Distance-based Self-Attention Network for Natural Language Inference

Jinbae Im and **Sungzoon Cho**
Department of Industrial Engineering
Seoul National University
Seoul, South Korea
jinbae@dm.snu.ac.kr, zoon@snu.ac.kr

Abstract

Attention mechanism has been used as an ancillary means to help RNN or CNN. However, the Transformer (Vaswani et al., 2017) recently recorded the state-of-the-art performance in machine translation with a dramatic reduction in training time by solely using attention. Motivated by the Transformer, Directional Self Attention Network (Shen et al., 2017), a fully attention-based sentence encoder, was proposed. It showed good performance with various data by using forward and backward directional information in a sentence. But in their study, not considered at all was the distance between words, an important feature when learning the local dependency to help understand the context of input text. We propose Distance-based Self-Attention Network, which considers the word distance by using a simple distance mask in order to model the local dependency without losing the ability of modeling global dependency which attention has inherent. Our model shows good performance with NLI data, and it records the new state-of-the-art result with SNLI data. Additionally, we show that our model has a strength in long sentences or documents.

1 Introduction

Sequence modeling has been employing Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN) mostly. More recently, models incorporating attention mechanisms have shown good performance in machine translation (Bahdanau et al., 2014; Sutskever et al., 2014), Natural Language Inference (NLI) (Liu et al., 2016), and Question Answering (QA) (Hermann et al., 2015; Sukhbaatar et al., 2015) etc. Attention mechanisms used to be exploited in conjunction with RNN or CNN as an ancillary means to help improve performance. Lately, Vaswani et al. (2017) presented the first fully attention-based model, which recorded the state-of-the-art result in machine translation. As a fully attention-based model can consider all words in a sentence at once, parallelization leads to great reduction in training time.

Motivated by Vaswani et al. (2017), Shen et al. (2017) proposed the first fully attention-based sentence encoder. Shen et al. (2017) recorded good performance in a variety of tasks. In particular, they recorded the state-of-the-art result with Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015) which is a representative dataset of NLI. The NLI task aims to classify the relationship between two sentences as entailment, contradiction, or neutral. One of the approaches to solving the NLI task is to use sentence-encoding based models.* Shen et al. (2017) presented a sentence-encoding based model reflecting directional information in a sentence. However, the distance between words was not considered at all in their model, and the directional information simply involved words before and after the reference word. Altogether, positional information of words was not fully taken into account. As a result, the difference of importance between the distant words and the nearby words was not appropriately reflected. Hence lo-

*The NLI task can be solved through two different approaches: sentence encoding-based models and joint models. The former separately encode each sentence, whereas the latter take into account the direct relationship between two sentences. Between them, sentence-encoding based models focus on training sentence encoder that can represent sentences in vector form well. We focus on the former approach, since the objective of our work is to develop an advanced sentence-encoding model.

cal dependency was not properly modeled, which in turn failed to capture the context information in long sentences.

To tackle this limitation, we propose Distance-based Self-Attention Network which introduces a distance mask which models the relative distance between words. In conjunction with a directional mask, the distance mask allows us to incorporate complete positional information of words in our model. Our Distance-based Self-Attention Network achieved good performance with NLI data, and recorded the state-of-the-art result with SNLI. Our model worked exceptionally well with long sentences, in particular. We also visualized the effect of the distance mask to show that our model can grasp both local dependency and global dependency.

2 Related Works

NLI tasks have been studied through models of various structures. Most of all, models combining attention with Long Short-Term Memory (LSTM) have performed well. Liu et al. (2016) improved the performance by adding the mean pooling vector to the conventional attention model in which attention is applied to hidden states of LSTM. Chen et al. (2017) used the input gates of the LSTM as attention weights to simplify the model structure. In Chen et al. (2017) and Ni and Bansal (2017), short-cut connections in stacked LSTM, in combination with max-pooling originally suggested by Conneau et al. (2017), were proven effective in improving performance, recording the state-of-the-art performance in MultiNLI. And Munkhdalai and Yu (2016a) used the memory for sentence encoding motivated by Neural Turing Machine (Graves et al., 2014).

Vaswani et al. (2017) was the first study to construct an end-to-end model with attention alone, and recorded the state-of-the-art performance in machine translation tasks. Vaswani et al. (2017)'s encoder-decoder framework consists of a multi-head attention and a position-wise feed forward network as a basic building block which is deeply stacked combined with residual connection. The multi-head attention projects the input sentences to multiple subspaces and then computes the scaled dot-product attention in each subspace. The results in each subspace are then concatenated and projected again. Position-wise feed forward network adds non-linearity to vector representations of each position. In this way, the fully attention-based model was constructed without using RNN or CNN, and the training cost was greatly reduced.

Shen et al. (2017), a very recent work, constructed a fully attention-based sentence encoder motivated by Vaswani et al. (2017). They proposed a multi-dimensional attention mechanism that computes the attention by each dimension through modification of additive attention. In addition, their model exploits directional attention as well as fusion gate motivated by bi-directional LSTM. Directional information was reflected by introducing a simple directional mask. By adding a directional mask to the logit of attention, words in a specific direction in the sentence were masked to avoid attention. The extent to which attention results are ultimately reflected was determined through fusion gate. In our study, we construct our model based on Vaswani et al. (2017)'s basic building block, as well as Shen et al. (2017)'s key model structures. In order to model the distance between words, which was not considered in their works, we transform the multi-head attention in Vaswani et al. (2017), in particular, to fit our objective. Details can be found in section 4.

3 Background

In Vaswani et al. (2017), the attention function is defined as follows by introducing the concept of query, key, and value. "An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key (Vaswani et al., 2017)." The two most commonly used attentions are additive attention (Bahdanau et al., 2014; Shang et al., 2015) and dot-product attention (Kim et al., 2016; Sukhbaatar et al., 2015; Vaswani et al., 2017).

3.1 Additive Attention

Let query, ith key, and ith value be q, k_i, and v_i respectively. ($q \in R^{d_k}$, $k_i \in R^{d_k}$, and $v_i \in R^{d_v}$)

Compatibility function of the query with the ith key is represented by the following equation 1.

$$f(q, k_i) = l_i = u^T \sigma(q + k_i), \tag{1}$$

where $u \in R^{d_k}$, and $\sigma(\cdot)$ is an activation function usually chosen as tanh.

And attention weight assigned to each ith value is computed by applying the softmax function to l_i and final output is weighted sum of value as following equations.

$$w_i = \frac{\exp(l_i)}{\sum_{j=1} \exp(l_j)} \tag{2}$$

$$\text{Output} = \sum_{i=1} w_i v_i \tag{3}$$

3.2 Dot-product Attention

Dot-product attention is the same as additive attention except for compatibility function. In dot-product attention, compatibility function is computed by the following equation 4 in place of the equation 1.

$$f(q, k_i) = l_i = \langle\, q, k_i \rangle \tag{4}$$

On implementation, dot-product attention is much faster and more space-efficient than additive attention due to optimized matrix multiplication.

In practice, however, additive attention outperforms dot product attention for large values of d_k. So Vaswani et al. (2017) used scaled dot-product attention instead of normal dot-product attention to prevent performance loss in large dimension as following equation 5.

$$f(q, k_i) = l_i = \frac{\langle\, q, k_i \rangle}{\sqrt{d_k}} \tag{5}$$

4 Proposed Model

4.1 Overall Architecture

Our model's overall architecture is shown in Figure 1. We follow the conventional architecture for training NLI data. First, the two input sentences, premise and hypothesis, are encoded as vectors, u and v respectively, through identical sentence encoders. For the encoded vectors u and v, the representation of relation between the two vectors is generated by the concatenation of u, v, $|u - v|$, and $u * v$. Thereafter, a probability for each of the 3-class is generated through the 300D ReLU layer and the 3-way softmax output layer. We configured the model with the setting of 1layer 300D as in Shen et al. (2017) to focus on the performance evaluation of the sentence encoder itself. Layer normalization (Ba et al., 2016) and dropout are applied to 300D ReLU layer.

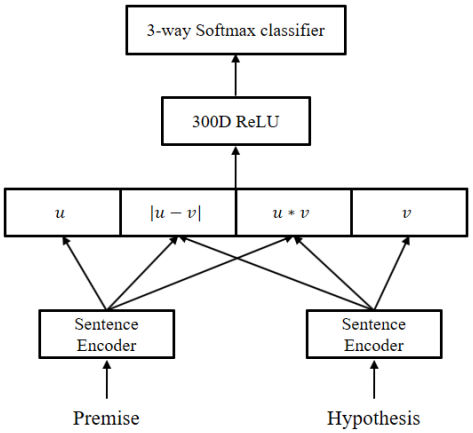


Figure 1: **Overall architecture**

4.2 Sentence Encoder

The sentence encoder structure proposed in this paper is shown in Figure 2. The term "Norm" in Figure 2 stands for layer normalization. The sentence encoder of Figure 2 encodes the premise and hypothesis in a vector form. We describe each component of our sentence encoder in detail in the following subsections.

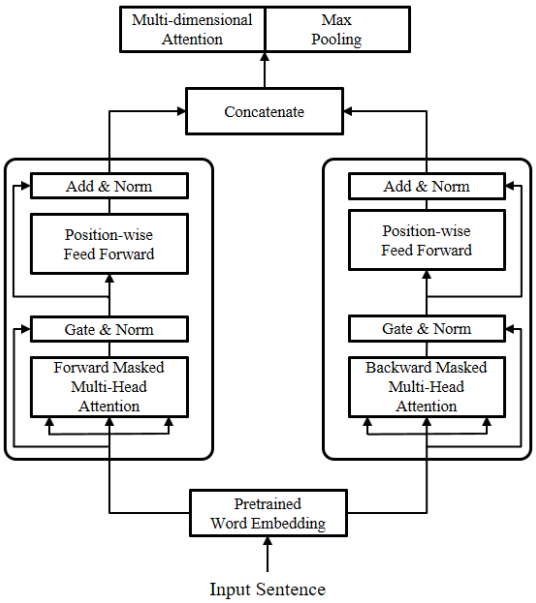


Figure 2: **Sentence encoder**

4.2.1 Word Embedding Layer

Let an input sentence be a sequence of discrete words $\mathbf{x} = [x_1, x_2, \cdots, x_n]$, where $x_i \in R^N$ is a one-hot representation of the word i, and N is the vocabulary size. These one-hot representations are transformed into dense representations by us-

ing the pre-trained word embedding.

Let $W_e \in R^{d_e \times N}$ be a pre-trained word embedding matrix. Then a sequence of dense word representations can be written as $\mathbf{w} = W_e\mathbf{x} = [w_1, w_2, \cdots, w_n]$, where $w_i \in R^{d_e}$ is dense representation of the word i.

4.2.2 Masked Multi-Head Attention

The masked multi-head attention is a variation of the multi-head attention employed by Vaswani et al. (2017). The scaled dot-product attention of Vaswani et al. (2017) is expressed as following:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (6)$$

where Q, K, V are matrices composed of a set of queries, keys, and values, respectively.

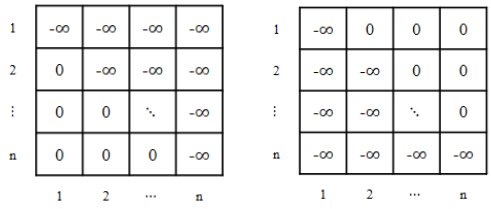
We transform equation 6 and express the masked attention as following:

$$\begin{aligned}&\text{Masked}(Q, K, V) \\ &= \text{softmax}(\frac{QK^T}{\sqrt{d_k}} + M_{dir} + \alpha M_{dis})V\end{aligned} \quad (7)$$

Here, $M_{dir} \in R^{n \times n}$ is the directional mask as proposed in Shen et al. (2017), while $M_{dis} \in R^{n \times n}$ is the distance mask proposed in this model. Hyper parameter α is the distance-alpha tuned through validation data.

M_{dir} consists of the forward mask and backward mask as explained in Figure 3. In the Forward Masked Multi-Head Attention phase, the forward mask is selected, and in the Backward Masked Multi-Head Attention phase, the backward mask. The forward masks prevent words that appear after a given word from being considered in the attention process, while backward masks prevent words that appear before from consideration by adding $-\infty$ to the logits before taking the softmax at the attention phase. The diagonal component of M_{dir} is also set to $-\infty$ so that each token does not consider itself to attention, and the information of each token is later transmitted through the fusion gate of section 4.2.3

M_{dis} is shown in the Figure 4. The (i, j) component of the distance mask is $-|i - j|$, representing the distance between $(i + 1)$th word and $(j + 1)$th word multiplied by -1. By multiplying this value by α and adding it to logit, the attention weight becomes smaller as distance increases.



(a) Forward mask (b) Backward mask

Figure 3: **Directional mask**

That is, the distance mask serves to concentrate on the local words around the reference word. Such a structure may appear similar to a CNN filter extracting a local feature. Yet, the big difference is that CNN only uses information in the window size, whereas our model considers all words in a sentence at once, concentrating on the local words by taking account of the relative distance between words.

By using the distance mask, the distance between words, not considered through the directional mask of Shen et al. (2017), was considered additionally, so the complete positional information of words was taken into consideration.*

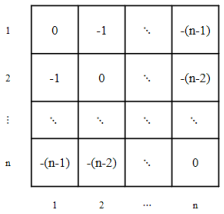


Figure 4: **Distance mask**

The masked multi-head attention can be expressed as following:

$$\begin{aligned}&\text{Masked\_Multi-Head}(Q, K, V) \\ &= \text{concat}(head_1, \cdots, head_h)W^O\end{aligned} \quad (8)$$

where $head_i = \text{Masked}(QW_i^Q, KW_i^K, VW_i^V)$, with h as the number of heads, W_i^Q, W_i^K, $W_i^V \in$

*In Vaswani et al. (2017), the positional information of the word was used through positional encoding. By adding the positional encoding vector to the word embedding vector, the embedding changed according to the absolute position of the word in the sentence. However, in sentence modeling, the relative position with respect to the other words is important, not the absolute position of the word. In other words, what words are placed in order before and after the word is important, not the absolute position of the word in a sentence. Therefore, we take the relative position directly into account in our model through the distance mask instead of the positional encoding which considers the relative position indirectly.

$R^{d_e \times d_e/h}$, and $W^O \in R^{d_e \times d_e}$. $Q, K, V \in R^{n \times d_e}$ are matrices created from n word embedding vectors of sentences and expressed as equation 9.

$$Q = K = V = \begin{bmatrix} - & w_1 & - \\ - & w_2 & - \\ & \vdots & \\ - & w_n & - \end{bmatrix} \quad (9)$$

The masked multi-head attention first projects Q, K, V into h subspaces, respectively, and performs masked attention of equation 7 for each Q, K, V projection combination. The h attention result is concatenated before projection.*

4.2.3 Fusion Gate

At the fusion gate, raw word embedding $S \in R^{n \times d_e}$ and the result of masked multi-head attention $H \in R^{n \times d_e}$ in equation 10 are used as input.

$$S = \begin{bmatrix} - & w_1 & - \\ - & w_2 & - \\ & \vdots & \\ - & w_n & - \end{bmatrix} H = \begin{bmatrix} - & h_1 & - \\ - & h_2 & - \\ & \vdots & \\ - & h_n & - \end{bmatrix} \quad (10)$$

First, we generate S^F, H^F by projecting S, H using $W^S, W^H \in R^{d_e \times d_e}$. Mathematically:

$$\begin{aligned} S^F &= SW^S \\ H^F &= HW^H \end{aligned} \quad (11)$$

Then create gate F as shown in equation 12 where $b^F \in R^{d_e}$.

$$\begin{aligned} \text{Gate}(S, H) &= F \odot S^F + (1 - F) \odot H^F \\ \text{where } F &= \text{sigmoid}(S^F + H^F + b^F) \end{aligned} \quad (12)$$

Finally, we obtain the gated sum by using F. It is common in many papers including Shen et al. (2017) to use raw S and H in gated sum. We, however, use the gated sum of S^F and H^F which resulted in a significant increase in accuracy.

*Multi-head attention (Vaswani et al., 2017) is fast and efficient because it is based on dot-product attention. However, multi-dimensional attention (Shen et al., 2017) has a disadvantage in that it consumes a lot of gpu memory because it requires several 4-dimensional tensors on implementation. So, in our model, the multi-head attention was used as a base structure instead of the multi-dimensional attention. In addition, the performance of the actual implementation was also better with multi-head attention.

4.2.4 Position-wise Feed Forward Networks

We used position-wise feed forward network structure of Vaswani et al. (2017) as it is. The position-wise feed forward network employs the same fully connected network to each position of sentence, in which the fully connected layer consists of two linear transformations, with the ReLU activation in between. Mathematically:

$$\text{FFN}(x) = max(0, xW_1^P + b_1^P)W_2^P + b_2^P \quad (13)$$

where $x \in R^{1 \times d_e}$, $W_1^P \in R^{d_e \times d_{ff}}$, $W_2^P \in R^{d_{ff} \times d_e}$, $b_1^P \in R^{d_{ff}}$, and $b_2^P \in R^{d_e}$.

The FFN function of the above equation 13 is applied to each position of the result of the fusion gate. Note that position-wise feed forward network is combined with the residual connection as shown in Figure 2. That is, FFN learns the residuals. In our model, d_{ff} was set to $4d_e$.

4.2.5 Pooling Layer

The vector representation of input sentence is generated through the pooling layer after the concatenation of the results of forward directional self attention and backward directional self attention. That is, the input of pooling layer is $U = [U^{fw}; U^{bw}] \in R^{n \times 2d_e}$ where each directional self attention output is $U^{fw} \in R^{n \times d_e}$, $U^{bw} \in R^{n \times d_e}$.

We use the multi-dimensional source2token self-attention of Shen et al. (2017) for our multi-dimensional self-attention.

For ith row vector of U, u_i, logit $l(u_i)$ is computed as following:

$$l(u_i) = ELU(u_i W_1^M + b_1^M)W_2^M + b_2^M \quad (14)$$

where $u_i = U_{i*} \in R^{1 \times 2d_e}$, $W_1^M, W_2^M \in R^{2d_e \times 2d_e}$, and $b_1^M, b_2^M \in R^{2d_e}$.

The calculations of logit consist of two linear transformations, with the Exponential Linear Units (ELU) activation function (Clevert et al., 2015) in between. Multi-dimensional attention differs from general attention in that the logit for an input vector is not a scalar but a vector with dimensions equal to the dimensions of the input vector. This allows each dimension of the input vector to have a scalar logit, and we can perform attention to n word tokens in each dimension, as illustrated below by equation 15, 16. Note that softmax is performed on the row dimension of L, not the column dimension.

$$M = \text{softmax}(L) \odot U$$
$$\text{where } L = \begin{bmatrix} - & l(u_1) & - \\ - & l(u_2) & - \\ & \vdots & \\ - & l(u_n) & - \end{bmatrix} \quad (15)$$

$$\text{Multi-dimensional}(U) = \sum_{i=1}^{n} M_{i*} \quad (16)$$

The $2d_e$-dimensional output vector of multi-dimensional attention and the $2d_e$-dimensional vector obtained by applying max pooling to U are concatenated to encode the input sentence as a $4d_e$-dimensional vector.

5 Experiments and Results

5.1 Dataset

The dataset used in the experiments are SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2017) datasets. The SNLI dataset consists of 549,367 / 9,842 / 9,824 (train / valid / test) premise and hypothesis pairs; and the MultiNLI dataset, 392,702 / 9,815 / 9,832 / 9,796 / 9,847 (train / valid_matched / valid_mismatched / test_matched / test_mismatched) sentence pairs. The two datasets have the same format, but sentences in the MultiNLI dataset are much longer than those in SNLI dataset. In addition, MultiNLI dataset consists of various genre information. If genres included in the train data are also found in valid (test) data, then the dataset is called "matched"; if valid (test) data includes genres that are not in the train data, then the dataset is called "mismatched".

5.2 Training Details

We used the Glove 840B 300D[1] ($d_e = 300$) for the pre-trained word embedding without any fine-tuning. This is to train the more universally usable sentence encoder.

Layer normalization (Ba et al., 2016) was applied to all linear projections of masked multi-head attention, fusion gate, and multi-dimensional attention. We applied residual dropout as used in Vaswani et al. (2017), with dropout to the output of masked multi-head attention and $S^F + H^F + b^F$ of fusion gate.

We set $h = 5$, $\alpha = 1.5$ in the masked multi-head attention, and the dropout probability was set to 0.1. Batch size was 64, and the model was trained with Adam (Kingma et al., 2014) optimizer, with a learning rate of 0.001. All models were implemented via Tensorflow on single Nvidia Geforce GTX 1080Ti GPU.

5.3 SNLI Results

Experimental results of SNLI data compared with the existing models on the SNLI leader-board[2] are shown in Table 1. Compared with the existing state-of-the-art model (Shen et al., 2017), the number of parameters and the training time increased, but our results show the new state-of-the-art record. We also looked at the model with distance mask removed to verify the effect of the distance mask proposed in this paper. Results show that the addition of the distance mask improved the performance without significantly affecting the training time or increasing the number of parameters.

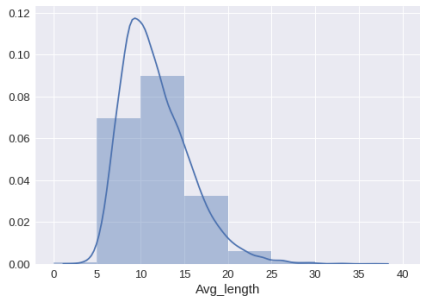


Figure 5: **SNLI average sentence length**

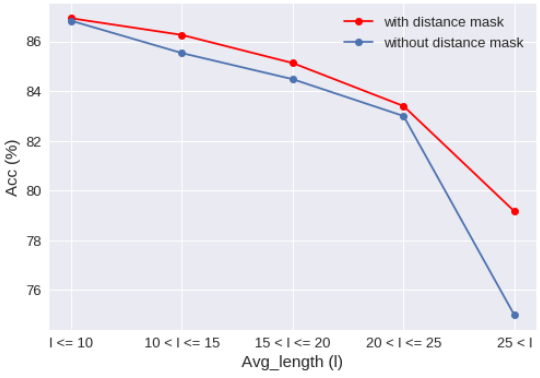


Figure 6: **With distance mask vs. Without distance mask.** Change of test accuracy on SNLI data w.r.t average length of sentence pair.

[1] https://nlp.stanford.edu/projects/glove/

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

Model Name	$\|\theta\|$	T(s)/epoch	Train Acc(%)	Test Acc(%)
Feature-based models				
Unlexicalized features (Bowman et al., 2015)			49.4	50.4
+Unigram and bigram features (Bowman et al., 2015)			99.7	78.2
Sentence encoding-based models				
100D LSTM encoders (Bowman et al., 2015)	220k		84.8	77.6
300D LSTM encoders (Bowman et al., 2016)	3.0m		83.9	80.6
1024D GRU encoders (Vendrov et al., 2015)	15m		98.8	81.4
300D Tree-based CNN encoders (Mou et al., 2015)	3.5m		83.3	82.1
300D SPINN-PI encoders (Bowman et al., 2016)	3.7m		89.2	83.2
600D Bi-LSTM encoders (Liu et al., 2016)	2.0m		86.4	83.3
300D NTI-SLSTM-LSTM encoders (Munkhdalai and Yu, 2016b)	4.0m		82.5	83.4
600D Bi-LSTM encoders+intra-attention (Liu et al., 2016)	2.8m		84.5	84.2
300D NSE encoders (Munkhdalai and Yu, 2016a)	3.0m		86.2	84.6
600D Deep Gated Attn. BiLSTM encoders (Chen et al., 2017)	11.6m		90.5	85.5
600D Directional Self-Attention Network (Shen et al., 2017)	2.4m	587	91.1	85.6
Our self-attention network (without distance mask)	4.7m	687	88.1	86.0
Our Distance-based Self-Attention Network	4.7m	693	89.6	**86.3**

Table 1: **Experimental results of different models on SNLI data.** $|\theta|$: number of parameters (excluding word embedding part). T(s)/epoch : average training time (second) per epoch.

The improvement of the test accuracy by introducing the distance mask is only by 0.3% point, potentially because SNLI data mostly consist of short sentences. Hence, we additionally examined how the effect of the distance mask changes as the average length of the two sentences of premise and hypothesis pair changes. The distribution of the average length of the two sentences of the SNLI test data is shown in Figure 5, and the effect of the distance mask according to the average length change can be seen from Figure 6. Figure 6 shows that the accuracy is similar until the average length is less than 25, yet the test accuracy of the model without the distance mask deteriorates drastically for data of an average length exceeding 25. This demonstrates that the distance mask has an advantage with long sentences or documents.

5.4 MultiNLI Results

The results of applying SNLI best model to MultiNLI dataset without additional parameter tuning are presented in Table 2. Note that matched-test accuracy and mismatched-test accuracy were obtained by submitting our test results to Kaggle open evaluation platforms: MultiNLI Matched Open Evaluation[3] and MultiNLI Mismatched Open Evaluation[4]. First, the average test accuracy difference is greater than 2% when compared to the Directional Self-Attention Network (Shen et al., 2017). This once again confirms our model's advantage in long sentences, given that the sentence is much longer in MultiNLI.

Compared with the result of RepEVAL 2017 (Nangia et al., 2017), we can see that the Distance-based Self-Attention Network performs well. When compared with the model of Chen et al. (2017), our model showed similar average test accuracy with much lower number of parameters. Also, considering that the model of Chen et al. (2017) is a complex LSTM model, our model has an advantage in training time as a fully attention-based model.

Ni and Bansal (2017) showed the best performance with 74.5% accuracy in Matched Test. However, it is a very deep structured LSTM model with 140.2m parameters. In our model, the inference layer is simply composed of 1 layer of 300D in order to focus on the training of sentence encoder. Both in Chen et al. (2017) and Ni and Bansal (2017) models, the inference layer was set very complex in order to improve the MultiNLI accuracy. Taking this into consideration, it can be seen that our Distance-based Self-Attention Network performs competitively given its simpler structure.

5.5 Case Study

A case study was conducted to investigate the role of each structure of the Distance-based Self-Attention Network. For this, a sentence "A lady stands outside of a Mexican market." is picked

[3] https://www.kaggle.com/c/multinli-matched-open-evaluation

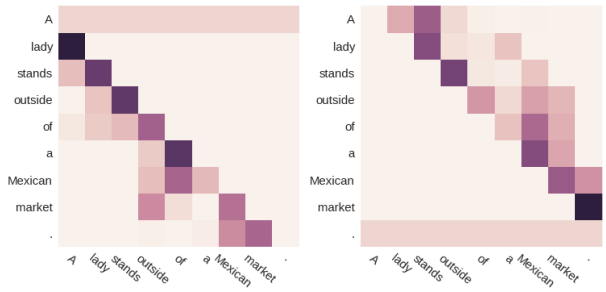
[4] https://www.kaggle.com/c/multinli-mismatched-open-evaluation

Model Name	SNLI Mix	$\|\theta\|$	Matched Test Acc(%)	Mismatched Test Acc(%)
Baseline				
CBOW (Williams et al., 2017)	O		66.2	64.6
BiLSTM (Williams et al., 2017)	O		67.5	67.1
RepEval 2017 (Nangia et al., 2017)				
Cha-level Intra-attention BiLSTM encoders (Yang et al., 2017)	O		67.9	68.2
BiLSTM + enhanced embedding + max pooling (Vu et al., 2017)	X		70.7	70.8
BiLSTM + Inner-attention (Balazs et al., 2017)	O		72.1	72.1
Deep Gated Attn. BiLSTM encoders (Chen et al., 2017)	X	11.6m	73.5	**73.6**
Shortcut-Stacked BiLSTM encoders (Ni and Bansal, 2017)	O	140.2m	**74.5**	73.5
Fully attention-based models				
Directional Self-Attention Network (Shen et al., 2017)	X	2.4m	71.0	71.4
Our Distance-based Self-Attention Network	X	4.7m	74.1	72.9

Table 2: **Experimental results of different models on MultiNLI data.** SNLI Mix : use of SNLI training dataset. $|\theta|$: number of parameters (excluding word embedding part).

among the premise sentences of SNLI test data. We focused on training encoders that can represent each sentence in a vector form well. Therefore, a case study was conducted on a single sentence, not a sentence pair.

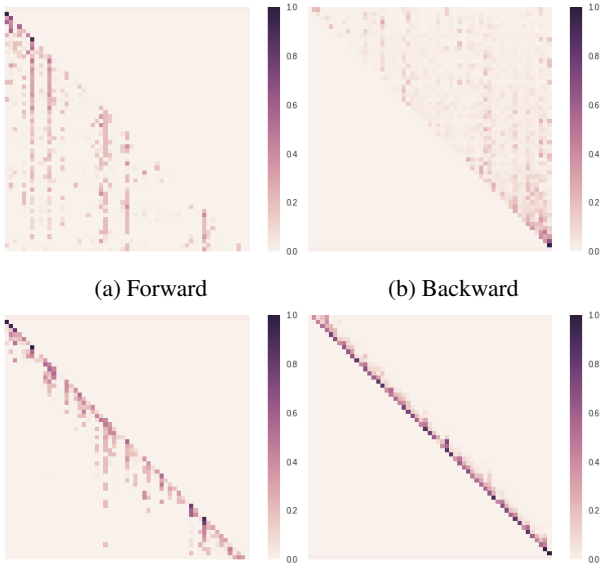
Masked Multi-Head Attention We first look at the attention weights in masked multi-head attention. Attention weights represent a n by n matrix corresponding to softmax$(\frac{QK^T}{\sqrt{d_k}} + M_{dir} + \alpha M_{dis})$ of equation 7, which is different for each head. Here we look at the average attention weights obtained by averaging the attention weights of each head. The attention weights for each head can be found in Appendix.



(a) Forward (b) Backward

Figure 7: **Masked multi-head average attention weights**

The row of the matrix of Figure 7 represents each word of the sentence, and the column represents the attention weights for each word at each row. It can be seen that the attention weights are heavier to the nearby words as compared to those distant from the reference word. At the same time, 'outside' in the forward mask and 'Mexican' in the backward mask have high attention weights for several words. From this, it can be seen that important word is considered in the attention process.



(a) Forward (b) Backward

(c) Forward (d) Backward

Figure 8: **Masked multi-head average attention weights : without/with distance mask.** (a), (b) : without distance mask. (c), (d) : with distance mask

Distance Mask We compared the masked multi-head average attention weights for the longest sentence example in the SNLI test data, with length of 57 words to further verify the effect of the distance mask. Panels (a) and (b) of Figure 8 show results without considering distance, while (c) and (d) show the results with the distance mask. In panels (a) and (b), very distant words are considered in the attention and the overall attention weights were reduced. This implies that each word does not focus on the important words in the attention process, but rather takes into account almost every word, resulting in noisier figures.

However, in panels (c) and (d), the neighboring words are seen more intensively, which im-

plies that the local dependency has been well captured by our model. In addition, as shown in panel (c), even if the word is far apart, it is still considered in the attention process if it is important. This demonstrates the effectiveness of the distance mask to identify local dependencies without losing the ability to grasp the global dependency.

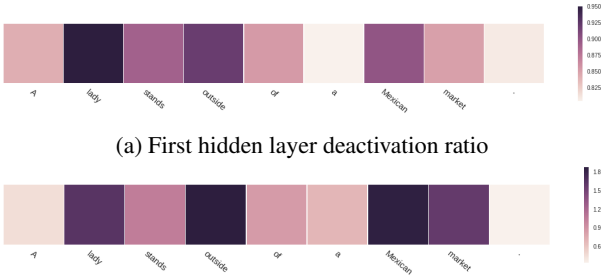
Fusion Gate We visualize the role of the fusion gate $F \in R^{n \times d_e}$ at forward directional self attention. Figure 9 represents the average gate value that averages d_e-dimensional gate value for each word. If look at the results of both extremes, keyword 'Mexican' has a low gate value, resulting in an output that greatly reflects the multi-head attention result. In contrast, 'of', '.', the words of little importance, have large gate values, which indicates that the original word embedding is greatly reflected, not the multi-head attention result.

Figure 9: **Fusion gate (forward)**

Position-wise FFN For the FFN function of equation 13, Figure 10(a) represents the deactivation ratio in the first hidden layer of position-wise ffn.

As shown in Figure 2, position-wise ffn is used in conjunction with a residual connection. That is, the final output of position-wise ffn for input x is the d_e-dimensional vector of LayerNorm$(x + \text{FFN}(x))$. Figure 10(b) visualizes the maximum value of this final output vector.

In Figure 10, keywords with a high deactivation ratio is shown in panel (a) and a high final max value in panel (b). In case of a word corresponding to a keyword, deactivation occurs frequently in (a), and residual learning is hardly achieved in the position-wise ffn, so that the output of the fusion gate is almost maintained. On the other hand, in case of non-important words, residual learning is performed in position-wise ffn because there is less deactivation in (a), so that the max value of final output becomes smaller in (b). This results in preventing non-important words from consideration in the subsequent pooling layer. In summary, position-wise ffn plays a key role in ensuring that non-critical words are paid less attention to in pooling layers.



(a) First hidden layer deactivation ratio

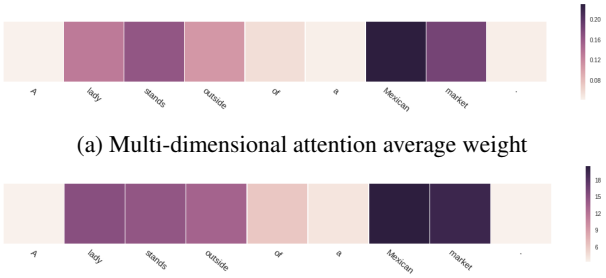
(b) Final output max value (+residual connection)

Figure 10: **Position-wise ffn (forward)**

Pooling Layer For the multi-dimensional attention corresponding to Figure 11(a), we visualized the attention weights averaged for each word, where attention weights correspond to softmax$(L) \in R^{n \times 2d_e}$ in equation 15.

In max pooling, the max value is selected for each column of $U \in R^{n \times 2d_e}$. Thus, in Figure 11(b), we visualize the percentage at which each word is selected in the max pooling operation for the $2d_e$ dimension.

It can be seen that panels (a) and (b) of Figure 11 are similar on the whole. In other words, both multi-dimensional attention and max pooling utilize information about key words intensively. A similar result can be expected by using only one of the pooling layers. However, experiment results show that using both multi-dimensional attention and max pooling layer gives better performance.



(a) Multi-dimensional attention average weight

(b) Max pooling ratio (%)

Figure 11: **Pooling layer**

6 Conclusion

In this paper, we propose the Distance-based Self-Attention Network reflecting the distance between words. By reflecting the word distance information, our model learns the local dependency without losing the ability to capture the global dependency. This was achieved through a simple distance mask, so that the performance of the

NLI task could be improved while maintaining the number of parameters and training time. In particular, we recorded the new state-of-the-art performance for SNLI data. The introduction of the distance mask improves the performance with longer sentences.

As the research on universal sentence encoders using NLI data was proposed by Conneau et al. (2017), we plan to carry out research on fully attention-based networks for universal sentence embedding as future work. We will also study the fully attention-based network in image data and speech data. Especially, regarding image data, capsule network (Sabour et al., 2017) recently proposed, and as research on new structure to replace CNN is going on, our future work will move in similar directions.

Acknowledgments

We would like to thank Hyejin Lee, Hyunjoong Kim, Taewook Kim, Jinwon An, Inbeom Park, Minki Chung, and many others in SNUDM center, for critical feedback and discussions.

This work was supported by the BK21 Plus Program(Center for Sustainable and Innovative Industrial Systems, Department of Industrial Engineering & Institute for Industrial Systems Innovation, Seoul National University) funded by the Ministry of Education, Korea (No. 21A20130012638), the National Research Foundation (NRF) grant funded by the Korea government (MSIP) (No. 2011- 0030814), and the Institute for Industrial Systems Innovation of SNU.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. In *NIPS 2016 Deep Learning Symposium*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR 2015*.

Jorge A. Balazs, Edison Marrese-Taylor, Pablo Loyola, and Yutaka Matsuo. 2017. Refining raw sentence representations for textual entailment recognition via attention. In *Proceedings of RepEval 2017: The Second Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

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 2015*, pages 632–642. Association for Computational Linguistics.

Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of ACL 2016*, pages 1466–1477. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Recurrent neural network-based sentence encoder with gated attention for natural language inference. In *Proceedings of RepEval 2017: The Second Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

Djork-Arn Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). In *Proceedings of ICLR 2016*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of EMNLP 2017*, pages 681–691. Association for Computational Linguistics.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Karl Moritz Hermann, Tom Koisk, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of NIPS 2015*, pages 1693–1701.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of AAAI 2016*, pages 2741–2749.

Kingma, Diederik, and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of ICLR 2015*.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of ACL 2016*, page 130. Association for Computational Linguistics.

Tsendsuren Munkhdalai and Hong Yu. 2016a. Neural semantic encoders. In *Proceedings of EACL 2017*, pages 397–407. Association for Computational Linguistics.

Tsendsuren Munkhdalai and Hong Yu. 2016b. Neural tree indexers for text understanding. In *Proceedings of EACL 2017*, pages 11–21. Association for Computational Linguistics.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R. Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. In *Proceedings of RepEval 2017: The Second Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

Yixin Ni and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of RepEval 2017: The Second Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of ACL 2015*, pages 1577–1586. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1709.04696*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Proceedings of NIPS 2015*, pages 2440–2448.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS 2014*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2015. Order-embeddings of images and language. In *Proceedings of ICLR 2016*.

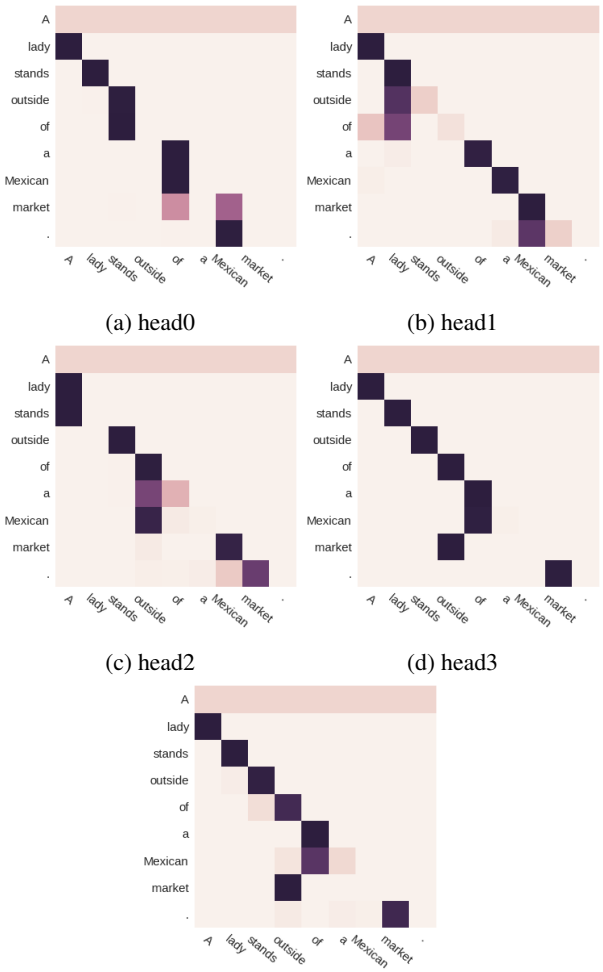
Hoa Trong Vu, Thuong-Hai Pham, Xiaoyu Bai Marc Tanti, Lonneke van der Plas, and Albert Gatt. 2017. Lct-maltas submission to repeval 2017 shared task. In *Proceedings of RepEval 2017: The Second Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Han Yang, Marta R, Costa-juss, and Jos A. R. Fonollosa. 2017. Character-level intra attention network for natural language inference. In *Proceedings of RepEval 2017: The Second Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

Appendix

Masked Multi-Head Attention The attention weights for each head in the masked multi-head attention are shown in Figures 12 and 13. Figure 12 shows the result of using a forward directional mask, and Figure 13 is the result of using a backward directional mask. It can be seen that the attention weights are different for each head. This allows our model to capture various dependencies between words in a sentence.

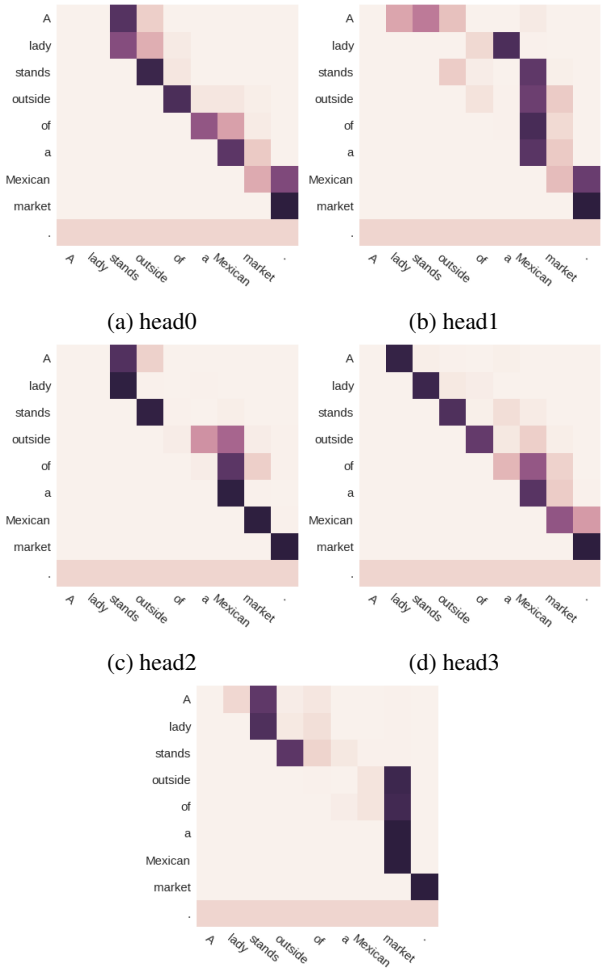


(a) head0 (b) head1

(c) head2 (d) head3

(e) head4

Figure 12: **Forward masked multi-head attention weights**



(a) head0 (b) head1

(c) head2 (d) head3

(e) head4

Figure 13: **Backward masked multi-head attention weights**