Semantics-aware BERT for Language Understanding

Zhuosheng Zhang[1,2,3,*], **Yuwei Wu**[1,2,3,4,*], **Hai Zhao**[1,2,3,†],
Zuchao Li[1,2,3], **Shuailiang Zhang**[1,2,3], **Xi Zhou**[5], **Xiang Zhou**[5]
[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]College of Zhiyuan, Shanghai Jiao Tong University, China
[5]CloudWalk Technology, Shanghai, China
{zhangzs,will8821}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,
{charlee,zsl123}@sjtu.edu.cn, {zhouxi,zhouxiang}@cloudwalk.cn

Abstract

The latest work on language representations carefully integrates contextualized features into language model training, which enables a series of success especially in various machine reading comprehension and natural language inference tasks. However, the existing language representation models including ELMo, GPT and BERT only exploit plain context-sensitive features such as character or word embeddings. They rarely consider incorporating structured semantic information which can provide rich semantics for language representation. To promote natural language understanding, we propose to incorporate explicit contextual semantics from pre-trained semantic role labeling, and introduce an improved language representation model, Semantics-aware BERT (SemBERT), which is capable of explicitly absorbing contextual semantics over a BERT backbone. SemBERT keeps the convenient usability of its BERT precursor in a light fine-tuning way without substantial task-specific modifications. Compared with BERT, semantics-aware BERT is as simple in concept but more powerful. It obtains new state-of-the-art or substantially improves results on ten reading comprehension and language inference tasks.

1 Introduction

Recently, deep contextual language model (LM) has been shown effective for learning universal language representations, achieving state-of-the-art results in a series of flagship natural language understanding (NLU) tasks. Some prominent examples are Embedding from Language models (ELMo) (Peters et al., 2018), Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) and Generalized Autoregressive Pretraining (XLNet) (Yang et al., 2019). Providing fine-grained contextual embedding, these pre-trained models could be either easily applied to downstream models as the encoder or used for fine-tuning.

Despite the success of those well pre-trained language models, we argue that current techniques which only focus on language modeling restrict the power of the pre-trained representations. The major limitation of existing language models lies in only taking plain contextual features for both representation and training objective, rarely considering explicit contextual semantic clues. Even though well pre-trained language models can implicitly represent contextual semantics more or less (Clark et al., 2019), they can be further enhanced by incorporating external knowledge. To this end, there is a recent trend of incorporating extra knowledge to pre-trained language models (Zhang et al., 2019; Sun et al., 2019).

A number of 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). Through their observation, deep learning models pay great attention to non-significant words and ignore important ones. For attractive question answering challenge (Rajpurkar et al., 2016), we observe a number of answers produced by previous models are semantically incomplete (As shown in Section 6.2), which suggests that the current NLU models suffer from insufficient contextual semantic representation and learning.

Actually, NLU tasks share the similar task purpose as sentence contextual semantic analysis. Briefly, semantic role labeling (SRL) over a sentence is to discover *who did what to whom*, *when and why* with respect to the central meaning of the sentence, which naturally matches the task target of NLU. For example, in question answering tasks, questions are usually formed with who, what, how, when and why, which can be conveniently formulized into the predicate-argument relationship in terms of contextual semantics.

* These authors contribute equally. † Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100) and Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

In human language, a sentence usually involves various perspectives of meaning, while neural models encode sentence into embedding representation, with little consideration of the modeling of multiple semantic structures. Thus we are motivated to enrich the sentence contextual semantics in different perspectives of predicate-specific argument sequence by presenting SemBERT: Semantics-aware BERT, which is a fine-tuned BERT with explicit contextual semantic clues. The proposed SemBERT learns the representation in a fine-grained manner and takes both strengths of BERT on plain context representation and explicit semantics for deeper meaning representation.

Our model consists of three components: 1) an out-of-shelf semantic role labeler to annotate the input sentences with a variety of semantic role labels; 2) an sequence encoder where a pre-trained language model is used to build representation for input raw texts and the semantic role labels are mapped to embedding in parallel; 3) a semantic integration component to integrate the text representation with the contextual explicit semantic embedding to obtain the joint representation for downstream tasks.

The proposed SemBERT will be directly applied to typical NLU tasks. Our model is evaluated on 11 benchmark datasets involving natural language inference, question answering, semantic similarity and text classification. SemBERT obtains new state-of-the-art on SNLI and also obtains significant gains on the GLUE benchmark and SQuAD 2.0. Ablation studies and analysis verify that our introduced explicit semantics is essential to the further performance improvement and SemBERT essentially and effectively works as a unified semantics-enriched language representation model.

2 Background and Related Work

2.1 Language Modeling for NLU

Natural language understanding tasks require a comprehensive understanding of natural languages and the ability to do further inference and reasoning. A common trend among NLU studies is that models are becoming more and more sophisticated with stacked attention mechanisms or large amount of corpus (Joshi et al., 2019; Liu et al., 2019b), resulting in explosive growth of computational cost. Notably, well pre-trained contextual language models such as ELMo (Peters et al., 2018), GPT (Radford et al., 2018) and BERT (Devlin et al., 2018) have been shown powerful to boost NLU tasks to reach new high performance.

Distributed representations have been widely used as a standard part of NLP models due to the ability to capture the local co-occurence of words from large scale unlabeled text (Mikolov et al., 2013; Pennington et al., 2014). However, these approaches for learning word vectors only involve a single, context independent representation for each word with litter consideration of contextual encoding in sentence level. Thus recently introduced contextual language models including ELMo, GPT, BERT and XLNet fill the gap by strengthening the contextual sentence modeling for better representation, among which BERT uses a different pre-training objective, masked language model, which allows capturing both sides of context, left and right. Besides, BERT also introduces a *next sentence prediction* task that jointly pre-trains text-pair representations. The latest evaluation shows that BERT is powerful and convenient for downstream NLU tasks.

The major technical improvement over traditional embeddings of these newly proposed language models is that they focus on extracting context-sensitive features from language models. When integrating these contextual word embeddings with existing task-specific architectures, ELMo helps boost several major NLP benchmarks (Peters et al., 2018) including question answering on SQuAD, sentiment analysis (Socher et al., 2013), and named entity recognition (Sang and De Meulder, 2003), while BERT especially shows effective on language understanding tasks on GLUE, MultiNLI and SQuAD (Devlin et al., 2018). In this work, we follow this line of extracting context-sensitive features and take pre-trained BERT as our backbone encoder for jointly learning explicit context semantics.

2.2 Explicit Contextual Semantics

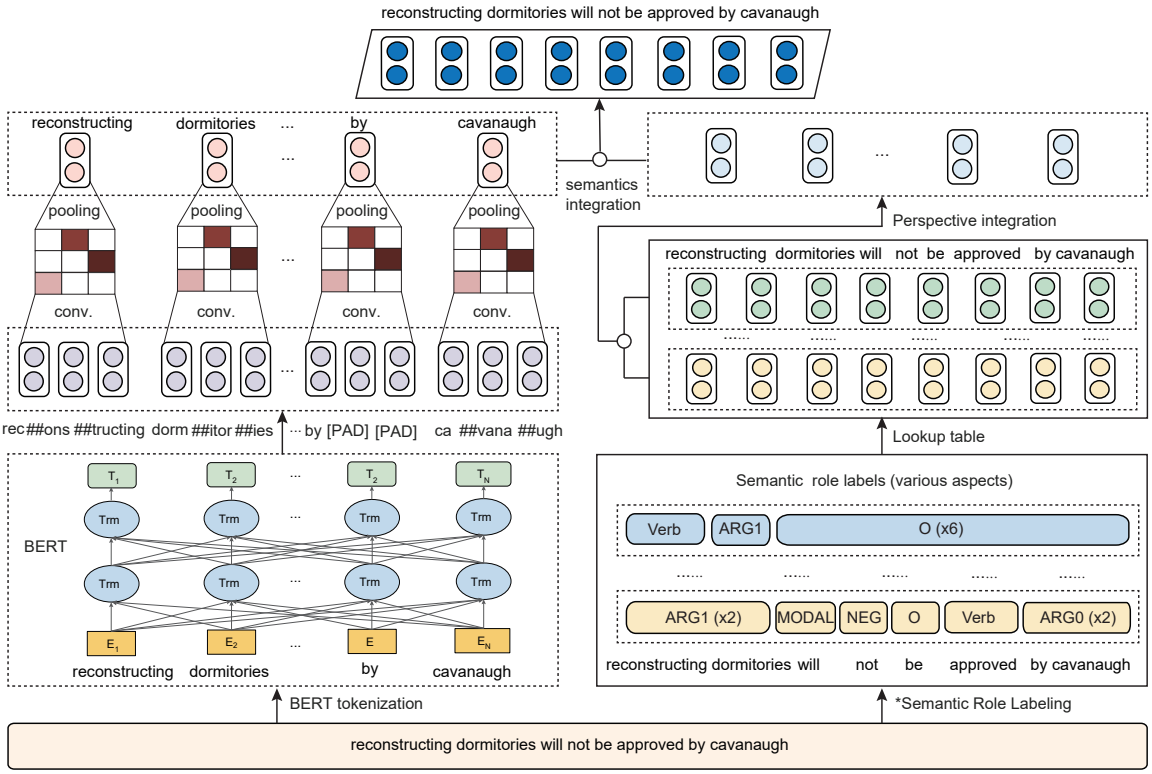
Although distributed representations including the latest advanced pre-trained contextual language models have already been strengthened by semantics to some extent from linguistic sense (Clark et al., 2019), we argue such implicit semantics may not be enough to support a powerful contextual representation for NLU, according to our observation on the semantically incomplete answer span generated by BERT on SQuAD, which motivates us to directly introduce explicit semantics.

There are a few formal semantic frames, including FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005), in which the latter is more popularly implemented in computational linguistics. Formal semantics generally presents the semantic relationship as predicate-argument structure. For example, given the following sentence with target verb (predicate) *sold*, all the arguments are labeled as follows,

[$_{ARG0}$ 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.

To parse the predicate-argument structure, we have an NLP task, semantic role labeling (SRL), which 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



For the text, {*reconstructing dormitories will not be approved by cavanaugh*}, it will be tokenized to a subword-level sequence, {*rec, ##ons, ##tructing, dorm, ##itor, ##ies, will, not, be, approved, by, ca, ##vana, ##ugh*}. Meanwhile, there are two kinds of word-level semantic structures,
[ARG1: reconstructing dormitories] [ARGM-MOD: will] [ARGM-NEG: not] be [V: approved] [ARG0: by cavanaugh]
[V: reconstructing] [ARG1: dormitories] will not be approved by cavanaugh

Figure 1: Semantics-aware BERT. The pre-trained labeler will not be fine-tuned in our framework.

subtasks one after another. Traditional systems relied on sophisticated handcraft features or some declarative constraints (Pradhan et al., 2005). Recently, Zhou and Xu (2015) and He et al. (2017) introduced end-to-end neural 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 NLU. The pioneering work on building an end-to-end neural system was presented by Zhou and Xu (2015), applying an LSTM model, which takes only original text as input 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.

3 Semantics-aware BERT

Figure 1 overviews our semantics-aware BERT framework. We omit rather extensive formulations of BERT and recommend readers to get the details from (Devlin et al., 2018). SemBERT is designed to be capable of handling multiple sequence inputs. In SemBERT, words in the input sequence are passed to semantic role labeler to fetch multiple predicate-derived structures of explicit semantics and the corresponding embeddings are aggregated after a linear layer to form the final semantic embedding. In parallel, the input sequence is segmented to subwords (if any) by BERT word-piece tokenizer, then the subword representation is transformed back to word level via a convolutional layer to obtain the contextual word representations. At last, the word representations and semantic embedding are concatenated to form the joint representation for downstream tasks.

3.1 Semantic Role Labeling

During the data pre-processing, each sentence is annotated into several semantic sequences using our pre-trained semantic labeler. We take PropBank style (Palmer et al., 2005) of semantic roles to annotate every token of input sequence with semantic labels. Given a specific sentence, there would be various perspectives of meaning [1]. As shown in Figure 1, for the text, [*reconstructing dormitories will not be approved by cavanaugh*], there are semantic structures in the view of the predicates in the sentence,

[ARG1: reconstructing dormitories] [ARGM-MOD: will] [ARGM-NEG: not] be [V: approved] [ARG0: by cavanaugh]

[1] We hypothesize that each predicate-specific argument sequence reflects different *perspective* of sentence meaning.

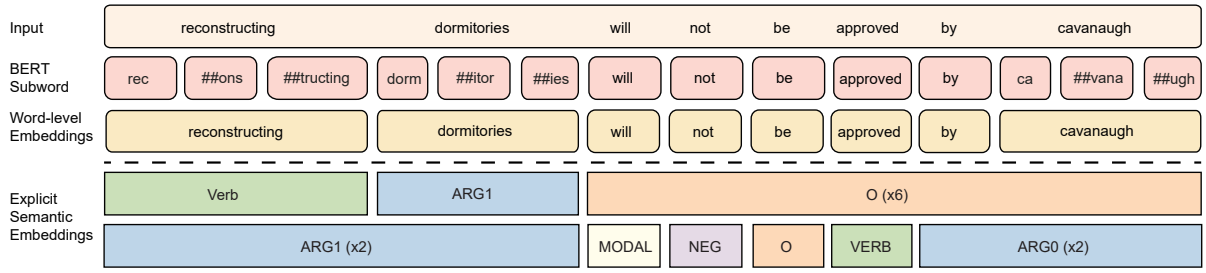


Figure 2: The input representation flow.

[V: reconstructing] [ARG1: dormitories] will not be approved by cavanaugh

To disclose the multidimensional semantics, we group the semantic labels and integrate them with text embeddings in the next encoding component. The input data flow is depicted in Figure 2.

3.2 Encoding

The raw text sequences and semantic role label sequences are firstly represented as embedding vectors to feed a pre-trained BERT. The input sentence $X = \{x_1, \ldots, x_n\}$ is a sequence of words of length n, which is first tokenized to word pieces (subword tokens). Then the transformer encoder captures the contextual information for each token via self-attention and produces a sequence of contextual embeddings.

For m label sequences related to each predicate (perspective), we have $T = \{t_1, \ldots, t_m\}$ where t_i contains n labels denoted as $\{label_1^i, label_2^i, ..., label_n^i\}$. Since our labels are in word-level, the length is equal to the original sentence length n of X. We regard the semantic signals as embeddings and use a lookup table to map these labels to vectors $\{v_1^i, v_2^i, ..., v_n^i\}$ and feed a BiGRU layer to obtain the label representations for m label sequences in latent space, $e(t_i) = BiGRU(v_1^i, v_2^i, \ldots, v_n^i)$ where $0 < i \leqslant m$. For m perspectives, let L_i denote the label perspectives for token x_i, we have $e(L_i) = \{e(t_1), \ldots, e(t_m)\}$. We concatenate the m perspectives of label representation and feed them to a fully connected layer to obtain the refined joint representation e^t in dimension d:

$$\begin{aligned} e'(L_i) &= W_2\left[e(t_1), e(t_2), \ldots, e(t_m)\right] + b_2, \\ e^t &= \{e'(L_1), ..., e'(L_n)\}, \end{aligned} \tag{1}$$

where W_2 and b_2 are trainable parameters.

3.3 Integration

This integration module fuses the lexical text embedding and label representations. As the original pre-trained BERT is based on a sequence of subwords, while our introduced semantic labels are on words, we need to align these different sized sequences. Thus we group the subwords for each word and use convolutional neural network (CNN) with a max pooling to obtain the representation in word-level. We select CNN because of fast speed and our preliminary experiments show that it also gives better results than RNNs in our concerned tasks where we think the local feature captured by CNN would be beneficial for subword-derived LM modeling.

We take one word for example. Supposing that word x_i is made up of a sequence of subwords $[s_1, s_2, ..., s_l]$, where l is the number of subwords for word x_i. Denoting the representation of subword s_j from BERT as $e(s_j)$, we first utilize a Conv1D layer,

$$e_i' = W_1\left[e(s_i), e(s_{i+1}), \ldots, e(s_{i+k-1})\right] + b_1 \tag{2}$$

where W_1 and b_1 are trainable parameters and k is the kernel size. We then apply ReLU and max pooling to the output embedding sequence for x_i:

$$\begin{aligned} e_i^* &= ReLU(e_i'), \\ e(x_i) &= MaxPooling(e_1^*, e_2^*, ..., e_{l-k+1}^*), \end{aligned} \tag{3}$$

Therefore, the whole representation for word sequence X is represented as $e^w = \{e(x_1), \ldots e(x_n)\} \in \mathbb{R}^{n \times d_w}$ where d_w denotes the dimension of word embedding.

The aligned context and distilled semantic embeddings are then merged by a fusion function $h = e^w \diamond e^t$, where $\diamond$ represents concatenation operation[2].

4 Model Implementation

Now, we introduce the specific implementation parts of our SemBERT. SemBERT could be a forepart encoder for a wide range of tasks and could also become an end-to-end model with only a linear layer for prediction. For simplicity, we only show the straightforward SemBERT that directly gives the predictions after fine-tuning[3].

4.1 Semantic Role Labeler

To obtain the semantic labels, we use a pre-trained SRL module to predict all predicates and corresponding arguments in one shot. Following the state-of-the-art model in (He et al., 2017), our semantic role labeler is trained on English *OntoNotes v5.0* benchmark

[2]We also tried summation, multiplication and attention mechanisms, but our experiments show that concatenation is the best.

[3]We only use *single* model for each task without jointly training and parameter sharing.

Method	Classification		Natural Language Inference			Semantic Similarity			Avg.
	CoLA (mc)	SST-2 (acc)	MNLI m/mm(acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	STS-B (pc)	-
Leaderboard									
ALICE large	65.3	95.2	88.0/87.7	95.7	83.1	92.0	74.1	90.3	83.9
MT-DNN++(BigBird)	65.4	95.6	87.9/87.4	95.8	85.1	91.1	72.7	89.6	83.8
Snorkel MeTaL	63.8	96.2	87.6/87.2	93.9	80.9	91.5	73.1	90.1	83.2
BERT + BAM	61.5	95.2	86.6/85.8	93.1	80.4	91.3	72.5	88.6	82.3
BERT on STILTs	62.1	94.3	86.4/85.6	92.7	80.1	90.2	71.9	88.7	82.0
In literature									
BiLSTM+ELMo+Attn	36.0	90.4	76.4/76.1	79.9	56.8	84.9	64.8	75.1	70.5
GPT	45.4	91.3	82.1/81.4	88.1	56.0	82.3	70.3	82.0	72.8
GPT on STILTs	47.2	93.1	80.8/80.6	87.2	69.1	87.7	70.1	85.3	76.9
MT-DNN	61.5	95.6	86.7/86.0	-	75.5	90.0	72.4	88.3	82.2
$\text{BERT}_{\text{BASE}}$	52.1	93.5	84.6/83.4	-	66.4	88.9	71.2	87.1	78.3
$\text{BERT}_{\text{LARGE}}$	60.5	94.9	86.7/85.9	92.7	70.1	89.3	72.1	87.6	80.5
Our implementation									
$\text{SemBERT}_{\text{BASE}}$	57.8	93.5	84.4/84.0	90.9	69.3	88.2	71.8	87.3	80.9
$\text{SemBERT}_{\text{LARGE}}$	62.3	94.6	87.6/86.3	94.6	84.5	91.2	72.8	87.8	82.9

Table 1: Results on GLUE benchmark. All the results are obtained from (Liu et al., 2019a), (Radford et al., 2018) and the GLUE leaderboard (https://gluebenchmark.com/leaderboard) at the time of submitting SemBERT (15 April, 2019). We exclude the problematic WNLI set and do not show the accuracy of the datasets have F1 scores to save space. *mc* and *pc* denote the Matthews correlation and Pearson correlation, respectively.

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. In our implementation, there are 104 labels in total. We use *O* for non-argument words and *Verb* label for predicates.

4.2 Task-specific Fine-tuning

In Section 3, we have described how to obtain the semantics-aware BERT representations. Here, we show how to adapt SemBERT to classification, regression and span-based MRC tasks. We transform the fused contextual semantic and LM representations h to a lower dimension and obtain the prediction distributions. Note that this part is basically the same as the implementation in BERT without any modification, to avoid extra influence and focus on the intrinsic performance of SemBERT. We outline here to keep the completeness of the implementation.

For classification and regression tasks, h is directly passed to a fully connection layer to get the class logits or score, respectively. The training objectives are CrossEntropy for classification tasks and Mean Square Error loss for regression tasks.

For span-based reading comprehension, h is passed to a fully connection layer to get the start logits s and end logits e of all tokens. The score of a candidate span from position i to position j is defined as $s_i + e_j$, and the maximum scoring span where $j \geq i$ is used as a prediction[5]. For prediction, we compare the score of the pooled first token span: $s_{\texttt{null}} = s_0 + e_0$ to the score of the best non-null span $\hat{s_{i,j}} = \texttt{max}_{j \geq i}(s_i + e_j)$. We predict a non-null answer when $\hat{s_{i,j}} > s_{\texttt{null}} + \tau$, where the threshold τ is selected on the dev set to maximize F1.

5 Experiments

5.1 Setup

Our implementation is based on the PyTorch implementation of BERT[6]. We use the pre-trained weights of BERT and follow the same fine-tuning procedure as BERT without any modification, and all the layers are tuned with moderate model size increasing, as the extra SRL embedding volume is less than 15% of the original encoder size. We use Adam as our optimizer with an initial learning rate in {8e-6, 1e-5, 2e-5, 3e-5} with warm-up rate of 0.1 and L2 weight decay of 0.01. The batch size is selected in {16, 24, 32}. The maximum number of epochs is set in [2, 5] depending on tasks. Texts are tokenized using wordpieces, with maximum length of 384 for SQuAD and 200 for other tasks. The dimension of SRL embedding is set to 10. Hyper-parameters were selected using the dev set.

5.2 Tasks and Datasets

Our evaluation is performed on ten NLU benchmark datasets involving natural language inference, machine reading comprehension, semantic similarity and text classification. Some of these tasks are available from the recently released GLUE benchmark (Wang et al.,

[4]This result nearly reaches the SOTA in (He et al., 2018).

[5]All the candidate scores are normanized by softmax.

[6]https://github.com/huggingface/pytorch-transformers.

Model	Params (M)	Shared (M)	Rate
MT-DNN	3,060	340	9.1
BERT on STILTs	335	-	1.0
BERT	335	-	1.0
SemBERT	340	-	1.0

Table 2: Parameter Comparison on LARGE models. The numbers are from GLUE leaderboard.

Model	EM	F1
#1 BERT + DAE + AoA†	85.9	88.6
#2 SG-NET†	85.2	87.9
#3 BERT + NGM + SST†	85.2	87.7
U-Net (Sun et al., 2018)	69.2	72.6
RMR + ELMo + Verifier (Hu et al., 2018)	71.7	74.2
Our implementation		
$\text{BERT}_{\text{LARGE}}$	80.5	83.6
$\text{SemBERT}_{\text{LARGE}}$	82.4	85.2
$\text{SemBERT}^{*}_{\text{LARGE}}$	84.8	87.9

Table 3: Exact Match (EM) and F1 scores on SQuAD 2.0 test set for single models. † denotes the top 3 single submissions from the leaderboard at the time of submitting SemBERT (11 April, 2019). The top results from the SQuAD leaderboard do not have public model descriptions available, and are allowed to use any public data for system training. We therefore further adopt synthetic self training[7] for data augmentation, denoted as $\text{SemBERT}^{*}_{\text{LARGE}}$.

2018), which is a collection of nine NLU tasks. We also extend our experiments to two widely-used tasks, SNLI (Bowman et al., 2015) and SQuAD 2.0 (Rajpurkar et al., 2018) to show the superiority.

Reading Comprehension As a widely used MRC benchmark dataset, SQuAD 2.0 (Rajpurkar et al., 2018) combines the 100,000 questions in SQuAD 1.1 (Rajpurkar et al., 2016) with over 50,000 new, unanswerable questions that are written adversarially by crowdworkers to look similar to answerable ones. For SQuAD 2.0, systems must not only answer questions when possible, but also abstain from answering when no answer is supported by the paragraph.

Natural Language Inference Natural Language Inference involves reading a pair of sentences and judging the relationship between their meanings, such as entailment, neutral and contradiction. We evaluate on 4 diverse datasets, including Stanford Natural Language Inference (SNLI) (Bowman et al., 2015), Multi-Genre Natural Language Inference (MNLI) (Nangia et al., 2017), Question Natural Language Inference (QNLI) (Rajpurkar et al., 2016) and Recognizing Textual Entailment (RTE) (Bentivogli et al., 2009).

[7] https://nlp.stanford.edu/seminar/details/jdevlin.pdf

Model	Dev	Test
In literature		
GPT (Radford et al., 2018)	-	89.9
DRCN (Kim et al., 2018)	-	90.1
MT-DNN (Liu et al., 2019a)	91.4	91.1
Our implementation		
$\text{BERT}_{\text{BASE}}$	90.8	90.7
$\text{BERT}_{\text{LARGE}}$	91.3	91.1
$\text{SemBERT}_{\text{BASE}}$	91.2	91.0
$\text{SemBERT}_{\text{LARGE}}$	92.3	91.6

Table 4: Accuracy on SNLI dataset. Previous state-of-the-art result is marked by †. Both our SemBERT and BERT are single models, fine-tuned based on the pre-trained models.

Semantic Similarity Semantic similarity tasks aim to predict whether two sentences are semantically equivalent or not. The challenge lies in recognizing rephrasing of concepts, understanding negation, and handling syntactic ambiguity. Three datasets are used, including Microsoft Paraphrase corpus (MRPC) (Dolan and Brockett, 2005), Quora Question Pairs (QQP) dataset (Chen et al., 2018) and Semantic Textual Similarity benchmark (STS-B) (Cer et al., 2017).

Classification The Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2018) is used to predict whether an English sentence is linguistically acceptable or not. The Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) provides a dataset for sentiment classification that needs to determine whether the sentiment of a sentence extracted from movie reviews is positive or negative.

5.3 Results

Table 1 shows results on the GLUE benchmark datasets, showing SemBERT gives substantial gains over BERT and outperforms all the previous state-of-the-art models in literature. Since SemBERT takes BERT as the backbone with the same evaluation procedure, the gain is entirely owing to newly introduced explicit contextual semantics. Though recent dominant models take advance of multi-tasking, knowledge distillation, transfer learning or ensemble, our single model is lightweight and competitive, even yields better results with simple design and less parameters. Model parameter comparison is shown in Table 2. We observe that without multi-task learning like MT-DNN[8], our model still achieves remarkable results.

Table 3 shows the results for reading comprehension on SQuAD 2.0 test set[9]. SemBERT boosts the strong BERT baseline essentially on both EM and F1. It also

[8] Since MT-DNN is a multi-task learning framework with shared parameters on 9 task-specific layers, we count their 340M shared parameters for nine times for fair comparison.

[9] There is a restriction of submission frequency for online SQuAD 2.0 evaluation, we do not submit our base models.

Question	Baseline	SemBERT
What is the prize offered for finding a solution to P=NP?	US	US $1,000,000
What monastery did the Saint-Evroul monks establish in Italy?	Sant	Sant'Eufemia
What is a very seldom used unit of mass in the metric system?	The ki	metric slug
What is the lone MLS team that belongs to southern California?	Galaxy	LA Galaxy
How many people does the Greater Los Angeles Area have?	17.5 million	over 17.5 million

Table 5: The comparison of answers from baseline and our model. In these examples, answers from SemBERT are the same as the ground truth.

Model	**SNLI**	**SQuAD 2.0**	
	Dev	**EM**	**F1**
$\text{BERT}_{\text{LARGE}}$	91.3	79.6	82.4
$\text{BERT}_{\text{LARGE}}$+SRL	91.5	80.3	83.1
$\text{SemBERT}_{\text{LARGE}}$	92.3	80.9	83.6

Table 6: Analysis on SNLI and SQuAD 2.0 datasets.

outperforms all the published works and achieves comparable performance with a few unpublished models from the leaderboard.

Table 4 shows SemBERT also achieves a new state-of-the-art on SNLI benchmark and even outperforms all the ensemble models[10] by a large margin.

6 Analysis

6.1 Ablation Study

To evaluate the contributions of key factors in our method, we perform an ablation study on the SNLI and SQuAD 2.0 dev sets as shown in Table 6. Since SemBERT absorbs contextual semantics in a deep processing way, we wonder if a simple and straightforward way integrating such semantic information may still work, thus we concatenate the SRL embedding with BERT subword embeddings for a direct comparison, where the semantic role labels are copied to the number of subwords for each original word, without CNN and pooling for word-level alignment. From the results, we observe that the concatenation would yield an improvement, verifying that integrating contextual semantics would be quite useful for language understanding. However, SemBERT still greatly outperforms the simple BERT+SRL model just like the latter outperforms the original BERT by a large performance margin, which shows that SemBERT works more effectively for integrating both plain contextual representation and contextual semantics at the same time.

6.2 Model Prediction

To have an intuitive observation of the predictions of SemBERT, we show a list of prediction examples on SQuAD 2.0 from baseline BERT and SemBERT in Table 5. The comparison indicates that our model could extract more semantically accurate answer, yielding more exact match answers while those from the baseline BERT model are often semantically incomplete. This shows that utilizing explicit semantics is potential to guide the model to produce meaningful predictions. Intuitively, the advance would attribute to better awareness of semantic role spans, which guides the model to learn the patterns like *who did what to whom* explicitly.

Through the comparison, we observe SemBERT might benefit from better span segmentation through span-based SRL labeling. We conduct a case study on our best model of SQuAD 2.0, by transforming SRL into segmentation tags to indicate which token is inside or outside the segmented span. The result is 83.69(EM)/87.02(F1), which shows that the segmentation indeed works but marginally beneficial compared with our complete architecture.

It is worth noting that we are motivated to use the SRL signals to help the model to capture the span relationships inside sentence, which results in both sides of semantic label hints and segmentation benefits across semantic role spans to some extent. The segmentation could also be regarded as the awareness of semantics even with better semantic span segmentations. Intuitively, this indicates that our model evolves from BERT subword-level representation to intermediate word-level and final semantic representations.

6.3 Infulence of Accuracy of SRL

Our model relies on a semantic role labeler that would influence the overall model performance. To investigate influence of the accuracy of the labeler, we degrade our labeler by randomly turining specific proportion [0, 20%, 40%] of labels into random error ones as cascading errors. The F1 scores of SQuAD are respectively [87.93, 87.31, 87.24]. This advantage can be attributed to the concatenation operation of BERT hidden states and SRL representation, in which the lower dimensional SRL representation (even noisy) would not affect the former one intensely. This result indicates that the LM can not only benefit from high-accuracy labeler but also keep robust against noisy labels.

Besides the wide range of tasks verified in this work, SemBERT could also be easily adapted to other languages. As SRL is a fundamental NLP task, it is convenient to train a labeler for main languages as CoNLL 2009 provides 7 SRL treebanks. For those without available treebanks, unsupervised SRL methods can

[10] https://nlp.stanford.edu/projects/snli/. As ensemble models are commonly composed of multiple heterogeneous models and resources, we exclude them in our table to save space.

be effectively applied. For out-of-domain issue, the datasets (GLUE and SQuAD) that we are working on cover quite diverse domains, and experiments show that our method still works.

7 Conclusion

This paper proposes a novel semantics-aware BERT network architecture for fine-grained language representation. Experiments on a wide range of NLU tasks including natural language inference, question answering, machine reading comprehension, semantic similarity and text classification show the superiority over the strong baseline BERT. Our model has surpassed all the published works in all of the concerned NLU tasks. This work discloses the effectiveness of semantics-aware BERT in natural language understanding, which demonstrates that explicit contextual semantics can be effectively integrated with state-of-the-art pre-trained language representation for even better performance improvement. Recently, most works focus on heuristically stacking complex mechanisms for performance improvement, instead, we hope to shed some lights on fusing accurate semantic signals for deeper comprehension and inference through a simple but effective method[11].

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *ACL-PASCAL*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[11] After this work was done, we noticed the preprints of XLNet and RoBERTa. It is also potential to incorporate explicit semantics to other LMs which is left for future work.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *IWP2005*.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *ACL*.

Luheng He, Kenton Lee, Mike Lewis, Luke Zettlemoyer, Luheng He, Kenton Lee, Mike Lewis, Luke Zettlemoyer, Luheng He, and Kenton Lee. 2017. Deep semantic role labeling: What works and whats next. In *ACL*.

Minghao Hu, Yuxing Peng, Zhen Huang, Nan Yang, Ming Zhou, et al. 2018. Read+ verify: Machine reading comprehension with unanswerable questions. *arXiv preprint arXiv:1808.05759*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Seonhoon Kim, Jin-Hyuk Hong, Inho Kang, and Nojun Kwak. 2018. Semantic sentence matching with densely-connected recurrent and co-attentive information. *arXiv preprint arXiv:1805.11360*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *ACL*.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. In *RepEval*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

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 *CoNLL*.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H Martin, and Daniel Jurafsky. 2005. Semantic role labeling using different syntactic views. In *ACL*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Technical report*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Fu Sun, Linyang Li, Xipeng Qiu, and Yang Liu. 2018. U-net: Machine reading comprehension with unanswerable questions. *arXiv preprint arXiv:1810.06638*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *2018 EMNLP Workshop BlackboxNLP*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

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