to the two marked entity mentions (Zeng et al., 2014; dos Santos et al., 2015). For the given sentence in Fig. 1, the relative distances of word "and" to entity e_1 "drinks" and e_2 "diabetes" are -1 and 6, respectively. Every relative distance is mapped to a randomly initialized position vector in $\mathbb{R}^{d_{\mathrm{p}}}$, where d_{p} is a hyper-parameter. For a given word i, we obtain two position vectors $\mathbf{w}_{i,1}^{\mathrm{p}}$ and $\mathbf{w}_{i,2}^{\mathrm{p}}$, with regard to entities e_1 and e_2, respectively. The overall word embedding for the i-th word is $\mathbf{w}_i^{\mathrm{M}} = [(\mathbf{w}_i^{\mathrm{d}})^\intercal, (\mathbf{w}_{i,1}^{\mathrm{p}})^\intercal, (\mathbf{w}_{i,2}^{\mathrm{p}})^\intercal]^\intercal$.

Using a sliding window of size k centered around the i-th word, we encode k successive

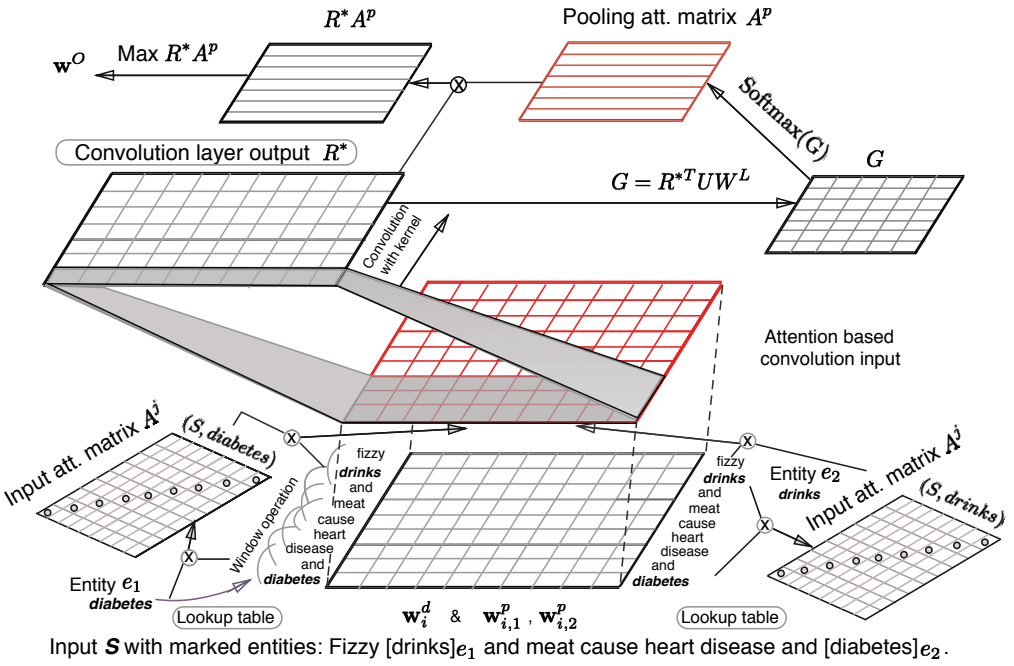

Figure 1: Schematic overview of our Multi-Level Attention Convolutional Neural Networks

words into a vector $\mathbf{z}_i \in \mathbb{R}^{(d_\mathrm{w}+2d_\mathrm{p})k}$ to incorporate contextual information as

$$\mathbf{z}_i = [(\mathbf{w}^{\mathrm{M}}_{i-(k-1)/2})^\intercal, ..., (\mathbf{w}^{\mathrm{M}}_{i+(k-1)/2})^\intercal]^\intercal \quad (4)$$

An extra padding token is repeated multiple times for well-definedness at the beginning and end of the input.

3.3 Input Attention Mechanism

While position-based encodings are useful, we conjecture that they do not suffice to fully capture the relationships of specific words with the target entities and the influence that they may bear on the target relations of interest. We design our model so as to automatically identify the parts of the input sentence that are relevant for relation classification.

Attention mechanisms have successfully been applied to sequence-to-sequence learning tasks such as machine translation (Bahdanau et al., 2015; Meng et al., 2015) and abstractive sentence summarization (Rush et al., 2015), as well as to tasks such as modeling sentence pairs (Yin et al., 2015) and question answering (Santos et al., 2016). To date, these mechanisms have generally been used to allow for an alignment of the input and output sequence, e.g. the source and target sentence in machine translation, or for an alignment between two input sentences as in sentence similarity scoring and question answering.

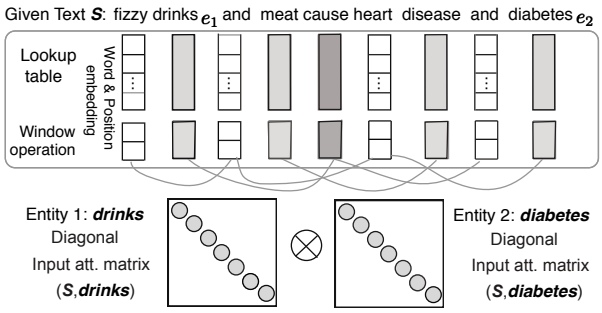

Figure 2: Input and Primary Attention

In our work, we apply the idea of modeling attention to a rather different kind of scenario involving heterogeneous objects, namely a sentence and two entities. With this, we seek to give our model the capability to determine which parts of the sentence are most influential with respect to the two entities of interest. Consider that in a long sentence with multiple clauses, perhaps only a single verb or noun might stand in a relevant relationship with a given target entity.

As depicted in Fig. 2, the input representation layer is used in conjunction with diagonal attention matrices and convolutional input composition.

Contextual Relevance Matrices. Consider the example in Fig. 1, where the non-entity word "cause" is of particular significance in determining the relation. Fortunately, we can exploit the fact

that there is a salient connection between the words "cause" and "diabetes" also in terms of corpus co-occurrences. We introduce two diagonal attention matrices A^j with values $A^j_{i,i} = f(e_j, w_i)$ to characterize the strength of contextual correlations and connections between entity mention e_j and word w_i. The scoring function f is computed as the inner product between the respective embeddings of word w_i and entity e_j, and is parametrized into the network and updated during the training process. Given the A^j matrices, we define

$$\alpha^j_i = \frac{\exp(A^j_{i,i})}{\sum_{i'=1}^{n} \exp(A^j_{i',i'})}, \tag{5}$$

to quantify the relative degree of relevance of the i-th word with respect to the j-th entity ($j \in \{1, 2\}$).

Input Attention Composition. Next, we take the two relevance factors α^1_i and α^2_i and model their joint impact for recognizing the relation via simple averaging as

$$\mathbf{r}_i = \mathbf{z}_i \frac{\alpha^1_i + \alpha^2_i}{2}. \tag{6}$$

Apart from this default choice, we also evaluate two additional variants. The first (Variant-1) concatenates the word vectors as

$$\mathbf{r}_i = [(\mathbf{z}_i\, \alpha^1_i)^\intercal, (\mathbf{z}_i\, \alpha^2_i)^\intercal]^\intercal, \tag{7}$$

to obtain an information-enriched input attention component for this specific word, which contains the relation relevance to both entity 1 and entity 2.

The second variant (Variant-2) interprets relations as mappings between two entities, and combines the two entity-specific weights as

$$\mathbf{r}_i = \mathbf{z}_i \frac{\alpha^1_i - \alpha^2_i}{2}, \tag{8}$$

to capture the relation between them.

Based on these $\mathbf{r}_i$, the final output of the input attention component is the matrix $R = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n]$, where n is the sentence length.

3.4 Convolutional Max-Pooling with Secondary Attention

After this operation, we apply convolutional max-pooling with another secondary attention model to extract more abstract higher-level features from the previous layer's output matrix R.

Convolution Layer. A convolutional layer may, for instance, learn to recognize short phrases such as trigrams. Given our newly generated input attention-based representation R, we accordingly apply a filter of size d^{c} as a weight matrix W_{f} of size $d^{\mathrm{c}} \times k(d^{\mathrm{w}} + 2d^{\mathrm{p}})$. Then we add a linear bias B_{f}, followed by a non-linear hyperbolic tangent transformation to represent features as follows:

$$R^* = \tanh(W_{\mathrm{f}} R + B_{\mathrm{f}}). \tag{9}$$

Attention-Based Pooling. Instead of regular pooling, we rely on an attention-based pooling strategy to determine the importance of individual windows in R^*, as encoded by the convolutional kernel. Some of these windows could represent meaningful n-grams in the input. The goal here is to select those parts of R^* that are relevant with respect to our objective from Section 3.1, which essentially calls for a relation encoding process, while neglecting sentence parts that are irrelevant for this process.

We proceed by first creating a correlation modeling matrix G that captures pertinent connections between the convolved context windows from the sentence and the relation class embedding W^{L} introduced earlier in Section 3.1:

$$G = R^{*\intercal}\, U\, W^{\mathrm{L}}, \tag{10}$$

where U is a weighting matrix learnt by the network.

Then we adopt a softmax function to deal with this correlation modeling matrix G to obtain an attention pooling matrix A^{p} as

$$A^{\mathrm{p}}_{i,j} = \frac{\exp(G_{i,j})}{\sum_{i'=1}^{n} \exp(G_{i',j})}, \tag{11}$$

where $G_{i,j}$ is the (i,j)-th entry of G and $A^{\mathrm{p}}_{i,j}$ is the (i,j)-th entry of A^{p}.

Finally, we multiply this attention pooling matrix with the convolved output R^* to highlight important individual phrase-level components, and apply a max operation to select the most salient one (Yin et al., 2015; Santos et al., 2016) for a given dimension of the output. More precisely, we obtain the output representation $\mathbf{w}^{\mathrm{O}}$ as follows in Eq. (12):

$$\mathbf{w}^{\mathrm{O}}_i = \max_j (R^* A^{\mathrm{p}})_{i,j}, \tag{12}$$

where $\mathbf{w}^{\mathrm{O}}_i$ is the i-th entry of $\mathbf{w}^{\mathrm{O}}$ and $(R^* A^{\mathrm{p}})_{i,j}$ is the (i,j)-th entry of $R^* A^{\mathrm{p}}$.

3.5 Training Procedure

We rely on stochastic gradient descent (SGD) to update the parameters with respect to the loss function in Eq. (2) as follows:

$$\theta' = \theta + \lambda \frac{d(\sum_{i=1}^{|\mathcal{S}|} [\delta_\theta(S_i, y) + (1 - \delta_\theta(S_i, \hat{y}_i^-))])}{d\theta} + \lambda_1 \frac{d(\beta||\theta||^2)}{d\theta} \quad (13)$$

where λ and λ_1 are learning rates, and incorporating the β parameter from Eq. (2).

4 Experiments

4.1 Experimental Setup

Dataset and Metric. We conduct our experiments on the commonly used SemEval-2010 Task 8 dataset (Hendrickx et al., 2010), which contains 10,717 sentences for nine types of annotated relations, together with an additional "Other" type. The nine types are: Cause-Effect, Component-Whole, Content-Container, Entity-Destination, Entity-Origin, Instrument-Agency, Member-Collection, Message-Topic, and Product-Producer, while the relation type "Other" indicates that the relation expressed in the sentence is not among the nine types. However, for each of the aforementioned relation types, the two entities can also appear in inverse order, which implies that the sentence needs to be regarded as expressing a different relation, namely the respective inverse one. For example, Cause-Effect(e_1,e_2) and Cause-Effect(e_2,e_1) can be considered two distinct relations, so the total number $|\mathcal{Y}|$ of relation types is 19. The SemEval-2010 Task 8 dataset consists of a training set of 8,000 examples, and a test set with the remaining examples. We evaluate the models using the official scorer in terms of the Macro-F1 score over the nine relation pairs (excluding Other).

Settings. We use the word2vec skip-gram model (Mikolov et al., 2013a) to learn initial word representations on Wikipedia. Other matrices are initialized with random values following a Gaussian distribution. We apply a cross-validation procedure on the training data to select suitable hyperparameters. The choices generated by this process are given in Table 2.

Parameter	Parameter Name	Value
d^{p}	Word Pos. Emb. Size	25
d^{c}	Conv. Size	1000
k	Word Window Size	3
λ	Learning rate	0.03
λ_1	Learning rate	0.0001

Table 2: Hyperparameters.

4.2 Experimental Results

Table 3 provides a detailed comparison of our Multi-Level Attention CNN model with previous approaches. We observe that our novel attention-based architecture achieves new state-of-the-art results on this relation classification dataset. Att-Input-CNN relies only on the primal attention at the input level, performing standard max-pooling after the convolution layer to generate the network output $\mathbf{w}^{\mathrm{O}}$, in which the new objective function is utilized. With Att-Input-CNN, we achieve an F1-score of 87.5%, thus already outperforming not only the original winner of the SemEval task, an SVM-based approach (82.2%), but also the well-known CR-CNN model (84.1%) with a relative improvement of 4.04%, and the newly released DRNNs (85.8%) with a relative improvement of 2.0%, although the latter approach depends on the Stanford parser to obtain dependency parse information. Our full dual attention model Att-Pooling-CNN achieves an even more favorable F1-score of 88%.

Table 4 provides the experimental results for the two variants of the model given by Eqs. (7) and (8) in Section 3.3. Our main model outperforms the other variants on this dataset, although the variants may still prove useful when applied to other tasks. To better quantify the contribution of the different components of our model, we also conduct an ablation study evaluating several simplified models. The first simplification is to use our model without the input attention mechanism but with the pooling attention layer. The second removes both attention mechanisms. The third removes both forms of attention and additionally uses a regular objective function based on the inner product $s = r \cdot w$ for a sentence representation r and relation class embedding w. We observe that all three of our components lead to noticeable improvements over these baselines.

4.3 Detailed Analysis

Primary Attention. To inspect the inner workings of our model, we considered the primary attention matrices of our multi-level attention model

Classifier	F1
Manually Engineered Methods	
SVM (Rink and Harabagiu, 2010)	82.2
Dependency Methods	
RNN (Socher et al., 2012)	77.6
MVRNN (Socher et al., 2012)	82.4
FCM (Yu et al., 2014)	83.0
Hybrid FCM (Yu et al., 2014)	83.4
SDP-LSTM (Xu et al., 2015b)	83.7
DRNNs (Xu et al., 2016)	85.8
SPTree (Miwa and Bansal, 2016)	84.5
End-To-End Methods	
CNN+ Softmax (Zeng et al., 2014)	82.7
CR-CNN (dos Santos et al., 2015)	84.1
DepNN (Liu et al., 2015)	83.6
depLCNN+NS (Xu et al., 2015a)	85.6
STACK-FORWARD$*$	83.4
VOTE-BIDIRECT$*$	84.1
VOTE-BACKWARD$*$	84.1
Our Architectures	
Att-Input-CNN	87.5
Att-Pooling-CNN	**88.0**

Table 3: Comparison with results published in the literature, where '$*$' refers to models from Nguyen and Grishman (2015).

for the following randomly selected sentence from the test set:

> The disgusting scene was retaliation against her brother Philip who rents the [room]$_{e_1}$ inside this apartment [house]$_{e_2}$ on Lombard street.

Fig. 3 plots the word-level attention values for the input attention layer to act as an example, using the calculated attention values for every individual word in the sentence. We find the word "inside" was assigned the highest attention value, while words such as "room" and "house" also are deemed important. This appears sensible in light of the ground-truth labeling as a Component-Whole(e_1,e_2) relationship. Additionally, we observe that words such as "this", which are rather irrelevant with respect to the target relationship, indeed have significantly lower attention scores.

Most Significant Features for Relations. Table 5 lists the top-ranked trigrams for each relation class y in terms of their contribution to the score for determining the relation classification. Recall the definition of $\delta_\theta(x, y)$ in Eq. (1). In the network, we trace back the trigram that contributed most to the correct classification in terms of $\delta_\theta(S_i, y)$ for each sentence S_i. We then rank all such trigrams in the sentences in the test set according to their total contribution and list the top-ranked trigrams.[*] In Table 5, we see that these are indeed very informative for deducing the relation. For example, the top trigram for Cause-Effect(e_2,e_1) is "are caused by", which strongly implies that the first entity is an effect caused by the latter. Similarly, the top trigram for Entity-Origin(e_1,e_2) is "from the e_2", which suggests that e_2 could be an original location, at which entity e_1 may have been located.

Classifier	F1
Att-Input-CNN (Main)	87.5
Att-Input-CNN (Variant-1)	87.2
Att-Input-CNN (Variant-2)	87.3
Att-Pooling-CNN (regular)	88.0
– w/o input attention	86.6
– w/o any attention	86.1
– w/o any attention, w/o δ-objective	84.1

Table 4: Comparison between the main model and variants as well as simplified models.

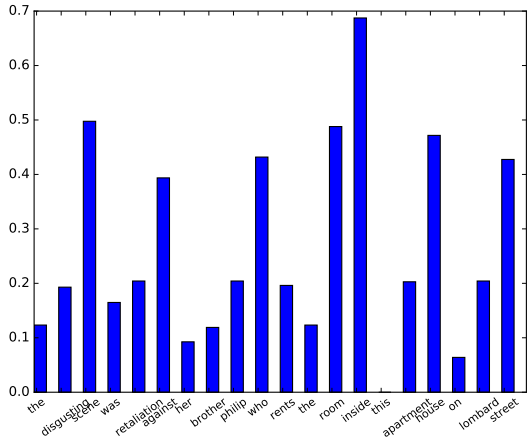

Figure 3: Input Attention Visualization. The value of the y-coordinate is computed as $100 * (A_i^t - \min_{i \in \{1,\ldots,n\}} A_i^t)$, where A_i^t stands for the overall attention weight assigned to the word i.

Error Analysis. Further, we examined some of the misclassifications produced by our model. The following is a typical example of a wrongly classified sentence:

[*]For Entity-Destination(e_2,e_1), there was only one occurrence in the test set.

Relation	(e_1, e_2)	(e_2, e_1)
Cause-Effect	e_1 caused a, caused a e_2, e_1 resulted in, the cause of, had caused the, poverty cause e_2	e_2 caused by, e_2 from e_1, is caused by, are caused by, was caused by, been caused by
Component-Whole	e_1 of the, of a e_2, of the e_2, in the e_2, part of the	with its e_2, e_1 consists of, e_1 has a, e_1 comprises e_2
Content-Container	in a e_2, was hidden in, inside a e_2, was contained in	e_1 with e_2, filled with e_2, e_1 contained a, full of e_2,
Entity-Destination	e_1 into the, e_1 into a, was put inside, in a e_2	had thrown into
Entity-Origin	from this e_2, is derived from, from the e_2, away from the	e_1 e_2 is, the e_1 e_2, for e_1 e_2, the source of
Instrument-Agency	for the e_2, is used by, by a e_2, with the e_2, a e_1 e_2	e_1 use e_2, with a e_2, by using e_2
Member-Collection	of the e_2, in the e_2, a member of, from the e_2	a e_1 of, e_1 of various, e_1 of e_2, the e_1 of
Message-Topic	on the e_2, e_1 asserts the, e_1 points out, e_1 is the	the e_1 of, described in the, the topic for, in the e_2
Product-Producer	e_1 made by, made by e_2, from the e_2, by the e_2	has constructed a, came up with, has drawn up, e_1 who created

Table 5: Most representative trigrams for different relations.

> A [film]$_{e_1}$ revolves around a [cadaver]$_{e_2}$ who seems to bring misfortune on those who come in contact with it.

This sentence is wrongly classified as belonging to the "Other" category, while the ground-truth label is Message-Topic(e_1,e_2). The phrase "revolves around" does not appear in the training data, and moreover is used metaphorically, rather than in its original sense of turning around, making it difficult for the model to recognize the semantic connection.

Another common issue stems from sentences of the form "... e_1 e_2 ...", such as the following ones:

> The size of a [tree]$_{e_1}$ [crown]$_{e_2}$ is strongly ...
> Organic [sesame]$_{e_1}$ [oil]$_{e_2}$ has an ...
> Before heading down the [phone]$_{e_1}$ [operator]$_{e_2}$ career ...

These belong to three different relation classes, Component-Whole(e_2,e_1), Entity-Origin(e_2,e_1), and Instrument-Agency(e_1,e_2), respectively, which are only implicit in the text, and the context is not particularly helpful. More informative word embeddings could conceivably help in such cases.

Convergence. Finally, we examine the convergence behavior of our two main methods. We plot the performance of each iteration in the Att-Input-CNN and Att-Pooling-CNN models in Fig. 4. It can be seen that Att-Input-CNN quite smoothly converges to its final F1 score, while for the Att-Pooling-CNN model, which includes an additional attention layer, the joint effect of these two attention layer induces stronger back-propagation effects. On the one hand, this leads to a seesaw phenomenon in the result curve, but on the other hand it enables us to obtain better-suited models with slightly higher F1 scores.

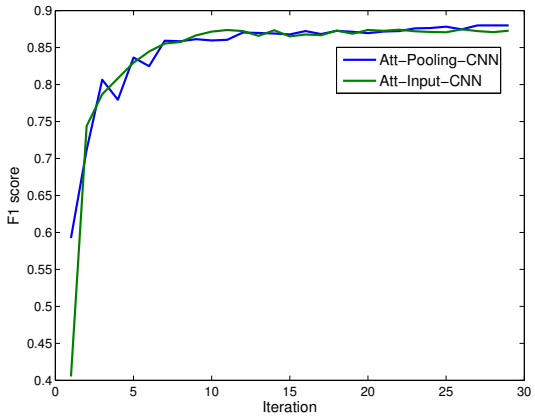

Figure 4: Training Progress of Att-Input-CNN and Att-Pooling-CNN across iterations.

5 Conclusion

We have presented a CNN architecture with a novel objective and a new form of attention mechanism that is applied at two different levels. Our results show that this simple but effective model is able to outperform previous work relying on substantially richer prior knowledge in the form of structured models and NLP resources. We expect this sort of architecture to be of interest also beyond the specific task of relation classification, which we intend to explore in future work.

Acknowledgments

The research at IIIS is supported by China 973 Program Grants 2011CBA00300, 2011CBA00301, and NSFC Grants 61033001, 61361136003, 61550110504. Prof. Liu is supported by the China 973 Program Grant 2014CB340501 and NSFC Grants 61572273 and 61532010.

References

Nguyen Bach and Sameer Badaskar. 2007. A review of relation extraction. *Online at* `http://www.cs.cmu.edu/%7Enbach/papers/A-survey-on-Relation-Extraction.pdf`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Elizabeth Boschee, Ralph Weischedel, and Alex Zamanian. 2005. Automatic information extraction. In *Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, VA*, pages 2–4.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics.

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

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2005. Unsupervised feature selection for relation extraction. In *Proceedings of IJCNLP*.

Jiaqiang Chen, Niket Tandon, and Gerard de Melo. 2015. Neural word representations from large-scale commonsense knowledge. In *Proceedings of WI 2015*.

Cıcero 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*, volume 1, pages 626–634.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, page 415. Association for Computational Linguistics.

Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1372–1376, Seattle, Washington, USA, October. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38. Association for Computational Linguistics.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the 42nd Annual Meeting of the Association for Computation Linguistics*, page 22. Association for Computational Linguistics.

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*, pages 285–290.

Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu. 2015. Encoding source language with convolutional neural network for machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7the International Joint Conference on Natural Language Processing*, pages 20–30.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR Workshop*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Raymond J Mooney and Razvan C Bunescu. 2005. Subsequence kernels for relation extraction. In *Advances in Neural Information Processing Systems*, pages 171–178.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 68–74.

Thien Huu Nguyen and Ralph Grishman. 2015. Combining neural networks and log-linear models to improve relation extraction. *arXiv preprint arXiv:1511.05926*.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1, pages 697–704. Association for Computational Linguistics.

Longhua Qian, Guodong Zhou, Fang Kong, and Qiaoming Zhu. 2009. Semi-supervised learning for semantic relation classification using stratified sampling strategy. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

Bryan Rink and Sanda Harabagiu. 2010. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 256–259. Association for Computational Linguistics.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of EMNLP 2015*.

Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

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*, pages 1201–1211.

Fabian M Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 712–717. ACM.

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2011. Deriving a Web-scale common sense fact database. In *Proceedings of the Twenty-fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*, pages 152–157, Palo Alto, CA, USA. AAAI Press.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of IJCAI*, volume 11, pages 2764–2770.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. *Proceedings of EMNLP 2015*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015b. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (to appear)*.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with data augmentation. *arXiv preprint arXiv:1601.03651*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. ABCNN: attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Mo Yu, Matthew Gormley, and Mark Dredze. 2014. Factor-based compositional embedding models. In *NIPS Workshop on Learning Semantics*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, 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 pages*, pages 73–78.

Zhu Zhang. 2004. Weakly-supervised relation classification for information extraction. In *In Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*.

Guodong Zhou, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 427–434. Association for Computational Linguistics.