I Know What You Want: Semantic Learning for Text Comprehension

Zhuosheng Zhang[1,2], **Yuwei Wu**[1,2], **Zuchao Li**[1,2], **Shexia He**[1,2], **Hai Zhao**[1,2,*]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering,
Shanghai Jiao Tong University, Shanghai, 200240, China
{zhangzs, will8821, heshexia, charlee}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,
{zhouxi, zhouxiang}@cloudwalk.cn

Abstract

Who did what to whom is a major focus in natural language understanding, which is right the aim of semantic role labeling (SRL). Although SRL is naturally essential to text comprehension tasks, it is surprisingly ignored in previous work. This paper thus makes the first attempt to let SRL enhance text comprehension and inference through specifying verbal arguments and their corresponding semantic roles. In terms of deep learning models, our embeddings are enhanced by semantic role labels for more fine-grained semantics. We show that the salient labels can be conveniently added to existing models and significantly improve deep learning models in challenging text comprehension tasks. Extensive experiments on benchmark machine reading comprehension and inference datasets verify that the proposed semantic learning helps our system reach new state-of-the-art.

Introduction

Text comprehension (TC) is challenging for it requires computers to read and understand natural language texts to answer questions or make inference, which is indispensable for advanced context-oriented dialogue and interactive systems. This paper focuses on two core text comprehension tasks, *machine reading comprehension* (MRC) and *textual entailment (TE)*.

One of the intrinsic challenges for TC is the semantic learning. Though deep learning has been applied to a variety of natural language processing (NLP) tasks with remarkable performance (Zhang et al. 2018; Zhu et al. 2018; Zhang and Zhao 2018), recent studies have found deep learning models might not really understand the natural language texts (Mudrakarta et al. 2018) and vulnerably suffer from adversarial attacks (Jia and Liang 2017). Typically, an MRC model pays great attention to non-significant words and ignores important terms and actions. For example, most of the highly attributed words such as *there, be, how* are usually less important in questions. To help model better understand natural language, we are motivated to discover an effective way to distill the semantics inside the input sentence explicitly, such as semantic role labeling, instead of completely relying on uncontrollable model parameter learning or manual pruning techniques.

Semantic role labeling (SRL) is a shallow semantic parsing task aiming to discover *who* did *what* to *whom*, *when* and *why*, which naturally matches the task target of text comprehension. For MRC, questions are usually formed with *who*, *what*, *how*, *when* and *why*, whose predicate-argument relationship that is supposed to be from SRL is of the same importance as well. Besides, SRL has been proved to be beneficial to a wide range of NLP tasks, including discourse relation sense classification (Mihaylov and Frank 2016), machine translation (Shi et al. 2016) and question answering (Yih et al. 2016). All the previous successful work indicates that SRL may be hopefully integrated into reading comprehension and inference tasks.

Some work studied question answering (QA) driven SRL, like QA-SRL parsing (He, Lewis, and Zettlemoyer 2015; Fitzgerald, He, and Zettlemoyer 2018). They focus on detecting argument spans for a predicate and generating questions to annotate the semantic relationship. However, our task is quite different. In QA-SRL, the focus is commonly simple and short factoid questions that are less related to the context, let alone makes inference. Actually, text comprehension and inference are quite challenging tasks in NLP, requiring to dig the deep semantics between the document and comprehensive question which are usually raised or rewritten by humans, instead of shallow argument alignment around the same predicate in QA-SRL. In this work, to alleviate such an obvious shortcoming about semantics, we make the first attempt to explore integrative models for finer-grained text comprehension and inference. In this work, we propose an SRL-based enhancement framework for TC tasks, which boosts the strong baselines effectively. We implement an easy and feasible scheme to integrate SRL signals in downstream neural models in end-to-end manner. An example about how SRL helps MRC is illustrated in Figure 1. A series of detailed case studies are employed to analyze the robustness of the semantic role labeler. To our best knowledge, our work is the first attempt to apply SRL for text comprehension tasks, which have been ignored in previous works for a long time.

The rest of this paper is organized as follows. The next section reviews the related work. Section 3 will demonstrate our semantic learning framework and implementation. Task details and experimental results are reported in Section 4, followed by case studies and analysis in Section 5 and con-

* Corresponding author.

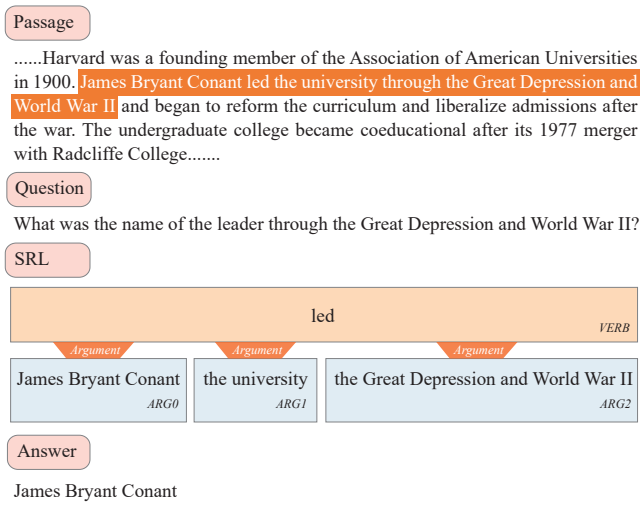

Figure 1: Semantic role labeling guides text comprehension.

Passage	There are three major types of rock: igneous, sedimentary, and metamorphic. The rock cycle is an important concept in geology which illustrates the relationships between these three types of rock, and magma. When a rock crystallizes from melt (magma and/or lava), it is an igneous rock. This rock can be weathered and eroded, and then redeposited and lithified into a sedimentary rock, or be turned into a metamorphic rock due to heat and pressure that change the mineral content of the rock which gives it a characteristic fabric. The sedimentary rock can then be subsequently turned into a metamorphic rock due to heat and pressure and is then weathered, eroded, deposited, and lithified, ultimately becoming a sedimentary rock. Sedimentary rock may also be re-eroded and redeposited, and metamorphic rock may also undergo additional metamorphism. All three types of rocks may be re-melted; when this happens, a new magma is formed, from which an igneous rock may once again crystallize.
Question	What changes the mineral content of a rock?
Answer	heat and pressure.

Table 1: A machine reading comprehension example.

Premise	A man parasails in the choppy water.	Label
Hypothesis	The man is competing in a competition.	Neutral
	The man parasailed in the calm water.	Contradiction
	The water was choppy as the man parasailed.	Entailment

Table 2: A textual entailment example.

clusion in Section 6.

Related Work

Text Comprehension

As a challenging task in NLP, text comprehension is one of the key problems in artificial intelligence, which aims to read and comprehend a given text, and then answer questions or make inference based on it. These tasks require a comprehensive understanding of natural languages and the ability to do further inference and reasoning. We focus on two types of text comprehension, document-based question-answering (Table 1) and textual entailment (Table 2). Textual entailment aims for a deep understanding of text and reasoning, which shares the similar genre of machine reading comprehension, though the task formations are slightly different.

In the last decade, the MRC tasks have evolved from the early cloze-style test (Hill et al. 2015; Hermann et al. 2015; Zhang, Huang, and Zhao 2018) to span-based answer extraction from passage (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018). The former has restrictions that each answer should be a single word in the document and the original sentence without the answer part is taken as the query. For the span-based one, the query is formed as questions in natural language whose answers are spans of texts. Notably, Chen, Bolton, and Manning (2016) conducted an in-depth and thoughtful examination on the comprehension task based on an attentive neural network and an entity-centric classifier with a careful analysis based on handful features. Then, various attentive models have been employed for text representation and relation discovery, including Attention Sum Reader (Kadlec et al. 2016), Gated attention Reader (Dhingra et al. 2017), Self-matching Network (Wang et al. 2017) and Attended over Attention Reader (Cui et al. 2017).

With the release of the large-scale span-based datasets (Rajpurkar et al. 2016; Nguyen et al. 2016; Joshi et al. 2017; Wang et al. 2018; Rajpurkar, Jia, and Liang 2018), which constrain answers to all possible text spans within the reference document, researchers are investigating the models with more logical reasoning and content understanding (Wang et al. 2018; Wang, Yan, and Wu 2018).

For the other type of text comprehension, natural language inference (NLI) is proposed to serve as a benchmark for natural language understanding and inference, which is also known as recognizing textual entailment (RTE). In this task, a model is presented with a pair of sentences and asked to judge the relationship between their meanings, including entailment, neutral and contradiction. Bowman et al. (2015) released Stanford Natural language Inference (SNLI) dataset, which is a high-quality and large-scale benchmark, thus inspiring various significant work.

Most of existing NLI models apply attention mechanism to jointly interpret and align the premise and hypothesis, while transfer learning from external knowledge is popular recently. Notably, Chen et al. (2017) proposed an enhanced sequential inference model (ESIM), which employed recursive architectures in both local inference modeling and inference composition, as well as syntactic parsing information, for a sequential inference model. ESIM is simple with satisfactory performance, and thus is widely chosen as the baseline model. Mccann et al. (2017) proposed to transfer the LSTM encoder from the neural machine translation (NMT) to the NLI task to contextualize word vectors. Pan et al. (2018) transfered the knowledge learned from the discourse marker prediction task to the NLI task to augment the semantic representation.

Semantic Role Labeling

Given a sentence, the task of semantic role labeling is dedicated to recognizing the semantic relations between the predicates and the arguments. For example, given the sentence, *Charlie sold a book to Sherry last week*, where the target verb (predicate) is *sold*, SRL yields the following outputs,

[$_{ARG0}$ Charlie] [$_{V}$ sold] [$_{ARG1}$ a book]
[$_{ARG2}$ to Sherry] [$_{AM-TMP}$ last week].

where $ARG0$ represents the seller (agent), $ARG1$ represents the thing sold (theme), $ARG2$ represents the buyer (recipient), $AM-TMP$ is an adjunct indicating the timing of the action and V represents the predicate.

Recently, SRL has aroused much attention from researchers and has been applied in many NLP tasks (Mihaylov and Frank 2016; Shi et al. 2016; Yih et al. 2016). SRL task is generally formulated as multi-step classification subtasks in pipeline systems, consisting of predicate identification, predicate disambiguation, argument identification and argument classification. Most previous SRL approaches adopt a pipeline framework to handle these subtasks one after another. Notably, Gildea and Jurafsky (2002) devised the first automatic semantic role labeling system based on FrameNet. Traditional systems relied on sophisticated handcraft features or some declarative constraints (Pradhan et al. 2005; Zhao et al. 2009), which suffer from poor efficiency and generalization ability. A recently tendency for SRL is adopting neural networks methods thanks to their significant success in a wide range of applications. Recently, (Zhou and Xu 2015; He et al. 2017) introduced end-to-end models for span-based SRL. These studies tackle argument identification and argument classification in one shot. Inspired by recent advances, we can easily integrate SRL into text comprehension. The pioneering work on building an end-to-end neural system was presented by Zhou and Xu (2015), applying an 8 layered LSTM model, which takes only original text information as input feature without using any syntactic knowledge, outperforming the previous state-of-the-art system. He et al. (2017) presented a deep highway BiLSTM architecture with constrained decoding, which is simple and effective, enabling us to select it as our basic semantic role labeler.

Semantic Role Labeling for Text Comprehension

For either of text comprehension tasks, we consider an end-to-end model as well as the semantic learning model. The former may be regarded as downstream model of the latter. Thus, our SRL augmented model will be an integration of two end-to-end models through simple embedding concatenation as shown in Figure 2.

We apply semantic role labeler to annotate the semantic tags (i.e. predicate, argument) for each token in the input sequence, and then the input sequence along with the corresponding SRL labels is fed to downstream models. We regard the SRL signals as SRL embeddings and employ a lookup table to map each label to vectors, similar to the implementation of word embedding. For each word x, a joint embedding $e^j(w)$ is obtained by the concatenation of word embedding $e^w(x)$ and SRL embedding $e^s(x)$,

$$e^j(w) = e^w(x) \oplus e^s(x)$$

where $\oplus$ is the concatenation operator. The downstream model is task-specific. In this work, we focus on the textual entailment and machine reading comprehension, which will be discussed latter.

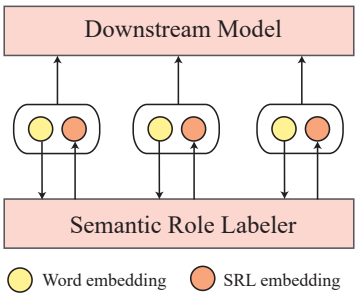

Figure 2: Overview of the semantic learning framework.

Semantic Role Labeler

Our concerned SRL includes two subtasks: predicate identification and argument labeling. While the CoNLL-2005 shared task assumes gold predicates as input, this information is not available in many applications, which requires us to identify the predicates for a input sentence at the very beginning. Thus, our SRL module has to be end-to-end, predicting all predicates and corresponding arguments in one shot.

We use spaCy[1] to tokenize the input sentence with part-of-speech (POS) tags and the verbs are marked as the binary predicate indicator for whether the word is the verb for the sentence.

Following (He et al. 2017), we model SRL as a span tagging problem using *BIO* encoding[2] and use an 8-layer deep BiLSTM with forward and backward directions interleaved. Different from the baseline model, we replace the GloVe embeddings with ELMo representations[3] due to the recent success of ELMo in NLP tasks (Peters et al. 2018).

In brief, the implementation of our SRL is a series of stacked interleaved LSTMs with highway connections (Srivastava, Greff, and Schmidhuber 2015). The inputs are embedded sequences of words concatenated with a binary indicator containing whether a word is the verbal predicate. Additionally, during inference, Viterbi decoding is applied to accommodate valid BIO sequences. The details are as follows.

[1]https://spacy.io/

[2]To represent a token at the beginning, interior, or outside of any span, respectively.

[3]The ELMo representation is obtained from `https://allennlp.org/elmo`. We use the original one for this work whose output size is 512.

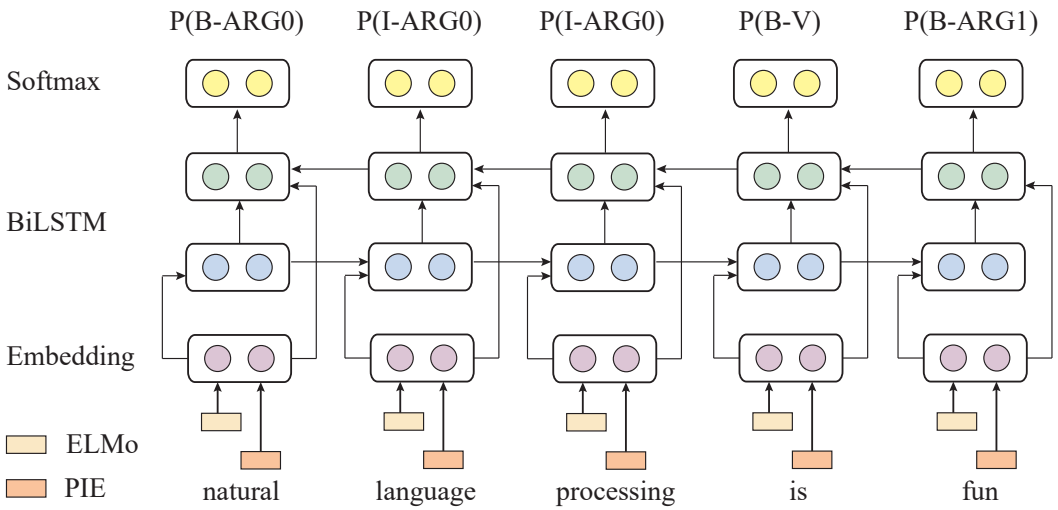

Figure 3: Semantic role labeler.

Word Representation The word representation of our SRL model is the concatenation of two vectors: an ELMo embedding $\boldsymbol{e}^{(l)}$ and predicate indicator embedding (PIE) $\boldsymbol{e}^{(p)}$. ELMo is trained from the internal states of a deep bidirectional language model (BiLM), which is pre-trained on a large text corpus with approximately 30 million sentences (Chelba et al. 2014). Besides, following (He et al. 2018b) who shows the predicate-specific feature is helpful in promoting the role labeling, we employ a predicate indicator embedding $\boldsymbol{e}^{(p)}$ to mark whether a word is a predicate when predicting and labeling the arguments. The final word representation is given by $\boldsymbol{e} = \boldsymbol{e}^{(l)} \oplus \boldsymbol{e}^{(p)}$, where $\oplus$ is the concatenation operator.

Encoder As commonly used to model the sequential input, BiLSTM is adopted for our sentence encoder. By incorporating a stack of distinct LSTMs, BiLSTM processes an input sequence in both forward and backward directions. In this way, the BiLSTM encoder provides the ability to incorporate the contextual information for each word.

Given a sequence of word representation $S = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N\}$ as input, the i-th hidden state $\boldsymbol{h}_i$ is encoded as follows:

$$\boldsymbol{h}_i^f = LSTM^{\mathcal{F}}\left(\boldsymbol{e}_i, \boldsymbol{h}_{i-1}^f\right),$$
$$\boldsymbol{h}_i^b = LSTM^{\mathcal{B}}\left(\boldsymbol{e}_i, \boldsymbol{h}_{i+1}^b\right),$$
$$\boldsymbol{h}_i = \boldsymbol{h}_i^f \oplus \boldsymbol{h}_i^b,$$

where $LSTM^{\mathcal{F}}$ denotes the forward LSTM transformation and $LSTM^{\mathcal{B}}$ denotes the backward LSTM transformation. $\boldsymbol{h}_i^f$ and $\boldsymbol{h}_i^b$ are the hidden state vectors of the forward LSTM and backward LSTM respectively.

This LSTM uses highway connections between layers and variational recurrent dropout. The encoded representation is then projected using a final dense layer followed by a softmax activation to form a distribution over all possible tags. The predicted SRL Labels are defined in PropBank (Palmer, Gildea, and Kingsbury 2005) augmented with B-I-O tag set to represent argument spans.

Model Implementation The training objective is to maximize the logarithm of the likelihood of the tag sequence, and we expect the correct output sequence matches with,

$$y^* = \underset{\widetilde{y} \in C}{argmax}\, s(x, \widetilde{y}) \quad (1)$$

where $\mathbf{C}$ is candidate label set.

Our semantic role labeler is trained on English *OntoNotes v5.0* benchmark dataset (Pradhan et al. 2013) for the CoNLL-2012 shared task, achieving an F1 of 84.6%[4] on the test set. At test time, we perform Viterbi decoding to enforce valid spans using BIO constraints[5]. For the following evaluation, the default dimension of SRL embeddings is 5 and the case study concerning the dimension is shown in the subsection *dimension of SRL Embedding*.

The model is run forward for every verb in the sentence. In some cases there is more than one predicate in a sentence, resulting in various semantic role sets whose number is equal to the number of predicates. For convenient downstream model input, we need to ensure the word and the corresponding label are matched one-by-one, that is, only one set for a sentence. To this end, we select the corresponding BIO sets with the most non-O labels as the semantic role labels. For sentences with no predicate, we directly assign *O* labels to each word in those sentences.

Text Comprehension Model

Textual Entailment Our basic TE model is the reproduced Enhanced Sequential Inference Model (ESIM) (Chen

[4]This result is comparable with the state-of-the-art (He et al. 2018a).

[5]The BIO format requires argument spans to begin with a B tag.

et al. 2017) which is a widely used baseline model for textual entailment. ESIM employs a BiLSTM to encode the premise and hypothesis, followed by an attention layer, a local inference layer, an inference composition layer. Slightly different from (Chen et al. 2017), we do not include extra syntactic parsing features and directly replace the pre-trained Glove word embedding with ELMo which are completely character based. Our SRL embedding is concatenated with ELMo embeddings and the joint embeddings are then fed to the BiLSTM encoders.

Machine Reading Comprehension Our baseline MRC model is an enhanced version of Bidirectional Attention Flow (Seo et al. 2017) following (Clark and Gardner 2018). The token embedding is the concatenation of pre-trained Glove word vectors, a character-level embedding from a convolutional neural network with max-pooling and pre-trained ELMo embeddings from language models (Peters et al. 2018). Our SRL enhanced model takes input of concatenating the token embedding with SRL embeddings. The embeddings of document and question are passed through a shared bi-directional GRU (BiGRU), followed by a bi-directional attention from (Seo et al. 2017) to obtain the context vectors. The contextual document and question representations are then passed to a residual self-attention layer. Then, the model predicts the start and end token of the answer. To this end, a BiGRU is applied, with a linear layer to compute answer start scores for each word. The hidden states are concatenated with the input and fed into a second bidirectional GRU and linear layer to predict answer end scores. Then, we apply a softmax operation to produce start and end probabilities, and we optimize the negative loglikelihood of selecting correct start and end tokens.

Evaluation

In this section, we evaluate the performance of SRL embeddings on two kinds of text comprehension tasks, *textual entailment* and *reading comprehension*. Both of the concerned tasks are quite challenging, and could be even more difficult considering that the latest performance improvement has been already very marginal. However, we present a new solution instead of heuristically stacking network design techniques. Namely, we show that SRL embeddings could be potential to give further advances due to its meaningful linguistic augments, which has not been studied yet for the concerned tasks.

Table 3 shows the hyper-parameters of our models. In our experiments, we basically follow the same hyper-parameters for each model as the original settings from their corresponding literatures (He et al. 2018b; Peters et al. 2018; Chen et al. 2017; Clark and Gardner 2018) except those specified (e.g. SRL embedding dimension). For both of the tasks, we also report the results by using pre-trained BERT (Devlin et al. 2018) as word representation in our baseline models [6]. The hyperparameters were selected using the Dev set, and the reported Dev and Test scores are averaged over 5 random seeds using those hyper-parameters.

SRL	Predicate embedding	100
	LSTM hidden units	300
	Dropout rate	0.1
	Batch size	80
	Optimizer	Adadelta
	Gradients clipping	1.0
	Learning rate	1.0 (ϵ = 0.95)
TE	LSTM hidden units	300
	Dropout rate	0.5
	Optimizer	Adam
	Gradients clipping	5.0
	Batch size	32
	Learning rate	0.001 (halved per epoch)
MRC	Glove embeddings	300
	Character embedding	30
	GRU hidden units	90
	Dropout rate	0.2
	Batch size	45
	Optimizer	Adadelta
	Max span length	17
	Learning rate	1.0

Table 3: Hyper-parameters of our models.

Textual Entailment

Textual entailment is the task of determining whether a *hypothesis* is *entailment, contradiction* and *neutral*, given a *premise*. The Stanford Natural Language Inference (SNLI) corpus (Bowman et al. 2015) provides approximately 570k hypothesis/premise pairs. We evaluate the model performance in terms of accuracy.

Results in Table 4 show that SRL embedding can boost the ESIM+ELMo model by +0.7% improvement [7]. With the semantic cues, the simple sequential encoding model yields substantial gains over the previous models, and our single $\text{BERT}_{\text{LARGE}}$ model also achieves a new state-of-the-art, even outperforms all the ensemble models in the leaderboard. This would be owing to more accurate and fine-grained information from effective explicit semantic cues.

To evaluate the contributions of key factors in our method, a series of ablation studies are performed on the SNLI dev and test set. The results are in Table 5. We observe both SRL and ELMo embeddings contribute to the overall performance. Note that ELMo is obtained by deep bidirectional language with 4,096 hidden units on a large-scale corpus,

[6] We use the last layer of BERT output. Since BERT is in subword-level while semantics role labels are in word-level, to use BERT in conjunction with our SRL embeddings, we need to keep them aligned. Therefore, we use the BERT embedding for the first subword of each word, which is slightly different from the original BERT: `https://github.com/google-research/bert`.

[7] Since ensemble systems are commonly integrated with multiple heterogeneous models and resources, we only show the results of single models to save space though our single model also outperforms the ensemble models

Model	Accuracy (%)
Deep Gated Attn. BiLSTM	85.5
Gumbel TreeLSTM	86.0
Residual stacked	86.0
Distance-based SAN	86.3
BCN + CoVe + Char	88.1
DIIN	88.0
DR-BiLSTM	88.5
CAFE	88.5
MAN	88.3
KIM	88.6
DMAN	88.8
ESIM + TreeLSTM	88.6
ESIM + ELMo	88.7
DCRCN	88.9
LM-Transformer	89.9
MT-DNN†	91.1
Baseline (ELMo)	88.4
+ SRL	89.1
Baseline ($\text{BERT}_{\text{BASE}}$)	89.2
+ SRL	89.6
Baseline ($\text{BERT}_{\text{LARGE}}$)	90.4
+ SRL	**91.3**

Table 4: Accuracy on SNLI test set. Models in the first block are sentence encoding-based. The second block embodies the joint methods while the last block shows our SRL based model. All the results except ours are from the SNLI Leaderboard. Previous state-of-the-art model is marked by †.

which requires a huge amount of training time with 93.6 million parameters. The output dimension of ELMo is 512. Compared with the massive computation and high dimension, the SRL embedding is much more convenient for training and much easier for model integration, giving the same level of performance gains.

Machine Reading Comprehension

To investigate the effectiveness of the SRL embedding in conjunction with more complex models, we conduct experiments on machine reading comprehension tasks. The reading comprehension task can be described as a triple $< D, Q, A >$, where D is a document (context), Q is a query over the contents of D, in which a span is the right answer A.

As a widely used benchmark dataset for machine reading comprehension, the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al. 2016) contains $100k$+ crowd sourced question-answer pairs where the answer is a span in a given Wikipedia paragraph. Two metrics are selected to evaluate the model performance: Exact Match (EM) and a softer metric F1 score, which measures the weighted average of the precision and recall rate at a character level.

Table 6 shows the results[8]. The SRL embeddings give an absolute +0.81% and relative +5.48% performance gains,

[8]Since the test set of SQuAD is not publicly available, our evaluations are based on dev set.

Model	Dev	Test
Our model	**89.11**	**89.09**
-ELMo	88.51	88.42
-SRL	88.89	88.65
-ELMo -SRL	88.39	87.96

Table 5: Ablation study. Since we use ELMo as the basic word embeddings, we replace ELMO with 300D Glove embeddings for the case *-ELMo*.

Model	**EM**	**F1**	**RERR**
Baseline (ELMo)	77.53	85.22	-
+SRL	**78.53**	**86.03**	**5.48%**
Baseline ($\text{BERT}_{\text{BASE}}$)	81.34	88.49	-
+SRL	**81.67**	**88.78**	**2.52%**
Baseline ($\text{BERT}_{\text{LARGE}}$)	84.20	90.94	-
+SRL	**84.52**	**91.16**	**2.43%**

Table 6: Exact Match (EM) and F1 scores on SQuAD dev set. RERR is short for relative error rate reduction of our model to the baseline evaluated on F1 score.

showing it is also quite effective for more complex document and question encoding.

Case Studies

From the above experiments, we see our semantic learning framework works effectively and the semantic role labeler boosts model performance, verifying our hypothesis that semantic roles are critical for text understanding. Though the semantic role labeler is trained on a standard benchmark dataset, *Ontonotes*, whose source ranges from news, conversational telephone speech, weblogs, etc., it turns out to be generally useful for text comprehension from probably quite different domains in both textual entailment and machine reading comprehension. To further evaluate the proposed method, we conduct several case studies as follows [9].

Dimension of SRL Embedding

The dimension of embedding is a critical hyper-parameter in deep learning models that may influence the performance. Too high dimension would cause severe over-fitting issues while too low dimension would also cause under-fitting results. To investigate the influence of the dimension of SRL embeddings, we change the dimension in the intervals [1, 2, 5, 10, 20, 50, 100]. Figure 4-5 show the results. We see that 5-dimension SRL embedding gives the best performance on both SNLI and SQuAD datasets.

Compare with POS/NER Tags

Part-of-speech (POS) and named entity (NE) tags have been used in various NLP tasks. To make comparsion between them, we conduct experiments on SNLI with modifications

[9]For simplicity, our case studies are based on ELMo-based models.

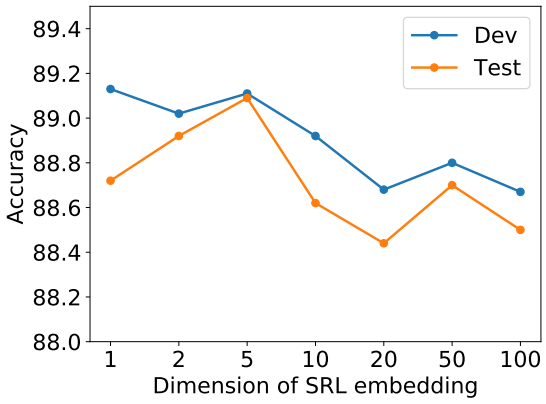

Figure 4: Result on SNLI of different embedding dimensions.

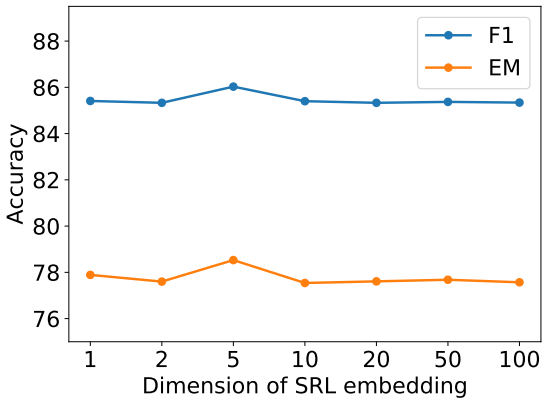

Figure 5: Result on SQuAD of different embedding dimensions.

on label embeddings using tags of SRL, POS and NE, respectively. Results in Table 7 show that SRL gives the best result, showing semantic roles contribute to the performance, which also indicates that semantic information matches the purpose of NLI task best.

Model Training

We are interested in how the SRL embeddings influence the model training procedure. We observe that our model converge much more quickly than baseline models without SRL information. Our model achieves the best result after nearly 10 epochs of training while for the baseline model, the iteration is about 20-epoch. Besides, for each epoch, the accuracy of our model is basically higher than the baseline. This shows SRL signals could accelerate model training with accurate hints. Owing to the semantic role indication, our models are able to gain the best performance with less training time.

Model	Dev	Test
Baseline	88.89	88.65
Word + SRL	89.11	89.09
Word + POS	88.90	88.68
Word + NE	89.14	88.51

Table 7: Comparison with different NLP tags.

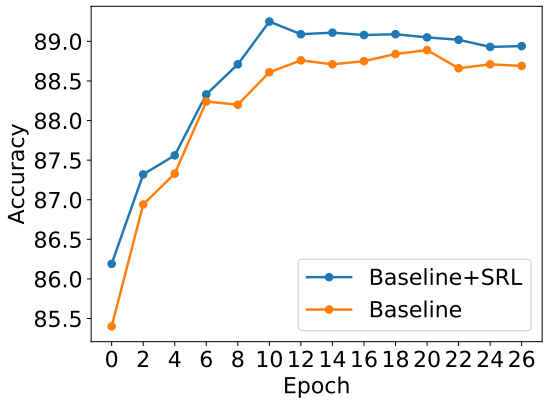

Figure 6: Learning curve on the SNLI dev set.

Conclusion

This paper presents a novel semantic learning framework for fine-grained text comprehension and inference. We show that our proposed method is simple yet powerful, which achieves a significant improvement over strong baseline models. This work discloses the effectiveness of SRL in text comprehension and inference and proposes an easy and feasible scheme to integrate SRL information in neural models. A series of detailed case studies are employed to analyze the robustness of the semantic role labeler. Though most recent works focus on heuristically stacking complex mechanisms for performance improvement, we hope to shed some lights on fusing accurate semantic signals for deeper comprehension and inference.

References

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. *EMNLP*.

Chelba, C.; Mikolov, T.; Schuster, M.; Qi, G.; Brants, T.; Koehn, P.; and Robinson, T. 2014. One billion word benchmark for measuring progress in statistical language modeling. *arXiv:1312.3005*.

Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced LSTM for natural language inference. *ACL*.

Chen, D.; Bolton, J.; and Manning, C. D. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. *ACL 2016*.

Clark, C., and Gardner, M. 2018. Simple and effective multi-paragraph reading comprehension. *ACL*.

Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2017. Attention-over-attention neural networks for reading comprehension. *ACL*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dhingra, B.; Liu, H.; Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2017. Gated-attention readers for text comprehension. *ACL*.

Fitzgerald, N.; He, L.; and Zettlemoyer, L. 2018. Large-scale qa-srl parsing. *ACL*.

Gildea, D., and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational Linguistics*.

He, L.; Lee, K.; Lewis, M.; Zettlemoyer, L.; He, L.; Lee, K.; Lewis, M.; Zettlemoyer, L.; He, L.; and Lee, K. 2017. Deep semantic role labeling: What works and whats next. *ACL*.

He, L.; Lee, K.; Levy, O.; and Zettlemoyer, L. 2018a. Jointly predicting predicates and arguments in neural semantic role labeling. *ACL*.

He, S.; Li, Z.; Zhao, H.; Bai, H.; and Liu, G. 2018b. Syntax for semantic role labeling, to be, or not to be. *ACL*.

He, L.; Lewis, M.; and Zettlemoyer, L. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. *EMNLP*.

Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. *NIPS*.

Hill, F.; Bordes, A.; Chopra, S.; and Weston, J. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv:1511.02301*.

Jia, R., and Liang, P. 2017. Adversarial examples for evaluating reading comprehension systems. *EMNLP*.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *ACL*.

Kadlec, R.; Schmid, M.; Bajgar, O.; and Kleindienst, J. 2016. Text understanding with the attention sum reader network. *ACL*.

Mccann, B.; Bradbury, J.; Xiong, C.; and Socher, R. 2017. Learned in translation: Contextualized word vectors. *NIPS*.

Mihaylov, T., and Frank, A. 2016. Discourse relation sense classification using cross-argument semantic similarity based on word embeddings. *CoNLL*.

Mudrakarta, P. K.; Taly, A.; Sundararajan, M.; and Dhamdhere, K. 2018. Did the model understand the question? *ACL*.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv:1611.09268v2*.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*.

Pan, B.; Yang, Y.; Zhao, Z.; Zhuang, Y.; Cai, D.; and He, X. 2018. Discourse marker augmented network with reinforcement learning for natural language inference. *ACL*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv:1802.05365*.

Pradhan, S.; Ward, W.; Hacioglu, K.; Martin, J.; and Jurafsky, D. 2005. Semantic role labeling using different syntactic views. *ACL*.

Pradhan, S.; Moschitti, A.; Xue, N.; Ng, H. T.; Björkelund, A.; Uryupina, O.; Zhang, Y.; and Zhong, Z. 2013. Towards robust linguistic analysis using OntoNotes. *CoNLL*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ questions for machine comprehension of text. *EMNLP*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for SQuAD. *ACL*.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2017. Bidirectional attention flow for machine comprehension. *ICLR*.

Shi, C.; Liu, S.; Ren, S.; Feng, S.; Li, M.; Zhou, M.; Sun, X.; and Wang, H. 2016. Knowledge-based semantic embedding for machine translation. *ACL*.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Training very deep networks. *NIPS*.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. *ACL*.

Wang, Y.; Liu, K.; Liu, J.; He, W.; Lyu, Y.; Wu, H.; Li, S.; and Wang, H. 2018. Multi-passage machine reading comprehension with cross-passage answer verification. *ACL*.

Wang, W.; Yan, M.; and Wu, C. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *ACL*.

Yih, W. T.; Richardson, M.; Meek, C.; Chang, M. W.; and Suh, J. 2016. The value of semantic parse labeling for knowledge base question answering. *ACL*.

Zhang, Z., and Zhao, H. 2018. One-shot learning for question-answering in gaokao history challenge. *COLING*.

Zhang, Z.; Li, J.; Zhu, P.; and Zhao, H. 2018. Modeling multi-turn conversation with deep utterance aggregation. *COLING*.

Zhang, Z.; Huang, Y.; and Zhao, H. 2018. Subword-augmented embedding for cloze reading comprehension. In *COLING*.

Zhao, H.; Chen, W.; Kazama, J.; Uchimoto, K.; and Torisawa, K. 2009. Multilingual dependency learning: Exploiting rich features for tagging syntactic and semantic dependencies. *CoNLL*.

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

Zhu, P.; Zhang, Z.; Li, J.; Huang, Y.; and Zhao, H. 2018. Lingke: A fine-grained multi-turn chatbot for customer service. In *COLING Demo*.