Going out on a limb: Joint Extraction of Entity Mentions and Relations without Dependency Trees

Arzoo Katiyar and **Claire Cardie**
Department of Computer Science
Cornell University
Ithaca, NY, 14853, USA
arzoo, cardie@cs.cornell.edu

Abstract

We present a novel attention-based recurrent neural network for joint extraction of entity mentions and relations. We show that attention along with long short term memory (LSTM) network can extract semantic relations between entity mentions without having access to dependency trees. Experiments on Automatic Content Extraction (ACE) corpora show that our model significantly outperforms feature-based joint model by Li and Ji (2014). We also compare our model with an end-to-end tree-based LSTM model (SPTree) by Miwa and Bansal (2016) and show that our model performs within 1% on entity mentions and 2% on relations. Our fine-grained analysis also shows that our model performs significantly better on AGENT-ARTIFACT relations, while SPTree performs better on PHYSICAL and PART-WHOLE relations.

1 Introduction

Extraction of entities and their relations from text belongs to a very well-studied family of structured prediction tasks in NLP. There are several NLP tasks such as fine-grained opinion mining (Choi et al., 2006), semantic role labeling (Gildea and Jurafsky, 2002), etc., which have a similar structure; thus making it an important and a challenging task.

Several methods have been proposed for entity mention and relation extraction at the sentence-level. These can be broadly categorized into – 1) pipeline models that treat the identification of entity mentions (Nadeau and Sekine, 2007) and relation classification (Zhou et al., 2005) as two separate tasks; and 2) joint models, also the more recent, which simultaneously identify the entity mention and relations (Li and Ji, 2014; Miwa and Sasaki, 2014). Joint models have been argued to perform better than the pipeline models as knowledge of the typed relation can increase the confidence of the model on entity extraction and vice versa.

Recurrent networks (RNNs) (Elman, 1990) have recently become very popular for sequence tagging tasks such as entity extraction that involves a set of contiguous tokens. However, their ability to identify relations between *non-adjacent* tokens in a sequence, e.g., the head nouns of two entities, is less explored. For these tasks, RNNs that make use of tree structures have been deemed more suitable. Miwa and Bansal (2016), for example, propose an RNN comprised of a sequence-based long short term memory (LSTM) for entity identification and a separate tree-based dependency LSTM layer for relation classification using shared parameters between the two components. As a result, their model depends critically on access to dependency trees, restricting it to sentence-level extraction and to languages for which (good) dependency parsers exist. Also, their model does not jointly extract entities and relations; they first extract all entities and then perform relation classification on all pairs of entities in a sentence.

In our previous work (Katiyar and Cardie, 2016), we address the same task in an opinion extraction context. Our LSTM-based formulation explicitly encodes distance between the head of entities into opinion relation labels. The output space of our model is quadratic in size of the entity and relation label set and we do not specifically identify the relation type. Unfortunately, adding relation type makes the output label space very sparse, making it difficult for the model to learn.

In this paper, we propose a novel RNN-based model for the joint extraction of entity mentions

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 917–928
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1085

and relations. Unlike other models, our model does not depend on any dependency tree information. Our RNN-based model is a multi-layer bi-directional LSTM over a sequence. We encode the output sequence from left-to-right. At each time step, we use an attention-like model on the previously decoded time steps, to identify the tokens in a specified relation with the current token. We also add an additional layer to our network to encode the output sequence from right-to-left and find significant improvement on the performance of relation identification using bi-directional encoding.

Our model significantly outperforms the feature-based structured perceptron model of Li and Ji (2014), showing improvements on both entity and relation extraction on the ACE05 dataset. In comparison to the dependency tree-based LSTM model of Miwa and Bansal (2016), our model performs within 1% on entities and 2% on relations on ACE05 dataset. We also find that our model performs significantly better than their tree-based model on the AGENT-ARTIFACT relation, while their tree-based model performs better on PHYSICAL and PART-WHOLE relations; the two models perform comparably on all other relation types. The very competitive performance of our non-tree-based model bodes well for relation extraction of non-adjacent entities in low-resource languages that lack good parsers.

In the sections that follow, we describe related work (Section 2); our bi-directional LSTM model with attention (Section 3); the training (Section 4); the experiments on ACE dataset (Section 5); results (Section 6); error analysis (Section 7) and conclusion (Section 8).

2 Related Work

RNNs (Hochreiter and Schmidhuber, 1997) have been recently applied to many sequential modeling and prediction tasks, such as machine translation (Bahdanau et al., 2015; Sutskever et al., 2014), named entity recognition (NER) (Hammerton, 2003), opinion mining (Irsoy and Cardie, 2014). Variants such as adding CRF-like objective on top of LSTMs have been found to produce state-of-the-art results on several sequence prediction NLP tasks (Collobert et al., 2011; Huang et al., 2015; Katiyar and Cardie, 2016). These models assume conditional independence at the output layer whereas the model we propose in this paper does not assume any conditional independence at the output layer, allowing it to model an arbitrary distribution over output sequences.

Relation classification has been widely studied as a stand-alone task, assuming that the arguments of the relations are known in advance. There have been several models proposed including feature-based models (Bunescu and Mooney, 2005; Zelenko et al., 2003) and neural network based models (Socher et al., 2012; dos Santos et al., 2015; Hashimoto et al., 2015; Xu et al., 2015a,b).

For joint-extraction of entities and relations, feature-based structured prediction models (Li and Ji, 2014; Miwa and Sasaki, 2014), joint inference integer linear programming models(Yih and Roth, 2007; Yang and Cardie, 2013), card-pyramid parsing (Kate and Mooney, 2010) and probabilistic graphical models (Yu and Lam, 2010; Singh et al., 2013) have been proposed. In contrast, we propose a neural network model which does not depend on the availability of any features such as part of speech (POS) tags, dependency trees, etc.

Recently, Miwa and Bansal (2016) proposed an end-to-end LSTM based sequence and tree-structured model. They extract entities via a sequence layer and relations between the entities via the shortest path dependency tree network. In this paper, we try to investigate recurrent neural networks with attention for extracting semantic relations between entity mentions without using any dependency parse tree features. We also present the *first* neural network based joint model that can extract entity mentions and relations along with the relation type. In our previous work (Katiyar and Cardie, 2016), as explained earlier, we proposed a LSTM-based model for joint extraction of opinion entities and relations, but no relation types. This model cannot be directly extended to include relation types as the output space becomes sparse making it difficult for the model to learn.

Recent advances in recurrent neural network has seen the application of attention on recurrent neural networks to obtain a representation weighted by the importance of tokens in the sequence model. Such models have been very frequently used in question-answering tasks (for recent examples, see Chen et al. (2016) and Lee et al. (2016)), machine translation (Luong et al., 2015; Bahdanau et al., 2015), and many other NLP applications. Pointer networks (Vinyals et al., 2015), an adaptation of attention models, use these token-level weights as pointers to the input elements.

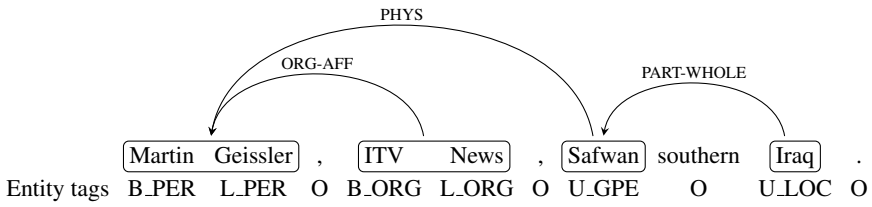

Figure 1: Gold standard annotation for an example sentence from ACE05 dataset.

Zhai et al. (2017), for example, have used these for neural chunking, and Nallapati et al. (2016) and Cheng and Lapata (2016), for summarization. However, to the best of our knowledge, these networks have not been used for joint extraction of entity mentions and relations. We present first such attempt to use these attention models with recurrent neural networks for joint extraction of entity mentions and relations.

3 Model

Our model comprises of a multi-layer bi-directional recurrent network which learns a representation for each token in the sequence. We use the hidden representation from the top layer for joint entity and relation extraction. For each token in the sequence, we output an entity tag and a relation tag. The entity tag corresponds to the entity type, whereas the relation tag is a tuple of pointers to related entities and their respective relation types. Figure 1 shows the annotation for an example sentence from the dataset. We transform the relation tags from entity level to token level. For example, we separately model the relation "ORG-AFF" for each token in the entity "ITV News". Thus, we model the relations between "ITV" and "Martin Geissler", and "News" and "Martin Geissler" separately. We employ a pointer-like network on top of the sequence layer in order to find the relation tag for each token as shown in Figure 2. At each time step, the network utilizes the information available about all output tags from the previous time steps in order to output the entity tag and relation tag *jointly* for the current token.

3.1 Multi-layer Bi-directional Recurrent Network

We use multi-layer bi-directional LSTMs for sequence tagging because LSTMs are more capable of capturing long-term dependencies between tokens, making it ideal for both entity mention and relation extraction.

Using LSTMs, we can compute the hidden state $\overrightarrow{h_t}$ in the forward direction and $\overleftarrow{h_t}$ in the backward direction for every token as below:

$$\overrightarrow{h}_t = LSTM(x_t, \overrightarrow{h}_{t-1})$$
$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1})$$

For every token t in the subsequent layer l, we combine the representations $\overrightarrow{h}_t^{l-1}$ and $\overleftarrow{h}_t^{l-1}$ from previous layer l-1 and feed it as an input. In this paper, we only use the hidden state from the last layer L for output layer and compute the top hidden layer representation as below:

$$z_t^{'} = \overrightarrow{V}\overrightarrow{h}_t^{(L)} + \overleftarrow{V}\overleftarrow{h}_t^{(L)} + c$$

$\overrightarrow{V}$ and $\overleftarrow{V}$ are weight matrices for combining hidden representations from the two directions.

3.2 Entity detection

We formulate entity detection as a sequence labeling task using BILOU scheme similar to Li and Ji (2014) and Miwa and Bansal (2016). We assign each token in the entity with the tag B appended with the entity type if it is the beginning of the entity, I for inside of an entity, L for the end of the entity or U if there is only one token in the entity. Figure 1 shows an example of the entity tag sequence assigned to the sentence. For each token in the sequence, we perform a softmax over all candidate tags to output the most likely tag:

$$y_t = \text{softmax}(Uz_t^{'} + b)$$

Our network structure as shown in Figure 2 also contains connections from the output y_{t-1} of the previous time step to the current top hidden layer. Thus our outputs are not conditionally independent from each other. In order to add connections from y_{t-1}, we transform this output k into a label embedding b_{t-1}^k [1]. We represent each label type

[1] We can also add relation label embeddings using the relation tag output from the previous time step.

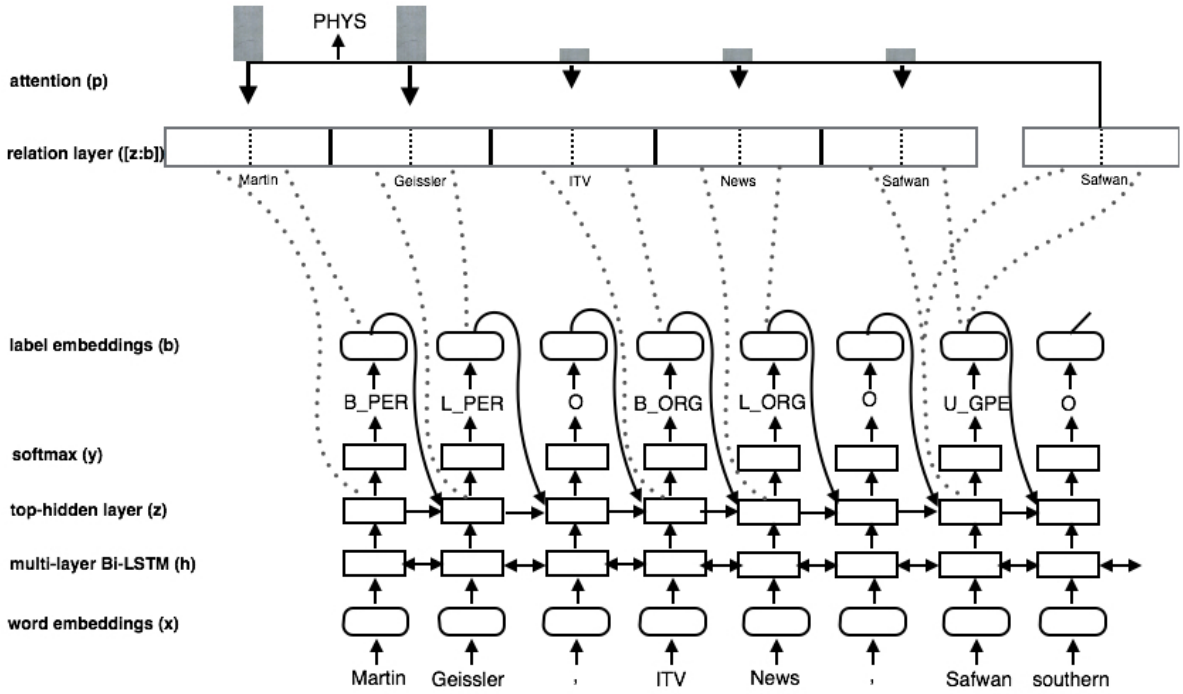

Figure 2: Our network structure based on bi-directional LSTMs for joint entity and relation extraction. This snapshot shows the network when encoding the relation tag for the word "Safwan" in the sentence. The dotted lines in the figure show that top hidden layer and label embeddings for tokens is copied into relation layer. The pointers at attention layer indicate the probability distribution over tokens, the length of the pointers is used to denote the probability value.

k with a dense representation b^k. We compute the output layer representations as:

$$z_t = LSTM([z_t^{'}; b_{t-1}^k], h_{t-1})$$
$$y_t = \text{softmax}(Uz_t + b^{'})$$

We decode the output sequence from left to right in a greedy manner.

3.3 Attention Model

We use attention model for relation extraction. Attention models, over an encoder sequence of representations z, can compute a soft probability distribution p over these learned representations, where d_i is the i^{th} token in decoder sequence. These probabilities are an indication of the importance of different tokens in the encoder sequence:

$$u_t^i = v^T \tanh(W_1 z + W_2 d_i)$$
$$p_t^i = \text{softmax}(u_t^i)$$

v is a weight matrix for attention which transforms the hidden representations into attention scores.

We use pointer networks (Vinyals et al., 2015) in our approach, which are a variation of these attention models. Pointer networks interpret these p_t^i as the probability distribution over the input encoding sequence and use u_t^i as pointers to the input elements. We can use these pointers to encode relation between the current token and the previous predicted tokens, making it fit for relation extraction as explained in Section 3.4.

3.4 Relation detection

We formulate relation extraction also as a sequence labeling task. For each token, we want to find the tokens in the past that the current token is related to along with its relation type. In Figure 1, "Safwan" is related to the tokens "Martin" as well as "Geissler" by the relation type "PHYS". For simplicity, let us assume that there is only one previous token the current token is related to when training, i.e., "Safwan" is related to "Geissler" via PHYS relation. We can extend our approach to output multiple relations as explained in Section 4.

We use pointer networks as described in Sec-

tion 3.3. At each time step, we stack the top hidden layer representations from the previous time steps $z_{\leq t}$[2] and its corresponding label embeddings $b_{\leq t}$. We only stack the top hidden layer representations for the tokens which were predicted as non-O's for previous time steps as shown in Figure 2. Our decoding representation at time t is the concatenation of z_t and b_t. The attention probabilities can now be computed as below:

$$u^t_{\leq t} = v^T \tanh(W_1[z_{\leq t}; b_{\leq t}] + W_2[z_t; b_t])$$
$$p^t_{\leq t} = \text{softmax}(u^t_{\leq t})$$

Thus, $p^t_{\leq t}$ corresponds to the probability of each token, in the sequence so far, being related to the current token at time step t. For the case of NONE relations, the token at t is related to itself.

We also want to find the type of the relations. In order to achieve this, we add an extra dimension to v corresponding to the size of relation types R space. Thus, u^i_t is no longer a score but a R dimensional vector. We then take softmax over this vector of size $O(|z_{\leq t}| \times R)$ to find the most likely tuple of pointer to the related entity and its relation type.

3.5 Bi-directional Encoding

Bi-directional LSTMs have been found to be able to capture context better than plain left-to-right LSTMs, based on their performance on various NLP tasks (Irsoy and Cardie, 2014). Also, Sutskever et al. (2014) found that their performance on machine translation task improved on reversing the input sentences during training. Inspired by these developments, we experiment with bi-directional encoding at the output layer. We add another top hidden layer on Bi-LSTM in Figure 2 which encodes the output sequence from right-to-left. The two encoding share the same multi-layer bi-directional LSTM except for the top hidden layer. Thus, we have two output layers in our network which output the entity tags and relation tags separately. At inference time, we employ heuristics to combine the output from the two directions.

[2]The notation $\leq$ is used to denote the stacking of the representations from the previous time steps. Thus, if z_t is a 2-dimensional matrix then $z_{\leq t}$ will be a 3-dimensional tensor. The size along the first dimension will now correspond to the number of 2-dimensional matrices stacked.

4 Training

We train our network by maximizing the log-probability of the correct entity E and relation R tag sequences *jointly* given the sentence S as below:

$$\begin{aligned}
&\log p(E, R|S, \theta) \\
=&\frac{1}{|S|}\sum_{i \in |S|} \log p(e_i, r_i|e_{<i}, r_{<i}, S, \theta) \\
=&\frac{1}{|S|}\sum_{i \in |S|} \log p(e_i|e_{<i}, r_{<i}) + \log p(r_i|e_{\leq i}, r_{<i})
\end{aligned}$$

Thus, we can decompose our objective into the sum of log-probabilities over entity sequence and relation sequence. We use the gold entity tags while training. As shown in Figure 2, we input the label embedding from the previous time step to the top hidden layer at the current time step along with the other recurrent inputs. During training, we pass the gold label embedding to the next time step which enables better training of our model. However, at test time when the gold label is not available we use the predicted label at previous time step as input to the current step.

At inference time, we can greedily decode the sequence to find the most likely entity $\widehat{E}$ and relation $\widehat{R}$ tag sequences:

$$(\widehat{E}, \widehat{R}) = \underset{E,R}{\text{argmax}}\, p(E, R)$$

Since, we add another top layer to encode tag sequences in the reverse order as explained in Section 3.5, there may be conflicts in the output. We select the positive and more confident label similar to Miwa and Bansal (2016).

Multiple Relations Our approach to relation extraction is different from Miwa and Bansal (2016). Miwa and Bansal (2016) present each pair of entities to their model for relation classification. In our approach, we use pointer networks to identify the related entities. Thus, for our approach described so far if we only compute the argmax on our objective then we limit our model to output only one relation label per token. However, from our analysis of the dataset, an entity may be related to more than one entity in the sentence. Hence, we modify our objective to include multiple relations. In Figure 2, token "Safwan" is related to both tokens "Martin" and "Geissler" of the entity "Martin Geissler", hence we assign probability of 0.5

to both these tokens. This can be easily expanded to include tokens from other related entities, such that we assign equal probability $\frac{1}{N}$ to all tokens[3] depending on the number N of these related tokens.

The log-probability for the entity part remain the same as in our objective discussed in Section 4, however we modify the relation log-probability as below:

$$\sum_{|j:\mathbf{r}'_{\mathbf{i,j}}>0|} \mathbf{r}'_{\mathbf{i,j}} \log p(\mathbf{r_{i,j}}|e_{\leq i}, \mathbf{r_{<i}}, S, \theta)$$

where, $\mathbf{r}'_{\mathbf{i}}$ is the true distribution over relation label space and $\mathbf{r_i}$ is the softmax output from our model. From empirical analysis, we find that $\mathbf{r}'_{\mathbf{i}}$ is generally sparse and hence using a cross entropy objective like this can be useful to find multiple relations. We can also use Sparsemax (Martins and Astudillo, 2016) instead of softmax which is more suitable for sparse distributions. However, we leave it for future work.

At inference time, we output all the labels with probability value above a certain threshold. We adapt this threshold based on the validation set.

5 Experiments

5.1 Data

We evaluate our proposed model on the two datasets from the Automatic Content Extraction (ACE) program – ACE05 and ACE04. There are 7 main entity types namely Person (PER), Organization (ORG), Geographical Entities (GPE), Location (LOC), Facility (FAC), Weapon (WEA) and Vehicle (VEH). For each entity, both entity mentions and its head phrase are annotated. For the scope of this paper, we only use the entity head phrase similar to Li and Ji (2014) and Miwa and Bansal (2016). Also, there are relation types namely Physical (PHYS), Person-Social (PER-SOC), Organization-Affiliation (ORG-AFF), Agent-Artifact (ART), GPE-Affiliation (GPE-AFF).

ACE05 has a total of 6 relation types including PART-WHOLE. We use the same data splits as Li and Ji (2014) and Miwa and Bansal (2016) such that there are 351 documents for training, 80 for development and the remaining 80 documents for the test set.

ACE04 has 7 relation types with an additional Discourse (DISC) type and split ORG-AFF relation type into ORG-AFF and OTHER-AFF. We perform 5-fold cross validation similar to Chan and Roth (2011) for fair comparison with the state-of-the-art.

5.2 Evaluation Metrics

In order to compare our system with the previous systems, we report micro F1-scores, Precision and Recall on both entities and relations similar to Li and Ji (2014) and Miwa and Bansal (2016). An entity is considered correct if we can identify its head and the entity type correctly. A relation is considered correct if we can identify the head of the argument entities and also the relation type. We also report a combined score when both argument entities and relations are correct.

5.3 Baselines and Previous Models

We compare our approach with two previous approaches. The model proposed by Li and Ji (2014) is a feature-based structured perceptron model with efficient beam-search. They employ a segment-based decoder instead of token-based decoding. Their model outperformed previous state-of-the-art pipelined models. Miwa and Sasaki (2014) (SPTree) recently proposed a LSTM-based model with a sequence layer for entity identification, and a tree-based dependency layer which identifies relations between pairs of candidate entities using the shortest dependency path between them. We also employed our previous approach (Katiyar and Cardie, 2016) for extraction of opinion entities and relations to this task. We found that the performance was not competitive with the two approaches mentioned above, performing upto 10 points lower on relations. Hence, we do not include the results in Table 1. Also, Li and Ji (2014) showed that the joint model performs better than the pipelined approaches. Thus, we do not include any pipeline baselines.

5.4 Hyperparameters and Training Details

We train our model using Adadelta (Zeiler, 2012) with gradient clipping. We regularize our network using dropout (Srivastava et al., 2014) with the drop-out rate tuned using development set. We initialized our word embeddings

[3] In this paper, we only identify mention heads and hence the span is limited to a few tokens. We can also include only the last token of the gold entity span in the gold probability distribution.

	Entity			Relation			Entity+Relation		
Method	P	R	F1	P	R	F1	P	R	F1
Li and Ji (2014)	.852	.769	.808	.689	.419	.521	.654	.398	.495
SPTree	.829	.839	.834	–	–	–	.572	.540	.556
SPTree[1]	.823	.839	.831	.605	.553	.578	.578	.529	.553
Our Model	.840	.813	.826	.579	.540	.559	.555	.518	.536

Table 1: Performance on ACE05 test dataset. The dashed ("–") performance numbers were missing in the original paper (Miwa and Bansal, 2016).

[1] We ran the system made publicly available by Miwa and Bansal (2016), on ACE05 dataset for filling in the missing values and comparing our system with theirs at fine-grained level.

	Entity			Relation			Entity+Relation		
Encoding	P	R	F1	P	R	F1	P	R	F1
Left-to-Right	.821	.812	.817	.622	.449	.522	.601	.434	.504
+Multiple Relations	.835	.811	.823	.560	.492	.524	.539	.473	.504
+Bi-directional (Our Model)	.840	.813	.826	.579	.540	.559	.555	.518	.536

Table 2: Performance of different encoding methods on ACE05 dataset.

with 300-dimensional word2vec (Mikolov et al., 2013) word embeddings trained on Google News dataset. We have 3 hidden layers in our network and the dimensionality of the hidden units is 100. All the weights in the network are initialized from small random uniform noise. We tune our hyperparameters based on ACE05 development set and use them for training on ACE04 dataset.

6 Results

Table 1 compares the performance of our system with respect to the baselines on ACE05 dataset. We find that our joint model significantly outperforms the joint structured perceptron model (Li and Ji, 2014) on both entities and relations, despite the unavailability of features such as dependency trees, POS tags, etc. However, if we compare our model to the SPTree models, then we find that their model has better recall on both entities and relations. In Section 7, we perform error analysis to understand the difference in the performance of the two models in detail.

We also compare the performance of various encoding schemes in Table 2. We compare the benefits of introducing multiple relations in our objective and bi-directional encoding compared to left-to-right encoding.

Multiple Relations We find that modifying our objective to include multiple relations improves the recall of our system on relations, leading to slight improvement on the overall performance on relations. However, careful tuning of the threshold may further improve precision.

Bi-directional Encoding By adding bi-directional encoding to our system, we find that we can significantly improve the performance of our system compared to left-to-right encoding. It also improves precision compared to left-to-right decoding combined with multiple relations objective.

We find that for some relations it is easier to detect them with respect to one of the entities in the entity pair. PHYS relation is easier identified with respect to GPE entity than PER entity. Thus, our bi-directional encoding of relations allows us to encode these relations with respect to both entities in the relation.

Table 3 shows the performance of our model on ACE04 dataset. We believe that tuning the hyperparameters of our model can further improve the results on this dataset. As also pointed out by Li and Ji (2014) that ACE05 has better annotation quality, we focused on ACE05 dataset for this work.

7 Error Analysis

In this section, we perform a fine-grained comparison of our model with respect to the SPTree (Miwa and Bansal, 2016) model. We compare the performance of the two models with respect to entities, relation types and the distance between the relation arguments and provide examples from the test set in Table 6.

	Entity			Relation			Entity+Relation		
Method	P	R	F1	P	R	F1	P	R	F1
Li and Ji (2014)	.835	.762	.797	.647	.385	.483	.608	.361	.453
SPTree	.808	.829	.818	–	–	–	.487	.481	.484
Our Model	.812	.781	.796	.502	.488	.493	.464	.453	.457

Table 3: Performance on ACE04 test dataset. The dashed ("–") performance numbers were missing in the original paper (Miwa and Bansal, 2016).

7.1 Entities

We find that our model has lower recall on entity extraction than SPTree as shown in Table 1. Miwa and Bansal (2016), in one of the ablation tests on ACE05 development set, show that their model can gain upto 2% improvement in recall by entity pretraining. Since we propose a joint-model, we cannot directly apply their pretraining trick on entities separately. We leave it for future work. Li and Ji (2014) mentioned in their analysis of the dataset that there were many "UNK" tokens in the test set which were never seen during training. We verified the same and we hypothesize that for this reason the performance on the entities depends largely on the pretrained word embeddings being used. We found considerable improvements on entity recall when using pretrained word embeddings, if available, for these "UNK" tokens. Miwa and Bansal (2016) also use additional features such as POS tags in addition to pretrained word embeddings at the input layer.

Relation Type	Method	R	P	F1
ART	SPTree	.363	.552	.438
	Our model	.431	.611	**.505**
PART-WHOLE	SPTree	.560	.538	**.548**
	Our model	.520	.538	.528
PER-SOC	SPTree	.671	.671	.671
	Our model	.657	.648	.652
PHYS	SPTree	.489	.513	**.500**
	Our model	.388	.426	.406
GEN-AFF	SPTree	.414	.640	.502
	Our model	.484	.516	.500
ORG-AFF	SPTree	.692	.704	.697
	Our model	.706	.700	.703

Table 4: Performance on different relation types in ACE05 test dataset. Numbers in the bracket denote the number of relations of each relation type in the test set.

7.2 Relation Types

We evaluate our model on different relation types and compare the performance with SPTree model in Table 4. Interestingly, we find that the performance of the two models is varied over different relation types. The dependency tree-based model significantly outperforms our joint-model on PHYS and PART-WHOLE relations, whereas our model is significantly better than tree-based model on ART relation. We show an example sentence (S1) in Table 6, where SPTree model identifies the entities in ART relation correctly but fails to identify ART relation. We compare the performance with respect to PHYS relation in Section 7.3.

		Relation		
Distance	Method	R	P	F1
≤ 7	SPTree	.589	.628	.608
	Our model	.591	.605	.598
> 7	SPTree	.275	.375	**.267**
	Our model	.153	.259	.192

Table 5: Performance based on the distance between entity arguments in relations for ACE05 test dataset.

7.3 Distance-based Analysis

We also compare the performance of the two models on relations based on the distance between the entities in a relation in Table 5. We find that the performance of both the models is very low for distance greater than 7. SPTree model can identify 36 relations out of 131 such relations correctly, while our model can only identify 20 relations in this category. We manually compare the output of the two systems on these cases on several examples to understand the gain of using dependency tree on longer distances. Interestingly, the majority of these relations belong to PHYS type, thus resulting in lower performance on PHYS as discussed in Section 7.2. We found that there were a few instances of co-reference errors as shown in S2 in Table 6. Our model identifies a PHYS relation between "here" and "baghdad", whereas the gold annotation has PHYS relation between "location" and "baghdad". We think that

S1 :	the [men]$_{\text{PER:ART-1}}$ held on the sinking [vessel]$_{\text{VEH:ART-1}}$ until the [passenger]$_{\text{PER:ART-2}}$ [ship]$_{\text{VEH:ART-2}}$ was able...
SPTree :	the [men]$_{\text{PER}}$ held on the sinking [vessel]$_{\text{VEH}}$ until the [passenger]$_{\text{PER}}$ [ship]$_{\text{VEH}}$ was able to reach them.
Our Model :	the [men]$_{\text{PER:ART-1}}$ held on the sinking [vessel]$_{\text{VEH:ART-1}}$ until the [passenger]$_{\text{PER:ART-2}}$ [ship]$_{\text{VEH:ART-2}}$ was able...
S2 :	[her]$_{\text{PER}}$ research was conducted [here]$_{\text{FAC}}$ at a [location]$_{\text{FAC:PHYS1}}$ well-known to [u.n.]$_{\text{ORG:ORG-AFF1}}$ [arms]$_{\text{WEA}}$ [inspectors]$_{\text{PER:ORG-AFF1}}$. 300 miles west of [baghdad]$_{\text{GPE:PHYS1}}$.
SPTree :	[her]$_{\text{PER}}$ research was conducted [here]$_{\text{GPE}}$ at a [location]$_{\text{LOC:PHYS1}}$ well-known to u.n. [arms]$_{\text{WEA}}$ [[inspectors]$_{\text{PER:PHYS1,PHY2}}$. 300 miles west of [baghdad]$_{\text{GPE:PHYS2}}$.
Our Model :	[her]$_{\text{PER}}$ research was conducted [here]$_{\text{FAC:PHYS1}}$ at a [location]$_{\text{GPE}}$ well-known to [u.n.]$_{\text{ORG:ORG-AFF1}}$ [arms]$_{\text{WEA}}$ [inspectors]$_{\text{PER:ORG-AFF1}}$. 300 miles west of [baghdad]$_{\text{GPE:PHYS1}}$.
S3 :	... [Abigail Fletcher]$_{\text{PER:PHYS1}}$, a [marcher]$_{\text{FAC:GEN-AFF2}}$ from [Florida]$_{\text{FAC:GEN-AFF2}}$, said outside the [president]$_{\text{PER:ART3}}$'s [[residence]$_{\text{FAC:ART3, PHYS1}}$.
SPTree :	... [Abigail Fletcher]$_{\text{PER:PHYS1}}$, a [marcher]$_{\text{FAC:GEN-AFF2}}$ from [Florida]$_{\text{FAC:GEN-AFF2}}$, said outside the [president]$_{\text{PER:ART3}}$'s [[residence]]$_{\text{FAC:ART3, PHYS1}}$.
Our Model :	... [Abigail Fletcher]$_{\text{PER}}$, a [marcher]$_{\text{FAC:GEN-AFF2}}$ from [Florida]$_{\text{FAC:GEN-AFF2}}$, said outside the [president]$_{\text{PER}}$'s residence.

Table 6: Examples from the dataset with label annotations from SPTree and our model for comparison. The first row for each example is the gold standard.

incorporating these co-reference information during both training and evaluation will further improve the performance of both systems. Another source of error that we found was the inability of our system to extract entities (lower recall) as in S3. Our model could not identify the FAC entity "residence". Hence, we think an improvement on entity performance via methods like pretraining might be helpful in identifying more relations. For distance less than 7, we find that our model has better recall but lower precision, as expected.

8 Conclusion

In this paper, we propose a novel attention-based LSTM model for joint extraction of entity mentions and relations. Experimentally, we found that our model significantly outperforms feature-rich structured perceptron joint model by Li and Ji (2014). We also compare our model to an end-to-end LSTM model by Miwa and Bansal (2016) which comprises of a sequence layer for entity extraction and a tree-based dependency layer for relation classification. We find that our model, without access to dependency trees, POS tags, etc performs within 1% on entities and 2% on relations on ACE05 dataset. We also find that our model performs significantly better than their tree-based model on the ART relation, while their tree-based model performs better on PHYS and PART-WHOLE relations; the two models perform comparably on all other relation types.

In future, we plan to explore pretraining methods for our model which were shown to improve recall on entity and relation performance by Miwa and Bansal (2016). We introduce bi-directional output encoding as well as an objective to learn multiple relations in this paper. However, this presents the challenge of combining predictions from the two directions. We use heuristics in this paper to combine the predictions. We think that using probabilistic methods to combine model predictions from both directions may further improve the performance. We also plan to use Sparsemax (Martins and Astudillo, 2016) instead of Softmax for multiple relations, as the former is more suitable for multi-label classification for sparse labels.

It would also be interesting to see the effect of reranking (Collins and Koo, 2005) on our joint model. We also plan to extend the identification of entities to full entity mention span instead of only the head phrase as in Lu and Roth (2015).

Acknowledgments

We thank Qi Li and Makoto Miwa for their help with the dataset and sharing their code for analysis. We also thank Xilun Chen, Xanda Schofield, Yiqing Hua, Vlad Niculae, Tianze Shi and the three anonymous reviewers for their helpful feedback and discussion.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '05, pages 724–731. https://doi.org/10.3115/1220575.1220666.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 551–560. http://dl.acm.org/citation.cfm?id=2002472.2002542.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1223.pdf.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 484–494. http://www.aclweb.org/anthology/P16-1046.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Sydney, Australia, pages 431–439. http://www.aclweb.org/anthology/W/W06/W06-1651.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Comput. Linguist.* 31(1):25–70. https://doi.org/10.1162/0891201053630273.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12:2493–2537. http://dl.acm.org/citation.cfm?id=1953048.2078186.

Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. *CoRR* abs/1504.06580. http://arxiv.org/abs/1504.06580.

Jeffrey L. Elman. 1990. Finding structure in time. *COGNITIVE SCIENCE* 14(2):179–211.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Comput. Linguist.* 28(3):245–288. https://doi.org/10.1162/089120102760275983.

James Hammerton. 2003. Named entity recognition with long short-term memory. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. Association for Computational Linguistics, Stroudsburg, PA, USA, CONLL '03, pages 172–175. https://doi.org/10.3115/1119176.1119202.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2015. Task-oriented learning of word embeddings for semantic relation classification. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Beijing, China, pages 268–278. http://www.aclweb.org/anthology/K15-1027.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR* abs/1508.01991. http://arxiv.org/abs/1508.01991.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 720–728. http://aclweb.org/anthology/D/D14/D14-1080.pdf.

Rohit J. Kate and Raymond J. Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Stroudsburg, PA, USA, CoNLL '10, pages 203–212. http://dl.acm.org/citation.cfm?id=1870568.1870592.

Arzoo Katiyar and Claire Cardie. 2016. Investigating lstms for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1087.pdf.

Kenton Lee, Tom Kwiatkowski, Ankur P. Parikh, and Dipanjan Das. 2016. Learning recurrent span representations for extractive question answering. *CoRR* abs/1611.01436. http://arxiv.org/abs/1611.01436.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*. pages 402–412. http://aclweb.org/anthology/P/P14/P14-1038.pdf.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 857–867. http://aclweb.org/anthology/D15-1102.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Lisbon, Portugal, pages 1412–1421. http://aclweb.org/anthology/D15-1166.

André F. T. Martins and Ramón F. Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. JMLR.org, ICML'16, pages 1614–1623. http://dl.acm.org/citation.cfm?id=3045390.3045561.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pages 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1105–1116. http://www.aclweb.org/anthology/P16-1105.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1858–1869. http://aclweb.org/anthology/D/D14/D14-1200.pdf.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes* 30.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-sequence rnns for text summarization. *CoRR* abs/1602.06023. http://arxiv.org/abs/1602.06023.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*. ACM, New York, NY, USA, AKBC '13, pages 1–6. https://doi.org/10.1145/2509558.2509559.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP-CoNLL '12, pages 1201–1211. http://dl.acm.org/citation.cfm?id=2390948.2391084.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. pages 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. pages 2692–2700. http://papers.nips.cc/paper/5866-pointer-networks.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 536–540. http://aclweb.org/anthology/D15-1062.

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 the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1785–1794. http://aclweb.org/anthology/D15-1206.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers. pages 1640–1649. http://aclweb.org/anthology/P/P13/P13-1161.pdf.

Wen-Tau Yih and D. Roth. 2007. Global inference for entity and relation identification via a linear programming formulation. In L. Getoor and B. Taskar, editors, *An Introduction to Statistical Relational Learning*, MIT Press.

Xiaofeng Yu and Wai Lam. 2010. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pages 1399–1407. http://dl.acm.org/citation.cfm?id=1944566.1944726.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR* abs/1212.5701. http://arxiv.org/abs/1212.5701.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.* 3:1083–1106. http://dl.acm.org/citation.cfm?id=944919.944964.

Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. 2017. Neural models for sequence chunking. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*. pages 3365–3371. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14776.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, Ann Arbor, Michigan, pages 427–434. https://doi.org/10.3115/1219840.1219893.