Phrase-level Self-Attention Networks for Universal Sentence Encoding

Wei Wu[†], **Houfeng Wang**[†‡], **Tianyu Liu**[†] and **Shuming Ma**[†]

[†]MOE Key Lab of Computational Linguistics, Peking University, Beijing, 100871, China
[‡]Collaborative Innovation Center for Language Ability, Xuzhou, Jiangsu, 221009, China
{wu.wei,wanghf,tianyu0421,shumingma}@pku.edu.cn

Abstract

Universal sentence encoding is a hot topic in recent NLP research. Attention mechanism has been an integral part in many sentence encoding models, allowing the models to capture context dependencies regardless of the distance between elements in the sequence. Fully attention-based models have recently attracted enormous interest due to their highly parallelizable computation and significantly less training time. However, the memory consumption of their models grows quadratically with sentence length, and the syntactic information is neglected. To this end, we propose Phrase-level Self-Attention Networks (PSAN) that perform self-attention across words inside a phrase to capture context dependencies at the phrase level, and use the gated memory updating mechanism to refine each word's representation hierarchically with longer-term context dependencies captured in a larger phrase. As a result, the memory consumption can be reduced because the self-attention is performed at the phrase level instead of the sentence level. At the same time, syntactic information can be easily integrated in the model. Experiment results show that PSAN can achieve the state-of-the-art transfer performance across a plethora of NLP tasks including sentence classification, natural language inference and sentence textual similarity.

1 Introduction

Following the success of word embeddings (Bengio et al., 2003; Mikolov et al., 2013), one of NLP's next challenges has become the hunt for universal sentence encoders. The goal is to learn a general-purpose sentence encoding model on a large corpus, which can be readily transferred to other tasks. The learned sentence representations are able to generalize to unseen combination of words, which makes them highly desirable for downstream NLP tasks, especially for those with relatively small datasets.

Previous models for sentence encoding typically rely on Recurrent Neural Networks (RNNs) (Hochreiter and Schmidhuber, 1997; Chung et al., 2014) or Convolutional Neural Networks (CNNs) (Kalchbrenner et al., 2014; dos Santos and Gatti, 2014; Kim, 2014; Mou et al., 2016) to produce context-aware representation. RNNs encode a sentence by reading words in sequential order, they are capable of learning long-term dependencies but are hard to parallelize and not time-efficient. CNNs focus on local or position-invariant dependencies but do not perform well on many tasks (Shen et al., 2017).

Fully attention-based neural networks have attracted wide interest recently, because they can model both dependencies while being more parallelizable and requiring significantly less time to train. Vaswani et al. (2017) proposed the multi-head attention to project a sentence to multiple semantic subspaces, then apply self-attention in each subspace and concatenate the attention results. Shen et al. (2017) proposed the directional self-attention, they apply forward and backward masks to the alignment score matrix to encode temporal order information, and computed attention at feature level to select the features that can best describe the word's meaning in given context. Effective as their models are, the memory required to store the alignment scores of all the token pairs grows quadratically with the sentence length. Furthermore, the syntactic property that is intrinsic to natural language is not considered at all.

Language is inherently tree structured, and the meaning of a sentence comes largely from composing the meanings of subtrees (Chomsky, 1957). Previous syntactic tree-based sentence encoders (Socher et al., 2013; Tai et al., 2015) mainly rely on recursive networks. Although the composition-

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3729–3738
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

ality can be explicitly modeled, their models need expensive recursion computation and are hard to be trained by batched gradient descent methods.

In this paper, we propose the Phrase-level Self-Attention Networks (PSAN), for RNN/CNN-free sentence encoding, it inherits all the advantages of fully attention-based models while requires much less memory consumption. In addition, syntactic information can be incorporated into the model more easily. In our model, every sentence is split into multiple phrases based on parse tree, self-attention is performed at the phrase level instead of the sentence level, thus the memory consumption reduces rapidly as the number of phrases increases. Furthermore, a gated memory component is employed to refine word representations hierarchically by incorporating longer-term context dependencies. As a result, syntactic information can be integrated into the model without expensive recursion computation. At last, multi-dimensional attention is applied on the refined word representations to obtain the final sentence representation.

Following Conneau et al. (2017), we trained our sentence encoder on the SNLI (Bowman et al., 2015) dataset, and evaluate the quality of the obtained universal sentence representations on a wide range of transfer tasks. The SNLI dataset is extremely suitable for training sentence encoders because it is the largest high-quality human-annotated dataset that involves reasoning about the semantic relationships within sentences.

The main contributions of our work can be summarized as follows:

- We propose the Phrase-level Self-Attention mechanism (PSA) for contextualization. The memory consumption can be reduced because self-attention is performed at the phrase level instead of the sentence level.
- A gated memory updating mechanism is proposed to refine each word representation hierarchically by incorporating different levels of contextual information along the parse tree.
- Our proposed PSAN model outperforms the state-of-the-art supervised sentence encoders on a wide range of transfer tasks with significantly less memory consumption.

2 Proposed Model

In this section, we introduce the Phrase-level Self-Attention Networks (PSAN) for sentence encoding. A phrase is a group of words that carry a specific idiomatic meaning and function as a constituent in the syntax of a sentence. Words in a phrase are syntactically and semantically related to each other. Therefore, it can be advantageous to learn a context-aware representation inside a phrase while filtering out information from outside the phrase using self-attention mechanism. In an attempt to better utilize the tree structure which is intrinsic to language, we propose the gated memory updating mechanism to combine different levels of context information. At last, an attention mechanism is utilized to summarize all the token representations into a fixed-length sentence vector.

2.1 Phrase Division

The phrase structure organizes words into nested constituents which can be successively divided into their parts as we move down the constituency-based parse trees. One phrase division shows only one aspect of context dependency. In order to capture different levels of context dependencies, we can split a sentence at different granularities. The number of levels T is a hyper-parameter to be tuned.

We can break down the nodes at T different layers in the parse tree to capture T levels of context dependencies[1], as illustrated in Figure 1.

2.2 Phrase-level Self-Attention

This is the core component of our model. It aims to learn a context-aware representation for each token inside a phrase. In order to filter out information that is semantically or syntactically distant, self-attention is performed at the phrase level instead of the sentence level.

Similar to directional self-attention network (DiSAN) (Shen et al., 2017), Phrase-level Self-Attention uses multi-dimensional attention to compute the alignment score for each dimension of token embedding. Therefore, it can select the features that can best describe a word's specific meaning in any given context.

Given a phrase $P \in \mathbb{R}^{l \times d}$ represented as a sequence of word embeddings $[\boldsymbol{p}_1, \ldots, \boldsymbol{p}_l]$, where l is the length of the phrase and d is the dimension of word embedding representation, we first compute the alignment score for each token pair in the

[1] To avoid the situation that the produced phrases are too small, a phrase will not be further divided if its length is smaller than 4.

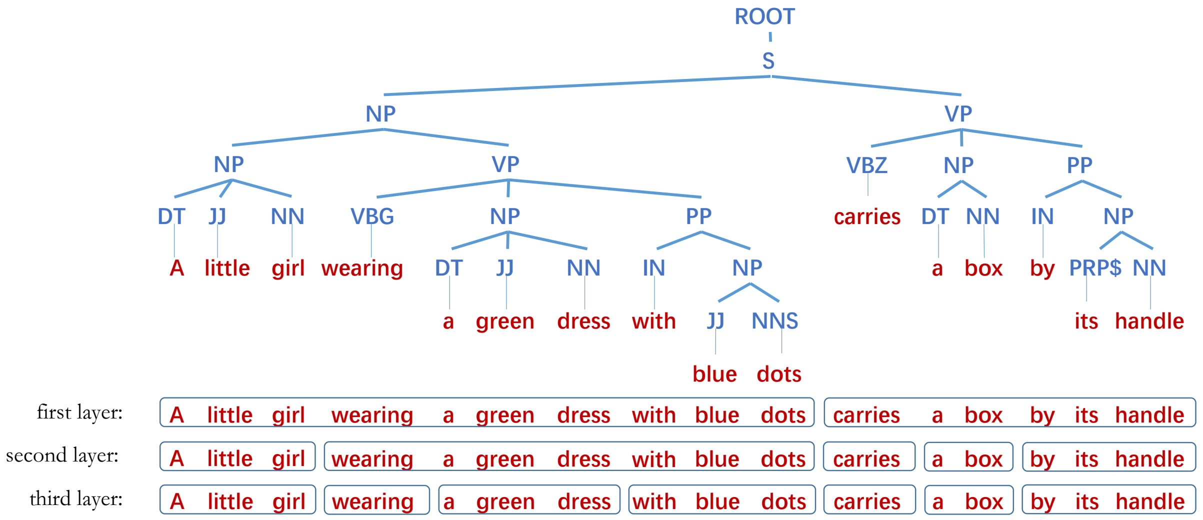


Figure 1: An example of phrase division, the sentence and its parse tree are from the SNLI training data. The division is started from the root of a parse tree. In this example, a phrase will not be further divided if it contains 3 or less words.

phrase:

$$\boldsymbol{a}_{ij} = \sigma\left(W^{a1}\boldsymbol{p}_i + W^{a2}\boldsymbol{p}_j + \boldsymbol{b}^a\right) + M_{ij}$$
$$M_{ij} = \begin{cases} 0, & i \neq j \\ -\infty, & i = j \end{cases} \tag{1}$$

where $\sigma(\cdot)$ is an activation function, $W^{a1}, W^{a2} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}^a \in \mathbb{R}^d$ are parameters to be learned, and M is a diagonal-diabled mask (Hu et al., 2017) that aims to prevent a word from being aligned with itself.

The output of the attention mechanism is a weighted sum of embeddings from all tokens for each token in the phrase:

$$\tilde{\boldsymbol{p}}_i = \sum_{j=1}^{l}\left[\frac{\exp\left(\boldsymbol{a}_{ij}\right)}{\sum_{k=1}^{l}\exp\left(\boldsymbol{a}_{ik}\right)} \odot \boldsymbol{p}_j\right] \tag{2}$$

where $\odot$ means point-wise product. Note that the alignment score for each token pair is a vector rather than a scalar in the multi-dimensional attention.

The final output of Phrase-level Self-Attention is obtained by comparing each input representation with its attention-weighted counterpart. We use a comparison function based on absolute difference and element-wise multiplication which was similar to Wang and Jiang (2016). This comparison function has the advantage of measuring the semantic similarity or relatedness of two sequences.

$$\boldsymbol{c}_i = \sigma\left(W^c\left[\left|\boldsymbol{p}_i - \tilde{\boldsymbol{p}}_i\right|; \boldsymbol{p}_i \odot \tilde{\boldsymbol{p}}_i\right] + \boldsymbol{b}^c\right) \tag{3}$$

where $W^c \in \mathbb{R}^{d\times 2d}$ and $\boldsymbol{b}^a \in \mathbb{R}^d$ are parameters to be learned. $\boldsymbol{c}_i$ is the representation for the i-th word in the phrase that captures local dependencies within the phrase.

At last, we put together the Phrase-level Self-Attention results for non-overlapping phrases from the same phrase division of a sentence. For the t-th phrase division we can get $C^{(t)} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{l_s}]$, the phrase-level self-attention results for the sentence from the t-th layer split, where l_s is the sentence length.

2.3 Gated Memory Updating

Above describes the Phrase-level Self-Attention (PSA) for one split of the parse tree. The parse tree can be split at different granularities. We propose a novel gated memory updating mechanism to refine each word representation hierarchically with longer-term dependencies captured in a larger granularity. Inspired by the idea of adaptive gate in highway networks (Srivastava et al., 2015), our memory mechanism add a gate to original memory networks (Weston et al., 2014; Sukhbaatar et al., 2015). This gate has the ability to determine the importance of the new input and the original memory in the memory updating.

$$C^{(t)} = PSA\left(M^{(t-1)}\right)$$
$$G^{(t)} = sigmoid\left(W^g\left[M^{(t-1)}; C^{(t)}\right] + \boldsymbol{b}^g\right)$$
$$M^{(t)} = G^{(t)} \odot \sigma\left(W^m\left[M^{(t-1)}; C^{(t)}\right] + \boldsymbol{b}^m\right) \tag{4}$$

where $W^g, W^m \in \mathbb{R}^{d\times 2d}$ and $\boldsymbol{b}^g, \boldsymbol{b}^m \in \mathbb{R}^d$ are parameters to be learned. Note that in order to share representation power and to reduce the number of parameters, the parameters of gated memory updating are shared among different layers.

2.4 Sentence Summarization

In this layer, self-attention mechanism is employed to summarize the refined representation of a sentence into a fixed-length vector. The self-attention mechanism can explore the dependencies among tokens within the whole sentence. As a result, global dependencies can also be incorporated in the model.

$$\begin{aligned} \boldsymbol{e}_i &= W^{e2}\sigma\left(W^{e1}\boldsymbol{m}_i^{(T)} + \boldsymbol{b}^{e1}\right) + \boldsymbol{b}^{e2} \\ \boldsymbol{v} &= \sum_{i=1}^{l}\left[\frac{\exp\left(\boldsymbol{e}_i\right)}{\sum_{j=1}^{l}\exp\left(\boldsymbol{e}_j\right)} \odot \boldsymbol{m}_i^{(T)}\right] \end{aligned} \tag{5}$$

where $W^g, W^m \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}^g, \boldsymbol{b}^m \in \mathbb{R}^d$ are parameters to be learned. After this step, the refined context-aware sentence representation is compressed into a fixed-length vector.

3 Experiments

In this section, we conduct a plethora of experiments to study the effectiveness of the PSAN model. Following Conneau et al. (2017), we train our sentence encoder using the SNLI dataset, and evaluate it across a variety of NLP tasks including sentence classification, natural language inference and sentence textual similarity.

3.1 Model Configuration

300-dimensional GloVe (Pennington et al., 2014) word embeddings (Common Crawl, uncased) are used to represent words. Following Parikh et al. (2016), out-of-vocabulary words are hashed to one of 128 random embeddings initialized by uniform distribution between (-0.05, 0.05). All the word embeddings remain fixed during training. Hidden dimension d is set to 300. All other parameters are initialized with Glorot normal initialization (Glorot and Bengio, 2010). Activation function $\sigma(\cdot)$ is ELU (Clevert et al., 2015) if not specified. Minibatch size is set to 16. The number of levels T is fixed to 3 in all of our experiments. The syntactic parse trees of SNLI are provided within the corpus. parse trees for all test corpus are produced by the Stanford PCFG Parser 3.5.2 (Klein and Manning, 2003), the same parser that produced parse trees for the SNLI dataset.

To train the model, Adadelta optimizer (Zeiler, 2012) with a learning rate of 0.75 is used on the SNLI dataset. The dropout (Srivastava et al., 2014) rate and L2 regularization weight decay factor γ are set to 0.5 and 5e-5. To test the model, the SentEval toolkit (Conneau and Kiela, 2018) is used as the evaluation pipeline for fairer comparison.

3.2 Training Setting

Natural language inference (NLI) is a fundamental task in the field of natural language processing that involves reasoning about the semantic relationship between two sentences, which makes it a suitable task to train sentence encoding models.

We conduct experiments on the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015). The dataset has 570k human-annotated sentence pairs, each labeled with one of the following pre-defined relationships: $Entailment$ (the premise entails the hypothesis), $Contradiction$ (they contradict each other) and $Neutral$ (they are irrelevant). Following previous work (Bowman et al., 2015; Mou et al., 2016), we remove the instances which annotators can not reach consensus on. In this way we get 549367/9842/9824 sentence pairs for train/validation/test set.

Following the siamese architecture (Bromley et al., 1993), we apply PSAN to both the premise and the hypothesis with their parameters tied. $\boldsymbol{v}^p$ and $\boldsymbol{v}^h$ are fixed-length vector representations for the premise and the hypothesis respectively. The final sentence-pair representation is formed by concatenating the original vectors with the absolute difference and element-wise multiplication between them:

$$\boldsymbol{v}^{inp} = \left[\boldsymbol{v}^p; \boldsymbol{v}^h; \left|\boldsymbol{v}^p - \boldsymbol{v}^h\right|; \boldsymbol{v}^p \odot \boldsymbol{v}^h\right] \tag{6}$$

At last, we feed the sentence-pair representation $\boldsymbol{v}^{inp}$ into a two layer feed-forward network and use a $softmax$ layer to make the classification. This is the de facto scheme for sentence encoders trained on SNLI. (Mou et al., 2016; Liu et al., 2016; Shen et al., 2017)

3.3 Evaluation Setting

To show the modeling capacity and robustness of our proposed model, we evaluate our model across a wide range of tasks that can be solved purely based on the encoded semantics. The set of tasks

dataset	size	task	output	# phrases / sent.			# words / phrase		
				$1st$	$2nd$	$3rd$	$1st$	$2nd$	$3rd$
MR	10662	sentiment	2	2.00	2.89	6.03	10.79	7.47	3.58
CR	3775	product reviews	2	1.99	3.22	6.02	10.11	6.25	3.34
MPQA	10606	opinion polarity	2	1.13	1.52	1.63	2.73	2.03	1.89
SUBJ	10000	subjectivity	2	1.98	3.29	4.51	5.61	3.40	2.48
SST2	70042	sentiment	2	1.95	3.35	5.03	5.53	3.22	2.15
SST5	11855	sentiment	5	2.00	3.53	6.10	10.08	5.71	3.31
TREC	5952	question type	6	2.00	3.73	5.59	5.03	2.98	1.99
SICK-E	9930	inference	3	1.93	3.40	4.92	5.01	2.85	1.97
SICK-R	9930	inference	[0, 5]	1.93	3.40	4.92	5.01	2.85	1.97
STS14	4500	semantic similarity	[0, 5]	1.96	3.58	5.12	5.34	2.92	2.04
MRPC	5803	paraphrase	2	1.99	3.31	4.55	5.59	3.37	2.65

Table 1: Statistics of the evaluation datasets. If the output is an integer, it represents the number of classes of the classification task. If the output is an interval, it represents the output range of the regression task. **# phrases / sent.** represents the average number of phrases per sentence for each layer of phrase division. **# words / phrase** represents the average number of words per phrase for each layer of phrase division.

Model	dim	$\|\theta\|$	SNLI	Micro	Macro
BiLSTM-Max	4096	40M	84.5	85.2	83.7
AdaSent	4096	36M	83 .4	82.0	80.9
TBCNN	300	3.5M	82.1	81.1	79.3
DiSAN	600	2.4M	85.6	84.7	83.4
PSAN	300	2.0M	**86.1**	**85.7**	**84.5**

Table 2: Performance on SNLI and transfer tasks of various sentence encoders. **dim**: the size of sentence representation. $|\theta|$: the number of parameters. Test accuracies on SNLI, micro and macro averages of accuracies of dev set on transfer tasks are chosen as evaluation metrics.

was selected based on what appears to be the community consensus regarding the appropriate evaluations for universal sentence representations. To facilitate comparison, we use the same sentence evaluation tool as Conneau et al. (2017) to automate evaluation on all the tasks mentioned in this paper.

The transfer tasks used in evaluation can be concluded in the following classes: sentence classification (MR, CR, MPQA, SUBJ, SST2, SST5, TREC), natural language inference (SICK-E, SICK-R), semantic relatedness (STS14) and paraphrase detection (MRPC). Table 1 presents some statistics about the datasets [2].

3.4 Baselines

We compare our model with the following supervised sentence encoders:

- **BiLSTM-Max** (Conneau et al., 2017) is a simple but effective baseline that performs max-pooling over a bi-directional LSTM.

- **AdaSent** (Zhao et al., 2015) forms a hierarchy of representations from words to phrases and then to sentences through recursive gated local composition of adjacent segments.

- **TBCNN** (Mou et al., 2015) is a tree-based CNN model where convolution is applied over the parse tree.

- **DiSAN** (Shen et al., 2017) is composed of a directional self-attention block with temporal order encoded, and a multi-dimensional attention that compresses the sequence into a vector representation.

4 Results and Analysis

4.1 Overall Performance

Experiment results of our model and four baselines are shown in Table 2. Micro and macro accuracies are two composite indicators for evaluating transfer performance of tasks whose metric is classification accuracy. Macro accuracy is the proportion of true results in the population of instances from all tasks. Micro accuracy is the arithmetic mean of dev accuracies for each task.

PSAN achieves the state-of-the-art performance

[2] For further information on the datasets, please refer to Conneau et al. (2017).

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	SICK-R	SICK-E	STS14
BiLSTM-max	79.9	**84.6**	92.1	89.8	83.3	88.7	**75.1/82.3**	0.885	86.3	.68/.65
AdaSent	77.0	82.0	89.9	87.2	82.3	85.6	72.6/80.0	0.855	83.1	.66/.62
TBCNN	75.4	81.6	89.1	85.9	79.4	83.7	72.0/78.6	0.839	82.1	.64/.61
DiSAN	79.7	84.1	**92.2**	89.5	82.9	88.3	75.1/81.8	0.860	85.1	.66/.64
PSAN	**80.0**	84.2	91.9	**89.9**	**83.8**	**89.1**	74.9/82.1	**0.891**	**86.9**	**.69/.67**

Table 3: Transfer test results for our model and various baselines. Classification accuracy is chosen as evaluation metric for datasets including MR, CR, SUBJ, MPQA, SST, TREC and SICK-E; Classification accuracy and F1-score are chosen for MRPC; Pearson correlation is chosen for SICK-R; Pearson and Spearman correlations are chosen for STS-14.

Model	**Acc(%)**
(1) PSA on the first layer only	84.9
(2) PSA on the second layer only	85.3
(3) PSA on the third layer only	84.6
(4) w/o PSA	85.3
(5) w/o syntactic division	85.5
(6) w/o gated memory updating	85.2
(7) w/o both	84.7
(8) Full Model	**86.1**

Table 4: Ablation studies on the SNLI dataset.

with considerably fewer parameters, outperforming a RNN-based model, a CNN-based model, a fully attention-based model and a model that utilize syntactic information. Especially when compared with previous best model *BiLSTM-Max*, PSAN can outperform their model with only 5% of their parameter numbers, demonstrating the effectiveness of our model at extracting semantically important information from a sentence.

In Table 3, we compare our model with baseline sentence encoders in each transfer task. PSAN can consistently outperform the baselines in almost every task considered. On the SICK dataset, which can be seen as an out-domain version of SNLI, our model can outperform the baselines by a large margin, demonstrating the semantic relationship learned on the SNLI can be well transfered to other domains. On the STS14 dataset, where sentence vectors can be more directly measured by the cosine distance, our model can also achieve the state-of-the-art performance, indicating that our learned sentence representations are of high quality.

4.2 Ablation Study

For thorough comparison, we implement seven extra baselines to analyze the improvements contributed by each part of our PSAN model:

- **PSA on the first/second/third layer only** only uses the Phrase-level Self-Attention at the first/second/third layer of phrase division.
- **w/o PSA** applies self-attention at the sentence level and uses the gated memory updating mechanism to refine each token representation hierarchically.
- **w/o syntactic division** divides each sentence equally into small blocks, and applies PSA within each block. The number of blocks equals the number of phrases in that layer.
- **w/o gated memory updating** concatenates the outputs of Phrase-level Self-Attention from three layers of phrase division and feeds the result to a feed-forward layer.
- **w/o both** applies self-attention at the sentence level, and uses sentence summarization to summarize the attention results into a fixed length vector.

The results are listed in Table 4. We can see that (2) performs best among (1), (2) and (3), demonstrating that the second layer split is more expressive, because the number of words per phrase in the second layer is the most suitable. It is neither too small to capture context dependencies, nor too large to filter out irrelevant noise. (8) outperforms (1), (2) and (3), showing that combining phrase-level information from different granularities can further improve performance.

We also experiment on models where the alignment matrix is calculated at the sentence level or at the syntactic-irrelevant block level. (5) performs quite well, showing that hierarchical refinement on smaller units can bring about reasonable

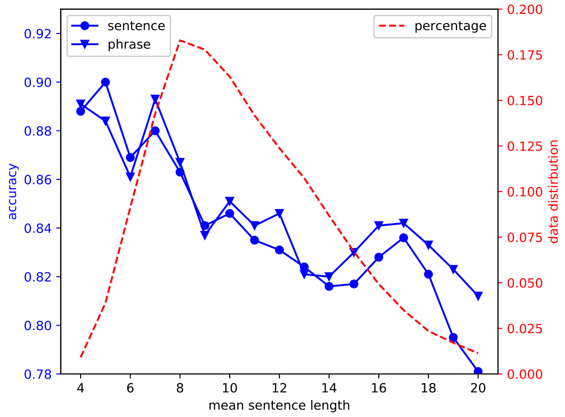


Figure 2: Fine-grained classification accuracies for PSAN and Sentence-level Self-Attention on the SNLI dataset are compared on the left, how data are distributed along sentence length is shown on the right.

performance gain. (8) outperforms (4) and (5), demonstrating syntactic information helps in sentence representation.

When comparing (6) with (8), we can tell that gated memory updating is a better method when used to refine token representation along the parse tree. We assume that memory updating resembles the tree structure of language in that larger phrase is composed in the knowledge of how smaller phrases are composed inside it.

Comparing (7) with (1), (2) and (3), we can find that performing self-attention at the phrase level is generally better than at the sentence level, indicating that reducing attention context into phrase level can effectively filter out words that are syntactically and semantically distant, thus focusing on the interaction with important words. Comparing (7) with (4), we can draw the conclusion that memory updating is effective even when the inputs to each layer are the same.

4.3 Analysis of Sentence Length

Long-term dependencies are typically hard to capture for sequential models like RNNs (Bengio et al., 1994; Hochreiter and Schmidhuber, 1997). We conduct experiments to see how performance changes as the sentence length increases. In Figure 2, we show the relationship between classification accuracy and the average length of sentence pair on the SNLI dataset. Sentence-level Self-Attention (w/o PSA model described in subsection 4.2) is used as a baseline for our model. PSAN outperforms Sentence-level Self-Attention model consistently for longer sentences of length 14 to 20. This demonstrates that incorporating syntactic information by performing self-attention at the phrase level and refining each word's representation hierarchically can help to capture long-term dependencies across words in a sentence.

Model	Memory(MB)	Acc(%)
(1) Multi-head	1508	87.1
(2) DiSAN	2943	87.7
(3) PSAN	1192	**89.1**

Table 5: Memory consumption and test accuracy of three fully attention-based models on the TREC dataset.

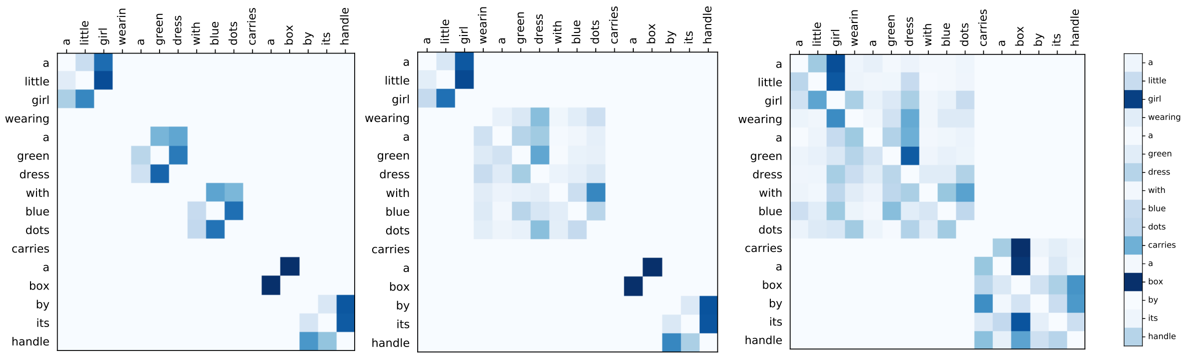
4.4 Analysis of Memory Consumption

We conduct experiments to analyze the memory consumption reduction resulted from Phrase-level Self-Attention. To this end, we re-implement two fully attention-based models (Vaswani et al., 2017; Shen et al., 2017) on the TREC dataset. To make fair comparison, the dimensions of sentence vectors are set to 300, the same number as our model. Table 5 lists the results. Our PSAN model can outperform the other two fully attention-based models, while being more memory efficient. reducing more than 20% of memory consumption.

4.5 Visualization and Case Study

In order to analyze the attention changing process and the importance of each word in the sentence vector, we visualize the attention scores in the alignment matrix of each layer in Phrase-level Self-Attention and sentence summarization layer. To facilitate the visualization of the multi-dimension attention vector, we use the $l2$ norm of the attention vector for representation.

In Figure 3, we can see that, the difference in attention weights between semantically important and unimportant words gets larger as the context becomes larger. This implies that token representation can be gradually refined by the gated memory updating mechanism. Furthermore, the alignment matrix of a phrase can be refined even if the phrase division does not change between layers. For instance, the word "girl" gets larger attention weight in the second layer division than in the first layer. This demonstrates that the memory



(a)

(b)

(c)

(d)

Figure 3: (a) / (b) / (c): attention weights of Phrase-level Self-Attention mechanism in the third / second / first layer phrase division; (d): attention weights of the sentence summarization layer.

updating mechanism can gradually pick out important words for sentence representation. Finally, nouns and verbs dominate the attention weights, while stop words like “a” and “its”, contribute little to the final sentence representation, this indicates that PSAN can effectively pick out semantically important words that are most representative for the meaning of the whole sentence.

5 Related Work

Recently, self-attention mechanism has been successfully applied to the field of sentence encoding, it utilizes the attention mechanism to relate elements at different positions from a single sentence. Due to its direct access to each token representation, both long-term and local dependencies can be modeled flexibly. Liu et al. (2016) leveraged the average-pooled word representation to attend words appear in the sentence itself. Cheng et al. (2016) proposed the LSTMN model for machine reading, an attention vector is produced for each of its hidden states during the recurrent iteration, thus empowering the recurrent network with stronger memorization capability and the ability to discover relations among tokens. Lin et al. (2017) obtained a fixed-size sentence embedding matrix by introducing self-attention. Different from the feature-level attention used in our model, their attention mechanism extracted different aspects of the sentence into multiple vector representations, and utilized a penalization term to encourage the diversity of different attention results.

Syntactic information can be useful for understanding a natural language sentence. Many previous researches utilized syntactic information to build sentence encoder from composing the meanings of subtrees. Tree-LSTM (Tai et al., 2015; Zhu et al., 2015) composed its hidden state from an input vector and the hidden states of arbitrarily many child units. In Tree-based CNN (Mou et al., 2015, 2016), a set of subtree feature detectors slide over the parse tree of a sentence, and a max-pooling layer is utilized to aggregate information along different parts of the tree.

Apart from the models that use parse information, there have been several researches that aimed to learn the hierarchical latent structure of text by recursively composing words into sentence representation. Among them, neural tree indexer (Munkhdalai and Yu, 2017b) utilized LSTM or attentive node composition function to construct full n-ary tree for input text. Gumbel Tree-LSTM (Choi et al., 2018) used Straight-Through Gumbel-Softmax estimator to decide the parent node among candidates dynamically. A major drawback of these models is that the recursion computation can be expensive and hard to be processed in batches.

6 Conclusion

We propose the Phrase-level Self-Attention Networks (PSAN), a fully attention-based model that can utilize syntactic information for universal sentence encoding. By applying self-attention at the phrase level, we can filter out distant and unrelated words and focus on modeling interaction between semantically and syntactically important words, a gated memory updating mechanism is utilized to incorporate different levels of contextual information along the parse tree. Empirical results on a wide range of transfer tasks demonstrate the effectiveness of our model.

Acknowledgments

Our work is supported by National Natural Science Foundation of China under Grant No.61433015 and the National Key Research and Development Program of China under Grant No.2017YFB1002101. The corresponding authors of this paper are Houfeng Wang.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a siamese time delay neural network. In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 737–744.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 551–561.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structure. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, February 2-7, 2018, New Orleans, Louisiana, USA.*

Noem Chomsky. 1957. Syntactic structures. *International Journal of American Linguistics*, 149(3):174196.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *CoRR*, abs/1511.07289.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

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 the 2017 Conference on Empirical Methods in Natural Language Processing*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pages 249–256.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/1705.02798*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional LSTM model and inner-attention. *CoRR*, abs/1605.09090.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2015. Discriminative neural sentence modeling by tree-based convolution. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Tsendsuren Munkhdalai and Hong Yu. 2017a. Neural semantic encoders. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*.

Tsendsuren Munkhdalai and Hong Yu. 2017b. Neural tree indexers for text understanding. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, RepEval@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 41–45.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. Disan: Directional self-attention network for rnn/cnn-free language understanding. *CoRR*, abs/1709.04696.

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 *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *CoRR*, abs/1505.00387.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *CoRR*, abs/1611.01747.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *CoRR*, abs/1410.3916.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 4069–4076.

Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1604–1612.