End-to-end Relation Extraction using Neural Networks and Markov Logic Networks

Sachin Pawar[1,2], **Pushpak Bhattacharyya**[2], **and Girish K. Palshikar**[1]

[1]TCS Research, Tata Consultancy Services, Pune
[2]Indian Institute of Technology Bombay, Mumbai
{sachin7.p,gk.palshikar}@tcs.com
pb@cse.iitb.ac.in

Abstract

End-to-end relation extraction refers to identifying boundaries of entity mentions, entity types of these mentions and appropriate semantic relation for each pair of mentions. Traditionally, separate predictive models were trained for each of these tasks and were used in a "pipeline" fashion where output of one model is fed as input to another. But it was observed that addressing some of these tasks jointly results in better performance. We propose a single, joint neural network based model to carry out all the three tasks of boundary identification, entity type classification and relation type classification. This model is referred to as "All Word Pairs" model (AWP-NN) as it assigns an appropriate label to each word pair in a given sentence for performing end-to-end relation extraction. We also propose to refine output of the AWP-NN model by using inference in Markov Logic Networks (MLN) so that additional domain knowledge can be effectively incorporated. We demonstrate effectiveness of our approach by achieving better end-to-end relation extraction performance than all 4 previous joint modelling approaches, on the standard dataset of ACE 2004.

1 Introduction

The task of relation extraction (RE) deals with identifying whether any pre-defined *semantic relation* holds between a pair of *entity mentions* in the given sentence. Pure relation extraction techniques (Zhou et al., 2005; Jiang and Zhai, 2007; Bunescu and Mooney, 2005; Qian et al., 2008) assume that for a sentence, gold-standard entity mentions (i.e. boundaries as well as types) in it are known. In contrast, end-to-end relation extraction deals with plain sentences without assuming any knowledge of entity mentions in them. The task of end-to-end relation extraction consists of three sub-tasks: i) identifying boundaries of entity mentions, ii) identifying entity types of these mentions and iii) identifying appropriate semantic relation for each pair of mentions. First two sub-tasks correspond to the Entity Detection and Tracking task defined by the the Automatic Content Extraction (ACE) program (Doddington et al., 2004) and the third sub-task corresponds to the Relation Detection and Characterization (RDC) task. ACE standard defined 7 entity types[1]: PER (person), ORG (organization), LOC (location), GPE (geo-political entity), FAC (facility), VEH (vehicle) and WEA (weapon). It also defined 7 coarse level relation types[2]: EMP-ORG (employment), PER-SOC (personal/social), PHYS (physical), GPE-AFF (GPE affiliation), OTHER-AFF (PER/ORG affiliation), ART (agent-artifact) and DISC (discourse).

Traditionally, the three sub-tasks of end-to-end relation extraction are carried out serially in a "pipeline" fashion. In this case, the errors in any sub-task affect subsequent sub-tasks. Another disadvantage of this "pipeline" approach is that it allows only one-way *information flow*, i.e. the knowledge about entities is used for identifying relations but not vice versa. Hence to overcome this problem, several approaches (Roth and Yih, 2004; Roth and Yih, 2002; Singh et al., 2013; Li and Ji, 2014) were proposed which carried out these sub-tasks jointly rather than in "pipeline" manner.

We propose a new approach which combines

[1]www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-edt-v4.2.6.pdf
[2]www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-rdc-v4.3.2.PDF

Entity Mention	Boundaries	Entity Type
His	$(0, 0)$	PER
sister	$(1, 1)$	PER
Mary Jones	$(2, 3)$	PER
United Kingdom	$(7, 8)$	GPE

Table 1: Expected output of end-to-end relation extraction system for entity mentions

Entity Mention Pair	Relation Type
His, sister	PER-SOC
His, Mary Jones	PER-SOC
sister, United Kingdom	PHYS
Mary Jones, United Kingdom	PHYS

Table 2: Expected output of end-to-end relation extraction system for relations

the powers of Neural Networks and Markov Logic Networks to jointly address all the three sub-tasks of end-to-end relation extraction. We design the "All Word Pairs" neural network model (AWP-NN) which reduces solution of these three sub-tasks to predicting an appropriate label for each word pair in a given sentence. End-to-end relation extraction output can then be constructed easily from these labels of word pairs. Moreover, as a separate prediction is made for each word pair, there may be some inconsistencies among the labels. We address this problem by refining the predictions of AWP-NN by using inference in Markov Logic Networks so that some of the inconsistencies in word pair labels can be removed at the sentence level.

The specific contributions of this work are : i) modelling boundary detection problem by introducing a special relation type WEM and ii) a single, joint neural network model for all three sub-tasks of end-to-end relation extraction. The paper is organized as follows. Section 2 provides a detailed problem definition. Section 3 describes our AWP-NN model in detail, followed by Section 4 which describes how the predictions of AWP-NN model are revised using inference in MLNs. Section 5 provides experimental results and analysis. Finally, we conclude in Section 6 with a short note on future work.

2 Problem Definition

Given a sentence as an input, an end-to-end relation extraction system should produce a list of entity mentions within it. For each entity mention, its boundaries and entity type should be identified. Also, for each pair of valid entity mentions, it should decide whether any pre-defined semantic relation holds between them.

Consider the sentence : $\texttt{His}_0\ \texttt{sister}_1\ \texttt{Mary}_2\ \texttt{Jones}_3\ \texttt{went}_4\ \texttt{to}_5\ \texttt{the}_6\ \texttt{United}_7\ \texttt{Kingdom}_8\ \texttt{.}_9$ Here, end-to-end relation extraction should produce the output as shown in the tables 1 and 2.

3 All Word Pairs Model (AWP-NN)

We propose a single, joint model for addressing all three sub-tasks of end-to-end relation extraction : i) identifying boundaries of entity mentions, ii) identifying entity types of these mentions and iii) identifying appropriate semantic relation for each pair of mentions. We refer to this model as AWP-NN, i.e. All Word Pairs model using Neural Networks. Here, annotations of all these three sub-tasks can be represented by assigning an appropriate label to each pair of words. It is not necessary to assign label to all possible word pairs; rather i^{th} word is paired with j^{th} word only when $j \geq i$. AWP-NN model is motivated from the table representation idea proposed by Miwa and Sasaki (2014) but differs significantly from it in following ways:

1. boundary identification is modelled with the help of a special relation type (WEM) instead of BIO (**B**egin, **I**nside, **O**ther) encoding or BILOU (**B**egin, **I**nside, **L**ast, **U**nit, **O**ther) encoding

2. neural network model for prediction of appropriate label for each word pair instead of structured prediction

Labels predicted by the AWP-NN model for each word pair can then be used to construct the end-to-end relation extraction output as described in tables 1 and 2.

Consider the example sentence from Section 2. Table 3 shows true annotations of all word pairs in this sentence as required for training the AWP-NN model. Labels used for these annotations can be grouped into the following 5 logical clusters:

1. PER, ORG, GPE, LOC, FAC, VEH and WEA : Represent entity type of *head word* of an entity mention when both the words in a word pair are the same

2. OTH : Represents words which are not head words of any entity mention and both the words in a word pair are the same

	His	sister	Mary	Jones	went	to	the	United	Kingdom	.
His	PER	PER-SOC	NULL	PER-SOC	NULL	NULL	NULL	NULL	NULL	NULL
sister		PER	NULL	NULL	NULL	NULL	NULL	NULL	PHYS	NULL
Mary			OTH	WEM	NULL	NULL	NULL	NULL	NULL	NULL
Jones				PER	NULL	NULL	NULL	NULL	PHYS	NULL
went					OTH	NULL	NULL	NULL	NULL	NULL
to						OTH	NULL	NULL	NULL	NULL
the							OTH	NULL	NULL	NULL
United								OTH	WEM	NULL
Kingdom									GPE	NULL
.										OTH

Table 3: Annotation of all word pairs as per the AWP-NN model

3. EMP-ORG, PHYS, OTHER-AFF, EMP-ORG-R, PHYS-R, OTHER-AFF-R[3], PER-SOC, GPE-AFF and ART : Represent relation type between *head words* of any two entity mentions

4. NULL : Indicates that no pre-defined semantic relation exists between the words in the word pair

5. WEM (Within Entity Mention) : Indicates that the words in the word pair belong to the same entity mention and one of the word is the *head word* of that mention

3.1 Features for the AWP-NN model

Previous work (Zhou et al., 2005; Jiang and Zhai, 2007; Bunescu and Mooney, 2005; Qian et al., 2008) in relation extraction establishes the importance of both lexical and syntactic features. Hence, we designed features to capture information about word sequences, POS tags and dependency structure. As each word pair constitutes a separate instance for classification, features are of three types: i) features characterizing individual word in a word pair, ii) features characterizing properties of both the words at a time and iii) features based on feedback, i.e. predictions of preceding instances.

3.1.1 Individual word features

These features are generated separately for both the words in a word pair.
1. Word itself and its POS tag
2. Previous word and previous POS tag
3. Next word and next POS tag
4. Parent / Governor of the word in the dependency tree, the corresponding dependency relation type and POS tag of the parent

[3]EMP-ORG-R, PHYS-R and OTHER-AFF-R correspond to relation types EMP-ORG, PHYS and OTHER-AFF in the reverse direction, respectively.

3.1.2 Word pair features

These features are generated for a word pair (say $\langle W_i, W_j \rangle$) as a whole.
1. Words distance (WD): Number of words in the sentence between the words W_i and W_j
2. Tree distance (TD): Number of words on the path leading from W_i to W_j in the sentence's dependency tree
3. Common Ancestor (CA): Lowest common ancestor of the two words in the dependency tree
4. Ancestor Position (AP): It indicates the position of the common ancestor with respect to the two words of a word pair. Different possible positions of the ancestor are - left of W_i, W_i itself, between W_i and W_j, W_j itself and right of W_j.
5. Dependency Path ($DP_1, DP_2, \cdots, DP_K$) : Sequence of dependency relation types (ignoring directions) on the dependency path leading from W_i to W_j in the sentence's dependency tree.

3.1.3 Feedback features

These features are based on predictions of the preceding instances. Unlike other sequence labelling problems such as Named Entity Recognition where each word gets a label and there is natural order / sequence of instances (i.e. words), there is no natural order / sequence of instances (i.e. word pairs) for AWP-NN model. Hence, for each instance we identify its two preceding instances and define two corresponding feedback features (FB_1 and FB_2). Let $\langle W_i, W_j \rangle$ be an instance representing a word pair in a sentence having N words such that $1 \leq i, j \leq N$ and $i \leq j$. There are following two cases for identifying two preceding instances of $\langle W_i, W_j \rangle$:

- If $i = j$ then both the preceding instances are same i.e. $\langle W_{i-1}, W_{i-1} \rangle$. Feedback features: $FB_1 = FB_2 = LabelOf(\langle W_{i-1}, W_{i-1} \rangle)$
- If $i < j$ then the preceding instances are

$\langle W_i, W_i \rangle$ and $\langle W_j, W_j \rangle$. Feedback features: $FB_1 = LabelOf(\langle W_i, W_i \rangle)$ and $FB_2 = LabelOf(\langle W_j, W_j \rangle)$

Label predictions of the preceding instances are then represented using one-hot encoding and used as features. During training, true labels of the preceding instances are used but while decoding, the predicted labels of these instances are used. Hence during decoding, predictions for word pairs of the form $\langle W_i, W_i \rangle$ (diagonal word pairs in the table 3) are obtained first, starting from $i = 1$ to N. Predictions of other word pairs can be obtained later, as predictions of their preceding instances would then be available.

3.2 Architecture of the AWP-NN model

Figure 1 shows various major components in the architecture of the AWP-NN model.

3.2.1 Embedding Layers

Most of the features used by the model are discrete in nature such as words, POS tags, dependency relation types and ancestor position. These discrete features have to be mapped to some numerical representation and *embedding* layers are used for this purpose. We have employed following embedding layers to represent various types of features:

Word embedding layer: It maps each word to a real-valued vector of some fixed dimensions. We initialize this layer with the *pre-trained* 100 dimensional GloVe word vectors[4] learned on Wikipedia corpus. All the different features which are expressed in the form of words (W_1, W_2, NW_1, PW_1, NW_2, PW_2, Pa_1, Pa_2 and CA in the figure 1) share the *same* word embedding layer. During training, the initial embeddings get *fine-tuned* for our supervised classification task.

POS embedding layer: It maps each distinct POS tag to some real-valued vector representation. All the different features which are expressed in the form of POS tags (T_1, T_2, NT_1, PT_1, NT_2, PT_2, PaT_1 and PaT_2 in the figure 1) share the *same* embedding layer.

Dependency relation type embedding layer: It maps each distinct dependency relation type to some real-valued vector representation. Both the features based on dependency types ($DR_1, DR_2, DP_1, \cdots, DP_K$ in the figure 1) also share the *same* embedding layer.

[4]http://nlp.stanford.edu/projects/glove/

AP embedding layer: It maps each distinct ancestor position to some real-valued vector representation.

WD/TD embedding layer: Even though word distance (WD) and tree distance (TD) are numerical features, we used embeddings to represent each distinct value for them as range of values of these features is large. It was observed to be better than directly providing them as inputs to the neural network.

In our experiments, we used 20 dimensions for POS embeddings, 40 for dependency relation type embeddings and 5 dimensions for AP, WD and TD embeddings. Unlike word embeddings these were initialized randomly during training.

3.2.2 Hidden Layers

First hidden layer is divided in 3 parts. First two parts of 60 nodes each are connected to only the features capturing first and second word, respectively. These nodes are expected to capture higher level abstract features of both the words separately. In order to force these two parts to learn similar abstract features, the weights matrix is shared among them. The third part of the first hidden layer consisting of 500 nodes is connected to all the input features except dependency path, i.e. individual word features of two words, word pair features and feedback features. Output of this part is further given as input to the second hidden layer of 250 units. Output of the second hidden layer is fed to the final softmax layer. Also, outputs of the first two parts of the first hidden layer are directly connected to the final softmax layer. As the dependency path is represented as a sequence of dependency relation types, it is fed to a separate LSTM layer. Output of the LSTM layer is directly connected to the final softmax layer. Softmax layer consists of 19 nodes, each representing one of the possible prediction label described earlier.

4 Inference using Markov Logic Networks

Pawar et al. (2016) presented an approach for end-to-end relation extraction which uses Markov Logic Networks (MLN) (Richardson and Domingos, 2006) to obtain *globally consistent* output by combining *local* outputs of individual classifiers. They developed separate classifiers for identifying mention boundaries, predicting entity types and

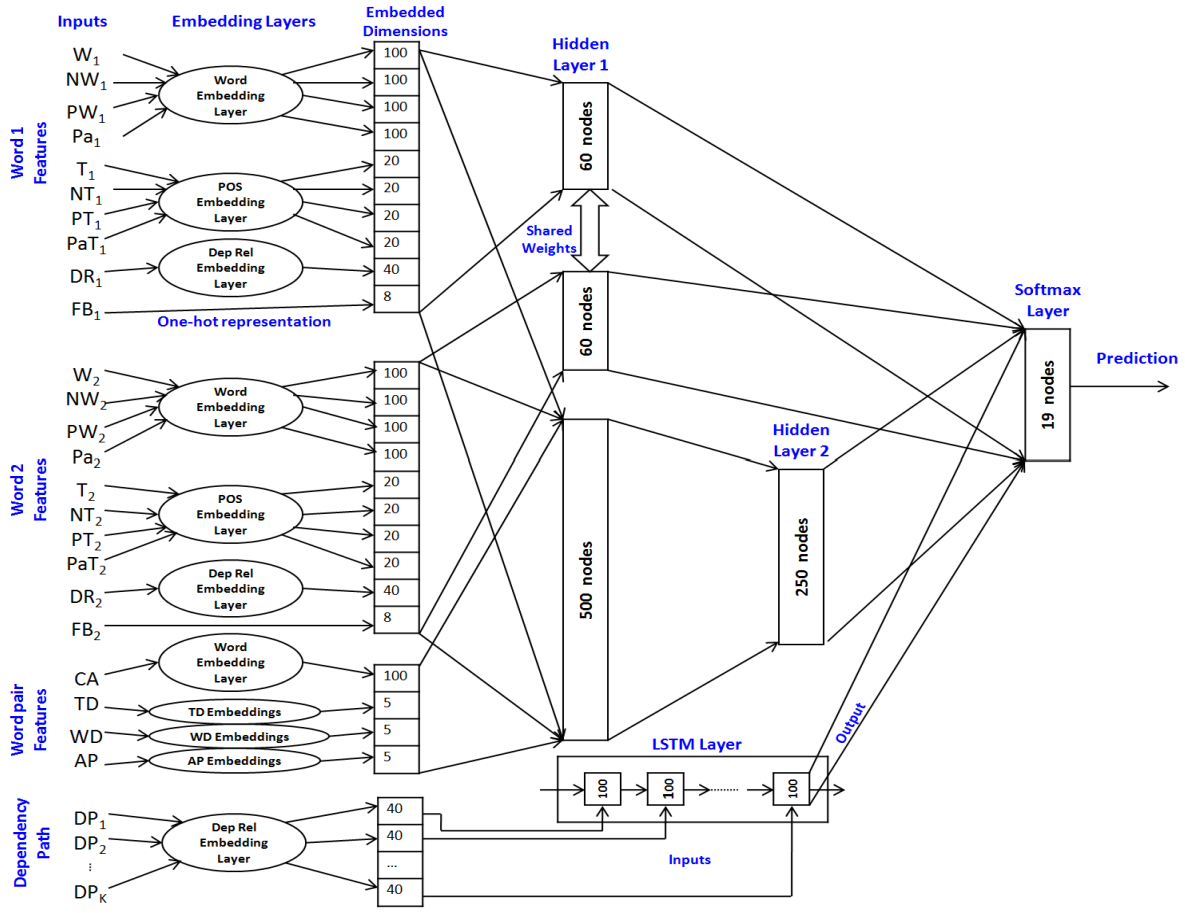

Figure 1: AWP-NN model architecture for predicting appropriate label for the given word pair. (W_1, W_2: words in the word pair; $NW_1, PW_1, NW_2, PW_2, NT_1, PT_1, NT_2, PT_2$: next and previous words/POS tags of W_1 and W_2; Pa_1, DR_1, Pa_2, DR_2: parents and corresponding dependency relation types of W_1 and W_2 in the dependency tree; PaT_1, PaT_2: POS tags of the parents of W_1 and W_2; FB_1, FB_2: Predictions of the preceding instances; CA: Lowest common ancestor of W_1 and W_2 in the dependency tree; TD: Tree distance; WD: Words distance; AP: Ancestor position; $DP_1, DP_2, \cdots, DP_K$: Sequence of dependency relation types on the dependency path leading from W_1 to W_2; Embedding layers for words, POS and dependency relations are shown separately for clarity, but are shared throughout the network.

predicting relation types. Outputs of these classifiers may be inconsistent. E.g., if PER-SOC relation is predicted by the local relation classifier for an entity pair and the local entity classifier predicts entity type as ORG for one of the entity mentions, then there is an inconsistency. Because PER-SOC relation can only exist between two PER entity mentions. Such domain knowledge can be easily incorporated in the form of first-order logic rules in MLN. For each sentence, predictions of individual classifiers are represented in an MLN as first-order logic rules where weights of these rules are proportional to the prediction probabilities. The consistency constraints among the relation types and entity types can be represented in the form of first-order logic rules with infinite weights. Now, the inference in such an MLN generates a globally consistent output with maximum weighted satisfiability of the rules.

AWP-NN is a single joint model which captures boundaries of mentions, their types and relations among them. As the same parameters are shared for all entity as well as relation type predictions, we expect the model to learn dependencies among relation and entity types. However, as it makes separate predictions for each word pair, there might be some inconsistencies among the labels as described above. We adopt the MLN-based approach of Pawar et al. (2016) for handling these inconsistencies and generate a globally consistent output. For this adoption, we consider the AWP-NN predictions for the words pairs where a word

Domains:
Let N be the number of words in the sentence in consideration.

$$word = \{1, 2, \cdots, N\}, etype = \{PER, ORG, LOC, GPE, WEA, FAC, VEH, OTH\}$$
$$rtype = \{EMPORG, GPEAFF, OTHERAFF, PERSOC, PHYS, ART, NULL, WEM\}$$

Evidence Predicates:
$ET(word, etype)$: AWP-NN predictions for word pairs $\langle W_i, W_j \rangle$ where $i = j$
$RT(word, word, rtype)$: AWP-NN predictions for word pairs $\langle W_i, W_j \rangle$ where $i < j$

Query Predicates:
$ETFinal(word, etype)$: Global predictions for word pairs $\langle W_i, W_j \rangle$ where $i = j$
$RTFinal(word, word, rtype)$: Global predictions for word pairs $\langle W_i, W_j \rangle$ where $i < j$

Some examples of *generic rules*:

$$RTFinal(x, y, EMPORG) \wedge ETFinal(x, PER) \Rightarrow (ETFinal(y, ORG) \vee ETFinal(y, GPE)).$$
$$RTFinal(x, y, PERSOC) \Rightarrow (ETFinal(x, PER) \wedge ETFinal(y, PER)).$$

Table 4: Domains and Predicates used for constructing MLN for any given sentence

is paired with itself (diagonal entries in table 3), as *entity* type predictions. Whereas all other word pairs where a word is paired with any subsequent word in the sentence, are considered as *relation* type predictions. Table 4 describes the domains and predicates required for generating an MLN for any given sentence. Unlike Pawar et al. (2016) which considers all predicted mentions in their $entity$ domain, we consider all words in our $word$ domain. But to keep the size of the MLN in check, we keep only those words in the $word$ domain which are part of *interesting* word pairs. A word pair is an *interesting* word pair, if it can potentially represent a relation i.e. if AWP-NN model assigns a probability more than some threshold (say 0.01) for any non-NULL relation type. All the *generic rules* (with infinite weights) described in (Pawar et al., 2016) are used for imposing constraints among the relation and entity types. Also, we added following additional *generic rules* for specifying constraints for our WEM relation type, which captures information about mention boundaries.

$$RTFinal(x, y, WEM) \Rightarrow$$
$$(ETFinal(x, OTH) \vee ETFinal(y, OTH)).$$
$$RTFinal(x, y, WEM) \Rightarrow$$
$$(!ETFinal(x, OTH) \vee !ETFinal(y, OTH)).$$

By definition, the WEM relation holds between a head word of an entity mention and other words of that entity mention. Additionally, head word of an entity mention is labelled with appropriate entity label and other words are labelled with entity type OTH. The above rules state that if there is WEM relation between two words x and y then at least one of them should have label OTH and at least one of them should have entity type label, i.e. a label from domain $etype$ other than OTH.

Similarly, all the *sentence-specific rules* (with finite weights proportional to AWP-NN prediction probabilities) described in (Pawar et al., 2016) are also generated for representing predictions of the AWP-NN model. We use *Constant Multiplier* (CM) as the weights assignment strategy. Following rule would be generated for each entity type E (from *etypes*) for each word pair $\langle W_i, W_i \rangle$, with the weight $10 \cdot Pr_{AWP-NN}(E|\langle W_i, W_i \rangle)$ where E_{max} is the predicted entity type with the highest probability:

$$ET(i, E_{max}) \Leftrightarrow ETFinal(i, E)$$

Similarly, following rule would be generated for each relation type R (from *rtypes*) for each word pair $\langle W_i, W_j \rangle$, with the weight $10 \cdot Pr_{AWP-NN}(R|\langle W_i, W_j \rangle)$ where R_{max} is the predicted relation type with the highest probability:

$$RT(i, R_{max}) \Leftrightarrow RTFinal(i, R)$$

Using these *generic* and *sentence-specific* rules, an MLN is constructed for each sentence. The best values of $ETFinal$ and $RTFinal$ (query predicates) for each word pair are obtained by using the

inference in this MLN with ET and RT as evidence predicates based on AWP-NN's predictions.

5 Experimental Analysis

5.1 Dataset

ACE 2004 dataset (Doddington et al., 2004) is the most widely used dataset[5] for reporting relation extraction performance. We use this dataset to demonstrate effectiveness of our approach for end-to-end relation extraction using AWP-NN model and MLN inference. We perform 5-fold cross-validation on this dataset where the folds are formed at the document level. We follow the same assumptions made by (Chan and Roth, 2011; Li and Ji, 2014; Pawar et al., 2016), which are - ignore the DISC relation, do not consider implicit relations (resulting due to intra-sentence co-references) as false positives and use coarse-level entity and relation types.

Direction of Relations: Out of 6 coarse-level relation types that we are considering, we need not model direction for relation types PER-SOC, GPE-AFF and ART. Because in case of these relations, given the entity types of their arguments, the direction of relation is not necessary or becomes implicit. As PER-SOC is a social relation between two PER entity mentions, the direction is not necessary. For GPE-AFF, as entity type of one of the arguments is always GPE, the direction becomes implicit. Also, the relation type ART always holds between an agent (PER, ORG or GPE) and an artifact (FAC, WEA or VEH), hence the direction is implicit. Whereas for relation like EMP-ORG which also represents subsidiary relationship between two ORG entity mentions, it is important to model the relation direction explicitly. Consider following sentence fragments:

- `..the fisheries section of the Gulf Coast Research Laboratory..`
- `..company that owned Road & Track..`

Here, EMP-ORG relation exists between ORG entity mentions `fisheries section` and `Gulf Coast Research Laboratory`. Whereas, EMP-ORG-R relation holds between `that` and `Road & Track`.

Hence, we consider 9 distinct relation types: EMP-ORG, EMP-ORG-R, PHYS, PHYS-R, OTHER-AFF, OTHER-AFF-R, PER-SOC, GPE-AFF and ART. Hence, the overall dataset contained 4074 instances[6] of valid relation types.

5.2 Implementation details

We used Keras (Chollet, 2015) for implementing our AWP-NN model. The model was trained for 40 epochs using batch size of 64 instances. We used *Dropout* (Srivastava et al., 2014) for regularization with probability 0.5 for hidden layers and 0.1 for embedding layers. We used the tool Alchemy[7] for MLN inference. The value of K (maximum length of dependency path, see Figure 1) was set to be 4, hence all word pairs having length of dependency path more than 4 were assumed to have NULL label.

5.3 Results

Table 5 shows the comparative performances (in terms of micro-F1 measure) for various approaches. The results are divided in three different sections:

1. **only entity extraction:** It includes boundary identification as well as entity type classification.
2. **only relation extraction:** It includes relation type classification for each pair of predicted entity mentions. It is a relaxed version of end-to-end relation extraction problem where correct relation label for an entity mention pair is counted as a true positive even if entity types of one or both the mentions are identified incorrectly.
3. **entity+relation extraction:** It is end-to-end relation extraction which includes boundary identification, entity type classification and relation type classification. Here, correct relation label for an entity mention pair is counted as a true positive only if boundaries and entity types of both the mentions are identified correctly.

It can be observed in the table 5 that end-to-end relation extraction performance of our AWP-NN model is better than all the 4 previous approaches (Chan and Roth, 2011; Li and Ji, 2014; Pawar et al., 2016; Miwa and Bansal, 2016) on the ACE 2004 dataset. However, the AWP-NN+MLN approach which uses MLN inference to revise AWP-NN predictions during decoding, achieves the best performance.

[5] We haven't yet acquired a more recent ACE 2005 dataset

[6] 279 instances of type DISC were not considered. Additionally, 21 relation instances were not contained in a single sentence as per our sentence detection algorithm.

[7] `https://alchemy.cs.washington.edu/`

Approach	Entity Extraction			Relation Extraction			Entity+Relation		
	P	R	F	P	R	F	P	R	F
(Chan and Roth, 2011)				42.9	38.9	40.8			
(Li and Ji, 2014)	83.5	76.2	79.7	64.7	38.5	48.3	60.8	36.1	45.3
(Pawar et al., 2016)	79.0	80.1	79.5	57.9	45.6	51.0	52.4	41.3	46.2
(Miwa and Bansal, 2016)	80.8	82.9	**81.8**				48.7	48.1	48.4
AWP-NN	81.1	79.7	80.4	60.3	48.1	53.5	55.6	44.4	49.3
AWP-NN + MLN	81.2	79.7	80.5	61.1	47.9	**53.7**	56.7	44.5	**49.9**

Table 5: Performance of various approaches on the ACE 2004 dataset. The numbers are micro-averaged and obtained after 5-fold cross-validation. Actual folds used by each algorithm may differ.

5.3.1 Statistical Significance

As neural networks are initialized randomly, if we train a neural network model multiple times, different predictions are obtained each time. Hence, it is important to establish the statistical significance of the performance. We train our AWP-NN model 30 times independently and obtain 30 different values for precision, recall and F1 score. The numbers shown in table 5 are average values over these 30 runs. Also, in order to establish that the F1 score of AWP-NN model is significantly higher than the best previous F1 score of 48.4% (by Miwa and Bansal (2016)), we conduct one tailed one sample t-test. Here, mean and standard deviation of sample of 30 F1 scores by AWP-NN are 49.3 and 0.44, respectively. This leads to p-value of 1.23×10^{-12}, hence establishing the statistical significance of AWP-NN's performance.

5.4 Analysis of results

5.4.1 Effect of using MLN

We analyzed the effect of using MLN by observing the individual sentences where errors of AWP-NN were being corrected by MLN. As an example, consider the following sentence:
`Lemieux`$_0$ `rescued`$_1$ `his`$_2$ `team`$_3$ `from`$_4$ `bankruptcy`$_5$ `last`$_6$ `season`$_7$ `by`$_8$ `exchanging`$_9$ `deferred`$_{10}$ `salary`$_{11}$ `for`$_{12}$ `an`$_{13}$ `ownership`$_{14}$ `stake`$_{15}$ `.`$_{16}$

End-to-end relation extraction output produced by the AWP-NN model for this sentence is shown in the tables 6 and 7. Only error in this output is that entity type of the mention `team` should be ORG instead of PER as it refers to some professional team. After MLN inference, the entity type of `team` is corrected to ORG. This happens because of high-confidence EMP-ORG relations between `Lemieux` and `team` and between `his` and `team`. As both `Lemieux` and `his` are of type PER with high confidence, global inference using MLN[8] forces type of `team` to be ORG to ensure compatibility of relation and entity types.

Entity Mention	Boundaries	Entity Type
`Lemieux`	$(0, 0)$	PER
`his`	$(2, 2)$	PER
`team`	$(3, 3)$	PER

Table 6: End-to-end relation extraction output (entity mentions) produced by the AWP-NN model

Entity Mention Pair	Relation Type
`Lemieux`, `team`	EMP-ORG
`his`, `team`	EMP-ORG

Table 7: End-to-end relation extraction output (relations) produced by the AWP-NN model

The AWP-NN model was able to outperform (see table 5) all 4 previous approaches without the help of MLN. One reason behind this may be that the AWP-NN model itself was sufficient to learn most of the dependencies among the entity and relation types. However, MLN helped to improve the performance of AWP-NN by 0.6 F1. Though considerable improvement was observed in the precision value, the recall improvement was not significant. In other words, MLN was observed to be more effective for reducing false positives than false negatives.

5.4.2 Difficult to identify entities

We observed that for some entity mentions, it is very difficult to identify their entity types as the key information required for identification lies outside the current sentence. Currently, our approach does not use any information outside the sentences, such as document level co-reference

[8] Detailed MLN rules & inference results for this sentence can be found at: `www.cse.iitb.ac.in/~sachinpawar/MLN/sentence.html`

information. Usually these difficult to classify entity mentions are pronoun mentions. Some examples are as follows:

1. `Though, I think that if they could stifle the entire peace process at the moment, then that is what they'd like to do.`
2. `It is a partially victory for both sides.`

Here, in the first sentence, it is difficult to identify (even for humans) whether entity type of `they` is PER (e.g. set of leaders) or GPE (e.g. countries). Also, in the second sentence, entity type of `sides` can be any of PER, ORG or GPE depending on the context. In future, we plan to capture document level information for correctly predicting types of such mentions.

6 Related Work

There have been multiple lines of research for jointly modelling and extracting entities and relations. Integer Linear Programming (ILP) based approaches (Roth and Yih, 2004; Roth and Yih, 2007) were the earliest ones. Here, various local decisions are associated with suitable "cost" values and they are represented using an integer linear program. The optimal solution to this integer linear program provides the best global output. Another significant lines of research were Probabilistic Graphical Models (Roth and Yih, 2002; Singh et al., 2013), Card-pyramid parsing (Kate and Mooney, 2010) and Structured Prediction (Li and Ji, 2014; Li et al., 2014; Miwa and Sasaki, 2014).

Four previous approaches (Miwa and Sasaki, 2014; Li and Ji, 2014; Pawar et al., 2016; Miwa and Bansal, 2016) are the most similar to our approach in the sense that they all address the problem of end-to-end relation extraction without assuming gold-standard entity mention boundaries like the earlier approaches. Our idea of labelling "all word pairs" is similar to the table representation idea of Miwa and Sasaki (2014). The major difference is that they identify boundaries of mentions through BIO encoding of labels whereas we try to capture boundaries by treating them as an additional relation type WEM. Also, they perform structured prediction with beam search to find optimal label assignment to the table, whereas we opt for neural network based classification. The idea of using MLNs to incorporate domain knowledge and perform joint inference to obtain globally consistent output was proposed by Pawar et al. (2016). The current state-of-the-art approach for end-to-end relation extraction is by Miwa and Bansal (2016), who employ LSTM-RNN based model for addressing this problem.

7 Conclusion and Future Work

We proposed a novel approach for end-to-end relation extraction which carries out its all three subtasks (identifying entity mention boundaries, their entity types and relations among them) jointly by using a neural network based model. We proposed a "All Word Pairs" neural network model (AWP-NN) which reduces solution of these three subtasks to predicting an appropriate label for each word pair in a given sentence. End-to-end relation extraction output is then constructed from these labels of word pairs. We further improved output of the AWP-NN model by using inference in Markov Logic Networks so that some of the inconsistencies in word pair labels can be removed at the sentence level.

We demonstrated effectiveness of our approaches (AWP-NN and AWP-NN+MLN) on the standard dataset of ACE 2004. They outperformed all 4 previously reported joint modelling approaches (Chan and Roth, 2011; Li and Ji, 2014; Pawar et al., 2016; Miwa and Bansal, 2016) for end-to-end relation extraction. Since all three subtasks share the same AWP-NN model parameters, many inter-task dependencies are captured effectively by the AWP-NN itself (without MLN) and this can be validated by the fact that AWP-NN itself performs better than all other joint models. However, MLN certainly helps to further improve the end-to-end relation extraction performance by correcting some errors in predictions of the AWP-NN model.

In future, we plan to incorporate some additional features (e.g. document level co-reference information) in the AWP-NN model for improving its performance further. Also, deeper analysis of the errors is required to have a better understanding about which characteristics are better captured by the AWP-NN model as compared to the MLN and vice versa. This will help these two to complement each other in a better way.

References

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

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*, pages 551–560, Portland, Oregon, USA, June. Association for Computational Linguistics.

Franois Chollet. 2015. keras. `https://github.com/fchollet/keras`.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In *LREC*, volume 2, page 1.

Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, Rochester, New York, April. Association for Computational Linguistics.

Rohit J. Kate and Raymond Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 203–212, Uppsala, Sweden, July. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland, June. Association for Computational Linguistics.

Qi Li, Heng Ji, Yu HONG, and Sujian Li. 2014. Constructing information networks using one single model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1846–1851, Doha, Qatar, October. Association for Computational Linguistics.

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)*, pages 1105–1116, Berlin, Germany, August. Association for Computational Linguistics.

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)*, pages 1858–1869, Doha, Qatar, October. Association for Computational Linguistics.

Sachin Pawar, Pushpak Bhattacharyya, and Girish Palshikar. 2016. End-to-end relation extraction using markov logic networks. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2016), LNCS 9624*. Springer.

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 (Coling 2008)*, pages 697–704, Manchester, UK, August. Coling 2008 Organizing Committee.

Matthew Richardson and Pedro Domingos. 2006. Markov Logic Networks. *Machine learning*, 62(1-2):107–136.

Dan Roth and Wen-tau Yih. 2002. Probabilistic reasoning for entity & relation recognition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. ACL.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In Hwee Tou Ng and Ellen Riloff, editors, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 1–8, Boston, Massachusetts, USA, May 6 - May 7. Association for Computational Linguistics.

Dan Roth and Wen-tau Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. *Introduction to statistical relational learning*, pages 553–580.

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*, pages 1–6. ACM.

Nitish Srivastava, Geoffrey E 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(1):1929–1958.

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)*, pages 427–434, Ann Arbor, Michigan, June. Association for Computational Linguistics.