Sentence Matching with Syntax- and Semantics-Aware BERT

Tao Liu[1], **Xin Wang**[2], **Chengguo Lv**[1], **Ranran Zhen**[1] and **Guohong Fu**[3*]
1. School of Computer Science and Technology, Heilongjiang University, China
2. CuraCloud Corporation, Seattle, WA, USA
3. Institute of Artificial Intelligence, Soochow University, China
taoooo1009@gmail.com, xinw@curacloudcorp.com
2004085@hlju.edu.cn, zenrran@gmail.com
ghfu@suda.edu.cn

Abstract

Sentence matching aims to identify the special relationship between two sentences, and plays a key role in many natural language processing tasks. However, previous studies mainly focused on exploiting either syntactic or semantic information for sentence matching, and no studies consider integrating both of them. In this study, we propose integrating syntax and semantics into BERT with sentence matching. In particular, we use an implicit syntax and semantics integration method that is less sensitive to the output structure information. Thus the implicit integration can alleviate the error propagation problem. The experimental results show that our approach has achieved state-of-the-art or competitive performance on several sentence matching datasets, demonstrating the benefits of implicitly integrating syntactic and semantic features in sentence matching.

1 Introduction

Sentence matching aims to determine the specific relationship between two sentences. It plays a key role in many natural language processing tasks such as natural language inference (Bowman et al., 2015), paraphrase identification (Wang et al., 2017), and answer selection (Yang et al., 2015).

Learning contextual representations is fundamental to many sentence matching tasks. Earlier works explore contextual features of sentences individually in a neural network classifier to predict sentence relationship (Bowman et al., 2015; Chen et al., 2017a). However, they do not consider the more fine-grained comparison information such as word-to-word correspondence between sentences. Recent studies achieve better results by taking into account contextual features and fine-grained interaction between sentences at the same time (Tay et al., 2018a; Kim et al., 2019; Yang et al., 2019a). More recently, with the development of pre-training language models (Radford et al., 2018; Devlin et al., 2019; Yang et al., 2019b), substantial progress has been made in sentence matching tasks. For example, BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) has proven to be a powerful representation of contexts for sentence relationship prediction, in that BERT representations can implicitly capture a rich hierarchy of linguistic information within sentences (Jawahar et al., 2019).

In addition to contextual representation, structural information such as syntactic structures and semantic roles has shown to be effective to enhance sentence matching tasks and other NLP systems (Chen et al., 2017a; Zhang et al., 2020). On the one hand, with the maturity of dependency parsing and semantic role labeling technologies in recent years, it is relatively easier to obtain reliable structural information for sentence matching. On the other hand, integrating structural information with pre-training language models has been a promising way to enhance sentence matching performance. Semantic role labeling (SRL) is expressed as predicate-argument relationships to explore who did what to whom, when and why for representing the central meaning of the sentence, which is the same as judging their relationship by understanding two sentences. Dependency parser can represent the dependency of words, which can compare from word level to reveal its syntactic structure and compare their difference. To enrich

*Corresponding author.

Proceedings of the 28th International Conference on Computational Linguistics, pages 3302–3312
Barcelona, Spain (Online), December 8-13, 2020

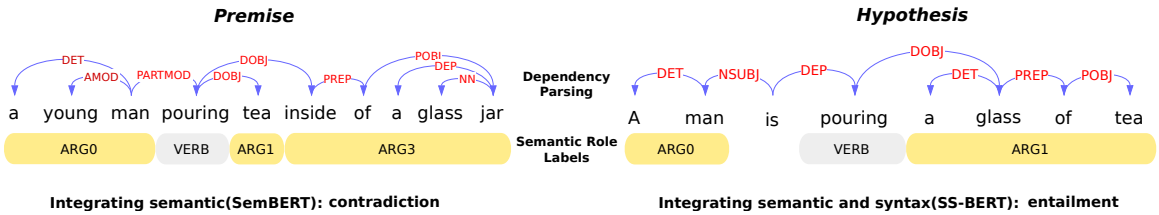

Figure 1: An example of the natural language inference task, the gold label is *entailment*, and the syntactic and semantic structural representation of two sentences are shown on the top and bottom respectively. In semantic role labeling, they have different predicate-argument structures, which may lead to the wrong prediction. However, in dependency parsing, they have the same subject, predicate and object (i.e. person, pour, and tea), which makes the two sentences have the same backbone. Thus, combining both syntax and semantics can help the model predict the correct result.

the contextual sentence representation of BERT, SemBERT (Zhang et al., 2020) has shown encouraging improvement, which first integrates the semantic labels into the BERT and obtains state-of-the-art performance on sentence matching. Moreover, SemBERT uses explicit method to integrate the semantic information, which highly depends on the accuracy of the semantic structure information. As the input of sentence matching, incorrect structure information may produce incorrect output, resulting in improper matching results.

However, most previous studies on sentence matching have exploited syntactic information or semantic information alone, without considering both structural information. For example, as illustrated in Figure 1, we show an example of applying the SemBERT (Zhang et al., 2020) on the natural language inference task. The gold label of this example is *entailment*, but SemBERT predicts their relationship is *contradiction*. As the SemBERT only integrates semantic structure information that focuses on the different predicate-argument structures, the deeper meaning contained in each argument is not involved.

We argue that syntactic and semantic structures are complementary cues for sentence matching, and can be combined with the BERT representations to boost the performance of various sentence matching tasks. To this end, we propose to design a framework called syntax- and semantics-aware BERT (SS-BERT), which uses BERT as the backbone and implicitly integrates syntactic and semantic information for sentence matching. In our model, the combination of syntax and semantics can improve the fault tolerance rate of the model to reduce the error propagation problem. Moreover, our method further enhances its syntactic and semantic awareness based on the strong ability of BERT to represent the context and enables SS-BERT to learn deeper meaning representation. On the other hand, we implicitly integrate syntax and semantics at the same time. The advantage of implicit integration is to reduce the sensitivity of the model to the structural information of output, and thus to alleviate the error propagation problem.

In summary, we mainly make the following three contributions in this work: (1) We propose a framework named SS-BERT that integrates both syntactic and semantic information for the sentence matching. To the best of our knowledge, we are the first to integrate both syntax and semantics into BERT for sentence matching; (2) We explore the differences between implicit and explicit integration. Our experiments show that implicit integration achieves better results with BERT. (3) We verify the performance of SS-BERT on three sentence matching tasks, and the results show that SS-BERT has achieved state-of-the-art or competitive performance on four datasets.

2 Related Work

Sentence Matching Early works encode sentences into contextual features individually and feed them to neural networks to predict sentences relationship (Bowman et al., 2015; Tan et al., 2016; Chen et al., 2017b; Conneau et al., 2017; Choi et al., 2018). However, these methods ignore the interactions between sentences, and it is difficult to determine their relationship only by simple contextual features. Recently,

BERT (Devlin et al., 2019) has made significant breakthroughs in sentence matching tasks. It has a stronger ability to learn context features and has been shown to implicitly capture structural information such as syntax and semantics (Jawahar et al., 2019). To further enrich contextual semantics for sentence matching tasks, Zhang et al. (2020) first integrated semantic information into BERT to enrich the sentence contextual semantics. They importing explicit contextual semantic labels information through a pre-trained semantic role labeling model to implicitly integrate into BERT. However, as mentioned above, they ignore dependency structures from dependency parsing, which is critical for the model to compare fine-grained dependency relationship between words within the two sentences for matching. Liu et al. (2019) presented a multi-task learning model named MT-DNN that can obtain a stronger generalized representation through large amounts of cross-task data. Its advantage is that it has a data enhancement effect which is able to reduce overfitting problem. However, MT-DNN requires longer training time. Furthermore, it is time and cost consuming to build large high-quality datasets for multi-task learning.

Structured Information Integration The structured information integration methods of syntax or semantics in the natural language processing tasks could be explicit or implicit. Some works (Gao et al., 2017; Xia et al., 2019; Zhang et al., 2019b; Zhang et al., 2019a) have proved that implicit incorporating syntactic or semantic information into downstream tasks can improve model performance. The core of these methods is to extract the hidden vector of the external parsing model or semantic role labeling model and integrate it with the output of the main network.

3 Syntax- and Semantics-Aware BERT

3.1 Overview

In this section, we describe the details of the SS-BERT. As shown in Figure 2, the SS-BERT has four components: (1) dependency parser; (2) BERT encoder; (3) semantic role labeling; (4) integration layer. In the following sections, we explain each component in detail.

In SS-BERT, words in the input sequence are passed to the dependency parser and semantic role labeling to obtain their hidden representation of sentences respectively. At the same time, the sentences input to BERT encoder and obtain contextual word representations. As the output of BERT is subword level, but dependency parser and semantic role labeling are word level, so we reformulate the word level to the subword level representation on the dependency parser and semantic role labeling[1]. At last, syntax- and semantics-aware word representation can be obtained by the integration layer that concatenated syntax-aware word representation, semantics-aware word representation, and output of BERT.

3.2 Dependency Parser

As biaffine parser is the state-of-the-art dependency parser (Dozat and Manning, 2017), we build our dependency parsing component based on this model, as shown on the left side of Figure 2. The standard biaffine parser can also be considered as an encoder-decoder model, where the encoder part is a three-layer bi-directional LSTM over the input words, and the decoder uses biaffine operations to score all candidate dependency arcs and finds the highest-scoring trees via dynamic programming.

Our dependency parser is built on the biaffine parser with some modifications. First, we reduce the dimension of the encoding layer bi-directional LSTM from 400 to 384 to keep the same as the output dimension of BERT. Note that, we remove Part-of-Speech (PoS) tagging embeddings, and remain word embedding in the embedding part. At last, we obtain the output of the last layer of bi-directional LSTM as a hidden representation. As we use the implicit integration methods, we do not need the decoder part of the dependency parser in SS-BERT.

Specifically, given an input sentence we first obtain its representation as embedding vectors $X_p = \{\boldsymbol{root}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, where n is the sentence length, and the root node is artificially added. Then we pass the embedding vectors to the biaffine parser encoder to obtain hidden representations of dependency parse denoted as P^R:

[1]We also tried transform subword level to word level on dependency parser and semantic role labeling, our experiments show that the former leads to the best results.

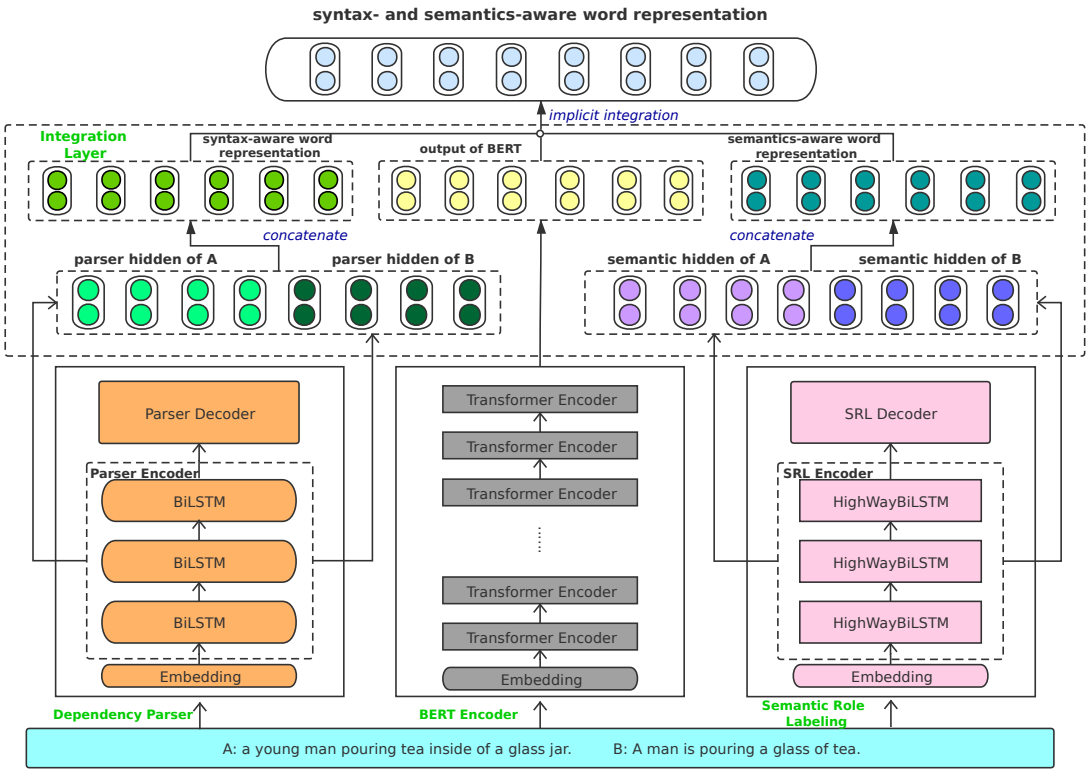

Figure 2: The structure of SS-BERT. From left to right are dependency parser, BERT encoder and semantic role labeling. We input sentences into three components respectively and obtain their output to integrate in the integration layer to obtain syntax- and semantics-aware word representation.

$$P^R = \{\boldsymbol{root}, \boldsymbol{p}_1, ..., \boldsymbol{p}_n\} = \text{Biaffine}(\boldsymbol{root}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_n) \tag{1}$$

3.3 BERT Encoder

In the SS-BERT, the input of the BERT encoder is a pair of sentences. The representation of the sentence pairs follows the default operation of BERT. The input sentences are segmented to subwords (if any) by the BERT word-piece tokenizer. Insert [CLS] at the beginning and [SEP] at the middle and end of the two sentences. For the sake of simplicity, formulations of BERT not be repeated here, please refer to (Devlin et al., 2019) for more details.

3.4 Semantic Role Labeling

We use a span-based mothed (He et al., 2018) as our semantic role labeling model, which is shown on the right side of Figure 2. In the encoder part, this method exploits highway bi-directional LSTM to obtain the deeper features and uses a beam pruning algorithm to obtain several candidate predicates and arguments from all options. The decoder part selects the best one from the score, which is calculated by the predicate, argument, and predicate-argument pairs. We keep the same settings as the original model. The output of the last highway bi-directional LSTM as hidden representations. Specifically, given an input sentence w_1,...,w_n, we obtain its embedding vectors $X_s = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ by embedding layer, where n is the sentence length. Then the hidden representation of semantics can be obtained by the output of semantic role labeling as follow:

$$S = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_n\} = \text{SRL}(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \tag{2}$$

3.5 Integration Layer

In the integration layer, we integrate the output of the dependency parser, BERT encoder, and semantic role labeling to obtain the syntax- and semantics-aware word representation. Given $P_A^R = \{\boldsymbol{root}_A, \boldsymbol{p}_1^a, ..., \boldsymbol{p}_m^a\}$ and $P_B^R = \{\boldsymbol{root}_B, \boldsymbol{p}_1^b, ..., \boldsymbol{p}_n^b\}$ are encoder outputs of dependency parsing model of two sentences, and their length are $m+1$ and $n+1$ respectively. Since each syntactic structure contains artificially added root nodes, the encoder outputs of dependency parsing model needs to be clipped to match the length of the original input sequence as follows:

$$P_A = \{\boldsymbol{p}_1^a, ..., \boldsymbol{p}_m^a\}, P_B = \{\boldsymbol{p}_1^b, ..., \boldsymbol{p}_n^b\} \tag{3}$$

where m and n represent the length after P_A^R and P_B^R are clipped respectively.

By getting the clipped encoder outputs of the dependency parsing model, we concatenate them according to the length. Then we obtain the encoder outputs of dependency parsing model of sentence pairs $P_{pair} \in \mathbb{R}^{d_{m+n} \times d_w^p}$, where d_{m+n} denotes the length of sentence pairs and d_w^p denotes the output size of parsing model.

$$P_{pair} = \{\boldsymbol{p}_1^a, ..., \boldsymbol{p}_m^a\ , \boldsymbol{p}_1^b, ..., \boldsymbol{p}_n^b\} \tag{4}$$

Given $B = \{\boldsymbol{o}_1, ..., \boldsymbol{o}_l\} \in \mathbb{R}^{d_l \times d_w^b}$ $(l \geq m+n)$ is an output of BERT encoder, where d_l denotes the length of sentence pairs in subword level and d_w^b denotes the output size of BERT. Since BERT encode based on subword level, we record the index of words that were segmented into subwords and store them in $H = \{\boldsymbol{h}_1, ..., \boldsymbol{h}_l\} \in \mathbb{R}^{d_l \times d_{m+n}}$. The formula of transform word level to subword level is follows:

$$\boldsymbol{p}_i^{sub} = \boldsymbol{h}_i \boldsymbol{p}_j^{pair} (1 \leq i \leq l; 1 \leq j \leq m+n) \tag{5}$$

where $P_{pair}^{sub} = \{\boldsymbol{p}_1^{sub}, ..., \boldsymbol{p}_l^{sub}\} \in \mathbb{R}^{d_l \times d_w^p}$. We project P_{pair}^{sub} into a sequence of vectors by a feed-forward linear layer to obtain the last syntax-aware word representations $P = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_l\} \in \mathbb{R}^{d_l \times d_w^b}$.

As with syntactic operation, for the sake of simplicity, we given $S_A = \{\boldsymbol{s}_1^a, ..., \boldsymbol{s}_m^a\}$ and $S_B = \{\boldsymbol{s}_1^b, ..., \boldsymbol{s}_n^b\}$ are encoder outputs of semantic role labeling model of two sentences. Firstly, we concatenate them according to the length. The specific operation is shown in formula (4) and obtain $S_{pair} = \{\boldsymbol{s}_1^a, ..., \boldsymbol{s}_m^a\ , \boldsymbol{s}_1^b, ..., \boldsymbol{s}_n^b\} \in \mathbb{R}^{d_{m+n} \times d_w^s}$, where d_w^s denotes the output size of semantic role labeling model.

We also transform word level to the subword level, the operation is the same as formula (5). Finally, we project S_{pair}^{sub} into a sequence of vectors by a feed-forward linear layer to obtain the last semantics-aware word representations $S = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_l\} \in \mathbb{R}^{d_l \times d_w^b}$.

At last, the encoder output of BERT, syntax-aware and semantics-aware word representations are concatenated together. $V = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_l\} \in \mathbb{R}^{d_l \times (d_w^b \times 3)}$ is the representation of the syntax- and semantics-aware word representation, the formula is as follows:

$$\boldsymbol{v}_i = [\boldsymbol{p}_i; \boldsymbol{o}_i; \boldsymbol{s}_i] (1 \leq i \leq l) \tag{6}$$

4 Experiment

4.1 Tasks and Datasets

Natural Language Inference Natural language inference tasks contain two sentences and judge their relationship between sentence pairs. In this task, we experiment on the SNLI (Bowman et al., 2015) and SciTail (Khot et al., 2018) datasets. SNLI dataset contains 570k manual labeled data from an image captioning corpus. We infer the relationship between premise and hypothesis, such as entailment, neutral and contradiction. We use the same data partitioning method as the original paper. SciTail is a textual entailment dataset derived from a science question answering dataset. This dataset contains 27k examples, and each set of data corresponds to only two kinds of relationships that entailment and neutral. In terms of data distribution, 10k data is entailment, and 17k data is neutral. The evaluation metric for SNLI and SciTail is accuracy.

Paraphrase Identification The paraphrase identification task is to determine whether two sentences express the same meaning. In this paper, we use Quora Question Pairs as our experiment dataset. The Quora Question Pairs contains 400k question pairs. We adopt the same data partitioning method as Wang et al. (2017). The evaluation metric is accuracy.

Answer Selection The answer selection is given a question and several candidate answers, we determine whether a candidate sentence is the correct answer to the question. WikiQA (Yang et al., 2015) is a dataset of a retrieval-based question answering based on Wikipedia. The training set has 20.4k, the development set has 2.7k, and the testing set has 6.2k. The evaluation metrics used in this dataset are mean average precision (MAP) and mean reciprocal rank (MRR).

4.2 Implementation Details

Experimental Setting All our experimental results are based on BERT fine-tune, and model parameters are selected on the development set. SS-BERT is based on the BERT of transformers (Wolf et al., 2019) with PyTorch (Paszke et al., 2019) implementation. We follow the original of BERT fine-tune method without making any changes. According to different tasks, our hyper-parameters setting are different. For the optimizer, we use the AdamW in the BERT and set the learning rate in {1e-5, 2e-5, 3e-5, 8e-6}. As for the learning rate decays, we use warm-up 0 or 0.1, and L2 weight decay 0.01 or 1e-8. We set epoch between 3 and 5, and the batch size is selected in {16, 32, 64}. We also set dropout at 0.1 or 0.3. To prevent gradient explosion, we set gradient clipping in {7.5, 10.0, 15.0}.

Biaffine Parser For the dependency parser, we use the biaffine parser proposed by Dozat and Manning (2017). We use original phrase-structure Penn Treebank (PTB) (Marcus et al., 1993) to convert by the Stanford Parser v3.3.0[2] to retrain a parser model. In the end, we achieve 95.43% UAS and 93.07% LAS performance on the PTB development set. Note that the syntax parser is not updated with the framework model.

Semantic Role Labeling We train a semantic role labeling model and use a representative span-based model by He et al. (2018). To ensure the best performance of the model, we do not change the hyper-parameters. We achieve 81.6% F1 on the development set of CoNLL-2005 (Carreras and Màrquez, 2005) benchmarks, and we have reached the same result as the original paper. And the semantic role labeling is not fine-tuned in our framework.

4.3 Results

Results on Natural Language Inference We evaluate the performance of our model on the SNLI dataset and the SciTail dataset. Table 1 shows the performance of SS-BERT results with other state-of-the-art models on SNLI datasets. On the development set, SS-BERT obtains 91.0%, 91.7% and 92.3% performances on $\text{BERT}_{\text{BASE}}$, $\text{BERT}_{\text{LARGE}}$ and BERT_{WWM} respectively. In particular, we have achieved the best development set performance on BERT_{WWM}. On the test set, we have achieved state-of-the-art performances on different versions of SS-BERT. Our results outperform the previous best result base on the $\text{BERT}_{\text{BASE}}$ model by 0.1% and on the $\text{BERT}_{\text{LARGE}}$ and BERT_{WWM}, we achieve 91.6% and 91.9%.

Compared with the multi-task learning model MT-DNN (Liu et al., 2019), our model achieves the best performance with fewer model parameters[3]. On the other hand, MT-DNN also requires longer training time and higher computational cost. Compared with the SemBERT (Zhang et al., 2020), we also use BERT as the backbone, but unlike SemBERT, the training data of semantic role labeling model is different, because it is difficult to obtain, the dataset for training our semantic model does not use CONLL-2012 (Pradhan et al., 2013) like SemBERT, we use a smaller CONLL-2005 (Carreras and Màrquez, 2005), which may cause the decoding quality of our semantic model is not as high as SemBERT. However, significant improvement can be seen through the implicit integration of syntax and semantics into BERT, which also verifies our view that based on syntactic and semantic integration is useful for sentence

[2]https://nlp.stanford.edu/software/lex-parser.html

[3]The parameters of the SS-BERT are 401M, but MT-DNN are 3060M.

Model	Dev(Acc%)	Test(Acc%)
GPT (Radford et al., 2018)	-	89.9
DRCN (Kim et al., 2019)	-	90.1
SemBERT$_{\text{BASE}}$ (Zhang et al., 2020)	91.2	91.0
SemBERT$_{\text{LARGE}}$ (Zhang et al., 2020)	92.3	91.6
MT-DNN$_{\text{BASE}}$ (Liu et al., 2019)	91.5	91.1
MT-DNN$_{\text{LARGE}}$ (Liu et al., 2019)	92.2	91.6
BERT$_{\text{BASE}}$	90.8	90.7
BERT$_{\text{LARGE}}$	91.2	91.0
SS-BERT$_{\text{BASE}}$ (Ours)	91.0	**91.2**
SS-BERT$_{\text{LARGE}}$ (Ours)	91.7	**91.6**
BERT$_{\text{WWM}}$	92.1	91.5
SemBERT$_{\text{WWM}}$ (Zhang et al., 2020)	92.2	91.9
SS-BERT$_{\text{WWM}}$ (Ours)	92.3	**91.9**

Table 1: Results for natural language inference on the SNLI dataset. The full ranking can be obtained from SNLI leaderboard[4].

Model	Dev(Acc%)	Test(Acc%)
RE2 (Yang et al., 2019a)	-	86.0
BigBird†	-	93.8
MT-DNN$_{\text{BASE}}$ (Liu et al., 2019)	95.7	94.1
MT-DNN$_{\text{LARGE}}$ (Liu et al., 2019)	96.3	95.0
BERT$_{\text{BASE}}$	94.6	92.9
BERT$_{\text{LARGE}}$	95.6	93.6
SS-BERT$_{\text{BASE}}$ (Ours)	94.6	**94.2**
SS-BERT$_{\text{LARGE}}$ (Ours)	96.1	**95.0**

Table 2: Results for natural language inference on the SciTail dataset. Results marked by † are from the official SciTail leaderboard[5].

matching tasks. Through experimental comparison, in the case of poor semantic decoding quality, our method can reach the same level as SemBERT.

Table 2 shows the performance of the representative models on the SciTail dataset. SciTail dataset is a binary classification task and relatively small, the variance of the prediction results of the model is relatively large. In the case of a model based on BERT$_{\text{BASE}}$, we achieve the current best performance of 94.2%. In the case of BERT$_{\text{LARGE}}$, SS-BERT has reached the same level as MT-DNN (Liu et al., 2019). Although MT-DNN has more model parameters and a large amount of cross-task training data, which lead to MT-DNN has more advantages in this regard. However, it also shows that our method can make up for the lack of generalization ability of small datasets by giving BERT the ability of syntax-aware and semantics-aware.

Results on Paraphrase Identification Table 3 compares the performance of the different models on the Quora question pairs dataset. Compared with the BERT baseline model, SS-BERT has improved performance. SS-BERT$_{\text{LARGE}}$ reached a very competitive 91.3% as a single model, which reached the same level as DRCN (ensemble). To the best of our knowledge, this is the best performance of paraphrase identification on the Quora question pairs dataset.

[4]https://nlp.stanford.edu/projects/snli/

[5]https://leaderboard.allenai.org/scitail/submissions/public

Model	Dev(Acc%)	Test(Acc%)
RE2 (Yang et al., 2019a)	-	89.2
SAN(Liu et al., 2018)	-	89.4
DRCN (Kim et al., 2019)	-	90.2
DRCN(ensemble) (Kim et al., 2019)	-	91.3
$\text{BERT}_{\text{BASE}}$	91.4	90.8
$\text{BERT}_{\text{LARGE}}$	91.6	91.2
SS-$\text{BERT}_{\text{BASE}}$ (Ours)	91.4	90.9
SS-$\text{BERT}_{\text{LARGE}}$ (Ours)	91.6	**91.3**

Table 3: Results for paraphrase identification on the Quora question pairs dataset.

Model	MAP	MRR
HCRN (Tay et al., 2018b)	0.743	0.756
RE2 (Yang et al., 2019a)	0.745	0.762
Comp-Clip + LM + LC (Yoon et al., 2019)	0.764	0.784
$\text{BERT}_{\text{BASE}}$	0.803	0.813
$\text{BERT}_{\text{LARGE}}$	0.822	0.835
SS-$\text{BERT}_{\text{BASE}}$ (Ours)	**0.834**	**0.848**
SS-$\text{BERT}_{\text{LARGE}}$ (Ours)	**0.855**	**0.867**

Table 4: Results for answer selection on the WikiQA test set.

Results on Answer Selection Results on the WikiQA dataset are listed in Table 4. Without using the BERT (Devlin et al., 2019), Yoon et al. (2019) and Yang et al. (2019a) achieve competitive performance. We achieve the 0.834 MAP on the development set and 0.848 MRR on the test set base on $\text{BERT}_{\text{BASE}}$. In the case of $\text{BERT}_{\text{LARGE}}$, we achieve the best performance of 0.855 MAP and 0.867 MRR on the development and test sets. Compared with the baseline models, our proposed method has significant improvement in performance.

5 Analysis

5.1 Evaluation of Implicit Integration

We explore the influence of different integration methods (i.e. explicit and implicit integration) on model performance. For explicit integration, we input the language structure information into an embedding layer respectively to obtain their embedding representation. The difference is that syntax is encoded using TreeGRU, while semantics is averaging the different perspectives embedding to obtain the final vector representations. For the sake of experiment fairness, we keep the same parameters as the implicit method. The details of our implicit integration method are summarized in section 3. The experiment is completed on the SNLI dataset based on $\text{BERT}_{\text{LARGE}}$ as the framework.

The experimental results are shown in Figure 3. Firstly, we verify the performance of the only explicit integration of syntax or semantics. The implicit integration of semantics has improved by 0.3% based on the baseline, but the effect of explicit integrating semantics is not visible. On the other hand, explicit or implicit integration into the syntax is equivalent and has not brought much improvement. In the same way, the explicit integration of semantics and the implicit integration of syntax have not brought much improvement. However, the implicit integration of semantics and the explicit integration of syntax have significant effects. Finally, we verify the performance of the simultaneous explicit integration of syntax and semantics. We find that it is improved compared to the baseline model. However, it is not as obvious as the implicit integration of syntax and semantics.

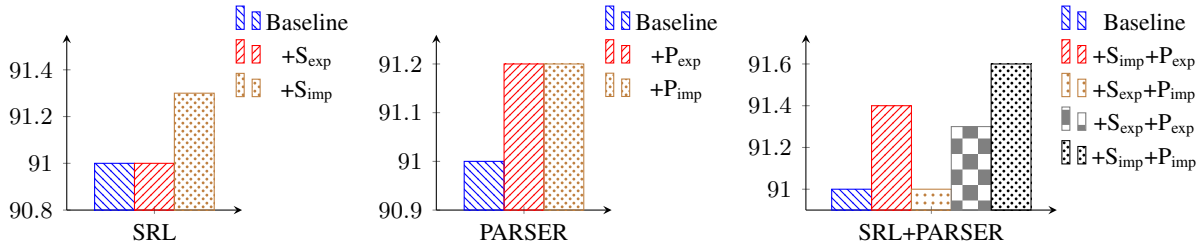

Figure 3: The influence of different feature integration methods on the SNLI test set. The baseline model is $\text{BERT}_{\text{LARGE}}$. *imp* indicates implicit integration, *exp* indicates explicit integration. From left to right, we compare the differences between implicit and explicit integration for semantic information, syntactic information, and the combination of semantic and syntactic information.

Model	**SNLI** (Acc%)	**SciTail** (Acc%)	**Quora** (Acc%)	**WikiQA** (MAP MRR)
$\text{BERT}_{\text{LARGE}}$	91.0	93.6	91.2	0.822 0.835
$\text{BERT}_{\text{LARGE}}$ + SRL	91.3	94.4	91.2	0.845 0.859
$\text{BERT}_{\text{LARGE}}$ + PARSER	91.2	94.3	91.0	0.837 0.850
SS-$\text{BERT}_{\text{LARGE}}$ (Ours)	**91.6**	**95.0**	**91.3**	**0.855 0.867**

Table 5: Ablation study results on the all test set.

Through the above experiments, we find that the effect of implicit integration semantics is better than explicit integration semantics. This shows that BERT is highly sensitive to the quality of the explicit predicate-arguments structure, which often requires us to provide a high-quality external semantic role labeler. Compared with implicit integration, this approach is less controllable and generalization. In syntax, the benefit of implicit integration is not obvious. However, when syntax and semantics are combined, BERT has the most significant ability to be aware of language feathers, especially implicit integration, and the model has the best performance. We analyze this because semantics and syntax are expressions of language from two perspectives and with different granularity.

Moreover, syntax and semantics can improve each other's fault tolerance rate to reduce the risk of error propagation. The above experimental results show that our proposed implicit integration of syntax and semantics is effective for sentence matching tasks.

5.2 Ablation Study

To evaluate the contributions of semantic and syntactic information in our method, we perform an ablation experiment on all datasets based on $\text{BERT}_{\text{LARGE}}$. As shown in Table 5, since SS-BERT is an implicit integration of semantic and syntactic structural information, we want to know whether the single implicit integration of semantics or syntax is useful for the sentence matching tasks. Our method is to implicitly integrate semantic and syntactic information, respectively, without changing the parameters. From the results, we can see that the performance of the model is improved compared with baseline, whether implicitly integrating syntax or semantics, which also indicates the importance of structural information for sentence matching tasks. On the other hand, SS-BERT has a great advantage over single integration. It can not only play the role of syntax and semantics itself but also improve the fault tolerance rate of SS-BERT. In general, SS-BERT is based on the strong contextual expressive ability of BERT and further combines the structural information of syntax and semantics to better inference sentence matching tasks.

6 Conclusions

In this paper, we propose syntax- and semantics-aware BERT(SS-BERT), which implicit integrates syntactic and semantic information for the first time. We have achieved state-of-the-art or competitive per-

formance on four datasets. Experiments show that our method of integrating syntax and semantics into BERT is effective. On the other hand, results indicate that implicit integration has advantages over explicit integration. Since SS-BERT is an end-to-end framework, it is expected to be applied to other natural language tasks in the future.

Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work was supported by National Natural Science Foundation of China (Grant No. 61672211, U1836222, 62076173).

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pages 632–642. Association for Computational Linguistics.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of CoNLL*, pages 152–164. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017a. Enhanced LSTM for natural language inference. In *Proceedings of ACL*, pages 1657–1668. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Recurrent neural network-based sentence encoder with gated attention for natural language inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 36–40. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5094–5101.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of EMNLP*, pages 670–680. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yuze Gao, Yue Zhang, and Tong Xiao. 2017. Implicit syntactic features for target-dependent sentiment analysis. In *Proceedings of IJCNLP*, pages 516–524, Taipei, Taiwan.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of ACL*, pages 364–369. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of ACL*, pages 3651–3657. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6586–6593.

Xiaodong Liu, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for natural language inference. *arXiv preprint arXiv:1804.07888*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of ACL*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of CoNLL*, pages 143–152. Association for Computational Linguistics.

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

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of ACL*, pages 464–473. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018a. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *Proceedings of EMNLP*, pages 1565–1575. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018b. Hermitian co-attention networks for text matching in asymmetrical domains. In *IJCAI*, pages 4425–4431.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Thomas Wolf, L Debut, V Sanh, J Chaumond, C Delangue, A Moi, P Cistac, T Rault, R Louf, M Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv, abs/1910.03771*.

Qingrong Xia, Zhenghua Li, Min Zhang, Meishan Zhang, Guohong Fu, Rui Wang, and Luo Si. 2019. Syntax-aware neural semantic role labeling. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 7305–7313. AAAI Press.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of EMNLP*, pages 2013–2018. Association for Computational Linguistics.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019a. Simple and effective text matching with richer alignment features. In *Proceedings of ACL*, pages 4699–4709, Florence, Italy. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. A compare-aggregate model with latent clustering for answer selection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2093–2096.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019a. Syntax-enhanced neural machine translation with syntax-aware word representations. In *Proceedings of NAACL*, pages 1151–1161. Association for Computational Linguistics.

Meishan Zhang, Peili Liang, and Guohong Fu. 2019b. Enhancing opinion role labeling with semantic-aware word representations from semantic role labeling. In *Proceedings of NAACL*, pages 641–646. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware BERT for language understanding. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.