Semantic Role Labeling with Associated Memory Network

Chaoyu Guan[†], Yuhao Cheng[†], Hai Zhao[†*]
[†]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[†]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
{351549709,cyh958859352}@sjtu.edu.cn
zhaohai@cs.sjtu.edu.cn

Abstract

Semantic role labeling (SRL) is a task to recognize all the predicate-argument pairs of a sentence, which has been in a performance improvement bottleneck after a series of latest works were presented. This paper proposes a novel syntax-agnostic SRL model enhanced by the proposed associated memory network (AMN), which makes use of inter-sentence attention of label-known associated sentences as a kind of memory to further enhance dependency-based SRL. In detail, we use sentences and their labels from train dataset as an associated memory cue to help label the target sentence. Furthermore, we compare several associated sentences selecting strategies and label merging methods in AMN to find and utilize the label of associated sentences while attending them. By leveraging the attentive memory from known training data, Our full model reaches state-of-the-art on CoNLL-2009 benchmark datasets for syntax-agnostic setting, showing a new effective research line of SRL enhancement other than exploiting external resources such as well pre-trained language models.

1 Introduction

Semantic role labeling (SRL) is a task to recognize all the predicate-argument pairs of a given sentence and its predicates. It is a shallow semantic parsing task, which has been widely used in a series of natural language processing (NLP) tasks, such as information extraction (Liu et al., 2016) and question answering (Abujabal et al., 2017).

Generally, SRL is decomposed into four classification subtasks in pipeline systems, consisting of predicate identification, predicate disambiguation, argument identification, and argument classification. In recent years, great attention (Zhou and Xu, 2015; Marcheggiani et al., 2017; He et al., 2017, 2018a,b) has been turned to deep learning method, especially Long Short-term Memory (LSTM) network for learning with automatically extracted features. (Zhou and Xu, 2015) proposed the first end-to-end recurrent neural network (RNN) to solve the SRL task. (Marcheggiani et al., 2017) studied several predicate-specified embedding and decoding methods. (He et al., 2017) delivered a full study on the influence of RNN training and decoding strategies. Whether to use the syntactic information for SRL is also studied actively (He et al., 2017, 2018b).

Since the recent work of (Marcheggiani et al., 2017), which surprisingly shows syntax-agnostic dependency SRL for the first time can be rival of syntax-aware models, SRL has been more and more formulized into standard sequence labeling task on a basis of keeping syntax unavailable. A series of work on SRL received further performance improvement following this line through further refining neural model design (He et al., 2018a). Different from all previous work, we propose to introduce an associated memory network which builds memory from known data through the inter-sentence attention to enhance syntax-agnostic model even further.

Inspired by the observation that people always refer to other similar problems and their solutions when dealing with a problem they have never seen, like query in their memory, we want to utilize similar known samples which include the associated sentences and their annotated labels to help model label target sentence. To reach such a goal, we adopt a memory network component, and use inter-sentence attention to fully exploit the information in memory.

[*] Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), National Natural Science Foundation of China (No. U1836222 and No. 61733011) and Key Project of National Society Science Foundation of China (No. 15-ZDA041).

Proceedings of NAACL-HLT 2019, pages 3361–3371
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

Based on Memory Network (Weston et al., 2014; Sukhbaatar et al., 2015), (Miller et al., 2016) proposed Key-Value Memory Network (KV-MemNN) to solve Question Answering problem and gain large progress. Our proposed method is similar to KV-MemNN, but with a different definition of key-value and different information distilling process. Thus, we propose a carefully designed inter-sentence attention mechanism to handle it.

Recently, there are also some attempts to make use of attention mechanism in SRL task. (Tan et al., 2018; Strubell et al., 2018) focus on self-attention, which only uses the information of the input sentence as the source of attention. (Cai et al., 2018) makes use of biaffine attention (Dozat and Manning, 2017) for decoding in SRL, which was the current state-of-the-art (SOTA) in CoNLL-2009 benchmark as this work was embarking. Different from all previous work, we utilize inter-sentence attention to help model leverage associated information from other known sentences in the memory.

To our best knowledge, this is the first time to use memory network in the SRL task. Our evaluation on CoNLL-2009 benchmarks shows that our model outperforms or reaches other syntax-agnostic models on English, and achieves competitive results on Chinese, which indicates that memory network learning from known data is indeed helpful to SRL task.

There are several SRL annotation conventions, such as PropBank (Bonial et al., 2012) and FrameNet (Baker et al., 1998). This paper focuses on the former convention. Under PropBank convention, there are two role representation forms, which are span-based SRL, such as CoNLL 2005 and CoNLL 2012 shared tasks, and dependency-based SRL, such as CoNLL 2009 shared task. The former uses span to represent argument, while the latter uses the headword of the span to represent the argument. As the latter has been more actively studied due to dependency style SRL for convenient machine learning, we will focus on dependency SRL only in this work.

Given a sentence **S**, the goal of dependency SRL task is to find all the predicate-argument pairs (p, a). The following shows an example sentence with semantic role labels marked in subscripts.

She$_{A0}$ has **lost**$_{v}$ **it**$_{A1}$ **just**$_{ARGM-MNR}$ as quickly.

Here, *v* means the predicate, *A0* means the agent, *A1* means the patient and *ARGM-MNR* means how an action *v* is performed.

In the rest of this paper, we will describe our model in Section 2. Then, the experiment set-up and results are given in Section 3. Related works about SRL and attention mechanism will be given in Section 4. Conclusions and future work are drawn in Section 5.

2 Model

An SRL system usually consists of four pipeline modules: predicate identification and disambiguation, argument identification and classification. Following most of previous work, we focus on the last two steps in standard SRL task: argument identification and classification. The predicate identification subtask is not needed in CoNLL-2009 shared task[1], and we follow previous work (He et al., 2018b) to handle the predicate disambiguation subtask. This work will only focus on the argument labeling subtask through sequence labeling formalization. We first describe our base model in Section 2.1. Then we introduce the proposed associated memory network including the inter-sentence attention design and label merging strategies in Section 2.2. The full model architecture is shown in Figure 1.

2.1 Base Model

Word Embedding

We use the concatenation of the following embeddings as the representation for every word. (1) Random-initialized word embedding $x_i^{re} \in \mathbb{R}^{d_{re}}$ (2) GloVe (Pennington et al., 2014) word embedding $x_i^{pe} \in \mathbb{R}^{d_{pe}}$ pre-trained on 6B tokens (3) Random-initialized part-of-speech (POS) tag embedding $x_i^{pos} \in \mathbb{R}^{d_{pos}}$ (4) Random-initialized lemma embedding $x_i^{le} \in \mathbb{R}^{d_{le}}$ (5) Contextualized word embedding derived by applying fully connected layer on ELMo embedding $x_i^{ce} \in \mathbb{R}^{d_{ce}}$ (Peters et al., 2018), and (6) Random-initialized predicate specified flag embedding $x_i^{pred} \in \mathbb{R}^{d_{pred}}$. The final representation of each word is:

$$x_i = x_i^{re} \circ x_i^{pe} \circ x_i^{pos} \circ x_i^{le} \circ x_i^{ce} \circ x_i^{pred}$$

where $\circ$ stands for concatenation operator.

[1] In CoNLL-2009 task, the predicates information is already identified when testing.

BiLSTM Encoder

LSTM network is known to handle the dependency over long sentence well, and can effectively model the context information when encoding. Therefore, we leverage a stacked BiLSTM network $LSTM_e$ to be our encoder. It takes word embedding sequence $\mathbf{x} = [x_i]_{i=1}^{n_\mathbf{S}}$ of sentence $\mathbf{S} = [w_i]_{i=1}^{n_\mathbf{S}}$ as input ($n_\mathbf{S}$ is the length of sentence), and outputs two different hidden states $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ for word w_i by processing the sequence in forward and backward directions. The final contextual representation of word w_i is the concatenation of two hidden states $h_i = \overrightarrow{h_i} \circ \overleftarrow{h_i}$.

Then, we use a final softmax layer after the BiLSTM encoding to predict the label of each word.

2.2 Associated Memory Network

Using the base model as backbone, we introduce an associated memory network (AMN) component for further performance improvement. The proposed AMN memorizes known associated sentences and their labels, then the useful clue in the memory will be delivered to the SRL module through an inter-sentence mechanism. AMN processing includes three steps, associated sentence selection, inter-sentence attention and label merging.

Associated Sentence Selection

We aim to utilize the associated sentences and their labels to help our model label the target sentences. For the sake of fairness, we only use the sentences in train dataset as our source. However, it is impossible to attend all the sentences in train dataset because of the extremely high computational and memory cost. Therefore, we propose a filter to select the most useful sentences from the given dataset (train dataset in this paper) when given the label-unknown sentence $\mathbf{S}$.

The filter algorithm is straightforward. First, We compute the distance of every two sentences. Then, we sort all the sentences in train dataset according to their distances with the target sentence $\mathbf{S}$, and select top m sentences $\{\mathbf{A}_j\}_{j=1}^{m}$ with the minimum distances and their label sequences $\{\mathbf{L}_j\}_{j=1}^{m}$ as our associated attention. m is the memory size.

As for the computation of distance between two sentences, we formally consider three types of distances, which are *edit distance (ED)*, *word moving distance (WMD)* and *smooth inverse frequency distance (SD)*, plus *random distance (RD)* as baseline. These distances are defined as follows,

- *edit distance* This method uses the edit distance of the POS tag sequences of two sentences as the distance value.
- *word moving distance* Following (Kusner et al., 2015), this method takes word moving distance of two sentences[2].
- *smooth inverse frequency distance* Following (Arora et al., 2017), we use Euclidean distance between the SIF embedding of two sentences as the distance value.
- *random distance* This method returns a random value for distance computation thus lead to selecting sentences randomly in the train dataset.

Inter-sentence Attention

This part aims to attain the inter-sentence attention matrix, which can be also regarded as the core memory part of the AMN. The input sentence $\mathbf{S}$ and associated sentences $\{\mathbf{A}_j\}_{j=1}^{m}$ first go through a stacked BiSLTM network $LSTM_a$ to encode the sentence-level information to each word representation[3]:

$$\mathbf{S}' = LSTM_a(\mathbf{S})$$

$$\mathbf{A}'_j = LSTM_a(\mathbf{A}_j) \;\; j \in \{1, 2, ..., m\}$$

where $\mathbf{S}' = [x'_i]_{i=1}^{n_\mathbf{S}}$ and $\mathbf{A}'_j = [x'_{j,k}]_{k=1}^{n_j}$ are the lists of new word representations, with each word representation is a vector $x' \in \mathbb{R}^{d_a}$, where d_a is the size of hidden state in $LSTM_a$.

Then, for each associated sentence $\mathbf{A}'_j$, we multiply it with the input sentence representation $\mathbf{S}'$ to get the raw attention matrix M_j^{raw}.

$$M_j^{raw} = \mathbf{S}'\mathbf{A}_j'^T$$

Every element $M_j^{raw}(i,k) = x'_i \cdot x_{j,k}'^T$ can be regarded as an indicator of similarity between the i^{th} word in input sentence $\mathbf{S}'$ and the k^{th} word in associated sentence $\mathbf{A}'_j$.

Finally, we perform softmax operation on every row in M_j^{raw} to normalize the value so that it can

[2] In this paper, we use relaxed word moving distance (rwmd) for efficiency

[3] Here we abuse the symbol $\mathbf{S}$ and $\mathbf{A}_j$ for meaning both the word sequence $[w_i]$ and the embedded sequence $[x_i]$

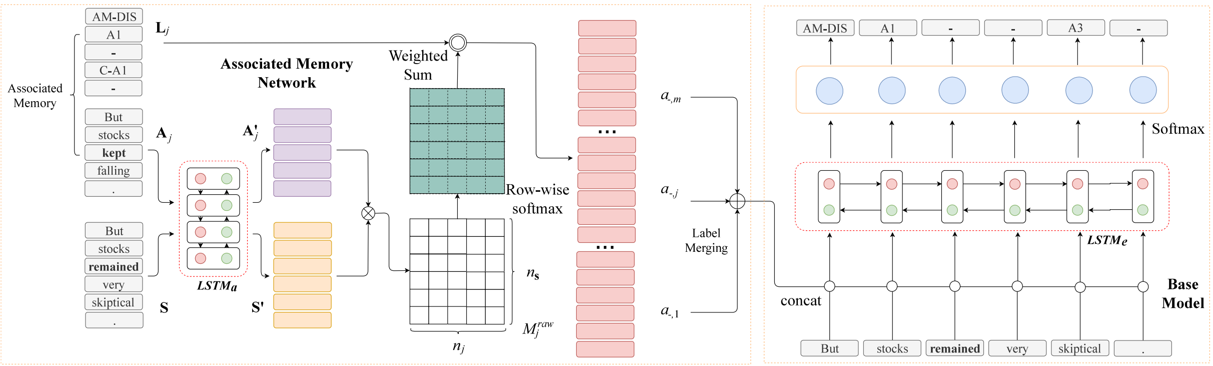

Figure 1: Semantic role labeling with associated memory network, where S is the input sentence with its length n_{S}. A_j is the j^{th} associated sentence of S with its label sequence L_j and its length n_j. S$'$ and A'_j are the result of $LSTM_1$ with S and A_j as input respectively. d_{ae} is the dimension of argument embedding. M_j^{raw} is the raw attention matrix of A_j and S. $a_{-,j} = [a_{1,j}, a_{2,j}, ..., a_{n_{\mathrm{S}},j}]$ is the associated-sentence-specified attention embedding.

be considered as probability from input sentence **S** to associated sentence $\mathbf{A}_j$.

$$\alpha_{i,j} = f([M_j^{raw}(i,1)..., M_j^{raw}(i,n_j)])$$

$$M_j = [\alpha_{1,j}, \alpha_{2,j}, ..., \alpha_{n_{\mathbf{S}},j}]$$

where $f(\cdot)$ stands for softmax function. $\alpha_{i,j}$ can be regarded as probability vector indicating the similarity between the i^{th} word in sentence **S** and every word in the associated sentence $\mathbf{A}'_j$.

Label Merging

In order to utilize the labels $\{\mathbf{L}_j\}_{j=1}^m$ of the associated sentences during decoding, a label merging needs to be done.

We use randomly initialized argument embedding $x^{ae} \in \mathbb{R}^{d_{ae}}$ to embed each argument label. Therefore, the label sequence $\mathbf{L}_j$ of associated sentence $\mathbf{A}_j$ can be written as $\mathbf{L}_j = [x_{j,k}^{ae}]_{k=1}^{n_j}$.We treat the probability vector $\alpha_{i,j}$ as weight to sum all the elements in L_j to get the associated-sentence-specified argument embedding $a_{i,j}$, which represents the attention embedding of word $w_i \in \mathbf{S}$ calculated from the j^{th} associated sentence $\mathbf{A}_j$ and label $\mathbf{L}_j$.

$$a_{i,j} = \alpha_{i,j} \cdot \mathbf{L}_j^T = \sum_{k=1}^{n_j} \alpha_{i,j}(k) x_{j,k}^{ae}$$

Because the associated sentences are different, the overall contributions of these argument embeddings should be different. We let the model itself learn how to make use of these argument embeddings. Following attention combination mechanism from (Libovickỳ and Helcl, 2017), we consider four ways to merge the label information.

1) Concatenation All the associated argument embedding are concatenated as the final attention embeddings.

$$a_i = a_{i,1} \circ a_{i,2} \circ ... \circ a_{i,m}$$

2) Average The average value of all the associated argument embeddings is used as the final attention embedding.

$$a_i = \frac{1}{m} \sum_{j=1}^m a_{i,j}$$

3) Weighted Average The weighted average of all the associated argument embedding is used as the final attention embedding. We calculate the mean value of every raw similarity matrix M_j^{raw} to indicate the similarity between input sentence **S** and associated sentence $\mathbf{A}_j$, and we use the softmax function to normalize them to get a probability vector β indicating the similarity of input sentence **S** towards all the associated sentences $\{\mathbf{A}_j\}_{j=1}^m$.

$$\beta = f([g(M_1^{raw}), ..., g(M_m^{raw})])$$

where $f(\cdot)$ stands for softmax function and $g(\cdot)$ represents the mean function. Then, we use the probability vector β as weight to sum all the associated-sentence-specified attention embedding $a_{i,j}$ to get the final attention embedding a_i of the i^{th} word w_i in input sentence **S**.

$$a_i = \sum_{j=1}^m \beta(j) a_{i,j}$$

4) Flat This method does not use $a_{i,j}$ information. First, we concatenate all the raw similarity matrix M_j^{raw} along the row.

$$M^{raw} = [M_1^{raw}, M_2^{raw}, ..., M_m^{raw}]$$

Then, we perform softmax operation on every row in M^{raw} to normalize the value so that it can be considered as probability from input sentence **S** to all associated sentences $\mathbf{A}_j$.

$$\gamma_i = f([M_{i,1}^{raw}, M_{i,2}^{raw}..., M_{i,n_{all}}^{raw}])$$

Name	**Meaning**	**Value**
d_{re}	random word embedding	100
d_{pe}	pre-trained word embedding	100
d_{pos}	POS embedding	32
d_{le}	lemma embedding	100
d_{ce}	contextualized embedding	128
d_{pred}	flag embedding	16
d_{ae}	argument embedding	128
m	memory size	4
k_e	$\#LSTM_e$ layers	2
k_a	$\#LSTM_a$ layers	3
d_e	$LSTM_e$ hidden state	512
d_a	$LSTM_a$ hidden state	512
r_d	dropout rate	0.1
l_r	learning rate	0.001

Table 1: Hyper-parameter settings (signal #x means number of x).

where $f(\cdot)$ stands for softmax operation. $n_{all} = \sum_{j=1}^{m} n_j$ is the total length of all m associated sentences.

We also concatenate the associated label information, and use γ_i as weight to sum the concatenated label sequence as final attention embedding.

$$\mathbf{L} = [\mathbf{L}_1, \mathbf{L}_2, ..., \mathbf{L}_j], \quad a_i = \gamma_i \cdot \mathbf{L}^T$$

After we have the final attention embedding a_i, we concatenate it with word embedding x_i as the input of the BiLSTM encoder $LSTM_e$.

3 Experiments

We conduct experiments on CoNLL-2009 (Hajič et al., 2009) English and Chinese dataset. We use the standard training, development and test data split provided by CoNLL-2009 shared task. The word lemma, word POS are the predicted ones given in CoNLL-2009 dataset. Adam optimizer (Kingma and Ba, 2014) is used for training to minimize the categorical cross entropy loss. All the hyper-parameters we use are listed in Table 1. All parameters are learned during training, and are randomly initialized except the pre-trained GloVe (Pennington et al., 2014) word embeddings.

For English, We independently determine the best distance calculating method and the best merging method one after another. First, we select a distance according to the results on development set and then we determine the merging method with the selected distance method. At last we explore the impact of memory size. For Chinese, we obtain the result with similar parameters as for the best model in English. The English and Chinese GloVe word embeddings are both trained on Wikipedia. The pretrained English ELMo model is from (Peters et al., 2018), and the Chinese one is from (Che et al., 2018), which is hosted at (Fares et al., 2017). The model is trained for maximum 20 epochs for the nearly best model based on development set results. We re-run our model using different initialized parameters for 4 times and report the average performance[4].

System (syntax-aware single)	P	R	F_1
(Zhao et al., 2009a)	-	-	86.2
(Zhao et al., 2009c)	-	-	85.4
(FitzGerald et al., 2015)	-	-	86.7
(Roth and Lapata, 2016)	88.1	85.3	86.7
(Marcheggiani and Titov, 2017)	89.1	86.8	88.0
(He et al., 2018b)	89.7	**89.3**	89.5
(Li et al., 2018)	**90.3**	89.3	**89.8**
System (syntax-aware ensemble)	P	R	F_1
(FitzGerald et al., 2015)	-	-	87.7
(Roth and Lapata, 2016)	90.3	85.7	87.9
(Marcheggiani and Titov, 2017)	**90.5**	**87.7**	**89.1**
System (syntax-agnostic single)	P	R	F_1
(Marcheggiani et al., 2017)	88.7	86.8	87.7
(He et al., 2018b)	89.5	87.9	88.7
(Cai et al., 2018)	89.9	**89.2**	**89.6**
(Li et al., 2018)	89.5	87.9	88.7
Ours (+ AMN + ELMo)	**90.0**	**89.2**	**89.6**

Table 2: Results on CoNLL-2009 English in-domain (WSJ) test set.

System (syntax-aware single)	P	R	F_1
(Zhao et al., 2009a)	-	-	74.6
(Zhao et al., 2009c)	-	-	73.3
(FitzGerald et al., 2015)	-	-	75.2
(Roth and Lapata, 2016)	76.9	73.8	75.3
(Marcheggiani and Titov, 2017)	78.5	75.9	77.2
(He et al., 2018b)	**81.9**	76.9	79.3
(Li et al., 2018)	80.6	**79.0**	**79.8**
System (syntax-aware ensemble)	P	R	F_1
(FitzGerald et al., 2015)	-	-	75.5
(Roth and Lapata, 2016)	79.7	73.6	76.5
(Marcheggiani and Titov, 2017)	**80.8**	**77.1**	**78.9**
System (syntax-agnostic single)	P	R	F_1
(Marcheggiani et al., 2017)	79.4	76.2	77.7
(He et al., 2018b)	**81.7**	76.1	78.8
(Cai et al., 2018)	79.8	78.3	79.0
Ours (+ AMN + ELMo)	80.0	**79.4**	**79.7**

Table 3: Results on CoNLL-2009 English out-of-domain (Brown) test set.

3.1 Results

For the predicate disambiguation, we use the same one from (He et al., 2018b) with the precisions

[4] Our implementation is publicly available at `https://github.com/Frozenmad/AMN_SRL`.

System (syntax-aware single)	P	R	F_1
(Zhao et al., 2009a)	80.4	75.2	77.7
(Roth and Lapata, 2016)	83.2	75.9	79.4
(Marcheggiani and Titov, 2017)	84.6	80.4	82.5
(He et al., 2018b)	84.2	**81.5**	82.8
(Li et al., 2018)	**84.8**	81.2	**83.0**
System (syntax-agnostic single)	P	R	F_1
(Marcheggiani et al., 2017)	83.4	79.1	81.2
(He et al., 2018b)	84.5	79.3	81.8
(Cai et al., 2018)	84.7	**84.0**	**84.3**
Ours (+ AMN + ELMo)	**85.0**	82.6	83.8

Table 4: Results on CoNLL-2009 Chinese test set.

of 95.01% and 95.58% on development and test sets. We compare our full model (using *edit distance* and *average method*) with the reported state-of-the-art models on both English and Chinese dataset. The results are in Tables 2, 3 and 4.

For English in-domain test, our model outperforms the syntax-agnostic model in (He et al., 2018b), whose architecture is quite similar to our base model. Our model achieves 89.6% in F_1 score, which is the same with current SOTA syntax-agnostic model (Cai et al., 2018). Besides, our result is competitive with existing syntax-aware and better than ensemble models.

The advantage is more salient on English out-of-domain test set. The F_1 score of our model is 79.7%, which is 0.7% higher than the current SOTA syntax-agnostic model (Cai et al., 2018). The result is also competitive with the best syntax-aware model (Li et al., 2018). The comparisons show that the proposed model has a greater generalization ability.

For Chinese, starting with the similar parameters as for the best model in English, we find that attending 5 associated sentences shows a better result on Chinese. Our model achieves 83.8% F_1 score, outperforming (He et al., 2018b) with an improvement of 2.0% in F_1 score. Our result is also competitive with that of (Cai et al., 2018).

Note that our method is not conflict with the one in (Cai et al., 2018), which leverages biaffine attention (Dozat and Manning, 2017) for decoding. However, due to experiment cycle, we are not able to combine these two methods together. We will leave the combination as future work.

In the following part, we conduct several ablation studies on our model. All the experiments are re-run 2-4 times and the average values are reported on CoNLL-2009 English development set.

System	P	R	F_1
WMD (Kusner et al., 2015)	**89.1**	87.1	88.1
SD (Arora et al., 2017)	88.5	**87.5**	88.0
RD	**89.1**	87.2	88.1
Base Model	88.7	86.9	87.8
ED	89.0	**87.5**	**88.3**

Table 5: Ablations about distance on CoNLL-2009 English development set. ED means edit distance, WMD means word moving distance, SD means SIF distance, RD means random distance.

System	P	R	F_1
Concatenation	88.9	86.6	87.7
Average	**89.0**	**87.5**	**88.3**
Weighted Average	88.7	87.4	88.1
Flat	88.4	86.9	87.7
None	88.7	86.9	87.8

Table 6: Ablations about label merging method on CoNLL-2009 English development set.

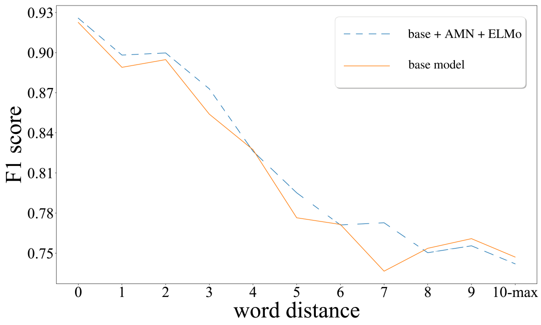

Figure 2: The comparison of our full model and base model with distance increases.

3.2 Ablation on Distance Method

Table 5 shows the performance of different distance calculating methods. All models use *average method* for label merging, and the memory size m is set to 4. It can be observed from Table 5 that edit distance performs best among all the distance calculating methods, with 88.3% F_1 score. All the distance calculating methods have surpassed the base model, showing that the proposed AMN is effective. Note that even the random distance model performs better than the base model, with an improvement of 0.3% in F_1 score, which shows that the proposed AMN can effectively extract useful information from even poorly related sentences. Besides, associated sentence selection methods based on word embeddings like WMD and SD have similar performance with random distance (RD), which shows simple word embedding may not be good enough signal indicator to measure semantic structure similarity in SRL task. On the contrary, we may also try to explain why even the random distance selection may work to some extent. As sentences always have core arguments label such as A0, A1 and A2, associated sentences even from random selection may also have such labels, which makes them helpful to enhance SRL over these labels. This may explain why our model with randomly selected associated sentences can distinguish core arguments better.

3.3 Ablation on Label Merging Method

Table 6 shows the performance of different label merging methods. All models use *edit distance* with 4 associated sentences. The result shows that *Average* label merging strategy gives the best performance, achieving 88.3% in F_1 score with an improvement of 0.5% compared to the baseline model.

Note that our weighted average model does not outperform the average model, which is a surprise to us. We speculate that the current weight calculation method needs to be more improved to fit the concerned task.

3.4 ELMo vs. AMN

Table 7 compares the performance contribution from ELMo and AMN. Our model can achieve better performance only using informative clue from training set in terms of AMN design, rather than focusing on external resource like ELMo. However, even though our baseline SRL has been enhanced by ELMo, it can still receive extra performance improvement from the propose AMN. Note that our enhancement from the proposed AMN keeps effective when ELMo is included (a 0.5% enhancement on baseline over the 0.3% enhancement on ELMo baseline)

3.5 Ablation on Memory Size

We show the effect of different memory size in Figure 3. Note that more associated sentences means more cost on time and space. We test memory size m from 2 to 6 (which reaches the limit under experiment setting in 11G GPU). We also fit the measured points with a linear function (the blue line in Figure 3). The performance of our model has a general trend of increasing when the memory size becomes larger, which shows the potential of the proposed AMN.

System (syntax-aware)	P	R	F_1
(He et al., 2018b)	86.8	85.8	86.3
(He et al., 2018b) + ELMo	87.7	87.0	87.3
(Li et al., 2018)	87.7	86.7	87.2
(Li et al., 2018) + ELMo	**89.2**	**87.6**	**88.4**
Ours (syntax-agnostic)	P	R	F_1
Base	86.9	85.0	86.0
Base + AMN	86.9	85.6	86.3
Base + ELMo	88.7	86.9	87.8
Ours + AMN + ELMo	**89.0**	**87.5**	**88.3**

Table 7: AMN vs. ELMo, the performance comparison on English development set.

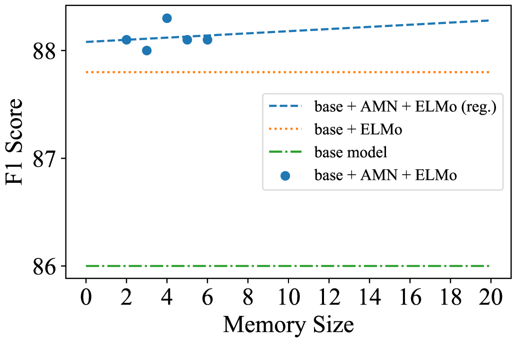

Figure 3: Model performance on English development set with different memory sizes, in which *base+AMN+ELMo* (*reg.*) indicates the general trend of our base model enhanced by the AMN when the memory size is enlarged.

3.6 Analysis on Confusion Matrix

To further understand the advance of the proposed method, we conduct an error type break down analysis. Figures 4 and 5 show the confusion matrices of labeling errors in the baseline model and our model on development set, respectively. We only show the main and most informative type of arguments. Every number in these figures stands for the times of occurrence. Comparing these two confusion matrixes shows that the proposed model makes fewer mistakes between core arguments such as $A0$, $A1$, and $A2$. AMN indeed helps when labeling them. It is also noted that, as in (He et al., 2017; Tan et al., 2018), the model still easily confuses ARG2 with AM-DIR, AM-LOC and AM-MNR.

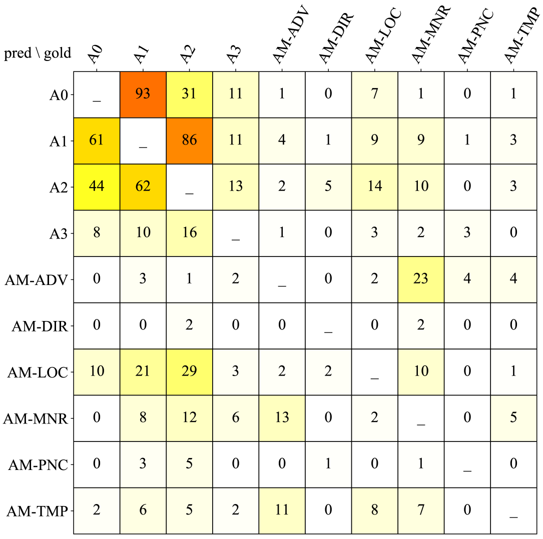

Figure 4: Confusion matrix for labeling errors in base model.

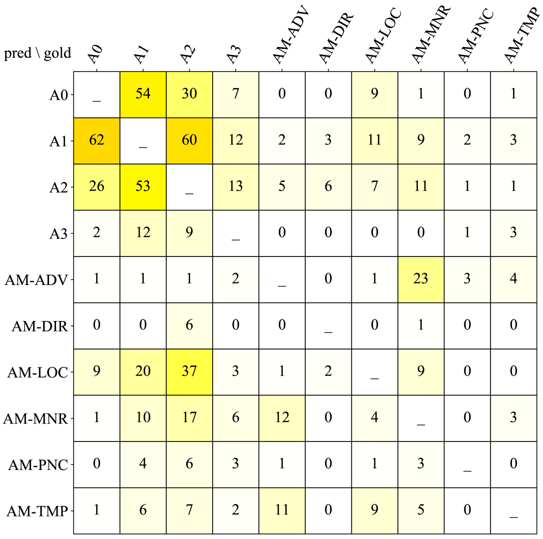

Figure 5: Confusion matrix for labeling errors in proposed model.

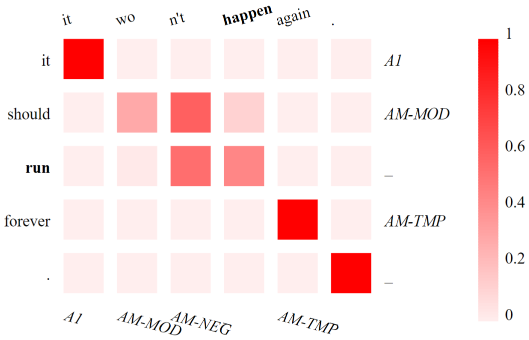

Figure 6: Visualization of similar matrix M. The input sentence is at the left of the matrix, with its golden argument label at the right. The associated sentence is at the top of the matrix, with its golden argument label at the bottom. Their predicate is bolded.

3.7 Analysis of Performance on Distance

We compare the performance concerning with the distance of argument and predicate on our best model and base model in Figure 2, from which we can observe that our model performs better nearly at any distance.

3.8 Case Study on AMN

To explore how the AMN works in the model, we visualize the similarity matrix M of some sentences from development set in Figure 6. The input sentence is

it$_{A1}$ **should**$_{AM-MOD}$ **run**$_{v}$ **forever**$_{AM-TMP}$.

And the associated sentence is

it$_{A1}$ **wo**$_{AM-MOD}$ **n\'t**$_{AM-NEG}$ **happen**$_{v}$ **again**$_{AM-TMP}$.

The current predicates are **run**, **happen** respectively. The visualization shows that inter-sentence attention can find and align the word in the similar context correctly, which shows that the proposed AMN is reasonable and effective.

4 Related Works

Early attempts (Pradhan et al., 2005; Zhao et al., 2009a,b, 2013; Roth and Woodsend, 2014) to the SRL task were mainly linear classifiers. The main focus was how to find proper feature templates that can best describe the sentences. (Pradhan et al., 2005) utilized a SVM classifier with rich syntactic features. (Toutanova et al., 2008) took the structural constraint into consideration by using a global reranker. (Zhao et al., 2009c) adopted a maximum entropy model with large scale feature template selection. (Roth and Woodsend, 2014) explored the distributional word representations as new feature to gain more powerful models.

Recently, a great attention has been paid on neural networks. (Zhou and Xu, 2015) proposed an end-to-end model using stacked BiLSTM network combined with CRF decoder without any syntactic input. (Marcheggiani et al., 2017) explored the predicate-specified encoding and decoding and also provided a syntax-agnostic LSTM model. (He et al., 2017) followed (Zhou and Xu, 2015) and analyzed all popular methods for initialization and regularization in LSTM network.

By considering that our approach also borrows power from the memory, the proposed inter-sentence attention in our AMN shares features with memory networks, which was proposed in (Weston et al., 2014) with motivation that memory may reduce the long-term forgetting issues. (Sukhbaatar et al., 2015) and (Miller et al., 2016) later further improved this work. However, we use quite different mechanisms to store the memory, and the effectiveness of our model needs a carefully designed attention mechanism to handle the sequence-level information distilling.

Attention mechanism was first used by (Bahdanau et al., 2014) in machine translation. Recently, (Tan et al., 2018) and (Strubell et al., 2018) proposed to use self-attention mechanism in SRL task. (Cai et al., 2018) leveraged the biaffine attention (Dozat and Manning, 2017) for better decoding performance. Different from all the existing work, we instead introduce an inter-sentence attention to further enhance the current state-of-the-art SRL.

5 Conclusions and Future Work

This paper presents a new alternative improvement on strong SRL baselines. We leverage memory network which seeks power from known data, the associated sentences, and thus is called associated memory network (AMN). The performance of our model on CoNLL-2009 benchmarks shows that the proposed AMN is effective on SRL task.

As to our best knowledge, this is the first attempt to use memory network in SRL task. There is still a large space to explore along this research line. For example, our weighted average method may need more carefully improved. Our model can be built over the biaffine attention which has been verified effective in (Cai et al., 2018)[5], and the encoder in our model can be improved with more advanced forms such as Transformer (Vaswani et al., 2017). At last, as this work is done on a basis of quite limited computational resources, only one piece of nVidia 1080Ti (11G graphic memory), much plentiful available computational resource will greatly enable us to explore more big model setting (i.e., larger memory size m) for more hopefully better performance improvement.

[5] As this paper is submitting, we get to know the work (Li et al., 2019), which has taken both strengths of biaffine and ELMo. We leave the verification of our proposed method over this new strong baseline in the future.

References

Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated template generation for question answering over knowledge graphs. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 1191–1200.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING)*, volume 1.

Claire Bonial, Jena Hwang, Julia Bonn, Kathryn Conger, Olga Babko-Malaya, and Martha Palmer. 2012. English propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntax-agnostic over syntax-aware? In *Proceedings of the 27th International Conference on Computational Linguistics (CoNLL)*, pages 2753–2765.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better ud parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 271–276.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 960–970.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 1–18.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018a. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 364–369.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 473–483.

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018b. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 2061–2071.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 957–966.

Zuchao Li, Shexia He, Jiaxun Cai, Zhuosheng Zhang, Hai Zhao, Gongshen Liu, Linlin Li, and Luo Si. 2018. A unified syntax-aware framework for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2401–2411.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dependency or span, end-to-end uniform semantic role labeling. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*.

Jindřich Libovickỳ and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 196–202.

Feifan Liu, Jinying Chen, Abhyuday Jagannatha, and Hong Yu. 2016. Learning for biomedical information extraction: Methodological review of recent advances. *arXiv preprint arXiv:1606.07993*.

Diego Marcheggiani, Anton Frolov, and Ivan Titov. 2017. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL)*, pages 411–420.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1506–1515.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1400–1409.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 1, pages 2227–2237.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H Martin, and Daniel Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 581–588.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 1192–1202.

Michael Roth and Kristian Woodsend. 2014. Composition of word representations improves semantic role labelling. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 407–413.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5027–5038.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, volume 2, pages 2440–2448.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4929–4936.

Kristina Toutanova, Aria Haghighi, and Christopher D Manning. 2008. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NIPS)*, pages 6000–6010.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Hai Zhao, Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009a. Multilingual dependency learning: Exploiting rich features for tagging syntactic and semantic dependencies. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 61–66.

Hai Zhao, Wenliang Chen, and Chunyu Kit. 2009b. Semantic dependency parsing of nombank and propbank: An efficient integrated approach via a large-scale feature selection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 30–39.

Hai Zhao, Wenliang Chen, Chunyu Kit, and Guodong Zhou. 2009c. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 55–60.

Hai Zhao, Xiaotian Zhang, and Chunyu Kit. 2013. Integrative semantic dependency parsing via efficient large-scale feature selection. *Journal of Artificial Intelligence Research*, 46:203–233.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent 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 (ACL-IJCNLP)*, volume 1, pages 1127–1137.