Semantics-Aware Inferential Network for Natural Language Understanding

Shuailiang Zhang,[1,2,3] **Hai Zhao,**[1,2,3*] **Junru Zhou,**[1,2,3]**, Xi Zhou,**[4] **Xiang Zhou**[4]

[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, China
[3]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
[4]CloudWalk Technology, Shanghai, China
zsl123@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,
zhoujunru@sjtu.edu.cn, {zhouxi,zhouxiang}@cloudwalk.cn

Abstract

For natural language understanding tasks, either machine reading comprehension or natural language inference, both semantics-aware and inference are favorable features of the concerned modeling for better understanding performance. Thus we propose a Semantics-Aware Inferential Network (SAIN) to meet such a motivation. Taking explicit contextualized semantics as a complementary input, the inferential module of SAIN enables a series of reasoning steps over semantic clues through an attention mechanism. By stringing these steps, the inferential network effectively learns to perform iterative reasoning which incorporates both explicit semantics and contextualized representations. In terms of well pre-trained language models as front-end encoder, our model achieves significant improvement on 11 tasks including machine reading comprehension and natural language inference.

Introduction

Recent studies (Zhang et al. 2020a; Mihaylov and Frank 2019; Sun et al. 2019; Zhang et al. 2019b, 2018) have shown that introducing extra common sense knowledge or linguistic knowledge into language representations may further enhance the concerned natural language understanding (NLU) tasks that latently have a need of reasoning ability, such as natural language inference (NLI) (Wang et al. 2019; Bowman et al. 2015) and machine reading comprehension (MRC) (Rajpurkar, Jia, and Liang 2018; Koisk et al. 2018). Zhang et al. (2020a) propose incorporating explicit semantics as a well-formed linguistic knowledge by concatenating the pre-trained language model embedding with semantic role labeling embedding, and obtains significant gains on the SNLI (Bowman et al. 2015) and GLUE benchmark (Wang et al. 2019). Mihaylov and Frank (2019) use semantic information to strengthen the multi-head self-attention model, and achieves substantial improvement on NarrativeQA (Koisk et al. 2018). In this work, we propose a Semantics-Aware Inferential Network (SAIN) to refine the use of semantic structures by decomposing text into different semantic structures for compositional processing in inferential network.

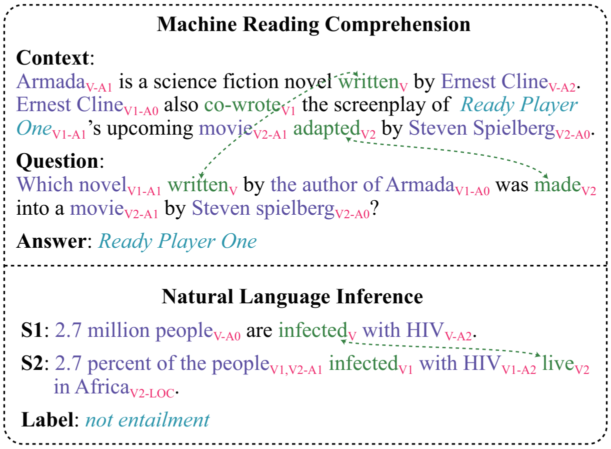
Machine Reading Comprehension
Context:
Armada$_{V\text{-}A1}$ is a science fiction novel written$_{V}$ by Ernest Cline$_{V\text{-}A2}$.
Ernest Cline$_{V1\text{-}A0}$ also co-wrote$_{V1}$ the screenplay of Ready Player One$_{V1\text{-}A1}$'s upcoming movie$_{V2\text{-}A1}$ adapted$_{V2}$ by Steven Spielberg$_{V2\text{-}A0}$.
Question:
Which novel$_{V1\text{-}A1}$ written$_{V}$ by the author of Armada$_{V1\text{-}A0}$ was made$_{V2}$ into a movie$_{V2\text{-}A1}$ by Steven spielberg$_{V2\text{-}A0}$?
Answer: Ready Player One
Natural Language Inference
S1: 2.7 million people$_{V\text{-}A0}$ are infected$_{V}$ with HIV$_{V\text{-}A2}$.
S2: 2.7 percent of the people$_{V1,V2\text{-}A1}$ infected$_{V1}$ with HIV$_{V1\text{-}A2}$ live$_{V2}$ in Africa$_{V2\text{-}LOC}$.
Label: not entailment

Figure 1: Examples in MRC and NLI with necessary semantic annotations. The connected predicates have important arguments to predict the answer.

Questions in NLU tasks are usually not compositional, so most existing inferential networks (Weston, Chopra, and Bordes 2014; Yu, Zha, and Yin 2019) input the same text at each reasoning step, which is not efficient enough to perform iterative reasoning. To overcome this problem, we use

[*] Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), Key Projects of National Natural Science Foundation of China (U1836222 and 61733011), Huawei-SJTU long term AI project, Cutting-edge Machine reading comprehension and language model.

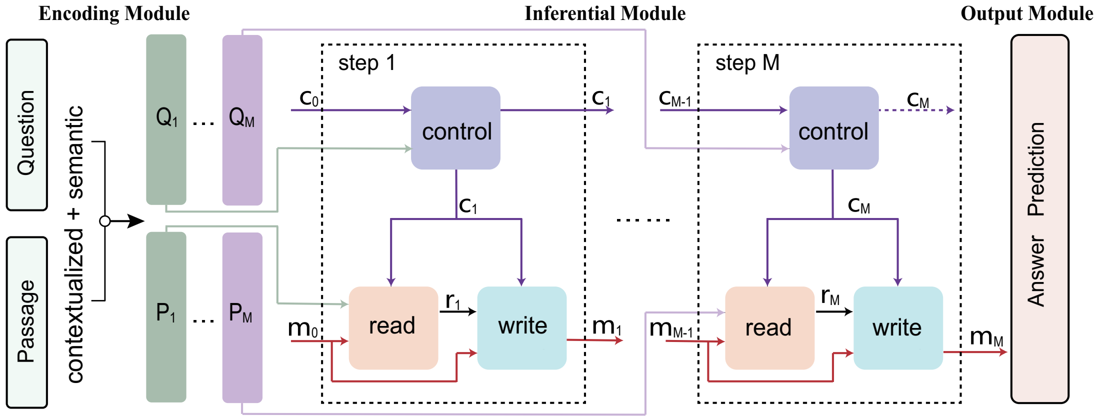

Figure 2: Overview of the framework. Here we only show the inputs and outputs of the first step and last step. The encoding module outputs M semantic representations that integrate both the contextualized and semantic embedding. The model attends to Q_i and P_i in step i. The final memory state m_M is used to predict the answer.

semantic role labeling to decompose the text into different semantic structures which are referred as different semantic representations of the sentence (Khashabi et al. 2018; Mihaylov and Frank 2019).

In Figure 1, to correctly answer the MRC question, the model needs to recognize that *the author of Armada* is *Ernest Cline* firstly, and then knows that *Ernest Cline*'s novel *Ready Player One* was made into a movie by *Steven spielberg*, which requires iteratively reasoning over the two predicates *written* and *made* because they have very similar arguments with the corresponding predicates *written* and *adapted* in the context. For the NLI example, if the model recognizes the predicate *infected* as the central meaning in S2 and ignores the true central word *live*, it probably makes wrong prediction *entailment* because S1 also has a similar structure predicated on *infected*. So it may be helpful to refine the use of semantic clues by integrating all the semantic information into the inference.

We are motivated to model these semantic structures by presenting SAIN, which introduces semantic information (Zhang et al. 2020a) into the multi-step reasoning model (Hudson and Manning 2018). In SAIN, there are a set of reasoning steps, each step attends to one predicate-argument structure and can be viewed as a cell consisting of three units: control unit, read unit and write unit, that operate over dull *control* and *memory* hidden states. The cells are recursively connected, where the result of the previous step acts as the context of next step. The interaction between the cells is regulated by structural constraints to perform iterative reasoning in an end-to-end way.

This work will focus on two typical NLU tasks, natural language inference and machine reading comprehension. Experiment results indicate that our proposed model achieves significant improvement over the strong baselines on these tasks and obtains the state-of-the-art performance on SNLI and MRQA datasets.

Approach

The model framework is shown in Figure 2. Our model includes: 1) contextualized encoding module which obtains the joint representation of the pre-trained language model embedding and semantic embedding. 2) inferential module which consists of a set of recurrent reasoning steps/cells, where each step/cell attends to one predicate-argument structure of one sentence. 3) output module which predicts the answer based on the final memory state of the inferential module.

For MRC task, given a passage (**P**) and a question (**Q**), the goal is to predict the answer from the given passage. For NLI task, given a pair of sentences, the goal is to judge the relationship between their meanings. Our model will be introduced with the background of MRC task, and the corresponding NLI implementation of our model can be regarded as a simplified case of the MRC, considering that passage and question in MRC task correspond to two sentences in NLI task.

Semantic Role Labeling

Semantic role labeling (SRL) is generally formulated as multi-step classification subtasks in pipeline systems to identify the semantic structures. There are a few of formal semantic frames, including FrameNet (Baker, Fillmore, and Lowe 1998) and PropBank (Palmer, Gildea, and Kingsbury 2005), which generally present the semantic relationship as predicate-argument structure. When several argument-taking predicates are recognized in one sentence, we obtain multiple semantic representations of the sentence. For example, given the context sentence in Figure 3 with target predicates *loves* and *eat*, there are two semantic structures labeled as follows,

[The cat]$_{ARG0}$ [loves]$_{V}$ [to eat fish]$_{ARG1}$.

[The cat]$_{ARG0}$ [loves to]$_{O}$ [eat]$_{V}$ [fish]$_{ARG1}$.

where ARG0, ARG1 represents the argument role 0, 1 of the predicate *V*, respectively.

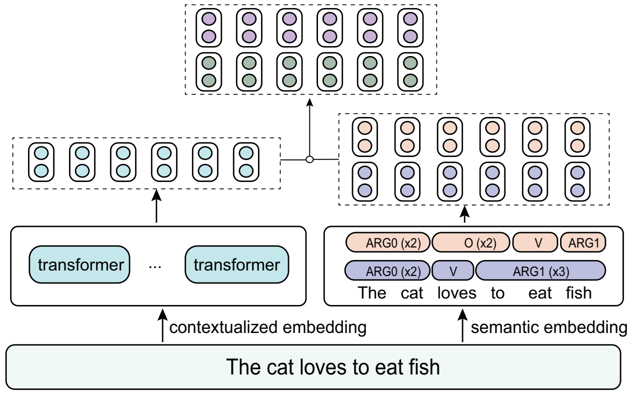

Figure 3: Different semantic representations of one sentence combined by contextualized embedding and semantic embedding.

Contextual Encoding

Semantic Embedding Given the sentence $X = \{x_1, ..., x_n\}$ with n words and m predicates ($m = 2$ in Figure 3), there come m corresponding labeled SRL sequences $\{L_1, L_2, ..., L_m\}$ with length n. Note this is done in data preprocessing and these labels are not updated with the following modules. These semantic role labels are mapped into vectors in dimension d_w where each sequence L_i is embedded as $E^{s_i} = \{e_1^i, ..., e_n^i\} \in R^{n \times d_w}$.

Contextualized Embedding With an adopted contextualized encoder, the input sequence $X = \{x_1, ..., x_n\}$ is embedded as $E^w = \{e_1, ..., e_{n_s}\} \in R^{n_s \times d_s}$, where d_s is hidden state size of the encoder and n_s is the tokenized sequence length.

Joint embedding Note that the input sequence may be tokenized into subwords. Then the tokenized sequence of length n_s is usually longer than the SRL sequence of length n. To align these two sequences, we extend the SRL sequence to length n_s by assigning the subwords the same label with original word. The aligned contextualized and semantic embeddings are then concatenated as the joint embedding for the sequence $E^{X_i} = [E^{s_i}; E^w] \in R^{n_s \times d}$, where $d = d_s + d_w$.

Different sentences have various numbers of predicate-argument structures, here we set the maximum number as M for ease of calculation. So for MRC, the passage and question both have M encoded representations where $E^P = \{E^{P_1}, ..., E^{P_M}\} \in R^{M \times |P| \times d}$ and $E^Q = \{E^{Q_1}, ..., E^{Q_M}\} \in R^{M \times |Q| \times d}$, where $|P|$, $|Q|$ are the length of passage and question.

Inferential Network

The inferential module performs explicit multi-step reasoning by stringing together M cells, where each attends to one semantic structure of the sentence. Each cell has three operation units: control unit, read unit and write unit, iteratively aggregating information from different semantic structures.

For MRC, each reasoning step attends to one semantic structure of each sentence from passage and question, respectively. So passage $E^{P_i} = \{p_{i,1}, ..., p_{i,|P|}\}$ and question $E^{Q_i} = \{q_{i,1}, ..., q_{i,|Q|}\}$ are the input sequences for step i. Besides, we use biLSTM to get the overall question representation $bq_i = [\overrightarrow{q_{i,1}}; \overleftarrow{q_{i,|Q|}}] \in R^{2d}$.

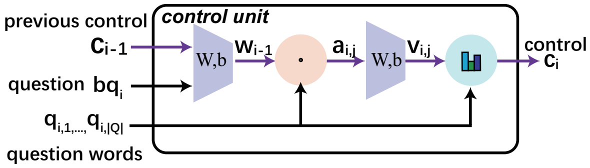

Figure 4: The control unit.

Reasoning Cell The reasoning cell is a recurrent cell designed to capture information from different semantic structures. For each step $i = 1, ..., M$ in the reasoning process, the i^{th} cell maintains two hidden states: **control** $\boldsymbol{c_i}$ and **memory** $\boldsymbol{m_i}$, with dimension d. The control $\boldsymbol{c_i}$ retrieves information from E^{Q_i} by calculating a soft attention-based weighted average of the question words. The memory $\boldsymbol{m_i}$ holds the intermediate results from the reasoning process up to the i^{th} step by integrating the preceding hidden state $\boldsymbol{m_{i-1}}$ with the new information $\boldsymbol{r_i}$ retrieved from the passage E^{P_i}.

There are three units in each cell: control unit, read unit and write unit, which work together to perform iterative reasoning. The control unit retrieves the information from the question, updating the control hidden state $\boldsymbol{c_i}$. The read unit extracts relevant information from the passage and outputs extracted information $\boldsymbol{r_i}$. The write unit integrates $\boldsymbol{c_i}$ and $\boldsymbol{r_i}$ into the memory $\boldsymbol{m_{i-1}}$, producing a new memory $\boldsymbol{m_i}$. In the following, we give the details of these three units. All the vectors are of dimension d unless otherwise stated.

The **control unit** (Figure 4) attends to the i^{th} semantic structure of the question E^{Q_i} at step i and updates the control state $\boldsymbol{c_i}$ accordingly. Firstly, we combines the overall question representation $\boldsymbol{bq_i}$ and preceding reasoning operation $\boldsymbol{c_{i-1}}$ into $\boldsymbol{w_i}$ through a linear layer. Subsequently, we calculate the similarity between $\boldsymbol{w_i}$ and each question word $\boldsymbol{q_{i,j}}$, and pass the result through a softmax layer, yielding an attention distribution over the question words. Finally, we sum the words over this distribution to get the new control $\boldsymbol{c_i}$. The calculation details are as follows:

$$\begin{aligned} w_i &= W^{d \times 2d}[c_{i-1}, bq_i] + b^d \\ a_{i,j} &= W^{1 \times d}(w_i \odot q_{i,j}) + b^1 \\ v_{i,j} &= \mathit{Softmax}(a_{i,j}), j = 1, ..., |Q| \\ c_i &= \sum_{j=1}^{|Q|} v_{i,j} \cdot q_{i,j} \end{aligned} \tag{1}$$

where $W^{d \times 2d}$, $W^{1 \times d}$, b^d and b^1 are learnable parameters, $|Q|$ is the question length.

The **read unit** (Figure 5) inspects the i^{th} semantic structure of the passage E^{P_i} at step i and retrieves the information $\boldsymbol{r_i}$ to update the memory. Firstly, we compute the interaction between every passage word $p_{i,p}$ and the memory

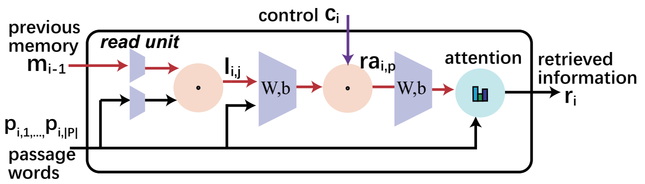

Figure 5: The read unit.

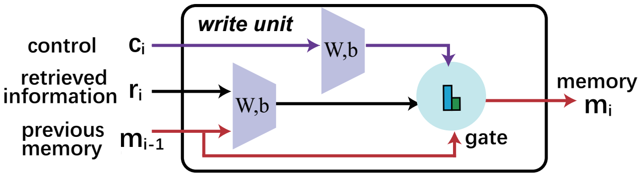

Figure 6: The write unit.

$\boldsymbol{m_{i-1}}$, resulting in $I_{i,p}$ which measures the relevance of the passage word to the preceding memory. Then, $I_{i,p}$ and $p_{i,p}$ are concatenated and passed through a linear transformation, yielding $\hat{I}_{i,p}$ which considers both the new information from E^{P_i} and the information related to the prior intermediate result. Finally, aiming to retrieve the information relevant to the question, we measure the similarity between $\hat{I}_{i,p}$ and c_i and pass the result through a softmax layer which produces an attention distribution over the passage words. This distribution is used to get the weighted average $\boldsymbol{r_i}$ over the passage. The calculation is detailed as follows:

$$
\begin{aligned}
I_{i,p} &= [W_1^{d\times d} m_{i-1} + b_1^d] \odot [W_2^{d\times d} p_{i,p} + b_2^d] \\
\hat{I}_{i,p} &= W^{d\times 2d}[I_{i,p}, p_{i,p}] + b^d \\
ra_{i,p} &= W^{d\times d}(c_i \odot \hat{I}_{i,p}) + b^d \\
rv_{i,p} &= \mathit{Softmax}(ra_{i,p}), p = 1, ..., |P| \\
r_i &= \sum_{p=1}^{|P|} rv_{i,p} \cdot p_{i,p}
\end{aligned}
$$

where all the W and b are learnable parameters, $|P|$ is the passage length.

The **write unit** (Figure 6) is responsible for integrating the information retrieved from the read unit $\boldsymbol{r_i}$ with the preceding memory $\boldsymbol{m_{i-1}}$, guided by the i^{th} reasoning operation $\boldsymbol{c_i}$ from the question. Specificly, a sigmoid gate is used when combining the previous memory state $\boldsymbol{m_{i-1}}$ and the new memory candidate $\boldsymbol{m_i^r}$. The calculation details are as follows:

$$
\begin{aligned}
m_i^r &= W^{d\times 2d}[r_i, m_{i-1}] + b^d \\
\hat{c}_i &= W^{1\times d} c_i + b^1 \\
m_i &= \sigma(\hat{c}_i) m_{i-1} + (1 - \sigma(\hat{c}_i)) m_i^r
\end{aligned}
\tag{2}
$$

Output Module

For MRC, the output module predicts the final answer to the question based on the set of memory states $\{m_1,...,m_{\mathrm{M}}\}$ produced by the inferential module. For MRC, we calculate the similarity between the i^{th} memory $m_i \in R^d$ and each passage word $p_{i,p}$ in i^{th} semantic passage representation E^{P_i}, resulting in $\hat{E}^{P_i}$, $i = 1, ..., \mathrm{M}$. We concatenate $\hat{E}^{P_1}, ..., \hat{E}^{P_M}$ as the final passage representation $\hat{E}^P \in R^{|P|\times Md}$ which is then passed to a linear layer to get the start and end probability distribution p_s, p_e on each position. Finally, a cross entropy loss is computed:

$$
\begin{aligned}
mp_{i,p} &= ReLU(m_i) \cdot p_{i,p} \\
\hat{E}^{P_i} &= [mp_{i,1}, ..., mp_{i,|P|}] \in R^{|P|\times d} \\
E &= [\hat{E}^{P_1}, ..., \hat{E}^{P_M}] \in R^{|P|\times Md} \\
[p_s, p_e] &= EW^{Md\times 2} \in R^{|P|\times 2} \\
Loss &= \frac{1}{2} CE(p_s, y_s) + \frac{1}{2} CE(p_e, y_e)
\end{aligned}
$$

where y_s and y_e are the true start and end probability distribution. p_s, p_e, y_s and y_e are all with size $R^{|P|}$. $CE(\cdot)$ indicates the cross entropy function.

For NLI, the final memory state m_{M} is activated by the Tanh function and passed to a linear layer to produce the probability distribution over the labels: $p = Tanh(m_{\mathrm{M}}) \cdot W^{d\times N} \in R^N$. Cross entropy is used as the metric: Loss $= \mathit{CE}(p, y)$, where N is the number of labels. $p \in R^N$ is the predicted probability distribution over the labels and $y \in R^N$ is the true label distribution.

Experiments

Data and Task Description

Machine Reading Comprehension We evaluate our model on extractive MRC such as SQuAD (Rajpurkar, Jia, and Liang 2018) and MRQA[1] (Fisch et al. 2019) where the answer is a span of the passage. MRQA is a collection of existing question-answering related MRC datasets, such as SearchQA (Dunn et al. 2017), NewsQA (Trischler et al. 2017), NaturalQuestions (Kwiatkowski et al. 2019), TriviaQA (Joshi et al. 2017), etc. They are transformed into SQuAD style extractive MRC.

Natural Language Inference Given a pair of sentences, the target is to judge the relationship between their meanings, such as entailment, neural and contradiction. We evaluate on 4 diverse datasets, including SNLI (Bowman et al. 2015), MNLI (Williams, Nangia, and Bowman 2018), QNLI (Rajpurkar et al. 2016) and RTE (Bentivogli et al. 2009).

Implementation Details

To obtain the semantic role labels, we use the SRL system of (He et al. 2017) as implemented in AllenNLP (Gardner et al. 2018) that splits sentences into tokens and predicts SRL tags such as *ARG0*, *ARG1* for each verb. We use *O* for non-argument words and *V* for predicates. The dimension of SRL embedding is set to 30 and performance does not change significantly when setting this number to 10, 50 or

[1] https://github.com/mrqa/MRQA-Shared-Task-2019.

	NewsQA	TriviaQA	SearchQA	HotpotQA	NaturalQA	**(Avg.)**
MTL$_{\text{base}}$ (Fisch et al. 2019)	66.8	71.6	76.7	76.6	77.4	73.8
MTL$_{\text{large}}$ (Fisch et al. 2019)	66.3	74.7	79.0	79.0	79.8	75.8
CLER (Takahashi et al. 2019)	69.4	75.6	79.0	79.8	79.8	76.7
BERT$_{\text{large}}$ (Joshi et al. 2019)	68.8	77.5	81.7	78.3	79.9	77.3
HLTC (Su et al. 2019)	72.4	76.2	79.3	80.1	80.6	77.7
SemBERT* (Zhang et al. 2020a)	69.1	78.6	82.4	78.6	80.3	77.8
SpanBERT (Joshi et al. 2019)	73.6	83.6	84.8	83.0	82.5	81.5
BERT$^*{}_{\text{base}}$	66.2	71.5	77.0	75.0	77.5	73.4
BERT$^*{}_{\text{large}}$	69.2	77.4	81.5	78.2	79.4	77.2
SpanBERT*	73.0	83.1	83.5	82.5	81.9	80.9
RoBERTa*	73.3	83.2	83.7	82.8	82.1	81.1
Our Models						
SAIN$_{\text{BERT}_{\text{base}}}$	68.3	72.6	78.2	77.7	78.9	75.2
SAIN$_{\text{BERT}_{\text{large}}}$	72.1	80.1	83.4	79.4	82.0	79.4
SAIN$_{\text{SpanBERT}}$	75.1	85.2	85.4	84.4	83.8	82.8
SAIN$_{\text{RoBERTa}}$	**75.4**	**85.5**	**85.7**	**84.5**	**84.3**	**83.1**

Table 1: Performance (F1) on five MRQA tasks. Results with * are our implementations. Avg indicates the average score of these datasets. All these results are from single models.

Model	MNLI-m/mm		QNLI	RTE	SNLI	**(Avg.)**	SQuAD 1.1		SQuAD 2.0		**(Avg.)**
	Acc	Acc	Acc	Acc	Acc		EM	F1	EM	F1	
BERT$_{\text{base}}$	84.6	83.4	89.3	66.4	90.7	82.9	80.8	88.5	77.1*	80.3*	81.7
BERT$_{\text{large}}$	86.7	85.9	92.7	70.1	91.1	85.3	84.1	90.9	80.0	83.3	84.5
SemBERT$_{\text{base}}$	84.4	84.0	90.9	69.3†	91.0*	83.9	–	–	–	–	–
SemBERT$_{\text{large}}$	87.6	86.3	94.6	70.9†	91.6	86.2	84.5*	91.3*	80.9	83.6	85.1
Our Models											
SAIN$_{\text{BERT}_{\text{base}}}$	84.9	85.0	92.1	72.0	91.3	85.1	82.2	89.3	79.4	82.0	83.2
SAIN$_{\text{BERT}_{\text{large}}}$	**88.7**	**87.5**	**95.6**	**73.9**	**91.9**	**87.5**	**85.4**	**92.1**	**82.8**	**85.4**	**86.4**

Table 2: Experiment results on NLI and SQuAD datasets. The results of BERT and SemBERT are from (Devlin et al. 2019) and (Zhang et al. 2020a). † indicates the results of SemBERT without random restarts and distillation. Results with * are our implementations. Avg indicates the average score of these datasets. All these results are from single models.

100. According to the experimental results, it is a reasonable configuration that sets the maximum number of predicate-argument structures (reasoning steps) M to 3 and 4 for MRC and NLI tasks, respectively.

Our model framework is based on the Pytorch implementation of transformers[2]. We use Adam as our optimizer with initial learning rate 1e-5 and warm-up rate of 0.1. The batch size is set to 8.

Overall Results

Our main comparison models are the BERT baselines (BERT (Devlin et al. 2019) and SpanBERT (Joshi et al. 2019)) and SemBERT (Zhang et al. 2020a). SemBERT improves the language representation by concatenating the BERT embedding and semantic embedding, where embeddings from different predicate-argument structures are simply fused as one semantic representation by using one linear layer. We compare our model to these baselines on 11 benchmarks including 5 MRQA datasets, 4 NLI tasks and 2 SQuAD datasets in Tables 1 and 2.

SAIN vs. BERT/SpanBERT baselines Compared to BERT, our model achieves 2.2% (79.4 vs. 77.2), 2.2% (87.5 vs. 85.3), 1.7% (88.8% vs. 87.1 %) average improvement on MRQA, NLI and SQuAD datasets. Our model also outperforms other BERT based models CLER (Takahashi et al. 2019) and HLTC (Su et al. 2019) on MRQA. We also compare with SpanBERT on MRQA datasets and our model outperforms this baseline by 1.3% (82.8 vs. 81.5) in average F1 score. To the best of our knowledge, we achieve state-of-the-art performance on MRQA (dev sets) and SNLI.

SAIN vs. SemBERT Our SAIN outperforms SemBERT on all tasks, including 1.6% (79.4 vs. 77.8), 1.3% (87.5 vs. 86.2) and 1.3% (86.4 vs. 85.1) average improvement on MRQA, NLI and SQuAD datasets. We attribute the superiority of our SAIN to its more refined use of semantic clues in terms of inferential network rather than SemBERT which simply encodes all predicate-argument structures into one embedding.

[2] https://github.com/huggingface/transformers.

	RTE	SQuAD 1.1	SQuAD 2.0
SAIN	**71.8**	**89.1**	**82.2**
w/o IM	69.5 (-2.3)	87.5 (-1.6)	80.7 (-1.5)
w/o SI	69.9 (-1.9)	87.4 (-1.7)	80.8 (-1.4)
w/o IR	70.4 (-1.4)	88.0 (-1.1)	81.0 (-1.2)

Table 3: Ablation study on RTE, SQuAD 1.1 and SQuAD 2.0 (F1), using BERT$_{\text{base}}$ as contextual encoder. IM, SI and IR is defined in Section .

	RTE	SQuAD 1.1
full model	**71.8**	**89.1**
w/o control unit	68.5 (-3.3)	88.0 (-1.1)
w/o read unit	69.0 (-2.8)	87.4 (-1.7)
w/o write unit	69.2 (-2.6)	88.1 (-1.0)
w/o memory gate	70.4 (-1.4)	87.9 (-1.2)
w/o mem-con separation	68.0 (-3.8)	86.9 (-2.2)
w/o question attention	68.2 (-3.6)	87.7 (-1.4)

Table 4: Ablations of inferential module on RTE and SQuAD 1.1, using BERT$_{\text{base}}$ as contextual encoder.

Ablation Study

Semantics and Inference To evaluate the contribution of semantics and inference in our model, we perform ablation studies on the RTE and SQuAD dev sets as shown in Table 3. Here we focus on these components: (1) the whole inferential module (IM); (2) the semantic information (SI); (3) iterative reasoning (IR) that different reasoning cells attend to different predicate-argument structures. To evaluate their contribution, we perform experiments respectively by: (1) IM: removing the inferential module and simply combining the BERT embedding with semantic embeddings; (2) SI: removing all the semantic embeddings; (3) IR: combining multiple semantic embeddings as one and different reasoning steps taking the same semantic embedding.

As displayed in Table 3, the ablation on all evaluated components results in performance drop which indicates that both semantics and inference are indispensable for the model.

Analysis of Inferential Module To gain better insight into the relative contribution of three units, we perform extensive ablation studies in Figure 4. We remove the control, read and write unit by replacing $\mathbf{c_i}$, $\mathbf{r_i}$ and $\mathbf{m_i}$ with $\mathbf{bq_i}$, $\mathbf{m_{i-1}}$ and $\mathbf{r_i}$, respectively. All three ablations result in performance drop. Without control and read unit, the model cannot iteratively retrieve information from question and passage, which has been proved necessarily helpful for the performance. Without the write unit, the model assigns the newly retrieved information $\mathbf{r_i}$ to new memory $\mathbf{m_i}$ directly, which has no interaction with the control state $\mathbf{c_i}$. This also results in performance drop.

Control and Memory To further explore the influence of the control and memory, (1) we remove the memory gate, in which new memory $\mathbf{m_i}$ is computed by averaging the retrieved information $\mathbf{r_i}$ and previous memory $\mathbf{m_{i-1}}$. As expected, the performance gets worse. (2) Furthermore, we use one hidden state that plays both the roles of the control and memory (w/o mem-con separation in Table 4), iteratively attending and integrating information from the question and passage. This approach leads to a sharp performance drop(%3.8 on RTE, %2.2 on SQuAD). These results indicate that maintaining the separation setting and retrieving information from the interaction between memory and control are necessary for the model's accuracy.

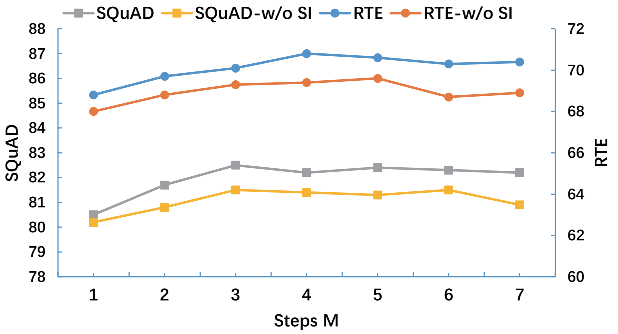

Figure 7: Results on the dev sets of SQuAD 2.0 and RTE when selecting different reasoning steps M, using BERT$_{\text{base}}$ as contextual encoder. SQuAD/RTE-w/o SI indicates the results without using any semantic information.

Question Attention This ablation (w/o question attention) shows that using attention over the question words in control unit (see Figure 4) is helpful for model performance. Using $\mathbf{w_i}$ in equation 1 instead of the attention based control state $\mathbf{c_i}$ leads to significant performance drop. This result illustrates the importance and usefulness of decomposing the question into multiple structures, such that a single cell is faced with learning one semantic structure or a few words in the question at a time, rather than modeling the question at once. This is intuitively shown in Figure 8.

Influence of Semantic Information

To further investigate the influence of semantic information, Figure 7 shows the performance comparison of whether to use the semantic information with different numbers of reasoning steps M (from 1 to 7). The highest performance is achieved when M is set to 3 on SQuAD, 4 on RTE. The results indicate that semantic information consistently contributes to the performance increase, although the inferential network is strong enough.

To investigate influence of the accuracy of the labeler, we randomly tune specific proportion [0, 20%, 40%] of labels into random error ones. The scores of SQuAD 2.0 and RTE are respectively [85.4, 83.2, 82.6] and [73.4, 71.8, 71.2], which indicate that the model benefits from high-accuracy labeler but can still maintain the performance even using some noisy labels.

To investigate the influence of SRL sequence ordering, we randomly shuffle the order to see the difference. Three groups of comparative experiments were carried out on

Passage: (S1) *Steel pipes and storage vessels used to store and transmit both gaseous and liquid oxygen will act as a fuel;* (S2) *and therefore the design and manufacture of oxygen systems requires special training to ensure that ignition sources are minimized.*

Question: *What does the transport and storage demand for safety in dealing with oxygen?*
Golden Answer: *special training*

SemBERT: *fuel* **SAIN**: *special training*

Table 5: One example that is correctly predicted by SAIN, but wrongly predicted by SemBERT.

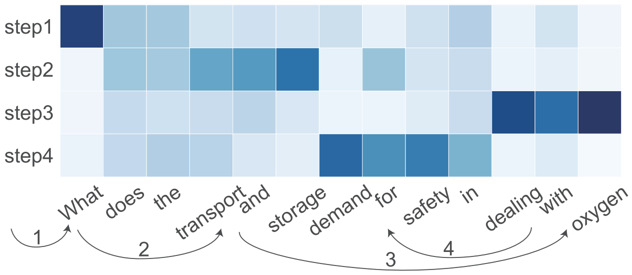

Figure 8: Transformation of attention distribution at each reasoning step, showing how the model iteratively retrieves information from the question.

SQuAD 2.0 and RTE. The scores of SQuAD 2.0 and RTE are respectively [85.3, 85.6, 85.5] and [73.3, 73.5, 73.4], which indicate that there is no significant difference when using various orderings of the SRL substructures.

Case Study

To obtain better insight into the underlying reasoning processes, we study the visualization of the attention distributions during the iterative computation, and provide examples in Table 5 and Figure 8. Table 5 shows a relatively complex question that is correctly answered by our model, but wrongly predicted by SemBERT (Zhang et al. 2020a). In this example, there is misleading contextual similarity between words "*store and transmit*" in sentence S1 and "*transport and storage*" in the question which may lead the model to wrong answer in S1, such as "*fuel*" by SemBERT. To overcome this misleading, the model needs to recognize the central connection predicates "*demand*" and "*requires*" between the question and passage, then extract the correct answer "*special training*" in S2.

Figure 8 shows how our model retrieves information from different semantic structures of the question in each reasoning step. The model first focuses on the word "*what*", working to retrieve a noun. Then it focuses on the arguments "*transport*" and "*storage*" in step 2 but gets around these words in step 3, and attends to the second verb phrase "*dealing with oxygen*", taking the model's attention away from sentence S1. Finally, the model focuses on the main meaning of the question: "*demand for security*" and predicts the correct answer "*special training*" in sentence S2, with respect to the semantic similarity between words "*demand for safety*" and "*requires to ensure*". This example intuitively explains why our model benefits from the iterative reasoning where each step only attends to one semantic representation.

Related Work

Semantic Information for MRC With the development of neural models(Zhang et al. 2019a; Zhou et al. 2020; Zhang et al. 2020b, 2019c), using semantic information to enhance the question answering system is one effective method to boost the performance. Narayanan and Harabagiu (2004) first stress the importance of semantic roles in dealing with complex questions. Shen and Lapata (2007) introduce a general framework for answer extraction which exploits semantic role annotations in the FrameNet (Baker, Fillmore, and Lowe 1998) paradigm. Yih et al. (2013) propose to solve the answer selection problem using enhanced lexical semantic models. More recently, Zhang et al. (2020a) propose to strengthen the language model representation by fusing explicit contextualized semantics. Mihaylov and Frank (2019) apply linguistic annotations to a discourse-aware semantic self-attention encoder which is employed for reading comprehension on narrative texts. (Weber et al. 2019) integrate Markov Logic Networks and Probabilistic Soft Logic with word embeddings to strengthen the model. In this work, we propose to use inferential model to recurrently retrieve different predicate-argument structures.

Inferential Network To support inference in neural network, exiting models either rely on structured rule-based matching methods (Sun, Cheng, and Qu 2018) or multilayer memory networks (Weston, Chopra, and Bordes 2014; Liu and Perez 2017), which either lack end-to-end design or no prior structure to subtly guide the reasoning direction. On Visual QA tasks, Hudson and Manning (2018) introduce an iterative network that separates memory and control to improve interpretability for compositional question. Our work leverages such separate design, dedicating to inferential NLU tasks.

To overcome the difficulty of applying inferential network into general NLU tasks, and passingly refine the use of multiple semantic structures, we propose SAIN which naturally decomposes text into different semantic structures for compositional processing in inferential network. Finally, we integrate semantics and inferential model in a creative way.

Conclusion

This work focuses on two typical NLU tasks, machine reading comprehension and natural language inference by refining the use of semantic clues and inferential model. The proposed semantics-aware inferential network (SAIN) is capable of taking multiple semantic structures as input of an inferential network by closely integrating semantics and reasoning steps in a creative way. Experiment results on 11 benchmarks, including 4 NLI tasks and 7 MRC tasks, show that our model outperforms all previous strong baselines, which consistently indicate the general effectiveness of our model.

References

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The Berkeley FrameNet Project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL)*, 86–90. doi:10.3115/980845.980860. URL https://www.aclweb.org/anthology/P98-1013.

Bentivogli, L.; Dagan, I.; Dang, H. T.; Giampiccolo, D.; and Magnini, B. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *In Proc Text Analysis Conference (TAC09)*.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 632–642. doi:10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075.

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 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 4171–4186. doi:10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Dunn, M.; Sagun, L.; Higgins, M.; Güney, V. U.; Cirik, V.; and Cho, K. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. URL http://arxiv.org/abs/1704.05179.

Fisch, A.; Talmor, A.; Jia, R.; Seo, M.; Choi, E.; and Chen, D. 2019. MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*, 1–13. doi:10.18653/v1/D19-5801. URL https://www.aclweb.org/anthology/D19-5801.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N. F.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 1–6. doi:10.18653/v1/W18-2501. URL https://www.aclweb.org/anthology/W18-2501.

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 (ACL)*, 473–483. doi:10.18653/v1/P17-1044. URL https://www.aclweb.org/anthology/P17-1044.

Hudson, D. A.; and Manning, C. D. 2018. Compositional Attention Networks for Machine Reasoning. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

Joshi, M.; Chen, D.; Liu, Y.; Weld, D. S.; Zettlemoyer, L.; and Levy, O. 2019. SpanBERT: Improving Pre-training by Representing and Predicting Spans. In *CoRR*, volume abs/1907.10529.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Khashabi, D.; Khot, T.; Sabharwal, A.; and Roth, D. 2018. Question Answering as Global Reasoning over Semantic Abstractions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Koisk, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K. M.; Melis, G.; and Grefenstette, E. 2018. The NarrativeQA Reading Comprehension Challenge. In *Transactions of the Association for Computational Linguistics (TACL)*, volume 6, 317328. ISSN 2307-387X. doi:10.1162/tacl_a_00023. URL http://dx.doi.org/10.1162/tacl_a_00023.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Kelcey, M.; Devlin, J.; Lee, K.; Toutanova, K. N.; Jones, L.; Chang, M.-W.; Dai, A.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: a Benchmark for Question Answering Research. In *Transactions of the Association of Computational Linguistics (TACL)*. URL https://tomkwiat.users.x20web.corp.google.com/papers/natural-questions/main-1455-kwiatkowski.pdf.

Liu, F.; and Perez, J. 2017. Gated End-to-End Memory Networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 1–10. URL https://www.aclweb.org/anthology/E17-1001.

Mihaylov, T.; and Frank, A. 2019. Discourse-Aware Semantic Self-Attention for Narrative Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, (EMNLP-IJCNLP)*, 2541–2552. doi:10.18653/v1/D19-1257. URL https://www.aclweb.org/anthology/D19-1257.

Narayanan, S.; and Harabagiu, S. 2004. Question Answering Based on Semantic Structures. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*, 693–701. URL https://www.aclweb.org/anthology/C04-1100.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. In *Computational Linguistics (CL)*, 71–106. doi:10.1162/0891201053630264. URL https://www.aclweb.org/anthology/J05-1004.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 784–789. doi:10.18653/v1/P18-2124. URL https://www.aclweb.org/anthology/P18-2124.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2383–2392.

Shen, D.; and Lapata, M. 2007. Using Semantic Roles to Improve Question Answering. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-CoNLL)*, 12–21. URL https://www.aclweb.org/anthology/D07-1002.

Su, D.; Xu, Y.; Winata, G. I.; Xu, P.; Kim, H.; Liu, Z.; and Fung, P. 2019. Generalizing Question Answering System with Pre-trained Language Model Fine-tuning. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (EMNLP)*, 203–211. doi:10.18653/v1/D19-5827. URL https://www.aclweb.org/anthology/D19-5827.

Sun, Y.; Cheng, G.; and Qu, Y. 2018. Reading Comprehension with Graph-based Temporal-Casual Reasoning. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 806–817. URL https://www.aclweb.org/anthology/C18-1069.

Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; and Wu, H. 2019. ERNIE: Enhanced Representation through Knowledge Integration. In *CoRR*, volume abs/1904.09223. URL http://arxiv.org/abs/1904.09223.

Takahashi, T.; Taniguchi, M.; Taniguchi, T.; and Ohkuma, T. 2019. CLER: Cross-task Learning with Expert Representation to Generalize Reading and Understanding. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 183–190. URL https://www.aclweb.org/anthology/D19-5824.

Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; and Suleman, K. 2017. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 191–200.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *The International Conference on Learning Representations (ICLR)*. URL http://arxiv.org/abs/1804.07461.

Weber, L.; Minervini, P.; Mnchmeyer, J.; Leser, U.; and Rocktschel, T. 2019. NLProlog: Reasoning with Weak Unification for Question Answering in Natural Language. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* doi:10.18653/v1/p19-1618. URL http://dx.doi.org/10.18653/v1/P19-1618.

Weston, J.; Chopra, S.; and Bordes, A. 2014. Memory Networks. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1112–1122. doi:10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

Yih, W.-t.; Chang, M.-W.; Meek, C.; and Pastusiak, A. 2013. Question Answering Using Enhanced Lexical Semantic Models. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1744–1753. URL https://www.aclweb.org/anthology/P13-1171.

Yu, J.; Zha, Z.; and Yin, J. 2019. Inferential Machine Comprehension: Answering Questions by Recursively Deducing the Evidence Chain from Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2241–2251. doi:10.18653/v1/P19-1217. URL https://www.aclweb.org/anthology/P19-1217.

Zhang, S.; Zhao, H.; Wu, Y.; Zhang, Z.; Zhou, X.; and Zhou, X. 2019a. DCMN+: Dual Co-Matching Network for Multi-choice Reading Comprehension. *CoRR* abs/1908.11511. URL http://arxiv.org/abs/1908.11511.

Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019b. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1441–1451. doi:10.18653/v1/P19-1139. URL https://www.aclweb.org/anthology/P19-1139.

Zhang, Z.; Wu, Y.; Li, Z.; and Zhao, H. 2018. Explicit Contextual Semantics for Text Comprehension. In *Proceedings of the 33nd Pacific Asia Conference on Language, Information and Computation (PACLIC 33)*. URL https://arxiv.org/abs/1809.02794.

Zhang, Z.; Wu, Y.; Li, Z.; and Zhao, H. 2019c. Explicit Contextual Semantics for Text Comprehension. In *Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33)*.

Zhang, Z.; Wu, Y.; Zhao, H.; Li, Z.; Zhang, S.; Zhou, X.; and Zhou, X. 2020a. Semantics-aware BERT for Language Understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Zhang, Z.; Zhang, Y.; ; Zhao, H.; Zhou, X.; and Zhou, X. 2020b. Composing Answer from Multi-spans for Reading Comprehension. *arXiv preprint arXiv:2009.06141* .

Zhou, J.; Zhang, Z.; Zhao, H.; and Zhang, S. 2020. LIMIT-BERT : Linguistics Informed Multi-Task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4450–4461. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.findings-emnlp.399. URL https://www.aclweb.org/anthology/2020.findings-emnlp.399.