Inter-Weighted Alignment Network for Sentence Pair Modeling

Gehui Shen Yunlun Yang Zhi-Hong Deng*
Key Laboratory of Machine Perception (Ministry of Education),
School of Electronics Engineering and Computer Science, Peking University,
Beijing 100871, China
{jueliangguke, incomparable-lun, zhdeng}@pku.edu.cn

Abstract

Sentence pair modeling is a crucial problem in the field of natural language processing. In this paper, we propose a model to measure the similarity of a sentence pair focusing on the interaction information. We utilize the word level similarity matrix to discover fine-grained alignment of two sentences. It should be emphasized that each word in a sentence has a different importance from the perspective of semantic composition, so we exploit two novel and efficient strategies to explicitly calculate a weight for each word. Although the proposed model only use a sequential LSTM for sentence modeling without any external resource such as syntactic parser tree and additional lexicon features, experimental results show that our model achieves state-of-the-art performance on three datasets of two tasks.

1 Introduction

Given two pieces of sentences S and T, sentence pair modeling (SPM) is a fundamental task whose applications include question answering (Lin, 2007), natural language inference (Bowman et al., 2015), paraphrase identification (Socher et al., 2011a) and sentence completion (Wan et al., 2016) and so on. In general, each of the two sentences are firstly mapped to a representation, and then a model is designed to determine the relation between them. Traditional methods use lexicon features such as Bag-of-Words(BOW) to map sentences. As we know, features design and selection are time-consuming and high dimensional features may suffer from sparsity because of the variation of linguistic. Recently, deep learning techniques have been applied to develop end-to-end models for NLP tasks, such as sentence modeling (Socher et al., 2011b; Kim, 2014), relation classification (Socher et al., 2012) and machine translation (Sutskever et al., 2014). These works show that deep learning models can be comparable with hand-crafted features based models and often outperform them.

Existing DNN models are based on pre-trained word embeddings which map each word to one low dimensional vector and compose word embeddings to represent sentence. Some models are developed directly from the sentence models. They obtain single vector representation for each sentence separately and then determine the relation based on two vectors (Huang et al., 2013; Qiu and Huang, 2015; Palangi et al., 2016). Because of the absence of interaction, these models can not achieve state-of-the-art performance.

Inspired by attention mechanism in computer vision and machine translation, some elaborate models have been proposed (Rocktäschel et al., 2016; Zhou et al., 2016; Wang and Jiang, 2016) which take interaction information into consideration. Meanwhile, to grasp the fine-grained information for semantic similarity, some prior works (Pang et al., 2016; He and Lin, 2016) firstly compute a word level similarity matrix according to word representation, and utilize multiple convolution layers and extract features from the similarity matrix in a perspective of image recognition.

In this paper, we focus on solving SPM problem by measuring semantic similarity between two sentences. We propose a new deep learning model based on two facts that previous works always neglected. As we know, in the aspect of semantic, each word in the sentence is of different importance. When calculating a sentence representation we should endow each word with a weight indicating its importance. Taking following sentences

*Corresponding author

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1179–1189
Copenhagen, Denmark, September 7–11, 2017.

as an example:

A: a man with a red helmet is riding a motorbike along a roadway.

B: a man is riding a motorbike along a roadway.

C: a man with a red helmet is riding a bicycle along a roadway.

We can see that sentence A is more similar with sentence B than with sentence C while a conventional model probably makes an opposite conclusion because the phrase "with a red helmet" will bias the meaning of A to C meanwhile the difference between "motorbike" and "bicycle" is not large enough. If the model can realize that the phrase "with a red helmet" has little effect on semantic composition, the mistake will be avoided. Since we have to analyse a pair of sentences, the weights should be related to not only the sentence itself, but also its partner. From this point, we propose a novel inter-weighted layer to measure the importance of each word.

On the other hand, the more similar two sentences are, the more probably we can align each word of sentence S with several words of sentence T, and vice versa. On account of the variety of expression, the position and length of two aligned parts are very likely different, so we apply soft-alignment mechanism and build an effective alignment layer.

In summary, our contributions are as follows:

1. We propose an Inter-Weighted Alignment Network (IWAN) for SPM, which builds an alignment layer to compute similarity score according to the degree of alignment.

2. Considering the importance of each word in a sentence is different, we argue that an inter-weighted layer for evaluating the weight of each word is crucial to semantic composition. We propose two strategies for calculating weights. Experimental results demonstrate their effectiveness.

3. Experimental results on semantic relatedness benchmark dataset SICK and two answer selection datasets show that proposed model achieves state-of-the-art performances without any external information.

2 Related Work

2.1 Sentence Models

For sentence modeling, RNN (Elman, 1990; Mikolov et al., 2010) and CNN (Kim, 2014) are both powerful and widely used. RNN models a sentence sequentially by updating the hidden state which represents context recurrently. As sentence length grows, RNN will suffer from gradient vanishing problem. However, gated mechanism, such as Long Short Term Memory(LSTM) (Hochreiter and Schmidhuber, 1997) is introduced to address it. RecNN exploits syntatic information and models sentences under a tree structure. Gated mechanism can also improve the performance of RecNN (Tai et al., 2015). CNN can extract and combine important local context meanwhile model sentences in a hierarchical way (Kim, 2014; Kalchbrenner et al., 2014). All of the above models can be adapted to SPM by modeling two sentences separately.

2.2 Attentive Models

Hermann et al. (2015) firstly introduces attention mechanism into question answering under an RNN architecture. Rocktäschel et al. (2016) applies a similar model to natural language inference which attends over the premise conditioned on the hypothesis. Zhou et al. (2016) combines attention mechanism with tree-structured RecNN encoder. Some prior works (Wang et al., 2016b; Parikh et al., 2016; Wang et al., 2017) compute soft-alignment representation for each word in sentences attentively with word level similarity and then compose the alignment representations to determine the relation. Our model is also under this framework however we focus on explicitly calculating weights for each word to get more reasonable semantic composition.

2.3 Similarity Matrix Based Models

Pang et al. (2016) adopts CNN on word level similarity matrix to extract fine-grained matching patterns from different text granularity. He and Lin (2016) uses a similar architecture with a 19-layer CNN in order to make full use of its power. Yin and Schütze (2015) proposes a hierarchical architecture to model different granularity representation and computes several similarity matrices for interaction.

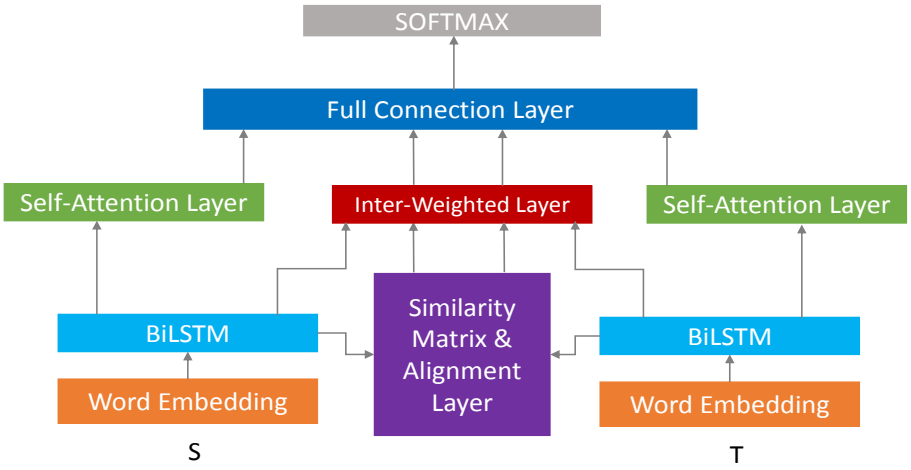

Figure 1: The architecture of IWAN. The blocks with same color have shared parameters.

3 Proposed Model

Given two sentences S and T, we aim to calculate a score to measure their similarity. Figure 1 shows the architecture of IWAN model. To learn representations with context information, we firstly use a bi-direction LSTM sentence model which takes word embeddings as inputs to obtain a context-aware representation for each position (Sec. 3.1). The context-aware representations are used to compute the word level similarity matrix (Sec. 3.2). Inspired by attention mechanism, we exploit soft-alignment to find semantic counterpart in one sentence for each position in the other and compute a weighted sum vector of one sentence as the alignment representation of each position of the other with an alignment layer (Sec. 3.3). Meanwhile, taking the context-aware representation of S and T as inputs, we apply an inter-weighted layer to compute a weight for each position in S and T. We argue that this weight can indicate the importance in semantic interaction and a weighted summation of the representations at each position is more interpretable than other composition method including max or average pooling and LSTM layer. We propose two strategies for computing those weights (Sec. 3.4). The weighted vectors are fed to full connection layers and a softmax layer is used to give the final prediction(Sec. 3.5).

As Figure 1 illustrates, our model is symmetric about S and T. So for simplicity, we only describe the left part of IWAN model which is mainly about modeling S from here. Right part is exactly same except the roles of S and T exchange.

3.1 BiLSTM Layer

With pre-trained d dimension word embedding, we can obtain sentence matrices $S_e = [s_e^1, \ldots, s_e^m]$ and $T_e = [t_e^1, \ldots, t_e^n]$ where $s_e^i \in \mathbb{R}^d$ is embedding of the i-th word in sentence S. m and n are the length of S and T respectively. In order to capture contextual information, we run a bi-direction LSTM (Hochreiter and Schmidhuber, 1997) on two matrices. Let hidden layer dimension of LSTM be u. Given the word embedding x_t at time step t, previous hidden vector h_{t-1} and cell state c_{t-1}, LSTM recurrently computes h_t and c_t as follows:

$$
\begin{aligned}
g_t &= \phi(W_g x_t + V_g h_{t-1} + b_g), \\
i_t &= \sigma(W_i x_t + W_i h_{t-1} + b_i), \\
f_t &= \sigma(W_f x_t + W_f h_{t-1} + b_f), \\
o_t &= \sigma(W_o x_t + W_o h_{t-1} + b_o), \\
c_t &= g_t \odot i_t + c_{t-1} \odot f_t, \\
h_t &= c_t \odot o_t.
\end{aligned}
$$

where all $W \in \mathbb{R}^{u \times d}$, $V \in \mathbb{R}^{u \times u}$ and $b \in \mathbb{R}^u$. σ is *sigmoid* function and ϕ is *tanh* function. $\odot$ indicates the element-wise multiplication of two vectors. The input gates i, forget gates f and output gates o control information flow self-adaptively, moreover cell state c_t can memorize long-distance information. h_t is regarded as the representation of time step t.

We feed S_e and T_e separately into a parameter shared LSTM sentence model. If we run an LSTM model on the sequence of S_e from left to right, we can get the forward hidden vector $S_{fh} = [s_{fh}^1, \ldots, s_{fh}^m]$. For applying bi-direction LSTM, we also run another LSTM backward and get

$S_{bh} = [s^1_{bh}, \ldots, s^m_{bh}]$. Then we concatenate them to one vector representation. So after bi-direction LSTM layer, we obtain $S_h = [s^1_h, \ldots, s^m_h]$ and $T_h = [t^1_h, \ldots, t^n_h]$ where $s^i_h = \begin{bmatrix} s^i_{fh} \\ s^i_{bh} \end{bmatrix}$.

3.2 Word Level Similarity Matrix

As mentioned above, the word level similarity matrix is crucial to making use of the interaction information. Pang et al. (2016) and Wang et al. (2016b) compute the similarity matrix between two word embeddings. We have argued that word embedding can not express the word meaning in context. From the view of RNN, s^i_{fh} contains the most semantic information about i-th word in S and less about the leftmost words, while s^i_{bh} also contains the most semantic information about i-th word in S and less about the rightmost words. Therefore, the hidden vector of BiLSTM keeps the most information of corresponding word as well as integrated with the context information. Computing similarity matrix between BiLSTM hidden vectors is expected to improve the interaction results. We regard the inner dot of two vectors as their similarity. For the similarity matrix M, its element M_{ij} indicates the similarity between s^i_h and t^j_h:

$$M_{ij} = s^{iT}_h \cdot t^j_h.$$

3.3 Alignment Layer

We design the alignment layer for an intuitive idea: more similar S and T are, more probably we can find semantic counterpart in T for each part in S, and vice versa. To some degree, people are likely to find semantic correspondences between two sentences and evaluate their similarity. He and Lin (2016) are also inspired by similar intuition, but they use deep CNN to recognize the alignment patterns implicitly. However, for each sentence, we explicitly calculate the alignment representation and alignment residual which we believe are good indicators of sentence pair similarity.

For calculating the alignment representation, we apply attention mechanism (Bahdanau et al., 2014) to conduct a soft-alignment. The original attention mechanism outputs the alignment weights from an extra full connection layer while we think the inner dot can represent the semantic relatedness adequately. Therefore, we consider the i-th row of M as the similarity between the i-th position of S and each position in T and normalize it as follows:

$$\alpha_{ij} = \frac{exp(M_{ij})}{\sum_{k=1}^{n} exp(M_{ik})}, \quad i = 1, \ldots, m$$

while we also normalize each column of M for T counterpart. α_{ij} always belongs to $[0, 1]$ and can be regarded as weight. Then the alignment representation $S_a = [s^1_a, \ldots, s^m_a]$ is computed as a weighted sum of $\{t^j_h\}$:

$$s^i_a = \sum_{k=1}^{n} \alpha_{ik} t^k_h, \quad i = 1, \ldots, m$$

For T counterpart, the alignment representation is $T_a = [t^1_a, \ldots, t^n_a]$.

In order to measure the gap between the alignment representation and original representation, a *direct* strategy is to compute the absolute value of their difference: $s^i_r = |s^i_h - s^i_a|$. We call $S_r = [s^1_r, \ldots, s^m_r]$ alignment residual which is considered as alignment feature for subsequent processing.

We also utilize an *orthogonal decomposition* strategy which is first proposed by Wang et al. (2016b): the component s^i_p of s^i_h parallel to s^i_a represents the alignment component and component s^i_o orthogonal to s^i_a represents alignment residual. We compute these two component as follows:

$$s^i_p = \frac{s^i_h \cdot s^i_a}{s^i_a \cdot s^i_a} s^i_a, \quad parallel\ component$$

$$s^i_o = s^i_h - s^i_p, \quad orthogonal\ component$$

Then we can replace S_r with S_p and S_o to measure the degree of alignment where $S_p = [s^1_p, \ldots, s^m_p]$ and $S_o = [s^1_o, \ldots, s^m_o]$.

3.4 Inter-Weighted Layer

3.4.1 Inter-Attention Layer

(Lin et al., 2017) firstly proposes a self-attention sentence model which explicitly computes a weight for each word and uses the weighted summation of word representations as sentence embedding. Inspired by this work, we apply a full connection neural network to measure the importance to semantic interaction of every word. We extend the self-attention model to inter-attention layer in order to compute the weights combined with interaction information which composing the alignment representation benefits from. As the

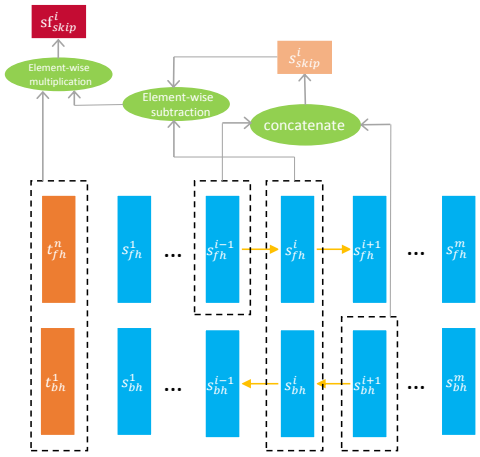

Figure 2: The illustration of computing sf^i_{skip}.

name suggests, these weights of S are not only dependent on S but also T and the parameters of the inter-attention layer are shared for S and T.

Formally, we take S_h and T_h as inputs and the inter-attention layer outputs a vector w_s with size m for S:

$$w_s = softmax(w_2 tanh(W_1 \begin{bmatrix} S_h \\ (t_{avg} \otimes e_m) \end{bmatrix})),$$

where $t_{avg} = \frac{1}{n}\sum_{k=1}^{n} t_h^k$ and $S_h \in \mathbb{R}^{2u \times m}$. We calculate the average of $\{t_h^k\}$ as the representation of T. We also try to replace average operator with a self-attention layer (Lin et al., 2017) but get a worse performance. e_m is a vector of 1s with size m and $\otimes$ represents outer product operator. We feed the concatenated matrix containing pairwise information into a 2-layer neural network. The parameter $W_1 \in \mathbb{R}^{s \times 4u}$ projects inputs into a hidden layer with s units. The output layer is parameterized by a vector w_2 with size s and a $softmax$ operator ensures all the element of output sum up to 1. Then we can use w_s to sum up S_r, S_p and S_o weightedly across the position dimension:

$$s_{wr} = S_r * (w_s)^T,$$
$$s_{wp} = S_p * (w_s)^T,$$
$$s_{wo} = S_o * (w_s)^T.$$

We can get t_{wr}, t_{wp} and t_{wo} in the same way. We call these inter-features for final prediction.

3.4.2 Inter-Skip Layer

We also explore another novel strategy to compute w_s from the intuition that if the i-th word in S has a low contribution to semantic composition, we will obtain a similar representation s^i_{skip} if we feed all word embeddings sequentially except s^i_e into BiLSTM. Unfortunately, the $O(m^2)$ complexity of running BiLSTM model m times is too high so we exploit an approximate method to compute $\{s^i_{skip}\}$:

$$s^i_{skip} = \begin{bmatrix} s^{i-1}_{fh} \\ s^{i+1}_{bh} \end{bmatrix}$$

Then we compute a skip feature as following:

$$sf^i_{skip} = (s^i_{skip} - S^i_h) \odot t_h,$$

where $t_h = \begin{bmatrix} t^n_{fh} \\ t^1_{bh} \end{bmatrix}$ is the BiLSTM hidden representation of T. Figure 2 illustrates how to compute sf^i_{skip}. We think the difference between s^i_{skip} and s^i_h approximately reflects the contribution the i-th word makes to semantic composition. On the one hand, if the difference is small or even close to zero, the importance of correspond word should be small. On the other hand, if the difference (a vector) is not similar to the representation of T, correspond word is probably of less importance in measuring semantic similarity. From these two points, we think $sf_{skip} = [sf^1_{skip}, \ldots, sf^m_{skip}]$ is a good feature to measure word importance. The process of computing w_s is similar:

$$w_s = softmax(w_2 tanh(W_1 sf_{skip})),$$

We can use w_s outputted by inter-skip layer to obtain inter-features in the same way.

3.5 Output Layer

For more rich information, we combine alignment information with sentence embeddings of S and T for final prediction. We run the simple but effective self-attention (Lin et al., 2017) model on S_h to obtain its embedding s_{wh}:

$$s_{wh} = S_h * (softmax(w'_2 tanh(W'_1 * S_h)))^T,$$

where W'_1 and w'_2 are trainable. We compute s_{wh} and t_{wh} with parameter shared self-attention layer which is similar with the inter-attention layer except inputs.

Following Tai et al. (2015), we compute their element-wise product $h_\times = s_{wh} \odot t_{wh}$ and their absolute difference $h_+ = |s_{wh} - t_{wh}|$ as self-features. If we use *direct strategy*, we combine the features as follows:

$$h_{di} = [h^T_\times; h^T_+; s^T_{wr}; t^T_{wr}]^T.$$

Strategy	SICK			TrecQA		WikiQA	
	r	ρ	MSE	MAP	MRR	MAP	MRR
DI	0.8774	0.8229	0.2374	0.815	0.882	0.724	0.739
OD	**0.8810**	**0.8261**	**0.2289**	**0.822**	**0.889**	**0.730**	**0.744**

Table 1: Performances of our model with different strategies in alignment layer on three datasets.

If we use *orthogonal decomposition strategy*, we combine the features as follows:

$$h_{od} = [h_{\times}^T; h_{+}^T; s_{wp}^T; t_{wp}^T; s_{wo}^T; t_{wo}^T]^T.$$

Following previous works, the sentence pair modeling problem can always be considered as a classification task, so we finally calculate a probability distribution with a 2-layer neural network:

$$\hat{p_\theta} = softmax(V_2 ReLU(V_1 h + b_1) + b_2),$$

where h can be h_{di} or h_{od} and the hidden size is l. We use rectified linear units (ReLU) as activation function.

4 Experimental Setup

4.1 Dataset and Evaluation Metric

To evaluate the proposed model, we conduct experiments on two tasks: semantic relatedness and answer selection.

For semantic relatedness task, we use the Sentences Involving Compositional Knowledge (SICK) dataset (Marelli et al., 2014), which consists of 9927 sentence pairs in a 4500/500/4927 train/dev/test split. The sentences are derived from existing image and video description and each sentence pair has a relatedness score $y \in [1, 5]$, where the larger score indicates more similarity between two sentences. As the goal of this task is to calculate sentence pair similarity, we can directly evaluate our model on SICK. Following previous works, we use Pearson's Correlation r, Spearman's Correlation ρ and mean square error (MSE) as evaluation metrics.

For answer selection task, we experiment on two datasets: TrecQA and WikiQA. The TrecQA dataset (Wang et al., 2007) from the Text Retrieval Conferences has been widely used for the answer selection task during the past decade. The original TrecQA train dataset consists of 1,229 questions with 53,417 question-answer pairs, 82 questions with 1,148 pairs in development set, and 100 questions with 1,517 pairs in test set. Recent works (dos Santos et al., 2016; Rao et al., 2016; Wang et al., 2016b) removed questions in development and test set with no answers or with only positive/negative answers, thus there are 65 questions with 1,117 pairs in *Clean version* development set and 68 questions with 1,442 pairs in *Clean version* test set. Rao et al. (2016) has showed the performances on Original TrecQA and *Clean version* TrecQA are not comparable. Therefore, for a fair comparison, we only display the results on *Clean version* TrecQA which are posted on the website of Wiki of the Association for Computational Linguistics[1]. The open domain question-answering WikiQA (Yang et al., 2015) is constructed from real queries of Bing and Wikipedia. We follow Yang et al. (2015) to remove all questions with no correct candidate answers. The excluded WikiQA has 873/126/243 questions and 8627/1130/2351 question-answer pairs for train/dev/test split. To adapt our model to this task, we use semantic similarity to measure the probability of matching between a question and a candidate answer. We evaluate models by mean average precision (MAP) and mean reciprocal rank (MRR).

4.2 Training Details

For experiments on SICK, we follows Tai et al. (2015) to transform the relatedness score y to a sparse target distribution p:

$$p_i = \begin{cases} y - \lfloor y \rfloor, & i = \lfloor y \rfloor + 1 \\ \lfloor y \rfloor + 1 - y, & i = \lfloor y \rfloor \\ 0, & otherwise \end{cases}$$

for $1 \leq i \leq 5$. The training objective is to minimize the KL-divergence loss between p and $\hat{p_\theta}$:

$$loss = \frac{1}{|D|} \sum_{k=1}^{|D|} KL(p^{(k)} \parallel \hat{p_\theta}^{(k)})$$

where $|D|$ is the number of training examples.

We regard the answer selection problem as "yes" or "no" binary classification and the training objective is to minimize the negative log-likelihood in training stage:

$$loss = -\frac{1}{|D|} \sum_{k=1}^{|D|} log \hat{p_\theta}^{(k)}(y^{(k)}|x^{(k)})$$

where $x^{(k)}$ represents a question-answer pair and $y^{(k)}$ indicates whether the candidate answer is cor-

[1] https://www.aclweb.org/aclwiki/index.php?title=Question_Answering_(State_of_the_art)

rect to the question. In test stage, we sort candidate answers for same question in descending order by probability of "yes" category and calculate MAP and MRR.

In all experiments, we use 300-dimension GloVe word embeddings[2] (Pennington et al., 2014) and fix the embeddings during training. The LSTM hidden size u is set to 150. The hidden size of inter-attention and self-attention layer s and full connection network l are both set to 50. The L2 regularization strength is set to 3×10^{-5}. We train the model with Adagrad (Duchi et al., 2011) optimization algorithm with a learning rate of 0.05. The minibatch size is always 25. We exploit early stopping strategy according to MSE on development set for SICK and MAP on development set for TrecQA and WikiQA.

Model	r	ρ	MSE
Meaning Factory (Jiménez et al., 2014)	0.8268	0.7721	0.3224
ECNU (Zhao et al., 2014)	0.8414	-	-
BiLSTM (Tai et al., 2015)	0.8567	0.7966	0.2736
Tree-LSTM (Tai et al., 2015)	0.8676	0.8083	0.2532
MPCNN (He et al., 2015)	0.8686	0.8047	0.2606
PWIM (He and Lin, 2016)	0.8784	0.8199	0.2329
Att Tree-LSTM (Zhou et al., 2016)	0.8730	0.8117	0.2426
Skip-thought+COCO* (Kiros et al., 2015)	0.8655	0.7995	0.2561
MaLSTM*° (Mueller and Thyagarajan, 2016)	0.8822	0.8345	0.2286
IWAN-att (Proposed)	0.8810	0.8261	0.2289
IWAN-skip (Proposed)	**0.8833**	**0.8263**	**0.2236**

Table 2: Test results on SICK. The symbol * indicates the models with pre-training. The symbol ° indicates the models with data augmentation strategy.

Model	MAP	MRR
Wang and Ittycheriah (2015)	0.746	0.820
QA-LSTM (Tan et al., 2015)	0.728	0.832
Att-pooling (dos Santos et al., 2016)	0.753	0.851
LDC (Wang et al., 2016b)	0.771	0.845
MPCNN (He et al., 2015)	0.777	0.836
PWIM (He and Lin, 2016)	0.738	0.827
NCE-CNN (Rao et al., 2016)	0.801	0.877
BiMPM (Wang et al., 2017)	0.802	0.875
IWAN-att (Proposed)	**0.822**	**0.889**
IWAN-skip (Proposed)	0.801	0.861

Table 3: Test results on *Clean version* TrecQA.

4.3 Results

Firstly, we evaluate the effectiveness of two strategies in alignment layer. We use inter-attention model in inter-weighted layer and we find *orthogonal decomposition* (OD) strategy has a superior performance to *direct* (DI) strategy on all datasets. The comparison results are posted in Table 1. In following experiments, we always choose OD strategy in alignment layer.

[2]http://nlp.stanford.edu/projects/glove/

Model	MAP	MRR
NASM (Miao et al., 2016)	0.689	0.707
Att-pooling (dos Santos et al., 2016)	0.689	0.696
LDC (Wang et al., 2016b)	0.706	0.723
MPCNN (He et al., 2015)	0.693	0.709
PWIM (He and Lin, 2016)	0.709	0.723
NCE-CNN (Rao et al., 2016)	0.701	0.718
IARNN° (Wang et al., 2016a)	**0.734**	0.742
BiMPM (Wang et al., 2017)	0.718	0.731
IWAN-att (Proposed)	0.730	0.744
IWAN-skip (Proposed)	**0.733**	**0.750**

Table 4: Test results on WikiQA. The symbol ° indicates the models with data augmentation strategy.

Semantic Relatedness Table 3 shows the performances of our model and compared models on SICK dataset. IWAN-att and IWAN-skip represents our models using inter-attention layer and inter-skip layer respectively. IWAN-skip outperforms IWAN-att in all metrics by a small margin. The traditional feature engineering based models in first group have much poorer performances than deep learning models. MaLSTM (Mueller and Thyagarajan, 2016) benefits from the data argumentation strategy with Wordnet information and pre-training process. Ablation experiments (Mueller and Thyagarajan, 2016) illustrates a 0.04 degradation of Pearson's r without data argumentation strategy. Therefore it is unfair to compare with this model directly, but our models achieve a comparable performance with it. Our models both outperform all other deep learning models. IWAN-skip outperforms Attentive Tree-LSTM (Zhou et al., 2016) by 0.01 in Pearson's r, over 0.01 in Spearman's ρ and almost 0.02 in MSE, although it exploits syntactic parser information. Our model significantly outperforms sentence modeling based models with CNN or RNN which results from the absence of interaction information. He and Lin (2016) proposes Pairwise Word Interaction Model (PWIM) which constructs 19-layer CNN on similarity matrix to capture fine-grained interaction information and shows most competitive. However our model outperforms it in all metrics with much fewer parameters (about 0.65 million versus 1.7 million (He and

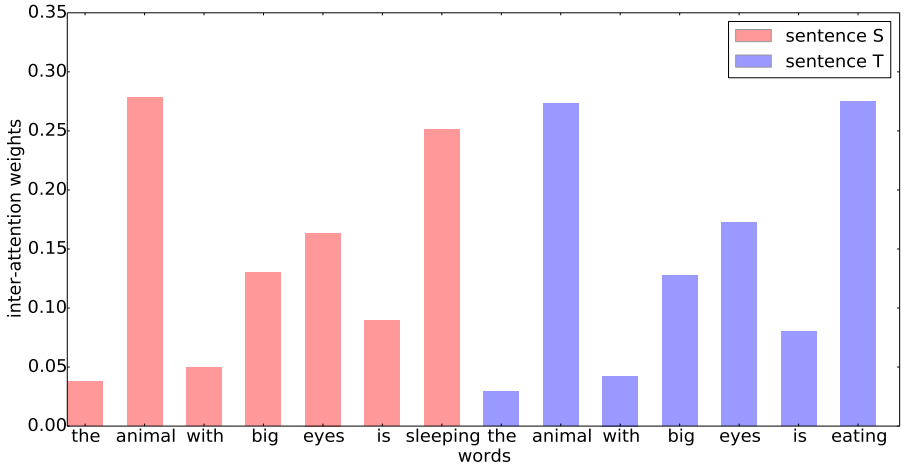

Figure 3: Visualization of weights outputted by inter-weighted layer of words in a sentence pair in SICK test set.

Lin, 2016)).

Answer Selection We compare our model with several state-of-the-art models on *Clean version* TrecQA and WikiQA in Table 3 and Table 4 respectively. Our two models both have a state-of-the-art performance on two datasets. IWAN-att outperform all previous works on TrecQA and make a significant improvement of state-of-the-art. IWAN-skip and IARNN (Wang et al., 2016a) which solves bias problem of attention mechanism beat all other models on WikiQA, while the latter is trained on an argumented dataset with negative sampling. Wang et al. (2016b) first proposes the orthogonal decomposition but their LDC model compute the similarity matrix between word embeddings which are lack of context information and IWAN-att outperforms it dramatically by 0.02-0.05 in MAP and MRR on both datasets. The PWIM (He and Lin, 2016) is still competitive on WikiQA but gets an inferior performance on TrecQA. However, our models both have state-of-the-art performances on three datasets which demonstrates our models have excellent generalization ability in different datasets.

4.4 Ablation Tests

Table 5 show the results of ablations tests on SICK dataset in r metric. From IWAN-att, we remove or replace one component at a time and evaluate performance of partial models. If removing inter-features, the r degrades with a 0.013 decline which proves the interaction information is crucial for sentence pair modeling. Whereas, the degradation from removing self-features is much

Ablation setting	**Pearson's r**
Full Model (IWAN-att)	0.8810
• w/o inter-features	-0.0130
• w/o self-features	-0.0070
• w/o BiLSTM layer	-0.0387
• w/o inter-attention layer	-0.0075
• Replace inter-attention weights with self-attention weights	-0.0063
• w/o parallel component	-0.0037
• w/o orthogonal component	-0.0046

Table 5: Ablation test on SICK dataset, removing each component separately.

smaller. We found a large decline when removing BiLSTM layer, which confirms our conjecture that context information is useful. It is worth mentioning that He and Lin (2016) posts the degradation of their model from removing BiLSTM is 0.1225 in r which is much larger than 0.0387 of us. Removing inter-attention layer means we perform a mean-pooling on inter-features instead of a weighted summation. A 0.0075 r degradation proves importance weighting can result in a significant improvement. If the weights are only about single sentence information, the performance still gets worse. The last two settings show both components from orthogonal decomposition are informative. More or less unexpected, parallel component is almost as useful as orthogonal component.

4.5 Visualization of Inter-Weighted Layer

In order to illustrate the effect of inter-weighted layer in proposed model, we take a sentence pair in SICK test set as an example and display the weights outputted by inter-attention layer of each word in Figure 3. The ground truth of this pair is

3.2 and the prediction given by IWAN-att model is 3.507 which is much more accurate than 4.356 given by the model without inter-attention layer. We can find the inter-attention layer gives very high weights over 0.25 (while the average weight is about 0.14) to "sleeping" and "eating" which are the only difference between two sentences. Therefore, the difference will be attended in following processing. Meanwhile, the weights of the article "the" and the preposition "with" which are not as important as other real words in semantic composition are much lower. These prove the inter-weighted mechanism is reasonable and effective.

5 Conclusion

This work proposes a weighted alignment model for sentence pair modeling. We utilize an alignment layer to measure the similarity of sentence pairs according to their degree of alignment. Moreover, we propose an inter-weighted layer to measure the importance of different parts in sentences. Two strategies for this layer have been explored which are both effective. The composition of alignment features can benefit from the inter-weighted weights. Experiment results shows that proposed models achieve the state-of-the-art performance on three datasets. In the future work, we will improve the inter-weighted layer with more sophisticated module and evaluate our model on other large scale datasets.

Acknowledgments

This work is partially supported by the National High Technology Research and Development Program of China (Grant No. 2015AA015403) and the National Natural Science Foundation of China (Grant No. 61170091). We would also like to thank the anonymous reviewers for their helpful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Hua He, Kevin Gimpel, and Jimmy J. Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1576–1586.

Hua He and Jimmy J. Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 937–948.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

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

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2333–2338.

Sergio Jiménez, George Dueñas, Julia Baquero, and Alexander F. Gelbukh. 2014. UNAL-NLP: combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 732–742.

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.

Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

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.

Jimmy J. Lin. 2007. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Trans. Inf. Syst.*, 25(2).

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *Proceedings of the International Conference on Learning Representations, ICLR 2017*.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 1–8.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1727–1736.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2786–2792.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 24(4):694–707.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2793–2799.

Ankur P. 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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2249–2255.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Xipeng Qiu and Xuanjing Huang. 2015. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1305–1311.

Jinfeng Rao, Hua He, and Jimmy J. Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 1913–1916.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proceedings of the International Conference on Learning Representations, ICLR 2016*.

Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 801–809.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1201–1211.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011b. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh,*

UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 151–161.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108.

Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2835–2841.

Bingning Wang, Kang Liu, and Jun Zhao. 2016a. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the jeopardy model? A quasi-synchronous grammar for QA. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 22–32.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1442–1451.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *CoRR*, abs/1702.03814.

Zhiguo Wang and Abraham Ittycheriah. 2015. Faq-based question answering via word alignment. *CoRR*, abs/1507.02628.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016b. Sentence similarity learning by lexical decomposition and composition. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1340–1349.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2013–2018.

Wenpeng Yin and Hinrich Schütze. 2015. Multigrancnn: An architecture for general matching of text chunks on multiple levels of granularity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 63–73.

Jiang Zhao, Tiantian Zhu, and Man Lan. 2014. ECNU: one stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 271–277.

Yao Zhou, Cong Liu, and Yan Pan. 2016. Modelling sentence pairs with tree-structured attentive encoder. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2912–2922.