An MRC Framework for Semantic Role Labeling

Anonymous ACL submission

Abstract

Semantic Role Labeling (SRL) aims at recognizing the predicate-argument structure of a sentence and can be decomposed into two subtasks: predicate disambiguation and argument labeling. Prior work deals with these two tasks independently, which ignores the semantic connection between the two tasks. In this paper, we propose to use the machine reading comprehension (MRC) framework to bridge this gap. We formalize predicate disambiguation as multiple-choice machine reading comprehension, where the descriptions of candidate senses of a given predicate are used as options to select the correct sense. The chosen predicate sense is then used to determine the semantic roles for that predicate, and these semantic roles are used to construct the query for another MRC model for argument labeling. In this way, we are able to leverage both the predicate semantics and the semantic role semantics for argument labeling. We also propose to select a subset of all the possible semantic roles for computational efficiency. Experiments show that the proposed framework achieves state-of-the-art results on both span and dependency benchmarks.

1 Introduction

Semantic Role Labeling (SRL) aims at recognizing the predicate-argument structure of a sentence. The classic PropBank-style SRL includes two tasks: predicate disambiguation and argument labeling. Predicate disambiguation determines the specific meaning of a predicate in a given context and argument labeling identifies the arguments of the predicate and assign them with the corresponding semantic roles, where each argument is a text span in the sentence. PropBank defines two types of semantic roles for argument labeling: core roles and non-core roles (Bonial et al., 2010). Core roles are required roles that are in a close relation to the main verb in a sentence, such as *agent* and *patient*. There are seven core roles in PropBank: A0-A5 and AA. Non-core roles are modifiers, such as location (LOC) and time (TMP). The specific meanings of predicates and core roles are defined in the frame files. For example, for the sentence in Figure 1, the sense id of the predicate "beaten" is "beat.02", and the roles of its three arguments are A1, A2 and TMP, whose arguments are respectively "The stock", "down" and "for two days". We can get the meaning of sense label "beat.02" and roles A1 and A2 from the frame files.

The stock has been beaten down for two days.
[A1] [beat.02][A2][TMP]

sense id	beat.02	
Sense	push, cause motion	
Roles	A0	causer of motion
	A1	thing moving
	A2	direction, destination

Figure 1: An example of SRL. A0, A1 and A2 are *semantic roles* for the sense id "beat.02". The meanings of A0, A1 and A2 are respectively "causer of motion", "thing moving" and "direction, destination".

In traditional methods, predicate disambiguation and argument labeling are usually solved as two independent tasks. These works usually rely on feature-based methods (He et al., 2018b; Roth and Lapata, 2016; Che and Liu, 2010b)) for predicate disambiguation, and use span-based (Ouchi et al., 2018; He et al., 2018a; Li et al., 2019b) or BIO-based (He et al., 2017; Strubell et al., 2018; Shi and Lin, 2019) methods for argument labeling. These methods treat different predicate senses and argument roles as different class categories, and then solve them through classification. However, since these approaches ignore the semantic information of both predicate senses and argument roles, they are unable to establish the semantic connection between the two tasks, i.e., argument roles are defined under predicate sense via the frame files. Some works (Cai et al., 2018; Conia and Navigli, 2020)

Input Sentence
The stock has been < p> beaten </p> down for two days.

Multiple-Choice MRC for Predicate Disambiguation
Question: What is the sense of predicate "beaten"?
A. (Cause) pulsating motion that often makes sound
B. push, cause motion
C. win over some competitor
Answer: B

Extractive MRC for Argument Labeling
Question for A0: What are the arguments with meaning "causer of motion"?
Answer: No Answer
Question for A1: What are the arguments with meaning "thing moving"?
Answer: the stock
Question for A2: What are the arguments with meaning "direction, destination"?
Answer: down
Question for TMP: What are the time modifiers of predicate "beaten"?
Answer: for two days

Figure 2: An illustration of our MRC framework for Semantic Role Labeling. The meanings of predicate senses and argument roles are used for multiple-choice and extractive MRC, respectively.

jointly deal with these two tasks, but still cannot establish the semantic connection. We bridge this gap with an MRC framework, and we hope that the results from predicate disambiguation will contribute to argument labeling.

For PropBank-style semantic role labeling, although the specific meanings of predicate senses and argument roles are provided in the frame files, this information is seldom used due to its huge number and lack of effective ways to utilize it. Inspired by recent success in formulating non-MRC NLP tasks as MRC tasks (Levy et al., 2017; Li et al., 2020c), we propose an MRC framework for SRL, which can effectively utilize the semantic information provided by frame files. First, we transform the predicate disambiguation task into multiple-choice machine reading comprehension, where the descriptions of candidate predicate senses are used as options to select the correct sense. Then, we use the result of predicate disambiguation (i.e., the predicate sense) to determine the meaning of each core role with respect to the predicate. Lastly, we transform argument labeling into extractive machine reading comprehension, where the description of each semantic role is used to construct the query to extract the answer span within the input sentence, which serves as the argument we want. In addition, we also propose an additional module to select a subset of all possible semantic roles to improve computational efficiency. We provide an example of the MRC framework regarding the example of Figure 1 in Figure 2.

We conduct experiments on CoNLL2005 (Carreras and Màrquez, 2005), CoNLL2009 (Hajič et al., 2009), and CoNLL2012 (Pradhan et al., 2013) benchmarks. Experimental results show that our model can achieve SOTA results on the three benchmarks.

In summary, our contributions are as follows:

- To the best of our knowledge, we are the first to exploit the semantic information of both the predicate senses and argument roles provided in the frame files.
- We propose a novel MRC framework to leverage this semantic information, where predicate disambiguation is formalized as multiple-choice MRC and argument labeling is formalized as extractive MRC.
- Our framework is able to obtain SOTA results on three benchmarks without using any syntactic information.

2 Related Work

2.1 Semantic Role Labeling

Early semantic role labeling methods focused on feature engineering (Zhao et al., 2009; Pradhan et al., 2005). Recently, neural network based models have been studied and achieved promising performance. Collobert et al. (2011) proposed a unified neural network architecture and can avoid task-specific engineering. Zhou and Xu (2015) proposed to use BiLSTM as an end-to-end system for SRL. Tan et al. (2018) applied self-attention (Vaswani et al., 2017) mechanism to directly draw the global dependencies of the inputs. Shi and Lin (2019) presented a BERT (Devlin et al., 2019) based model for semantic role labeling. Jindal et al. (2020) propose a parameterized neighborhood memory adaptive method for SRL. Kalyanpur et al. (2020);

Paolini et al. (2021); Blloshmi et al. (2021) cast SRL to a generative translation problem. Zhou et al. (2020); Marcheggiani and Titov (2020) incorporates syntactic information into SRL.

Some works also show that predicate disambiguation is helpful for argument labeling. Che et al. (2010) incorporated a word sense feature to improve the SRL performance. Che and Liu (2010a); Cai et al. (2018); Conia and Navigli (2020) jointly dealt with predicate disambiguation and argument labeling. These methods are different from ours and cannot use this semantic information of the sense label and role label.

2.2 Machine Reading Comprehension

According to the type of the answer, machine reading comprehension can be divided into the following four categories: extractive (Rajpurkar et al., 2016), multiple-choice (Lai et al., 2017), close style (Onishi et al., 2016), and free-form (Nguyen et al., 2016). Related to our work are extractive and multiple-choice MRC. For extractive reading comprehension such as SQuAD (Rajpurkar et al., 2016), the answer is a span in the text, and the MRC model (Seo et al., 2017) gets the answer by predicting the probability that the word is start or end. Some datasets such as DROP (Dua et al., 2019) have answers that include multiple spans, and the answers can be obtained by using BIO tagging (Segal et al., 2019). For multiple-choice reading comprehension where the answer is one of several options, a method (Pan et al., 2019) is to calculate the score for each option and then select the option with the highest score.

2.3 Formalizing Non-MRC Tasks as MRC

Over the past few years, some studies have tried to cast non-MRC tasks as MRC tasks. He et al. (2015) introduce the task of question-answer driven semantic role labeling without predefining an inventory of frames. Levy et al. (2017) show that relation extraction can be reduced to answering simple reading comprehension questions. McCann et al. (2018) frame ten tasks as question answering. Li et al. (2020c) propose to formulate named entity recognition as an MRC task. Other examples include joint entity relation extraction (Li et al., 2019a), coreference resolution (Wu et al., 2020), event extraction (Li et al., 2020a), entity linking (Gu et al., 2021), dependency parsing (Gan et al., 2021), text classification (Chai et al., 2020), etc.

Our approach to formalizing argument labeling as extractive MRC is similar to QA-SRL (He et al., 2015), but we focus on improving the performance of the model on Propbank-style SRL, while (He et al., 2015) aims to provide a new SRL annotation paradigm, and (He et al., 2015) neither uses the predicate sense definitions nor the argument role definitions provided in the frame files.

3 Method

3.1 Overview

An overview of our system is shown in Algorithm 1. Given a sentence $\mathbf{x} = \{x_1, ..., x_n\}$ and the predicate p, the predicate disambiguation task is to determine the predicate sense $s \in S$ of p, where S is the set of all predicate senses, and the argument labeling task is to find all the arguments $A = \{a_1, ..., a_k\}$ of p, where $a_i \in A$ is a text span in $\mathbf{x}$, and assigning them the corresponding semantic roles.

Our framework mainly consists of three modules: predicate disambiguation, role prediction, and argument labeling, all of which use RoBERTa (Liu et al., 2019) as the backbone and use two special symbols < p></p> to mark the predicate p in the input sentence $\mathbf{x}$. The predicate disambiguation module is intended to obtain the predicate sense of the predicate p. Note that we do not use the predicate sense for argument labeling directly, but only use it to get the meanings of the argument roles in the frame files. The role prediction module is to obtain the set of candidate roles for the predicate, and the main purpose of this module is to reduce the number of questions that need to be constructed when solving the argument labeling problem via an extractive MRC. The argument labeling module is used to obtain the arguments of the predicate, which is the core module in the whole framework.

3.2 Multiple-Choice MRC for Predicate Disambiguation

For the predicate disambiguation task, determining the sense label of the predicate involves two steps: identifying the lemma of the predicate, and determining the sense index of the predicate under this lemma. We use spaCy (Honnibal et al., 2020) to identify the lemma of the predicate. If the recognized lemma is not in the frame files, we use the lemma with the smallest edit distance of the predicate. After identifying the lemma, we can find all the senses defined under this lemma from the

Algorithm 1: MRC framework for SRL

Input: sentence $\mathbf{x} = \{x_1, ..., x_n\}$ with marked predicate p, frame files, annotation guidelines
Output: predicate sense $\hat{s}$, arguments A

Get the lemma l of p using SpaCy
Get all the predicate senses S_l of l and the corresponding descriptions D_l^s from the frame files
for s_i in S_l **do**
    Get the description d_i^s of sense s_i from D_l^s
    Concatenate d_i^s and $\mathbf{x}$ to get the input for RoBERTa
    Compute the score of s_i as the answer with Eq.(1)
end for
Select the highest scoring $\hat{s} \in S_l$ as the predicate sense of p
Get the candidate argument roles R_p of p from the role prediction module
for r_i in R_p **do**
    if r_i is core role **then**
        Get the description d_i^r of role r_i from the frame files with $\hat{s}$
    else
        Get the description d_i^r of role r_i from the annotation guidelines
    end if
    Construct query q_i using d_i^r and p
    Concatenate q_i and $\mathbf{x}$ to get the input for RoBERTa
    Calculate the probability that each word in $\mathbf{x}$ belongs to the BIO tags
end for
Decode with non-overlap constraint to get the arguments A of p
return $\hat{s}$, A

frame files, and then we choose the correct sense through multiple-choice reading comprehension.

Specifically, let S_l be all possible senses for the detected lemma. For each sense $s_i \in S_l$, the corresponding sense description is d_i^s. We treat d_i^s as option, and the input for the RoBERTa is the concatenation of d_i^s and $\mathbf{x}$. The probability score of s_i as the correct sense is calculated by:

$$P(s_i = 1 | d_i^s, \mathbf{x}, p) = \text{sigmoid}(\text{FFN}_\text{p}(h^d)) \quad (1)$$

where h^d is the context representation of the first input token from RoBERTa and FFN_p is a single layer feedforward neural network. We train the model using the binary cross-entropy loss function. [1] During inference, we choose the sense with the highest probability score among all the sense options as the answer.

3.3 Role Prediction

In semantic role labeling, most semantic roles do not have corresponding arguments given a specific input sentence. For example, in the CoNLL2005 dataset, there is a total number of 20 roles, but on average there are only 2.5 roles per predicate. Therefore, we use a role prediction module to avoid asking questions about impossible roles at the next argument labeling stage, reducing the amount of calculation required when using the MRC-based method.

Let R be the set of all semantic roles (in CoNLL 2005 the size of R is 20), the purpose of role prediction is to predict a set of possible roles $R_p \subseteq R$ for the predicate p. The input to RoBERTa is the sentence $\mathbf{x}$ with the marked predicate p. Let h^r be the context representation of the first token of the input sequence from RoBERTa, and $r_i \in R$ is the i-th role of R. We use the sigmoid function to calculate the probability that the predicate p has a role r_i:

$$P(r_i = 1 | \mathbf{x}, p) = \text{sigmoid}(\text{FFN}_{\text{r}_\text{i}}(\text{h}^\text{r})) \quad (2)$$

where $\text{FFN}_{\text{r}_\text{i}}$ is a single layer feedforward neural network. We use the binary cross entropy loss function to train the model. During inference, we only keep up to λN roles with the highest probability score, where N is the number of predicates in the dataset. [2] Note that here we select the roles with the top λN probability scores on the whole dataset, not on the input sentence. And in the argument labeling module, we use the predicted roles from the role prediction module instead of the gold roles for training.

3.4 Extractive MRC for Argument Labeling

We formalize argument labeling as extractive reading comprehension, where the meaning of argument role is used to construct the query, and since the answer may contain multiple spans, we use BIO

[1] We also tried to use softmax to get the probability of all senses, and then use the multi-class cross entropy loss for training, but we found the loss is unstable and hard to optimize.

[2] An alternative strategy is to use a fixed threshold, which performs similarly to ours. But our strategy can directly get the number of argument roles, which helps to analyze the amount of computation needed in argument labeling.

tagging to extract the arguments. [3] In ProbBank-style SRL, a role may be a norm role, a reference role, or a continuation role. A norm role is a standard role defined in the annotation guidelines, a reference role is a reference to some other arguments, and a continuation role is a continuation phrase of a previously started argument. For example, role A1 may be N-A1 or R-A1, or C-A1. Since the subcategories of N/R/C do not contain semantic information, we do not encode such information into the query of the MRC model. We use BIO tagging to get the arguments of the predicate, and the set of BIO tags is

$$T = \{B, I\} \times \{N, R, C\} \cup \{O\} \quad (3)$$

We use templates to construct the query of the MRC model. For core roles, our template is *"What are the X arguments of predicate Y with meaning Z?"*, where X is the role type(e.g., "A0"), Y is the predicate, and Z is the description of role X in the frame files. For non-core roles, our template is *"What are the W modifiers of predicate Y?"*, where W is the specific meanings of non-core roles defined in the annotation guidelines.

Specifically, let q_i represent the query corresponding to the predicate p and the role $r_i \in R_p$, the input of the MRC model is the concatenation of q_i and $\mathbf{x}$. The context representation of $\mathbf{x}$ in the input $< q_i, \mathbf{x} >$ pair is $\mathbf{h}^{r_i} = \{h_1^{r_i}, ..., h_n^{r_i}\}$, our goal is to predict $\mathbf{y}^{r_i} = \{y_1^{r_i}, ..., y_n^{r_i}\}$. Each $y_j^{r_i} \in \mathbf{y}^{r_i}$ belongs to one tag in the tag set T. For $y_j^{r_i}$, its probability distribution on BIO tag set is calculated by a softmax layer:

$$P(y_j^{r_i} = t|\mathbf{x}, p, r_i) \propto \exp(\mathrm{W}_{\boldsymbol{t}} h_j^{r_i} + \boldsymbol{b_t}) \quad (4)$$

where $t \in T$ is a BIO tag, $\mathrm{W}_{\boldsymbol{t}}$ and $\boldsymbol{b_t}$ are the corresponding parameters. We use multi-class cross entropy loss to train the model. And we use the method in section 3.5 to get the argument.

Note that at this stage, we use the predicate sense extracted at the predicate disambiguation stage to find the sense of each role selected at the role prediction stage. For example, suppose the predicate sense is "beat.02" and the semantic role is A0 as shown in Figure 1, we will immediately obtain the role's sense "causer of motion". In this way, the predicate sense can be leveraged for role sense detection, and thus further for semantic labeling, bridging the gap between the two tasks via an MRC framework.

[3] For dependency semantic role labeling, since pre-trained language models such as BERT split a word into multiple subwords, which is similar to span, BIO tagging is also applicable.

3.5 Constrained Decoding

There are many global constraints in semantic role labeling (Punyakanok et al., 2008; Li et al., 2020b), such as all arguments of the predicate cannot overlap and each core role should appear at most once for each predicate. Our MRC approach does not directly model these constraints and can not guarantee that the obtained results satisfy these constraints. For simplicity, we only consider the non-overlap arguments constraint. The previous approach of using BIO tagging (He et al., 2017; Shi and Lin, 2019) to extract arguments can naturally model the non-overlap constraint, since each word in $\mathbf{x}$ can only belong to one of the BIO tags, there will be no overlapping words between the argument elements. But in our MRC-based BIO tagging method, since we have R_p roles, each word has at most R_p BIO tags. We implement the non-overlap constraint by mapping the local role-related BIO tag of each word into a global BIO tag set.

Specifically, for the sentence $\mathbf{x} = \{x_1, ..., x_n\}$, the goal of constraint decoding is to obtain the corresponding tag sequence $\mathbf{y} = \{y_1, ..., y_n\}$, where $y_j \in \mathbf{y}$ belongs to the tag set T_p:

$$T_p = R_p \times \{B, I\} \times \{N, R, C\} \cup \{O\} \quad (5)$$

For tag $t_p \in T_p$, when it is a BI tag, it can be expressed as r_i-t, where $r_i \in R_p$ and $t \in T$. For BI tags, we add a role tag directly before the original BI tag. For example, the B-R tag of role A1 will be converted to A1-B-R, and then the score of the new tag is equal to the probability of the original tag:

$$\begin{aligned} s(y_j = t_p) &= s(y_j = r_i\text{-}t) \\ &= p(y_j^{r_i} = t) \end{aligned} \quad (6)$$

where $s(\cdot)$ is the score function. For O tags, we merge the O tags of different roles into one O tag, and the score of O tag after merging is the product of the O tag probabilities of all roles.

$$s(y_j = O) = \prod_{i=1}^{|R_p|} p(y_j^{r_i} = O) \quad (7)$$

During inference, for each word x_i, its tag y_j is the highest scoring tag in the new BIO tag set T_p.

$$y_j = \arg\max_{t_p \in T_p} s(y_j = t_p) \quad (8)$$

And we use the BIO tag sequence $\mathbf{y}$ to get all the arguments.

Model	Dev	WSJ	Brown
Shi and Zhang (2017)	-	93.4	82.4
Roth and Lapata (2016)	94.8	95.5	-
He et al. (2018b)	95.0	95.6	-
Shi and Lin (2019)+BERT	96.3	96.9	90.6
Ours+BERT	96.3	97.2	**91.9**
Ours+RoBERTa	**96.6**	**97.3**	91.3
Ours-semantics	96.2	96.7	89.9

Table 1: Predicate disambiguation results on CoNLL2009.

	CoNLL09 WSJ			CoNLL09 Brown		
Model	P	R	F1	P	R	F1
syntax-aware						
Cai and Lapata (2019)	91.7	90.8	91.2	83.2	81.9	82.5
Kasai et al. (2019)	90.3	90.0	90.2	81.0	80.5	80.8
Zhou et al. (2020)+BERT	91.2	91.2	91.2	85.7	86.1	85.9
Fei et al. (2021)+RoBERTa	92.9	92.8	92.8	-	-	-
syntax-agnostic						
Li et al. (2019b)	89.6	91.2	90.4	81.7	81.4	81.5
Conia and Navigli (2020)+BERT	92.5	92.7	92.6	-	-	85.9
Shi and Lin (2019)+BERT	92.4	92.3	92.4	85.7	85.8	85.7
Jindal et al. (2020)+BERT	90.0	91.5	90.8	83.5	86.5	85.0
Ours+BERT	93.3	92.7	93.0	87.5	86.6	87.0
Ours+RoBERTa	**93.5**	**93.1**	**93.3**	**87.7**	**86.6**	**87.2**

Table 2: Argument labeling results on CoNLL2009.

4 Experiments

4.1 Datasets

We conduct experiments on the CoNLL2005, CoNLL2009 and CoNLL2012 datasets. The CoNLL2005 and CoNLL2012 datasets are span-based SRL, where the arguments are constituents (spans) in the sentence, and the CoNLL2009 dataset is dependency-based SRL, where the arguments are syntactic heads. The CoNLL2005 dataset consists of sections of the Wall Street Journal part of the Penn TreeBank, where section 2-21 is used for training, section 24 is used for development, and section 23 is used for evaluation. In addition, it also includes three sections of the Brown corpus to test the robustness of the systems. The CoNLL2009 dataset uses the same corpus as CoNLL2005, but uses NomBank to extend the annotations. The CoNLL2012 dataset is extracted from the OntoNotes v5.0 corpus. The frame files are available as official resources in the three datasets and can be used by all systems.

4.2 Experiment Setup

For data preprocessing we follow (Li et al., 2019b). We use RoBERTa Large as the base encoder and we use two special symbols < p> and </p> to mark the predicate of the input sentence. We adopt Adam as optimizer, and the warmup rate is 0.05, the initial learning rate is 1e-5, the maximum number of epochs is 20, the number of tokens in each batch is 2048. λ is tuned on development set to ensure that the recall of the predicted roles is higher than 99%. All the experiments were conducted on a Tesla V100 GPU with 16GB memory.

Predicate disambiguation is evaluated using accuracy, and argument labeling is evaluated using micro F1. The evaluation of argument labeling in CoNLL2009 also includes the results of predicate disambiguation, where the predicate sense is treated as a special kind of argument of a virtual root node.

4.3 Main Results

Predicate Disambiguation We evaluate the performances of predicate disambiguation on the CoNLL2009 dataset as previous work on the CoNLL2005 and CoNLL2012 datasets did not consider predicate disambiguation. The error of lemma recognition is also included in the final results. In Table 1, we report the experimental results of our method when using BERT and RoBERTa as encoders. The model using RoBERTa achieves the best results on the development set and on the in-domain test set (WSJ), and the model using BERT achieves the best results on the out-of-domain test set (Brown). The performances of BERT and RoBERTa on the development and brown test sets are opposite, which indicates that the evaluation on the development set does not fully reflect the model's generalization ability.

To investigate the impact of the sense descriptions provided by the frame files, we also give the experimental results without using this semantic information in Table 1 ("-semantics"). In this setting, we also use RoBERTa, but the predicate sense description is replaced by the corresponding numeric label (e.g., "02" in "beat.02"). The experimental results show that the model performs worse when this semantic information is not available, especially in the out-of-domain Brown test set, where the accuracy decreases by 1.4%.

Argument Labeling Table 2 shows the results for dependency SRL, and Table 3 shows the experimental results for span SRL. Compared with the previous SOTA, [4] our improvement on the

[4] Fei et al. (2021) only reported the experimental results when using gold instead of predicted syntactic information, so we do not compare with it.

Model	CoNLL05 WSJ			CoNLL05 Brown			CoNLL12 Test		
	P	R	F1	P	R	F1	P	R	F1
syntax-aware									
Zhou et al. (2020)+BERT	89.0	88.8	88.9	81.9	81.0	81.4	-	-	-
Mohammadshahi and Henderson (2021)+BERT	89.1	88.7	88.9	83.9	82.5	83.2	-	-	-
Xia et al. (2020)+RoBERTa	88.4	88.8	88.6	83.1	83.3	83.2	-	-	-
Marcheggiani and Titov (2020)+RoBERTa	87.7	88.1	88.0	80.5	80.7	80.6	86.5	87.1	86.8
Fei et al. (2021)+RoBERTa	88.9	89.3	89.0	83.5	83.8	83.7	88.1	88.8	88.6
syntax-agnostic									
Li et al. (2019b)	87.9	87.5	87.7	80.6	80.4	80.5	85.7	86.3	86.0
Conia and Navigli (2020)+BERT	-	-	-	-	-	-	86.9	87.7	87.3
Blloshmi et al. (2021)+BART	-	-	-	-	-	-	87.8	86.8	87.3
Shi and Lin (2019)+BERT	88.6	89.0	88.8	81.9	82.1	82.0	85.9	87.0	86.5
Jindal et al. (2020)+BERT	88.7	88.0	87.9	80.3	80.1	80.2	86.3	86.8	86.6
Paolini et al. (2021)+T5	-	-	89.3	-	-	82.0	-	-	87.7
Ours+BERT	89.7	89.0	89.3	85.9	83.5	84.7	88.0	87.7	87.8
Ours+RoBERTa	**90.4**	**89.7**	**90.0**	**86.4**	**83.8**	**85.1**	**88.6**	**87.9**	**88.3**

Table 3: Argument labeling results on CoNLL2005 and CoNLL2012.

in-domain WSJ test sets of CoNLL2005 and CoNLL2009 is 0.7 and 0.7, respectively, on the out-of-domain Brown test set is 1.9 and 0.8, respectively, and on the CoNLL2012 test set is 0.6. Since our method is syntax-agnostic, we first compare it with the syntax-agnostic methods. Our method improves more on the out-of-domain Brown test set of CoNLL2005 and CoNLL2009 than on the in-domain WSJ test set, which indicates that our method has stronger generalization ability than the previous syntax-agnostic methods. The syntax-aware method (Mohammadshahi and Henderson, 2021) also performs better on the Brown test set compared to the syntax-agnostic methods (Shi and Lin, 2019), a similar phenomenon to ours. However, unlike the syntax-aware approach, our approach is syntax-agnostic and utilizes the semantic information provided in the frame files rather than the syntactic information of the sentence, and outperforms syntax-aware methods. This observation demonstrates that leveraging semantic information in frame files provides stronger generality than syntax-aware techniques for SRL.

5 Ablation studies

5.1 Effect of Predicate Disambiguation

Our framework uses a pipelined approach to connect the predicate disambiguation and the argument labeling task, so different predicate disambiguation accuracies may affect the results of argument labeling. Here we analyze the performance of the same argument labeling model with different predicate disambiguation accuracies. We obtain the results of different predicate disambiguation accuracies through randomly replacing part of the gold predicate senses with other predicate senses. Then we use the ordinarily trained argument labeling model to make predictions under different predicate disambiguation results. Figure 3 shows that the model performs monotonically worse as the predicate disambiguation accuracy decreases, with the rate of decline for recall being much faster than that for precision. This suggests that when an incorrect predicate disambiguation result is obtained, our argument labeling model usually does not lead to the correct argument. Thus, an accurate predicate disambiguation model is required.

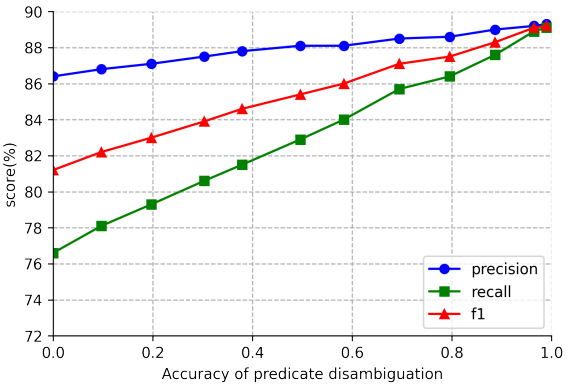

Figure 3: Experimental results of different predicate disambiguation accuracies.

5.2 Effect of Argument Role Semantics

We also study the performance of our MRC framework in the case where the query does not contain any semantics, and in this case, the query is replaced with a category label. We use RoBERTa base for our experiments. When role semantics

Recall	0.90	0.93	0.96	0.99
F1	87.3	88.4	88.2	88.6

Table 4: Experimental results of different role prediction recall scores.

decode	P	R	F1
overlap	85.4	**84.0**	84.7
non-overlap	**86.2**	83.9	**85.0**

Table 5: Experimental results of different decoding methods.

is not considered, the F1 scores on the development set and the out-of-domain Brown test set of CoNLL2005 are 88.2 and 83.2, respectively. when role semantics is considered, the F1 scores on the development set and the out-of-domain Brown test set of CoNLL2005 are 88.5 and 83.8, respectively. The experimental results show that taking semantics into account performs better than not taking semantics into account, especially when the domains of the training and test sets are different. And this proves that the semantics of the argument roles is useful in our framework.

5.3 Effect of Role Prediction

The main purpose of role prediction is to predict the possible argument roles of predicates, which can reduce the number of questions that need to be asked by the argument labeling module. Since role prediction is an upstream task of argument labeling, missing potential argument roles in the role prediction stage can lead to the error propagation problem. We mitigate this problem by ensuring that the recall of role prediction is higher than 99% and training the argument labeling model under the predicted roles. Table 4 shows the influence of different role prediction recall scores on argument labeling. It can be seen that when the recall is low, the F1 score of argument labeling will decrease significantly – 87.3 when recall is 0.90 versus 88.6 when recall is 0.99.

5.4 Computational Efficiency

In our MRC framework, to utilize the semantic information of labels (predicate senses and argument roles), we need to encode all <label, sentence> pairs using a pre-trained model, which can be computationally intensive if the number of labels is large, we mitigate this problem by filtering impossible labels. [5] In predicate disambiguation, we use lemma to filter impossible predicate senses, and in argument labeling, we use an additional role prediction module to filter impossible roles. Since the main computation in our framework is spent on the argument labeling module, here we give a rough analysis of the computational overhead it requires. In section 3.3, we select the λN roles with the highest probability scores in the dataset, which are used in the argument labeling module to construct queries, so λN reflects the amount of computation we need in the argument labeling module. When $\lambda = R$, this approach is equivalent to asking questions directly to all roles. In CoNLL2005, CoNLL2009, and CoNLL2012, the total number of semantic roles are 20, 20, 28, respectively, and the actual λs in the role prediction module are 5, 4.2, 5.5, respectively. The value of λ is much smaller than the number of all semantic roles, and this indicates that our model achieves approximately 4x, 4.8x and 5.1x speedups in CoNLL2005, CoNLL2009, and CoNLL2012 compared to asking questions directly to all roles.

[5] We also tried to decouple the label and sentence encoding to avoid encoding the same sentence multiple times, but it did not perform as well as the simple filtering strategy.

5.5 Effect of Constrained Decoding

In this section, we study the effect of the non-overlap constraint. Experimental results in Table 5 show that when not considering the non-overlap constraint, the model recall is higher, but the precision and F1 score are lower than when considering the non-overlap constraint. In our experiments, the F1 scores are improved most of the time after using constraint decoding, and all experimental results reported in this paper are obtained with constraint decoding.

6 Conclusion

In this paper, we propose an MRC-based framework for semantic role labeling. We formalize predicate disambiguation as multiple-choice reading comprehension and argument labeling as extractive reading comprehension. Besides, we also propose a role prediction module to reduce the computation caused by considering all roles in the dataset for argument labeling. Experimental results show that our framework can effectively use the semantic information of argument roles and achieve SOTA performance on three benchmarks.

References

Rexhina Blloshmi, Simone Conia, Rocco Tripodi, and Roberto Navigli. 2021. Generating senses and roles: An end-to-end model for dependency- and span-based semantic role labeling. In *IJCAI*.

Claire Bonial, Olga Babko-Malaya, Jinho D Choi, Jena Hwang, and Martha Palmer. 2010. 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, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765.

Rui Cai and Mirella Lapata. 2019. Semi-supervised semantic role labeling with cross-view training. In *EMNLP*.

Xavier Carreras and Lluís Màrquez. 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.

Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description based text classification with reinforcement learning. In *International Conference on Machine Learning*, pages 1371–1382. PMLR.

Wanxiang Che and Ting Liu. 2010a. Jointly modeling wsd and srl with markov logic. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 161–169.

Wanxiang Che and Ting Liu. 2010b. Using word sense disambiguation for semantic role labeling. In *2010 4th International Universal Communication Symposium*, pages 167–174. IEEE.

Wanxiang Che, Ting Liu, and Yongqiang Li. 2010. Improving semantic role labeling with word sense. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 246–249.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Simone Conia and R. Navigli. 2020. Bridging the gap in multilingual semantic role labeling: a language-agnostic approach. In *COLING*.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378.

Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. 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. Association for Computational Linguistics.

Leilei Gan, Yuxian Meng, Kun Kuang, Xiaofei Sun, Chun Fan, Fei Wu, and Jiwei Li. 2021. Dependency parsing as mrc-based span-span prediction. *arXiv preprint arXiv:2105.07654*.

Yingjie Gu, Xiaoye Qu, Z. Wang, Baoxing Huai, Nicholas Jing Yuan, and Xiaolin Gui. 2021. Read, retrospect, select: An mrc framework to short text entity linking. *ArXiv*, abs/2101.02394.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štepánek, et al. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages.

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 (Volume 2: Short Papers)*, 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 (Volume 1: Long Papers)*, pages 473–483.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 643–653.

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 (Volume 1: Long Papers)*, pages 2061–2071, Melbourne, Australia. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Ishan Jindal, Ranit Aharonov, Siddhartha Brahma, Huaiyu Zhu, and Yunyao Li. 2020. Improved semantic role labeling using parameterized neighborhood memory adaptation. *arXiv preprint arXiv:2011.14459*.

Aditya Kalyanpur, Or Biran, Tom Breloff, Jennifer Chu-Carroll, Ariel Diertani, Owen Rambow, and Mark Sammons. 2020. Open-domain frame semantic parsing using transformers. *arXiv preprint arXiv:2010.10998*.

Jungo Kasai, Dan Friedman, R. Frank, Dragomir R. Radev, and Owen Rambow. 2019. Syntax-aware neural semantic role labeling with supertags. In *NAACL*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.

Fayuan Li, Wei-Hua Peng, Y. Chen, Q. Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020a. Event extraction as multi-turn question answering. In *EMNLP*.

Tao Li, Parth Anand Jawale, Martha Palmer, and Vivek Srikumar. 2020b. Structured tuning for semantic role labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8402–8412.

Xiaoya Li, J. Feng, Yuxian Meng, Qinghong Han, F. Wu, and J. Li. 2020c. A unified mrc framework for named entity recognition. *ArXiv*, abs/1910.11476.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and J. Li. 2019a. Entity-relation extraction as multi-turn question answering. In *ACL*.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019b. Dependency or span, end-to-end uniform semantic role labeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6730–6737.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Diego Marcheggiani and Ivan Titov. 2020. Graph convolutions over constituent trees for syntax-aware semantic role labeling. In *EMNLP*.

B. McCann, N. Keskar, Caiming Xiong, and R. Socher. 2018. The natural language decathlon: Multitask learning as question answering. *ArXiv*, abs/1806.08730.

Alireza Mohammadshahi and J. Henderson. 2021. Syntax-aware graph-to-graph transformer for semantic role labelling. *ArXiv*, abs/2104.07704.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Takashi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Xiaoman Pan, Kai Sun, Dian Yu, Jianshu Chen, Heng Ji, Claire Cardie, and Dong Yu. 2019. Improving question answering with external knowledge. *arXiv preprint arXiv:1902.00993*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H Martin, and Dan Jurafsky. 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.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

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 (Volume 1: Long Papers)*, pages 1192–1202, Berlin, Germany. Association for Computational Linguistics.

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2019. A simple and effective model for answering multi-span questions. *arXiv preprint arXiv:1909.13375*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Peng Shi and Yue Zhang. 2017. Joint bi-affine parsing and semantic role labeling. In *2017 International Conference on Asian Language Processing (IALP)*, pages 338–341. IEEE.

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*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Zhixing Tan, Mingxuan Wang, J. Xie, Y. Chen, and X. Shi. 2018. Deep semantic role labeling with self-attention. In *AAAI*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

W. Wu, F. Wang, Arianna Yuan, F. Wu, and J. Li. 2020. Corefqa: Coreference resolution as query-based span prediction. In *ACL*.

Qingrong Xia, Rui Wang, Zhenghua Li, Y. Zhang, and Min Zhang. 2020. Semantic role labeling with heterogeneous syntactic knowledge. In *COLING*.

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

Jie Zhou and W. Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *ACL*.

Junru Zhou, Z. Li, and Hai Zhao. 2020. Parsing all: Syntax and semantics, dependencies and spans. In *FINDINGS*.