*Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 7129–7139
Marseille, 20-25 June 2022

# Syntax-driven Approach for Semantic Role Labeling

**Yuanhe Tian**♥, **Han Qin**♠, **Fei Xia**♥, **Yan Song**♠†
♥University of Washington ♠The Chinese University of Hong Kong (Shenzhen)
♥{yhtian, fxia}@uw.edu ♠hanqin@link.cuhk.edu.cn ♠songyan@cuhk.edu.cn

## Abstract

As an important task to analyze the semantic structure of a sentence, semantic role labeling (SRL) aims to locate the semantic role (e.g., agent) of noun phrases with respect to a given predicate and thus plays an important role in downstream tasks such as dialogue systems. To achieve a better performance in SRL, a model is always required to have a good understanding of the context information. Although one can use an advanced text encoder (e.g., BERT) to capture the context information, extra resources are also required to further improve the model performance. Considering that there are correlations between the syntactic structure and the semantic structure of the sentence, many previous studies leverage auto-generated syntactic knowledge, especially the dependencies, to enhance the modeling of context information through graph-based architectures, where limited attention is paid to other types of auto-generated knowledge. In this paper, we propose map memories to enhance SRL by encoding different types of auto-generated syntactic knowledge (i.e., POS tags, syntactic constituencies, and word dependencies) obtained from off-the-shelf toolkits. Experimental results on two English benchmark datasets for span-style SRL (i.e., CoNLL-2005 and CoNLL-2012) demonstrate the effectiveness of our approach, which outperforms strong baselines and achieves state-of-the-art results on CoNLL-2005.[‡]

**Keywords:** semantic role labeling, memory networks, syntactic information

## 1. Introduction

Semantic role labeling (SRL) is an important and fundamental task in natural language processing (NLP), since it provides a basic analysis of the semantic structure of the input sentence and plays an important role in downstream NLP tasks such as natural language inference (Saha et al., 2020), machine translation (Marcheggiani et al., 2018), and dialog generation (Xu et al., 2020). To achieve a good SRL performance, it is normally required to have precise understanding of the contextual information for the running text. Therefore, recent studies (Täckström et al., 2015; FitzGerald et al., 2015; Zhou and Xu, 2015; He et al., 2017; Tan et al., 2018; He et al., 2018; Ouchi et al., 2018; Shi and Lin, 2019; Guan et al., 2019; Li et al., 2020a; Li et al., 2020c) applied advanced text encoders (such as Bi-LSTM, Transformer (Vaswani et al., 2017), and pre-trained language models) that are proficient in effectively capturing contextual features and achieved great success on this task. Although such encoders are effective, it is always demanding by semantic tasks with extra knowledge. Therefore, to further enhance SRL performance, auto-generated knowledge especially syntactic ones, has been widely used and demonstrated to be more effective than only using the aforementioned encoders (Lewis et al., 2015; Wang et al., 2019; Kasai et al., 2019; Marcheggiani and Titov, 2020; Fei et al., 2021; Zhang et al., 2021b).

Among all different sources, word dependencies are the most intensively applied knowledge to SRL (Roth and Lapata, 2016; Marcheggiani and Titov, 2017; Strubell et al., 2018; Zhang et al., 2019b; Shi et al., 2020; Xia et al., 2020; Zhou et al., 2020), where limited attention is paid to other types of knowledge. In general, to leverage word dependencies, graph-based approaches (e.g., graph convolutional networks) (Marcheggiani and Titov, 2017; Xia et al., 2020) and multi-task learning (Strubell et al., 2018; Zhou et al., 2020; Paolini et al., 2021) approaches are often used and tend to be the standard solutions to this task. However, graph-based approaches are limited in requiring unlabeled graph structure input and thus need extra accommodations (e.g., use additional modules to leverage the label information on the edge of the graph) when they are applied to other types of knowledge, while multi-task learning approaches always perform parsing and SRL at the same time, requiring human annotated parses (difficult to be obtained) to train their models. Therefore, an appropriate framework is highly expected to incorporate different types of knowledge for this task with solving the aforementioned limitations.

In this paper, we propose a framework for SRL enhanced by map memories to leverage different types of auto-generated knowledge obtained from off-the-shelf toolkits for input sentences. Specifically, in the memory module, for each word in the input sentence, our approach extracts all its associated context words and their corresponding knowledge instances, which are then mapped to memory items to form (*key*, *value*) pairs, where each context word severs as the key and the corresponding knowledge instance serves as the value. Later, according to the contribution of the context word (i.e., key) and the corresponding knowledge instance (i.e., value) to SRL, our approach assigns different weights to memory items so as to discriminatively leverage the auto-generated knowledge. Finally, we compute the weighted sum of all memory items

[†] Corresponding author.

[‡] Our code is available at https://github.com/synlp/SRL-MM.

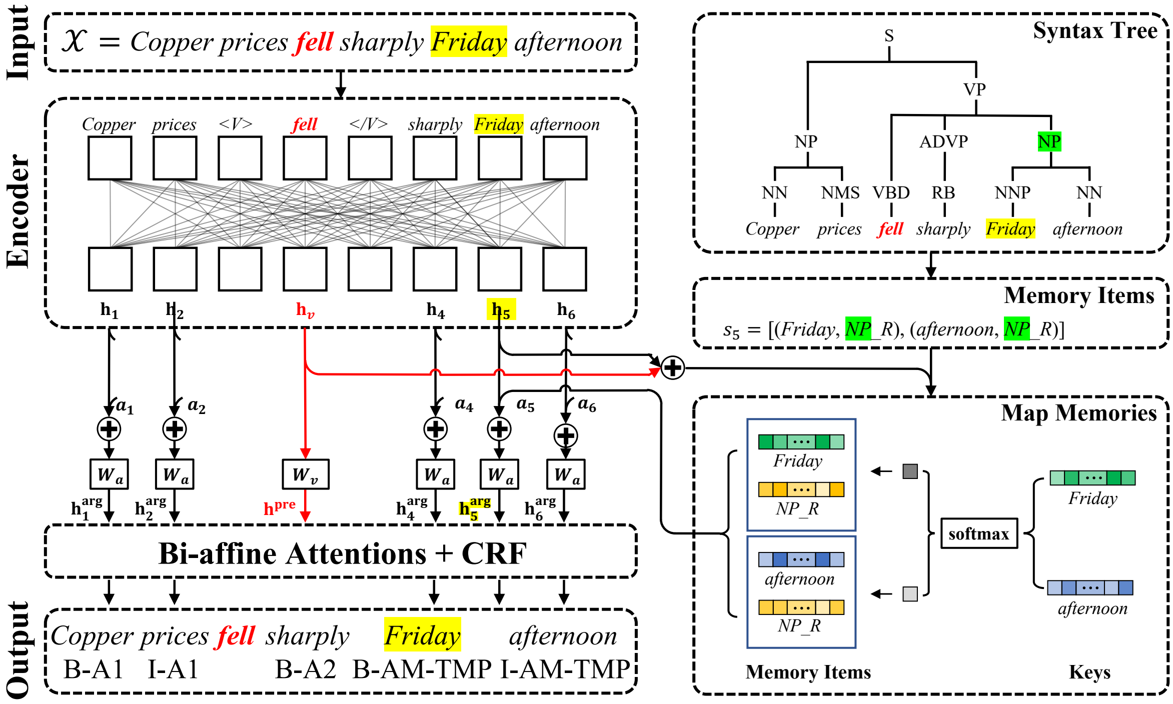

Input
$\mathcal{X}$ = Copper prices fell sharply Friday afternoon
Encoder
Copper prices <V> fell </V> sharply Friday afternoon
$\mathbf{h}_1$ $\mathbf{h}_2$ $\mathbf{h}_v$ $\mathbf{h}_4$ $\mathbf{h}_5$ $\mathbf{h}_6$
$a_1$ $a_2$ $a_4$ $a_5$ $a_6$
$W_a$ $W_v$
$\mathbf{h}_1^{arg}$ $\mathbf{h}_2^{arg}$ $\mathbf{h}^{pre}$ $\mathbf{h}_4^{arg}$ $\mathbf{h}_5^{arg}$ $\mathbf{h}_6^{arg}$
Bi-affine Attentions + CRF
Output
Copper prices fell sharply Friday afternoon
B-A1 I-A1 B-A2 B-AM-TMP I-AM-TMP
Syntax Tree
S
VP
NP
ADVP
NP
NN NMS VBD RB NNP NN
Copper prices fell sharply Friday afternoon
Memory Items
$s_5$ = [(Friday, NP_R), (afternoon, NP_R)]
Map Memories
Friday
NP_R
afternoon
NP_R
softmax
Friday
afternoon
Memory Items
Keys

Figure 1: The architecture of our proposed framework for SRL with map memories to leverage different types of auto-generated knowledge. The backbone model for SRL with an example input (where the given predicate, i.e., "*fell*", is highlighted in red color) and output is illustrated on the left. The process of extracting the memory items associated with a particular word, i.e., "*Friday*" (highlighted in yellow), from auto-generated knowledge, as well as the way of the memory module to encode the example memory items, are illustrated on the right.

and regard it as the output of the memory module to enhance encoder representations. In doing so, our approach leverages different types of auto-generated knowledge in a general way upon (*key*, *value*) represented information, which provides a more flexible framework than those graph-based approaches for the SRL task. In addition, by assigning different weights to memory items, our approach is able to distinguish potential noise (which could hurt model performance if it is not addressed appropriately) in the auto-generated knowledge. We test our approach on three types of widely used syntactic knowledge, namely, part-of-speech (POS) tags, syntactic constituents, and word dependencies, for SRL. Experimental results on two English benchmark datasets (i.e., CoNLL-2005 and CoNLL-2012) for span-style SRL demonstrate the effectiveness of our approach, which outperforms strong baselines and achieves state-of-the-art performance on CoNLL-2005.

## 2. The Approach

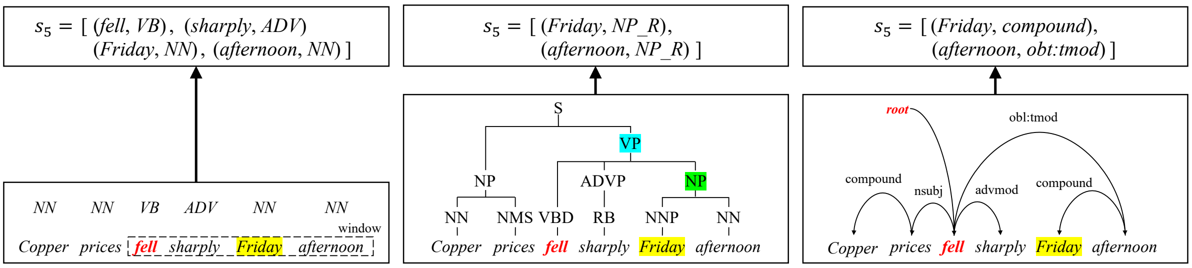
SRL aims to label the semantic roles of text spans with respect to a given predicate and it is conventionally performed as a sequence labeling task. We follow this convention and design the architecture of our approach illustrated in Figure 1, where the backbone semantic role tagger following the encoder-decoder architecture is illustrated on the left with an example input (the given predicate "*fell*" is highlighted in red color) and output. The extracted memory items for the particular word "*Friday*" (highlighted in yellow) from the auto-generated knowledge (i.e., the syntax constituency tree of the input sentence in this example) of the input sentence, as well as the map memories to leverage the knowledge, are illustrated on the right part. Formally, given an input word sequence (sentence) $\mathcal{X} = x_1 x_2 \cdots x_i \cdots x_n$ and its predicate $x_v \in \mathcal{X}$, the process of our approach for SRL with map memories (denoted by $\mathcal{M}$) is written by

$$\widehat{\mathcal{Y}} = \underset{\mathcal{Y}\in\mathcal{T}^n}{\arg\max}\, p\left(\mathcal{Y}|\mathcal{X}, x_v, \mathcal{M}\left(\mathcal{X}, x_v, \mathcal{S}\right)\right) \qquad (1)$$

where $\widehat{\mathcal{Y}} = \widehat{y}_1 \widehat{y}_2 \cdots \widehat{y}_i \cdots \widehat{y}_n$ is the corresponding output semantic role label sequence with $\widehat{y}_i$ denoting the semantic role label for the word $x_i$; $\mathcal{T}$ denotes the semantic role label set following the "BIO" scheme; $\mathcal{S}$ stands for the memory items extracted from the auto-generated knowledge. In the following text, we firstly elaborate the approach to construct the memory items based on different types of auto-generated knowledge (POS tags, syntactic constituents, and dependencies) for each word $x_i$. Next, we introduce the process of encoding the auto-generated knowledge through map memories. Finally, we illustrate the way to perform SRL with the proposed memory module.

### 2.1. Memory Item Construction

Following the paradigm in previous memory-based approaches for different NLP tasks (Sukhbaatar et al.,


s5 = [ (fell, VB), (sharply, ADV)
(Friday, NN), (afternoon, NN) ]
NN NN VB ADV NN NN
window
Copper prices fell sharply Friday afternoon
s5 = [ (Friday, NP_R),
(afternoon, NP_R) ]
S
VP
NP ADVP NP
NN NMS VBD RB NNP NN
Copper prices fell sharply Friday afternoon
s5 = [ (Friday, compound),
(afternoon, obt:tmod) ]
root
obl:tmod
compound
nsubj
advmod
compound
Copper prices fell sharply Friday afternoon

(a) POS Labels

(b) Syntactic Constituents

(c) Dependencies

Figure 2: An illustration of the way to construct the memory items (i.e., $s_5$) for an example word $x_5$ =“*Friday*” (highlighted in yellow background) based on the auto-generated (a) POS labels, (b) syntactic constituents, and (c) dependencies of the example sentence, where the given predicate, i.e., “*fell*”, is highlighted in red color.

2015; Miller et al., 2016; Mino et al., 2017; Xu et al., 2019; Tian et al., 2020a; Nie et al., 2020; Tian et al., 2020b; Chen et al., 2021), the memory item in our memory module corresponds to a (*key*, *value*) pair. Specifically, in our approach, the *key* stands for the context word and the *value* refers to the corresponding knowledge instance. In order to construct the memory items, we firstly use an off-the-shelf toolkit to generate different types of auto-generated knowledge of the input sentence, where in our paper, we consider three types of syntactic knowledge, namely, the POS labels, syntactic constituents, and dependencies, because they have been demonstrated to be effective in SRL (Marcheggiani and Titov, 2020; Xia et al., 2020; Zhou et al., 2020). Then, for each $x_i$, we extract its context words (i.e., keys) and the associated syntactic knowledge instances (i.e., values) according to the following procedures, which is illustrated with an example sentence and $x_5$ =“*Friday*” in Figure 2.

**POS Labels** For POS labels, we regard the word $x_i$ as the central word and apply a window of ±2 words[1] to extract its context words and regard them as the keys (denoted by $k_{i,j}$) in the memory items. Then, for each key (i.e., context word), we find its corresponding value (denoted by $v_{i,j}$) by extracting the auto-generated POS label of $k_{i,j}$ and thus obtain the memory item $s_{i,j} = (k_{i,j}, v_{i,j})$ accordingly. For example, in the example in Figure 2(a), for “*Friday*”, the ±2 word window covers its left and right two words. Therefore, the resulting context words (i.e., keys) are “*fell*”, “*sharply*”, “*Friday*”, and “*afternoon*” which are highlighted in the window box, and their corresponding knowledge instances are “*VB*”, “*ADV*”, “*NN*”, and “*NN*”, respectively. As a result, the memory items associated with the word $x_5$ =“*Friday*” are $s_5$ = [(*fell*, *VB*), (*sharply*, *ADV*), (*Friday*, *NN*), (*afternoon*, *NN*)].

**Syntactic Constituents** For syntactic constituents, we start with $x_i$ at the leaf of $\mathcal{X}$'s syntax tree, then search up through the tree to find the first syntactic node (denoted by $\mathbf{N}$) that dominates both $x_i$ and the give predicate $x_v$ in its corresponding text span. Then, we select the child node (denoted by $\mathbf{N}_c$) of $\mathbf{N}$ such that the text span of $\mathbf{N}_c$ contains $x_i$ and regard all words in that text span as the context words (i.e., keys) in the memory items. Afterwards, we regard the syntactic label $l$ of $\mathbf{N}_c$ as the value for all keys. In addition, because the relative position between the word $x_i$ and the predicate $x_v$ is important in distinguishing the semantic role label of $x_i$ (for SVO languages such as English and Chinese, the $x_i$ is more likely to be the agent if it appears at the left of the predicate), we further distinguish the values in the memory items when $x_i$ has different directional relation with $x_v$. Specifically, we attach a “*_L*” or “*_R*” suffix to the values in the memory items, if $x_i$ is at the left or right side of $x_v$, respectively. For example, in Figure 2(b), we start from “*Friday*” and find the first node that dominates both “*Friday*” and the predicate “*fell*” is the *VP* (highlighted in blue). Then, we find that the node *NP* (highlighted in green) is the child of the *VP* that dominates the word “*Friday*”. Afterwards, we select the two words covered by the *NP* node, namely, “*Friday*” and “*afternoon*”, to construct the keys of the memory items and use “*NP_R*” as their corresponding value (we attach the “*_R*” suffix to “*NP*” because “*Friday*” is on the right of “*fell*”). Therefore, the resulting memory items associated with $x_5$ =“*Friday*” are $s_5$ = [(*Friday*, *NP_R*), (*afternoon*, *NP_R*)].

**Dependencies** For dependencies, we find all context words (i.e., keys) for $x_i$ by collecting all its dependents and governor (as well as $x_i$ itself) from $\mathcal{X}$'s dependency tree. Then, for each key, we regard its in-bound dependency type as the corresponding value in the memory item. For example, as illustrated in Figure 2(c), for “*Friday*”, its context words (i.e., keys) are “*Friday*” and “*afternoon*” (the governor of “*Friday*”), and their corresponding in-bound dependency type are “*compound*” and “*obt:tmod*”.[2] Therefore, the memory items associated with $x_5$ =“*Friday*” are $s_5$ = [(*Friday*, *compound*), (*afternoon*, *obt:tmod*)].

[1] We use ±2 word window size because this is a widely used hyper-parameter in previous studies for NLP and it achieves the best performance in experiments.

[2] Note that, in this case, we do not have context words selected from the dependents since “*Friday*” does not have any dependents according to the dependency parse result.

As a result, we represent different types of auto-generated syntactic knowledge associated with $x_i$ by a uniformed structure, namely, a list of memory items $s_i$, where the keys and values in the memory items are the context words and the corresponding knowledge instances, respectively. This uniformed knowledge representation is then fed into the map memories to enhance the backbone SRL model as illustrated in Figure 1.

## 2.2. Map Memories

To leverage auto-generated knowledge, i.e., dependencies in particular, many studies (Marcheggiani and Titov, 2017; Chen et al., 2020; Xia et al., 2020; Yu et al., 2020; Tian et al., 2021; Qin et al., 2021) use graph-based architectures (e.g., graph convolutional networks and its variants), which normally require extra accommodations to be applied to other types of knowledge since they are not naturally represented in graphs. To model different types of knowledge in a uniformed architecture, we propose map memories to encode different types of knowledge resources (i.e., POS labels, syntactic constituents, and dependencies) represented by the keys and values in the memory items.

Specifically, for each word $x_i$ in the input, we firstly construct its associated memory item list $s_i$ according to the aforementioned process and use two matrices to map the keys and values in the memory items to their embeddings, where the embeddings for key $k_{i,j}$ and value $v_{i,j}$ in the $j$-th memory item are denoted by $\mathbf{e}^k_{i,j}$ and $\mathbf{e}^v_{i,j}$, respectively. Next, we compute the weight (denoted by $p_{i,j}$) for each memory item by

$$p_{i,j} = \frac{\exp\left((\mathbf{h}_i \oplus \mathbf{h}_v) \cdot \mathbf{e}^k_{i,j}\right)}{\sum_{j=1}^{m_i} \exp\left((\mathbf{h}_i \oplus \mathbf{h}_v) \cdot \mathbf{e}^k_{i,j}\right)} \quad (2)$$

where $\mathbf{h}_i$ and $\mathbf{h}_v$ are the hidden vectors for the word $x_i$ and the given predicate $x_v$ obtained from the encoder in the backbone model; $\oplus$ denotes the vector concatenation operation; $m_i$ is the total number of memory items associated with $x_i$. Then, we apply the weights to the corresponding memory items and obtain the weighted sum (denoted as $\mathbf{a}_i$) of both keys and values through

$$\mathbf{a}_i = \sum_{j=1}^{m_i} p_{i,j} \cdot \left(\mathbf{e}^k_{i,j} + \mathbf{e}^v_{i,j}\right) \quad (3)$$

where $\mathbf{a}_i$ is the output of the proposed memory module and it contains the weighted information of context words and knowledge instances.

In doing so, our approach is able to encode different types of auto-generated knowledge in a uniformed structure without requiring extra accommodations in model structure. In addition, compared with previous memory-based approaches (Sukhbaatar et al., 2015; Miller et al., 2016; Mino et al., 2017; Xu et al., 2019) that only leverage the information carried by values, our approach is able to discriminatively leverage both keys and values by assigning different weights to variant memory items and thus enhance the backbone model.

| Datasets | | Sent. # | Token # | Predicate # |
|---|---|---|---|---|
| WSJ | Train | 40K | 950K | 91K |
| | Dev | 1K | 33K | 3K |
| | Test | 2K | 57K | 5K |
| Brown | | 0.4K | 7K | 1K |
| CN12 | Train | 67K | 1,299K | 189K |
| | Dev | 8K | 163K | 24K |
| | Test | 8K | 170K | 24K |

Table 1: The statistics of the benchmark datasets used in our experiments for SRL, where the number of sentence, tokens, and predicates are reported.

## 2.3. SRL with Map Memories

Once the map memory module is built, it is straightforward to apply it to SRL through a backbone sequence labeling model. In our approach, we use a text encoder (e.g., Bi-LSTM, Transformer, BERT, etc.) to map all words $x_i$ in the input $\mathcal{X}$ to their corresponding hidden vectors $\mathbf{h}_i$. For each non-predicate word $x_i$ and $x_i \neq x_v$, we feed $\mathbf{h}_i$ into the memory module and obtain the corresponding output $\mathbf{a}_i$. Then, we concatenate $\mathbf{h}_i$ and $\mathbf{a}_i$ to obtain the knowledge enhanced word representation $\widetilde{\mathbf{h}}_i = \mathbf{h}_i \oplus \mathbf{a}_i$. Further more, we use a fully connected layer that is designed for non-predicate words to map $\widetilde{\mathbf{h}}_i$ to a new vector space by

$$\mathbf{h}^{\text{arg}}_i = ReLu\left(\mathbf{W}_a \cdot \widetilde{\mathbf{h}}_i + \mathbf{b}_a\right) \quad (4)$$

where $\mathbf{W}_a$ and $\mathbf{b}_a$ are the trainable matrix and bias vector, respectively, in the fully connected layer, and $\mathbf{h}^{\text{arg}}_i$ is the vector to be used to predict the semantic role label of $x_i$. Similarly, we apply another fully connected layer (with $\mathbf{W}_v$ and $\mathbf{b}_v$ trainable matrix and bias vector) to the hidden vector of the predicate obtained directly from the text encoder to compute the predicate representation $\mathbf{h}^{\text{pre}}$, which can be formalized as

$$\mathbf{h}^{\text{pre}} = ReLu\left(\mathbf{W}_v \cdot \mathbf{h}_v + \mathbf{b}_v\right) \quad (5)$$

Afterwards, we feed $\mathbf{h}^{\text{arg}}_i$ and $\mathbf{h}^{\text{pre}}$ to the bi-affine attention module where the score $o^y_i$ of labeling $x_i$ with a particular semantic role tag $y$ is computed by

$$o^y_i = \mathbf{h}^{\text{arg}}_i \cdot \mathbf{W}^y \cdot \mathbf{h}^{\text{pre}} \quad (6)$$

Herein, $\mathbf{W}^y$ is the trainable matrix for the semantic role tag $y \in \mathcal{T}$. Finally, we pass the scores over the tag set to the conditional random field (CRF) module to predict the semantic role tag $\widehat{y}_i$ for $x_i$.

# 3. Experimental Settings

## 3.1. Datasets

In the experiments, we follow previous studies (Täckström et al., 2015; He et al., 2017; Ouchi et al., 2018; Tan et al., 2018; Strubell et al., 2018; He et al., 2018; Li et al., 2020a; Zhang et al., 2021a) to evaluate

| Hyper-parameters | Values |
|---|---|
| Learning Rate | $5e-6, \mathbf{1e-5}, 3e-5, 5e-5$ |
| Warmup Rate | $\mathbf{0.06}, 0.1$ |
| Dropout Rate | $\mathbf{0.1}$ |
| Batch Size | $4, 8, \mathbf{16}$ |

Table 2: The hyper-parameters used in tuning our models, where the best one used in our final experiments are highlighted in boldface.

| Models | WSJ | Brown | CN12 |
|---|---|---|---|
| BERT-base | 87.93 | 81.51 | 85.92 |
| + $\mathcal{M}$ (POS) | 88.22 | 82.55 | 86.20 |
| + $\mathcal{M}$ (Syn.) | 88.43 | 82.70 | 86.37 |
| + $\mathcal{M}$ (Dep.) | **88.52** | **82.94** | **86.43** |
| BERT-large | 88.69 | 81.94 | 86.53 |
| + $\mathcal{M}$ (POS) | 88.84 | 82.86 | 86.79 |
| + $\mathcal{M}$ (Syn.) | 88.95 | 83.27 | 87.04 |
| + $\mathcal{M}$ (Dep.) | **89.02** | **83.34** | **87.12** |

(a) BERT

| Models | WSJ | Brown | CN12 |
|---|---|---|---|
| XLNet-base | 88.75 | 81.60 | 86.46 |
| + $\mathcal{M}$ (POS) | 88.92 | 83.57 | 86.59 |
| + $\mathcal{M}$ (Syn.) | 89.22 | 83.78 | 86.84 |
| + $\mathcal{M}$ (Dep.) | **89.34** | **83.89** | **87.02** |
| XLNet-large | 88.91 | 82.12 | 86.87 |
| + $\mathcal{M}$ (POS) | 89.31 | 83.97 | 87.39 |
| + $\mathcal{M}$ (Syn.) | 89.65 | 84.83 | 87.64 |
| + $\mathcal{M}$ (Dep.) | **89.80** | **85.02** | **87.67** |

(b) XLNet

Table 3: Experimental results of baselines and our approach with map memories (i.e., + $\mathcal{M}$) on the test set of WSJ, Brown, and CN12, where the base and large version of BERT and XLNet are used as the encoder. "POS", "Syn.", and "Dep." denote the POS labels, syntactic constituents, and dependencies used in our approach, respectively.

our approach on two span-style English SRL benchmark datasets, namely CoNLL-2005 (CN05) (Carreras and Màrquez, 2005) and CoNLL-2012 (CN12) (Pradhan et al., 2012).[3] Herein, CN05 has two datasets, namely WSJ and Brown, with WSJ and Brown used for in-domain and cross-domain settings. For all datasets, we use the standard train/dev/test splits and follow He et al. (2018) to pre-process the data. Table 1 reports the statistics (i.e., the number of sentences, tokens, and predicates) of all datasets used in this study.

### 3.2. Implementation

In the experiments, we use Stanford CoreNLP Toolkits (Manning et al., 2014) to obtain the three types of auto-generated knowledge (i.e., POS labels, syntactic constituents, and dependencies). Considering that a good text representation is the key to achieve outstanding model performance (Pennington et al., 2014; Komninos and Manandhar, 2016; Song et al., 2017; Peters et al., 2018; Song et al., 2018; Song and Shi, 2018; Raffel et al., 2019; Zhang et al., 2019a; Diao et al., 2020; Lewis et al., 2020; Song et al., 2021; Diao et al., 2021), we try pre-trained language models, i.e., the BERT base and large (Devlin et al., 2019) as well as XLNet (Yang et al., 2019), as the encoder in the backbone model. For both BERT and XLNet, we follow their default setting: 12 layers of self-attentions with 768 dimensional hidden vectors for their base version and use 24 layers of self-attentions with 1024 dimenstional hidden vectors for their large version).[4] To help the encoder identify the position of the predicate, we insert two special tokens, i.e., "`[V]`" and "`[/V]`", before and after the given predicate (as shown in Figure 1). We randomly initialize the embeddings of keys and values used in the memory modules and update them during training. For evaluation, we use the official evaluation script[5] from CoNLL-2005 to compute the F1 scores for all datasets. For all models, we try different combinations of hyper-parameters, select the one achieves the highest F1 score on the development set, and report its performance on the test set. Particularly, for the cross-domain experiments on Brown, we follow previous studies to train the models on the training set of WSJ, tune the hyper-parameters on the development of WSJ, and finally evaluate the model on Brown.

## 4. Results and Analyses

### 4.1. Overall Results

In the main experiments, we run BERT and XLNet baselines and our approach with map memories to incorporate auto-generated POS labels (POS), syntactic constituents (Syn.), and dependencies (Dep.), where one of the three types of knowledge is used at a time. The experimental results (F1 scores) of different models on the test sets. Here are some observations.

First, our approach works well with different types of auto-generated knowledge, where consistent improvements over the BERT and XLNet baselines are observed over all datasets, although the BERT and XLNet baselines have already achieved outstanding performance. Second, comparing the models with different types of knowledge, in most cases, the ones enhanced by syntactic constituents and dependencies obtain higher improvement over the baselines than the models enhanced by POS labels. This observation is

[3] We download CN05 from `https://www.cs.upc.edu/~srlconll/soft.html` and CN12 from `http://cemantix.org/data/ontonotes.html`.

[4] We download the cased version of BERT from `https://github.com/google-research/bert` and XLNet from `https://github.com/zihangdai/xlnet`.

[5] We use `srl-eval.pl` obtained from `https://www.cs.upc.edu/~srlconll/soft.html`.

| Models | WSJ | Brown | CN12 |
|---|---|---|---|
| He et al. (2017) | 84.6 | 73.6 | 83.4 |
| Tan et al. (2018) | 86.1 | 74.8 | 83.9 |
| Ouchi et al. (2018) | 87.6 | 78.7 | 86.2 |
| He et al. (2018) | 87.4 | 80.4 | 85.5 |
| †Strubell et al. (2018) | 86.04 | 76.54 | 83.38 |
| †Wang et al. (2019) | 88.2 | 79.3 | 86.4 |
| Shi and Lin (2019) (BERT) | 88.8 | 82.0 | 86.5 |
| Conia and Navigli (2020) (BERT) | - | - | 87.3 |
| †Shi et al. (2020) (BERT) | - | - | 85.9 |
| Li et al. (2020a) (RoBERTa) | 88.03 | 79.80 | 86.61 |
| †Xia et al. (2020) (RoBERTa) | 88.59 | 83.16 | - |
| †Marcheggiani and Titov (2020) (RoBERTa) | 87.9 | 80.6 | 86.8 |
| †Zhou et al. (2020) (XLNet) | 89.72 | 84.96 | - |
| Paolini et al. (2021) (T5) | 89.4 | 84.3 | 87.7 |
| Zhang et al. (2021a) | 87.9 | 82.1 | 86.6 |
| †Fei et al. (2021) (RoBERTa) | 89.04 | 83.67 | **88.59** |
| †BERT + $\mathcal{M}$ (Dep.) | 89.02 | 83.34 | 87.12 |
| †XLNet + $\mathcal{M}$ (Dep.) | **89.80** | **85.02** | 87.67 |

Table 4: The comparison of our best performing models (using BERT-large and XLNet-large encoders) with previous studies on the test set of all datasets. Models using extra syntactic features are marked by †.

intuitive because syntactic constituents and dependencies can provide more structure information of the input sentence than POS labels, so that they are more likely to appropriately guide the backbone SRL model to understand the structure of the input sentence, which can further lead to a higher performance.

## 4.2. Comparison with Previous Studies

We further compare our best performing models with recent previous studies on span-style SRL, where the test set results (F1 scores) on CN05 and CN12 datasets are reported in Table 4. It is observed that our approach with BERT encoder outperforms previous studies (Shi and Lin, 2019; Li et al., 2020a) with powerful pre-trained language models (i.e., BERT or RoBERTa) including the ones that are further enhanced by auto-generated syntax trees (Marcheggiani and Titov, 2020) and dependencies (Shi et al., 2020; Xia et al., 2020) on WSJ and Brown. Specifically, although our approach achieves inferior results on CN12 compared with Fei et al. (2021) that uses treeLSTM and label-aware GCN to leverage constituent and dependency trees at the same time, the effectiveness of our approach is still valid given that our approach outperforms their model performance on WSJ and Brown. Besides, compared with studies (Zhou et al., 2020; Conia and Navigli, 2020) that learn gold annotations of different tasks (e.g., parse trees, predicate disambiguation, and semantic role labels) via multi-task learning approaches, our approach with noisy auto-generated syntactic knowledge can still consistently outperform their models.

| | POS | Syn. | Dep. | WSJ | Brown | CN12 |
|---|---|---|---|---|---|---|
| BERT | × | × | × | 88.69 | 81.94 | 86.53 |
| | √ | √ | × | 88.99 | 83.33 | 87.06 |
| | √ | × | √ | 89.08 | 83.39 | 87.17 |
| | × | √ | √ | 89.13 | 83.49 | 87.27 |
| | √ | √ | √ | **89.19** | **83.58** | **87.34** |
| XLNet | × | × | × | 88.91 | 82.12 | 86.87 |
| | √ | √ | × | 89.70 | 84.92 | 87.67 |
| | √ | × | √ | 89.89 | 85.08 | 87.69 |
| | × | √ | √ | 89.92 | 85.11 | 87.74 |
| | √ | √ | √ | **89.98** | **85.19** | **87.80** |

Table 5: The test set results (F1 scores) of our models configured with different combinations of auto-generated knowledge, where BERT-large and XLNet-large encoders are used. "√" denotes a particular type of auto-generated knowledge is used in the memory module whereas "×" represents not.

## 4.3. Effect of Knowledge Ensemble

In the main experiments, we test our approach configured with a single type of auto-generated knowledge (i.e., one of POS labels, syntactic constituents, and dependencies). It is also important to explore how our model can perform if multiple types of knowledge are used. Therefore, we perform knowledge ensemble experiments on our models with BERT-large and XLNet-large encoder, where different combinations of POS labels, syntactic constituents, and dependencies are leveraged in the memory module. We report the experimental results of different models, as well as the baselines without the memory module, on the test set of WSJ, Brown, and CN12 in Table 5. It is observed that, our model with multiple types of knowledge can outperform the BERT and XLNet baselines, which demonstrates the effectiveness of our approach to leverage different types of knowledge at the same time. In addition, compared with models with single type of knowledge (see Table 3), the models with multiple types of knowledge can consistently obtain higher performance on both datasets, where the highest performance is obtained when all three types of knowledge are used. One possible explanation could be that the three types of syntactic knowledge provide cues for SRL from different perspectives, where the combination of them could further enhance the model's understanding to the input text and thus make correct predictions.

## 4.4. Ablation Study

The memory items in our approach consist of keys and values, where the keys stand for the associated context words and the values refer to the corresponding knowledge instances. In our full model, the information from both keys and values are leveraged (see Eq. (3)). To explore the contribution of the keys and values to SRL, we perform an ablation study on our model enhanced by auto-generated dependencies with BERT-

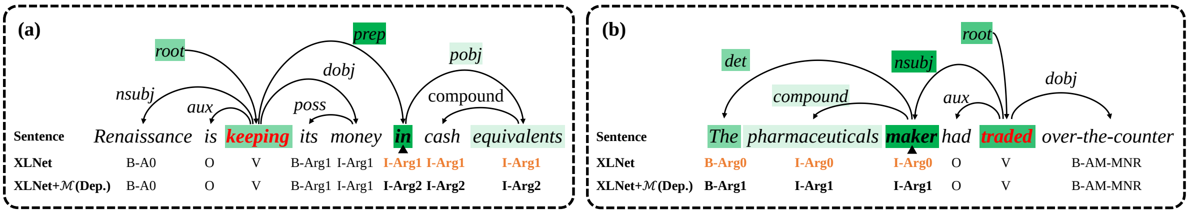

(a)
root
prep
dobj
pobj
nsubj
aux
poss
compound
Sentence Renaissance is keeping its money in cash equivalents
XLNet B-A0 O V B-Arg1 I-Arg1 I-Arg1 I-Arg1 I-Arg1
XLNet+$\mathcal{M}$(Dep.) B-A0 O V B-Arg1 I-Arg1 I-Arg2 I-Arg2 I-Arg2
(b)
root
det
nsubj
dobj
compound
aux
Sentence The pharmaceuticals maker had traded over-the-counter
XLNet B-Arg0 I-Arg0 I-Arg0 O V B-AM-MNR
XLNet+$\mathcal{M}$(Dep.) B-Arg1 I-Arg1 I-Arg1 O V B-AM-MNR

Figure 3: A case study with two example sentences and the predictions of the XLNet baseline and our approach (i.e., XLNet + $\mathcal{M}$ (Dep.)), where our approach correctly predicts the semantic role labels for all words whereas the baseline fails to do so (the incorrect predictions are highlighted in orange color). The given predicates are highlighted in red color. The weights assigned to keys and values in the memory items associated with a particular word marked by ▲ (i.e., "*in*" for (a) and "*maker*" for (b)) are visualized by the green backgrounds on the corresponding context words and knowledge instances, respectively, where deeper color refers to higher weights.

| | Keys | Values | WSJ | Brown | CN12 |
|---|---|---|---|---|---|
| BERT<br>+$\mathcal{M}$(Dep.) | √ | √ | **89.02** | **83.34** | **87.12** |
| | √ | × | 89.76 | 82.89 | 86.77 |
| | × | √ | 89.80 | 83.11 | 86.95 |
| | × | × | 88.69 | 81.94 | 86.53 |
| XLNet<br>+$\mathcal{M}$(Dep.) | √ | √ | **89.80** | **85.02** | **87.67** |
| | √ | × | 89.21 | 84.01 | 87.15 |
| | × | √ | 89.52 | 84.57 | 87.38 |
| | × | × | 88.91 | 82.12 | 86.87 |

Table 6: Experimental results of our models (with BERT-large and XLNet-large encoders) with map memories to encode auto-generated dependencies, where either keys (i.e., the context words) and values (i.e., the knowledge instances) are ablated in the memory module. The results of the full models and BERT and XLNet baselines are also reported for reference.

large and XLNet-large encoder, where either keys (i.e., context words) or values (i.e., knowledge instances) are ablated. We report the experimental results of the models in Table 6, where the results of the full models with both keys and values and the BERT-large and XLNet-large baselines are also reported for references. The observations are as follows. First, compared with the full model, the ablation of either keys and values results in a drop in model performance, which indicates that both context words and knowledge instances are important for SRL. Second, in most cases, the models that only leverage the values achieve higher performance than the ones that only leverage the keys. One possible explanation could be that the knowledge instances (values) carry more structure information than the context words (keys), which allows the model to have a better understanding to the input text and thus achieve higher performance.

### 4.5. Case Study

To give an detailed analysis on the way our approach to leverage auto-generated syntactic knowledge to improve SRL, we conduct a case study with two example input sentences, which are illustrated in Figure 3 with the given predicates (i.e., "*keeping*" for (a) and "*traded*" for (b)) highlighted in red color. For both examples, our approach (i.e., XLNet + $\mathcal{M}$ (Dep.)) can correctly predict the semantic role for each word while the XLNet baseline fails to do so (the incorrect predictions are highlighted in orange color). In addition, we visualize the weights assigned to the keys and values in the memory items associated with "*in*" and "*maker*" (marked by ▲) in the two examples on the corresponding context words and knowledge instances, respectively, where deeper color means higher weights.

In (a), the baseline incorrectly predicts that "*in*" is a component of the patient (i.e., *Arg1*) of the predicte "*keeping*", which may result from that the baseline mistakenly attaches the prepositional phrase "*in cash equivalents*" to the noun phrase "*its money*". On the contrary, our approach with dependency knowledge finds that "*keeping*" is the *prepositional modifier* (i.e., *prep*) of "*in*" and assigns the highest weight to it. Therefore, our approach can discriminatively leverage the dependency information and thus make correct predictions. Similarly, in (b), for the word "*maker*", our approach assigns high weights to the associated context word "*traded*", which is intransitive in this example. Thus, our approach can leverage that information carried by "*traded*" to correctly recognize that "*maker*" is the patient (i.e., Arg1) of "*traded*", because in general, the subject is more likely to be the patient, rather than the agent (i.e., Arg0) of an intransitive verb.

## 5. Related Work

SRL is an important task that attracts much attention from researchers in recent decades (Pradhan et al., 2005; Surdeanu et al., 2007; Johansson and Nugues, 2008; Toutanova et al., 2008; Punyakanok et al., 2008; Täckström et al., 2015; Zhou and Xu, 2015; Tan et al., 2018; He et al., 2018; Ouchi et al., 2018; Li et al., 2020a; Zhang et al., 2021a). To improve model performance, extra knowledge resources, especially the syntactic knowledge, such as POS tags, syntactic constituents, dependencies, and combinatory categorial grammar (CCG), are widely used and proved to be effective (Lewis et al., 2015; Roth and Lapata, 2016; Zhang et al., 2019b; Wang et al., 2019; Marcheggiani and Titov, 2020; Shi et al., 2020; Conia et al., 2021). In these studies, the dependency among the input words is the most widely used knowledge, where graph-based approaches are used to encode the dependency information (Marcheggiani and Titov, 2017; Xia et al., 2020; Fei et al., 2021). In addition, there are other studies (Strubell et al., 2018; Zhou et al., 2020; Conia and Navigli, 2020; Paolini et al., 2021) use multi-task learning approaches, which require the gold standard of different tasks (e.g., parse trees, predicate disambiguation, and semantic role labels) to train their model, or use associated memories (Guan et al., 2019; Li et al., 2020b) to leverage the information in the associated sentences. Compared with previous studies, this paper proposes a neural framework with map memories for SRL, which aims to encode different types of knowledge rather than associated sentences. Also, our approach uses auto-generated syntactic knowledge rather than the gold syntactic trees required by the multi-task learning approaches, so that our approach is more flexible when the human annotated parse trees are not available. Moreover, the proposed memory module assigns a weight to each context word and knowledge instance, so as to distinguish the potential noise in the auto-generated knowledge with respect to the current input and leverage them in a discriminative manner.

## 6. Conclusion

In this paper, we proposed a neural framework with map memories for SRL to encode different types of auto-generated knowledge (i.e., POS labels, syntactic constituents, and dependencies). Specifically, for each word in the input sentence, the memory module extracts its associated memory items from auto-generated knowledge obtained from an off-the-shelf toolkit and assign weights to them, where the keys (i.e., context words) and values (i.e., knowledge instances) in the memory items are weighted according to their contribution to SRL. In doing so, our approach can not only leverage different types of auto-generated knowledge in a uniformed structure, but also smartly address the potential noise in the auto-generated knowledge. Experimental results and further analyses demonstrate the effectiveness of our approach to leverage different types of knowledge, where our approach outperforms strong baselines and previous approaches on two English benchmark datasets for span-style SRL.

## 7. Acknowledgements

This work is supported by Shenzhen Science and Technology Program under the project "Fundamental Algorithms of Natural Language Understanding for Chinese Medical Text Processing" (JCYJ20210324130208022) and the Natural Science Foundation of Guangdong Province, China, under the project "Deep Learning based Chinese Combination Category Grammar Parsing and its Application in Relation Extraction".

## 8. Bibliographical References

Carreras, X. and Màrquez, L. (2005). Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan, June.

Chen, G., Tian, Y., and Song, Y. (2020). Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279, December.

Chen, G., Tian, Y., Song, Y., and Wan, X. (2021). Relation Extraction with Type-aware Map Memories of Word Dependencies. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2501–2512, Online, August.

Conia, S. and Navigli, R. (2020). Bridging the Gap in Multilingual Semantic Role Labeling: a Language-Agnostic Approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1396–1410, Barcelona, Spain (Online), December.

Conia, S., Bacciu, A., and Navigli, R. (2021). Unifying Cross-Lingual Semantic Role Labeling with Heterogeneous Linguistic Resources. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, June.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Diao, S., Bai, J., Song, Y., Zhang, T., and Wang, Y. (2020). ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740, November.

Diao, S., Xu, R., Su, H., Jiang, Y., Song, Y., and Zhang, T. (2021). Taming Pre-trained Language Models with N-gram Representations for Low-Resource Domain Adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3336–3349, Online, August.

Fei, H., Wu, S., Ren, Y., Li, F., and Ji, D. (2021). Better Combine Them Together! Integrating Syntactic Constituency and Dependency Representations for Semantic Role Labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 549–559, Online, August.

FitzGerald, N., Täckström, O., Ganchev, K., and Das, D. (2015). Semantic Role Labeling with Neural Network Factors. In *Proceedings of the 2015*

*Conference on Empirical Methods in Natural Language Processing*, pages 960–970, Lisbon, Portugal, September.

Guan, C., Cheng, Y., and Zhao, H. (2019). Semantic Role Labeling with Associated Memory Network. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3361–3371, Minneapolis, Minnesota, June.

He, L., Lee, K., Lewis, M., and Zettlemoyer, L. (2017). Deep Semantic Role Labeling: What Works and What's Next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada, July.

He, L., Lee, K., Levy, O., and Zettlemoyer, L. (2018). Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia, July.

Johansson, R. and Nugues, P. (2008). Dependency-based Semantic Role Labeling of PropBank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 69–78, Honolulu, Hawaii, October.

Kasai, J., Friedman, D., Frank, R., Radev, D., and Rambow, O. (2019). Syntax-aware Neural Semantic Role Labeling with Supertags. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 701–709, Minneapolis, Minnesota, June.

Komninos, A. and Manandhar, S. (2016). Dependency Based Embeddings for Sentence Classification Tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500, San Diego, California, June.

Lewis, M., He, L., and Zettlemoyer, L. (2015). Joint A* CCG Parsing and Semantic Role Labelling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1444–1454, Lisbon, Portugal, September.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July.

Li, T., Jawale, P. A., Palmer, M., and Srikumar, V. (2020a). Structured Tuning for Semantic Role Labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8402–8412, Online, July.

Li, Z., Guan, C., Zhao, H., Wang, R., Parnow, K., and Zhang, Z. (2020b). Memory Network for Linguistic Structure Parsing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2743–2755.

Li, Z., Zhao, H., Wang, R., and Parnow, K. (2020c). High-order Semantic Role Labeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1134–1151, Online, November.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June.

Marcheggiani, D. and Titov, I. (2017). Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark, September.

Marcheggiani, D. and Titov, I. (2020). Graph Convolutions over Constituent Trees for Syntax-Aware Semantic Role Labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3915–3928, Online, November.

Marcheggiani, D., Bastings, J., and Titov, I. (2018). Exploiting Semantics in Neural Machine Translation with Graph Convolutional Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana, June.

Miller, A., Fisch, A., Dodge, J., Karimi, A.-H., Bordes, A., and Weston, J. (2016). Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409.

Mino, H., Utiyama, M., Sumita, E., and Tokunaga, T. (2017). Key-value Attention Mechanism for Neural Machine Translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 290–295, Taipei, Taiwan, November.

Nie, Y., Tian, Y., Song, Y., Ao, X., and Wan, X. (2020). Improving Named Entity Recognition with Attentive Ensemble of Syntactic Information. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4231–4245, November.

Ouchi, H., Shindo, H., and Matsumoto, Y. (2018). A Span Selection Model for Semantic Role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium, October-November.

Paolini, G., Athiwaratkun, B., Krone, J., Ma, J., Achille, A., Anubhai, R., Santos, C. N. d., Xiang, B., and Soatto, S. (2021). Structured Prediction as Translation between Augmented Natural Languages. *arXiv preprint arXiv:2101.05779*.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (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, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., and Jurafsky, D. (2005). Semantic Role Labeling Using Different Syntactic Views. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 581–588, Ann Arbor, Michigan, June.

Sameer Pradhan, et al., editors. (2012). *Joint Conference on EMNLP and CoNLL - Shared Task*, Jeju Island, Korea, July.

Punyakanok, V., Roth, D., and Yih, W.-t. (2008). The Importance of Syntactic Parsing and Inference in Semantic Role Labeling. *Computational Linguistics*, 34(2):257–287.

Qin, H., Tian, Y., and Song, Y. (2021). Relation Extraction with Word Graphs from N-grams. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2860–2868, Online and Punta Cana, Dominican Republic, November.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *arXiv preprint arXiv:1910.10683*.

Roth, M. and Lapata, M. (2016). Neural Semantic Role Labeling with Dependency Path Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1202, Berlin, Germany, August.

Saha, S., Nie, Y., and Bansal, M. (2020). ConjNLI: Natural Language Inference Over Conjunctive Sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online, November.

Shi, P. and Lin, J. (2019). Simple BERT Models for Relation Extraction and Semantic Role Labeling. *arXiv preprint arXiv:1904.05255*.

Shi, T., Malioutov, I., and Irsoy, O. (2020). Semantic Role Labeling as Syntactic Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7551–7571, Online, November.

Song, Y. and Shi, S. (2018). Complementary Learning of Word Embeddings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4368–4374, 7.

Song, Y., Lee, C.-J., and Xia, F. (2017). Learning Word Representations with Regularization from Prior Knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152, August.

Song, Y., Shi, S., and Li, J. (2018). Joint Learning Embeddings for Chinese Words and Their Components via Ladder Structured Networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4375–4381.

Song, Y., Zhang, T., Wang, Y., and Lee, K.-F. (2021). ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

Strubell, E., Verga, P., Andor, D., Weiss, D., and McCallum, A. (2018). Linguistically-Informed Self-Attention for Semantic Role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium, October-November.

Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. (2015). End-to-end memory networks. *arXiv preprint arXiv:1503.08895*.

Surdeanu, M., Màrquez, L., Carreras, X., and Comas, P. R. (2007). Combination Strategies for Semantic Role Labeling. *Journal of Artificial Intelligence Research*, 29:105–151.

Täckström, O., Ganchev, K., and Das, D. (2015). Efficient Inference and Structured Learning for Semantic Role Labeling. *Transactions of the Association for Computational Linguistics*, 3:29–41.

Tan, Z., Wang, M., Xie, J., Chen, Y., and Shi, X. (2018). Deep Semantic Role Labeling with Self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tian, Y., Shen, W., Song, Y., Xia, F., He, M., and Li, K. (2020a). Improving Biomedical Named Entity Recognition with Syntactic Information. *BMC Bioinformatics*, 21:1471–2105.

Tian, Y., Song, Y., Xia, F., Zhang, T., and Wang, Y. (2020b). Improving Chinese Word Segmentation with Wordhood Memory Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285, Online, July.

Tian, Y., Chen, G., Song, Y., and Wan, X. (2021). Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and* *the 11th International Joint Conference on Natural Language Processing*.

Toutanova, K., Haghighi, A., and Manning, C. D. (2008). A Global Joint Model for Semantic Role Labeling. *Computational Linguistics*, 34(2):161–191.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention Is All You Need. In *Advances in neural information processing systems*, pages 5998–6008.

Wang, Y., Johnson, M., Wan, S., Sun, Y., and Wang, W. (2019). How to Best Use Syntax in Semantic Role Labelling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5338–5343, Florence, Italy, July.

Xia, Q., Wang, R., Li, Z., Zhang, Y., and Zhang, M. (2020). Semantic Role Labeling with Heterogeneous Syntactic Knowledge. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2979–2990, Barcelona, Spain (Online), December.

Xu, K., Lai, Y., Feng, Y., and Wang, Z. (2019). Enhancing Key-Value Memory Neural Networks for Knowledge Based Question Answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2937–2947, Minneapolis, Minnesota, June.

Xu, K., Tan, H., Song, L., Wu, H., Zhang, H., Song, L., and Yu, D. (2020). Semantic Role Labeling Guided Multi-turn Dialogue ReWriter. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6632–6639, Online, November.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763.

Yu, B., Xue, M., Zhang, Z., Liu, T., Wang, Y., and Wang, B. (2020). Learning to Prune Dependency Trees with Rethinking for Neural Relation Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3842–3852.

Zhang, H., Bai, J., Song, Y., Xu, K., Yu, C., Song, Y., Ng, W., and Yu, D. (2019a). Multiplex Word Embeddings for Selectional Preference Acquisition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5247–5256, Hong Kong, China, November.

Zhang, Y., Wang, R., and Si, L. (2019b). Syntax-Enhanced Self-Attention-Based Semantic Role Labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 616–626, Hong Kong, China, November.

Zhang, Z., Strubell, E., and Hovy, E. (2021a). Comparing Span Extraction Methods for Semantic Role Labeling. In *Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021)*, pages 67–77, Online, August.

Zhang, Z., Strubell, E., and Hovy, E. (2021b). On the Benefit of Syntactic Supervision for Cross-lingual Transfer in Semantic Role Labeling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6229–6246, Online and Punta Cana, Dominican Republic, November.

Zhou, J. and Xu, W. (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 (Volume 1: Long Papers)*, pages 1127–1137, Beijing, China, July.

Zhou, J., Li, Z., and Zhao, H. (2020). Parsing All: Syntax and Semantics, Dependencies and Spans. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4438–4449, Online, November.