Expert Systems With Applications 147 (2020) 113214

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Multiple premises entailment recognition based on attention and gate mechanism

Pin Wu[a,*], Zhidan Lei[a], Quan Zhou[a], Rukang Zhu[a], Xuting Chang[a], Junwu Sun[a], Wenjie Zhang[a], Yike Guo[a,b]

[a] Shanghai University, Shanghai, China
[b] Imperial College London, London, UK

ARTICLE INFO

Article history:
Received 20 March 2019
Revised 3 August 2019
Accepted 16 January 2020
Available online 16 January 2020

Keywords:
Natural language inference
Multiple premise entailment
Attention mechanism
Gate mechanism
Fine-tune

ABSTRACT

Multi-premise natural language inference provides important technical support for automatic question answering, machine reading comprehension and other application fields. Existing approaches for Multiple Premises Entailment (MPE) task are to convert MPE data into Single Premise Entailment (SPE) data format, then MPE is handled in the same way as SPE. This process ignores the unique characteristics of multi-premise, which will result in loss of semantics. This paper proposes a mechanism based on Attention and Gate Fusion Network (AGNet). AGNet adopts a "Local Matching-Integration" strategy to consider the characteristics of multi-premise. In this process, an attention mechanism combined with a matching gate mechanism can fully describe the relationship between the premise and hypothesis. A self-attention mechanism and a fusion gate mechanism can deeply exploit the relationship from the multi-premise. In order to avoid over-fitting problem, we propose a pre-training method for our model. In terms of computational complexity, AGNet has good parallelism, reduces the time complexity to $O(1)$ in the process of matching. The experiments show that our model has achieved new state-of-the-art results on MPE test set.

1. Introduction

Natural Language Inference (NLI) (Dagan & Glickman, 2004) is a fundamental research of natural language processing, aims to determine whether a natural language hypothesis H can be inferred from a premise P, their relationship will be one of these: (i) contradiction, (ii) neutral, or (iii) entailment.

After the Stanford Natural Language Inference (SNLI) corpus of 570K English sentence pairs (Bowman, Angeli, Potts, & Manning, 2015) was proposed, a variety of neural network models were allowed to perform on NLI. These models can be divided into two frameworks according to whether the sentence is encoded to a single vector or not: one is called "Encoding-Matching" framework (Liu, Sun, Lin, & Wang, 2016; Mou et al., 2015), focus on constructing sentence representation, mapping sentences of different lengths into a semantic space, calculating their cosine similarity to obtain the final classification. The second is Matching-Aggregation framework (Ghaeini et al., 2018; Parikh, Täckström, Das, & Uszkoreit, 2016; Wang & Jiang, 2016), it focuses on matching two sentences at the granularity of word, and then aggregating the matching results to obtain the final classification.

However, the above model framework is limited to deal with standard NLI problem, that is, the classification of one premise and one hypothesis, in this paper we call it Single Premise Entailment (SPE). But in the real world, an event will have multiple descriptions, which may come from multiple perspectives or multiple forms of expression. For example, multiple news articles describing a same event, social media posts by different individuals about a same event, or multiple witnesses report a same crime. In these cases, we hope to judge the authenticity of another statement (hypothesis) in multiple independent statements (premises), so the Multiple Premises Entailment (MPE) task is proposed.

As shown in Fig. 1, we summarize the MPE data has the following three characteristics: (i) Each premise is a complete sentence with complete semantics. This means that one premise sentence of SPE data cannot be divide into multiple parts to construct the multiple premise sentences. (ii) There is no order relationship among these premises. (iii) The information contained in each premise is asymmetrical. That is, all premises constitute a semantic scene, each sentence's description about the scene may be one-sided

* Corresponding author.
E-mail addresses: wupin@shu.edu.cn (P. Wu), leizd@outlook.com (Z. Lei), jzhou8763@shu.edu.com (Q. Zhou), zhurukang8763@shu.edu.com (R. Zhu), changxuting8763@shu.edu.com (X. Chang), shu2017sjw8763@shu.edu.com (J. Sun), wendy_zhang8763@shu.edu.com (W. Zhang), y.guo@imperial.ac.uk (Y. Guo).

https://doi.org/10.1016/j.eswa.2020.113214
0957-4174/

Premises:
1. Two girls sitting down and looking at a book.
2. A couple laughs together as they read a book on a train.
3. Two travelers on a train or bus reading a book together.
4. A woman wearing glasses and a brown beanie next to a girl with long brown hair holding a book.

Hypothesis:
Women smiling. ENTAILMENT

Premises:
1. A group of individuals performed in front of a seated crowd.
2. Woman standing in front of group with black folders in hand.
3. A group of women with black binders stand in front of a group of people.
4. A group of people are standing at the front of the room, preparing to sing.

Hypothesis:
A group having a meeting. CONTRADICTION

Premises:
1. Three men are working construction on top of a building.
2. Three male construction workers on a roof working in the sun.
3. One man is shirtless while the other two men work on construction.
4. Two construction workers working on infrastructure, while one worker takes a break.

Hypothesis:
A man smoking a cigarette. NEUTRAL

Fig. 1. The Multiple Premises Entailment Task.

information but not mutually exclusive, and one premise's description may be a subset of another premise. This means that each premise has a different level of contribution for the final classification.

Neural network models for MPE problem can also be divided into two strategies according to the process method of multiple-premise (Lai, Bisk, & Hockenmaier, 2017): one we call it "Concatenate-Matching", is to concatenate multiple premises into one premise, thus transforming MPE problem into SPE problem to process. The other is "Local Matching-Integration", it is to match each premise with the hypothesis separately to get multiple local classification results, and merge them together to get the final classification. The former attempts to solve the MPE with a large number of existing SPE process methods, but because of the first and second characteristics of multi-premise data, the long sentence composed of several premise sentences has complex semantics and syntax, is difficult to encode to get precise semantics. In the latter case, because of the third characteristic of multi-premise data, this strategy will fall into the trap of multiple local judgment errors, that is, every premise contains only partial information, so every result may be wrong, if blindly merge these results to classify, we will get a wrong final result.

This paper explores the second strategy and proposes AGNet to solve the MPE problem in combination with the "Matching-Aggregation" framework. For these problems mentioned above, we provide the following measures:

(1) Matching each premise and hypothesis separately can make full use of the details of each premise, we call this process as local matching. In the process, we believe that a dynamic combination of similarity information and difference information between two sentences can fully describe their relationship enough, so a matching gate mechanism is proposed to obtain the local matching features.
(2) To avoid model based the "Local Matching-Integration" strategy falling into the trap of multiple local judgment errors, we try to rationally integrate these local matching features. An interaction mechanism for all the local matching features is carried out to make each local feature has global semantic dependence with each other, we call this dependence as global relationship. We also hope that the really useful local features can reach the classification module more smoothly, a fusion gate mechanism is employed, so that the local matching features and the global relationship feature can be combined to make a final judgment.
(3) Since the scale of the training set of MPE is too small, a complex neural network is easy to fall into over-fitting. We propose to pre-train a part of our model with a large-scale SPE dataset. In order to solve the problem that the form of SPE data is different from the MPE data, a fine-tuning method suitable for our model is proposed.

We applied AGNet on MPE test set and got 65.5% accuracy, which was 40.2% and 80.1% respectively when the relationship is Neutral and Entailment, exceeding the existing best results by 1.0% and 2.2%. The new state-of-art result were obtained in the pre-trained model, exceeding the highest accuracy by 0.3%. Furthermore, based on the experimental results, we discuss some unique difficulties of MPE compared to SPE, demonstrate the importance of some process for multiple-premise and how to deal with them. We also perform an extensive analysis to clarify the strengths and weaknesses of our model.

2. Related work

Traditional natural language inference include similarity-based alignment (Adams, 2006; Jijkoun & de Rijke, 2005; Marneffe, Rafferty, & Manning, 2008), logical calculus and rules (Bayer, Burger, Ferro, Henderson, & Yeh, 2005). Similarity-based method is relatively simple to implement, but the assumption that "similarity means entailment" is unreliable. On this basis, the judgement based on lexical alignment is proposed, this method finds the similar parts of the pair and aligns them, then uses the degree of alignment as the basis for judging the relationship, the problem is it cannot do the complex alignment between different granularity in sentence pairs flexibly (Noh et al., 2015). The method based on logic calculus and rules uses the idea of mathematical proof and grammar tree, but the transformation from natural language to logical expression (Moldovan, Clark, Harabagiu, & Maiorano, 2003) is not robust enough, and the lack of linguistic knowledge leads to an accumulation of errors in reasoning process, which makes the model difficult to transfer to other knowledge areas (Akhmatova, 2005).

Using neural network models such as Recurrent Neural Network (RNN) (Giles, Kuhn, & Williams, 1994), Convolutional Neural Network (CNN) (LeCun et al., 1989), or attention mechanism to construct a classification function is the most accurate method that can be achieved at present. As a variant of RNN, Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) has excellent sequence processing ability, it is widely used in various NLP tasks. Attention mechanism is a component that extracts important semantics, it is first used in machine translation based on encoder and decoder mechanism (Kalchbrenner & Blunsom, 2013), which can realize the soft alignment of words between two sentences and obtain the weight of words in the original text to words in the translation. In recent years, the proposal to SPE or MPE problem has been almost explored in the combination of LSTM and attention.

Liu, Sun, Lin, & Wang (2016) takes the form of in-sentence attention, which is derived from the human habit of reading. When the brain reads a sentence, it can often form a rough intuition about which parts of the sentence are important. Inner-attention uses an average pooling (LeCun, Bengio, & Hinton, 2015) for LSTM encoded hidden vectors as a query vector, then calculate the relevant weight of the hidden vector to obtain the importance of words to the overall semantics, and finally weighted sum to obtain the sentence vector. Rocktäschel, Grefenstette, Hermann, Kočiskỳ, & Blunsom (2015) construct Word-to-word Attention on LSTM, that is, when the second LSTM reads each word in hypothesis, it

considers the weighted information output by the first LSTM reading premise, but from the premise to the hypothesis is a one-way process, which is inconsistent with the repeated comparison process when humans judge the relationship between two sentences. Parikh, Täckström, Das, & Uszkoreit (2016) use a bi-directional attention, multiply each word in the hypothesis by a weight matrix to match each word in the premise, and get the premise's alignment vector. The hypothesis's alignment vector is also obtained by the same operation, then calculate the alignment degree of the two alignment vectors to classify. On this basis, Ghaeini et al. (2018) measure the alignment degree with multiple perspectives, and employ a bidirectional LSTM to deduce the potential relationship between alignment vectors. Liu, Jiang, Yu, and Yu (2018) use gate mechanism, difference features and similarity features of sentence pairs are successively input into multiple LSTM for reasoning, it is said the mechanism could simulate the reading habit of human brain through memory storage. By this mechanism, the state-of-the-art results are achieved on MPE test set. These are the sources of inspiration for our model to use attention and gate mechanism to match sentence pairs. However, these methods still use the "Concatenate-Matching" strategy, are not suitable for solving MPE problem. LSTM is difficult to encode a long text (Tang, Müller, Rios, & Sennrich, 2018), after concatenation of multiple-premise, the premise will get longer, it is difficult for long-distance words in the sentence to get information interaction, which will result in loss of syntax and semantics. In terms of implementation, parallel computation is also difficult for these models.

Premise-wise sum of experts (SE) (Lai, Bisk, & Hockenmaier, 2017) takes the lead in exploring the relationship between multiple-premise. They use conditional LSTM (Rocktäschel, Grefenstette, Hermann, Kočiskỳ, & Blunsom, 2015) to match each premise and hypothesis respectively, that is, convert the MPE problem into multiple SPE problems. Thus, multiple classification results will be obtained, we call them local matching features or local results. Finally, the model votes to get the final classification result. This method takes advantage of the fact that data has multiple-premise, so it can do parallel matching regardless of the number of premise sentences. However, due to the third characteristic, namely the information asymmetry between these premises, the model does not further process the local matching results, which may lead to an error accumulation and thus obtain a wrong final classification result.

Multi-head self-attention mechanism (Vaswani et al., 2017) proved to be effective in obtaining the syntactic and semantic information from sentence. Compared to RNN or CNN, self-attention is flexible in modeling both long-range and local dependencies. In order to alleviate the large memory footprint in matrix calculation of long sentence, Shen, Zhou, Long, Jiang, & Zhang (2018b) propose Bi-directional Block self-attention (Bi-BloSA), which divides one sentence into multiple blocks and calculate attention in each block to get local features, then calculate attention between blocks to get long-distance correlation features. After that, two gate mechanisms are used to fuse local features and correlation features. This model provides us an inspiration on how to integrate multi-premise features.

3. Model

The main framework of AGNet is shown in Fig. 2. According to function, the model can be divided into input encoding layer (Section 3.1), attention layer (Section 3.2), matching layer (Section 3.3), inference layer (Section 3.4) and (classification layer (Section 3.5). According to the matching object, we divide our model into "Intra" part and "Inter" part. The "Intra" part matches each premise with hypothesis to get local matching features. The "Inter" part matches all the local matching features to obtain global relationship features. This will be detailed later.

In theory, the number of multiple premises can be any positive integer greater than 1, due to the format of the dataset MPE used in our experiments is 4 premises and 1 hypothesis, so this paper sets four premises for $p^1 = [p_1^1, \ldots, p_{m1}^1]$, $p^2 = [p_1^2, \ldots, p_{m2}^2]$, $p^3 = [p_1^3, \ldots, p_{m3}^3]$, and $p^4 = [p_1^4, \ldots, p_{m4}^4]$. Hypothesis is $h = [h_1, \ldots, h_n]$, Where $m1$, $m2$, $m3$ and $m4$ represent the length of the four premises respectively, n represents the length of the hypothesis. $p_i, h_j \in \mathbb{R}^r$ is embedded in the r-dimensional pre-trained word. Our purpose is to predict label $y \in$ [*entailment, contradiction, neutral*]. Since the four premises are handled separately in our model and the operations are same in Intra part, to avoid repetition, the following text uses p to represent any premise in Intra part, and m to represent the length.

3.1. Input encoding layer

The input encoding layer transforms a context-independent word embedding into a vector with specific task context semantics. We could choose bidirectional LSTM (Hochreiter & Schmidhuber, 1997) or self-attention (Vaswani et al., 2017) to do this. Compared with LSTM, self-attention ignores the positional information of words in a sentence, it is not necessary to process input words linearly as LSTM does, can make any two words directly calculate the relevant score by matrix, and then obtain the dependence of each word on the whole sentence through weighting calculation. We use self-attention to encode premise p:

$$q = W^q p, k = W^k p, v = W^v p \tag{1}$$

$$m_{ij}^{fw} = \begin{cases} 0, & i < j \\ -\infty, & otherwise \end{cases},$$

$$m_{ij}^{bw} = \begin{cases} 0, & i > j \\ -\infty, & otherwise \end{cases}, \quad \forall i, j \in [1, \ldots, m] \tag{2}$$

$$a_{ij} = {q_i}^T k_j + m_{ij}, \quad \forall i, j \in [1, \ldots, m] \tag{3}$$

$$\overline{p}_i = \sum_{j=1}^{m} \frac{exp(a_{ij})}{\sum_{k=1}^{m} exp(a_{kj})} v_j, \quad \forall i, j \in [1, \ldots, m] \tag{4}$$

Where W^q, W^k, $W^v \in \mathbb{R}^{r \times r}$ are weight parameters to be learned. The calculation process of attention mechanism can be regarded as a query process of q (query), k (key) and v (value), so a_{ij} is the relevant score of q_i and k_j. $\overline{p}_i \in \mathbb{R}^r$ is a new vector with weighted context semantics. In order for the context vectors to have order information, we utilize mask matrices $m^{fw} \in \mathbb{R}^{r \times r}$ and $m^{bw} \in \mathbb{R}^{r \times r}$ (Shen et al., 2018a), which are the forward and reverse mask matrices respectively. So actually the a_{ij} is a concatenation of a_{ij}^{fw} and a_{ij}^{bw}. In this paper. We employ multi-head self-attention, which is to concatenate multiple $\overline{p}_i$ obtained by different weight parameters, it is considered that we can learn different dependencies between vocabularies from different semantic subspaces.

For the convenience of the following description, we generalize the formulas (1) to (4) as follows:

$$\overline{p} = MaskedSelfAttention(p) \tag{5}$$

In some cases, it is not necessary to give the features order information, that is to say, m_{ij} in formula (2) can be all 0, then the formula expressed as:

$$\overline{p} = SelfAttention(p) \tag{6}$$

Replace the input p with h to perform the same operation can get $\overline{h} \in \mathbb{R}^{n \times r}$. It should be noted that the two calculation processes share the same weight parameters.

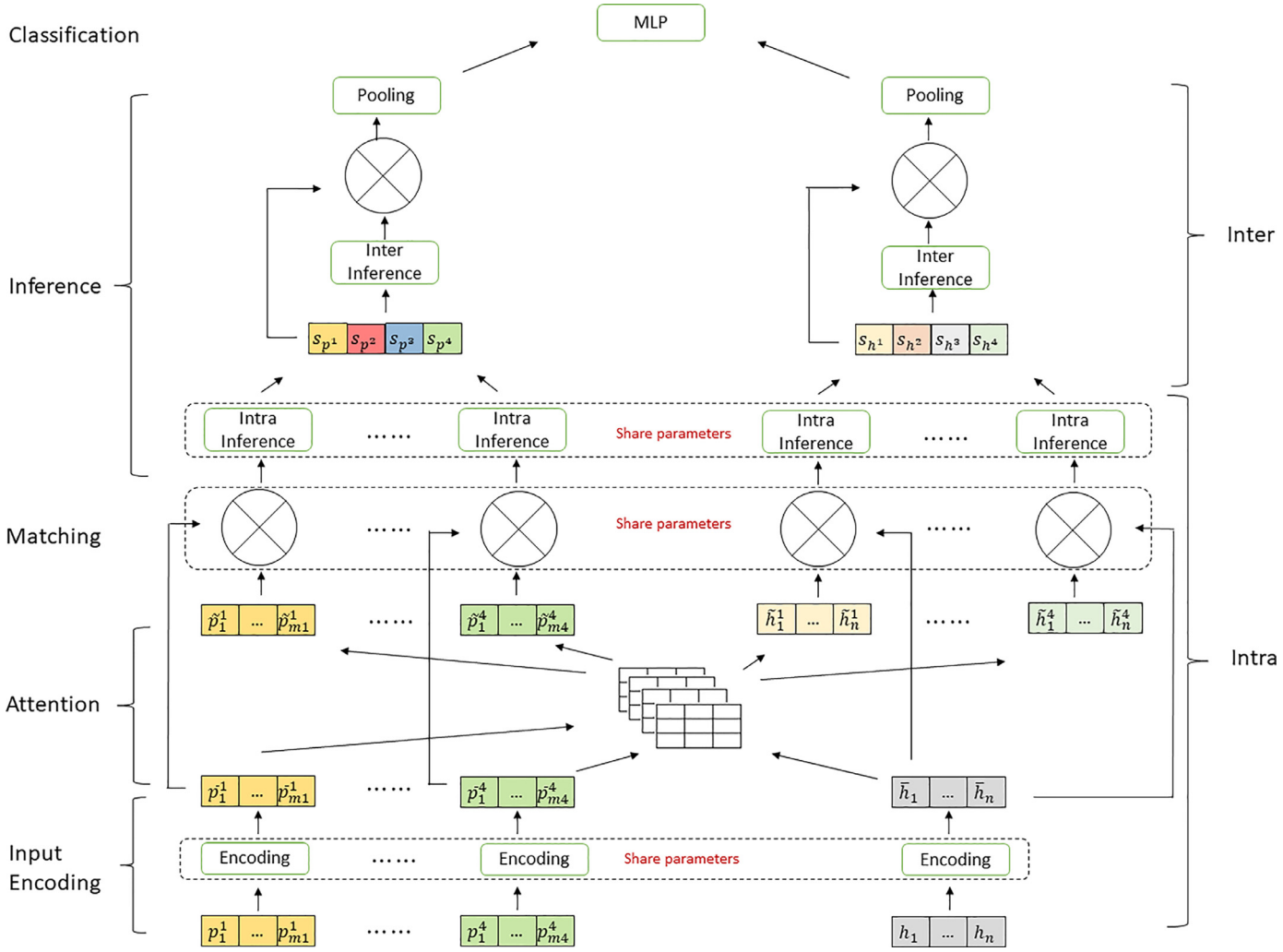

Fig. 2. A high-level view of AGNet model. The rectangle represents the tensor, the rounded rectangle represents the operation on the tensor, and the operation parameters in the same dashed box are shared. The ⊗ represents gate mechanism.

3.2. Attention layer

Finding the corresponding alignment vector for the context vector is widely used in machine translation (Bahdanau, Cho, & Bengio, 2014) and many other NLP tasks. The alignment vectors in matching process can help model to get the alignment degree of two sentences. Unlike self-attention, which is used to extract semantic and syntactic information from sentence, this attention mechanism in this module is to extract the correlation between premise and hypothesis:

$$e_{ij} = \overline{p}_i^T \overline{h}_j, \quad \forall i \in [1, m], \quad \forall j \in [1, n] \tag{7}$$

$$\tilde{p}_i = \sum_{j=1}^{n} \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} \overline{h}_j, \quad \forall i \in [1, m] \tag{8}$$

$$\tilde{h}_j = \sum_{i=1}^{m} \frac{exp(e_{ij})}{\sum_{k=1}^{m} exp(e_{kj})} \overline{p}_i, \quad \forall j \in [1, n] \tag{9}$$

e_{ij} is a scalar which indicates the correlation score of $\overline{p}_i$ and $\overline{h}_j$. $\tilde{p}_i \in \mathbb{R}^r$ is the weighted summaries of the hypothesis in terms of each word in the premise, it expresses the importance of each word in the hypothesis to the context vector $\overline{p}_i$. The same reason can understand $\tilde{h}_j \in \mathbb{R}^r$. Please note in this step, the attention is calculated between the context vectors of each premise and the context vectors of the hypothesis, so we will obtain $\tilde{p}^1 \in \mathbb{R}^{m1 \times r}$, $\tilde{p}^2 \in \mathbb{R}^{m2 \times r}$, $\tilde{p}^3 \in \mathbb{R}^{m3 \times r}$, $\tilde{p}^4 \in \mathbb{R}^{m4 \times r}$ and $\tilde{h}^1$, $\tilde{h}^2$, $\tilde{h}^3$, $\tilde{h}^4 \in \mathbb{R}^{n \times r}$ respectively.

3.3. Matching layer

In the matching layer, we evaluate the degree of alignment between context vectors and alignment vectors by two criteria: similarity and difference.

We use two operations to get the relationship between two vectors in one semantic space, that is, the product of element-by-element multiplication to reflect the degree of similarity, and the difference of element-by-element subtraction to reflect the degree of difference (Mou et al., 2015; Chen et al., 2016). We use a matching gate mechanism to combine the two:

$$mul_i = \overline{p}_i \odot \tilde{p}_i, \quad \forall i \in [1, \ldots, m] \tag{10}$$

$$sub_i = \overline{p}_i - \tilde{p}_i, \quad \forall i \in [1, \ldots, m] \tag{11}$$

$$G_1 = \sigma(W^{g1}[mul_i; sub_i] + b^{g1}), \quad \forall i \in [1, \ldots, m] \tag{12}$$

$$c_{p,i} = G_1 \odot sub_i + (1 - G_1) \odot mul_i, \quad \forall i \in [1, \ldots, m] \tag{13}$$

where $\odot$ and - represent multiplication and subtraction respectively, the brackets [;] denote concatenation, σ means sigmoid, an activation function (LeCun et al., 2015). $W^{g1} \in \mathbb{R}^{2r\times r}$ and $b^{g1} \in \mathbb{R}^r$ are weight parameters to be learned. $mul_i \in \mathbb{R}^r$ is considered to emphasize similar information for both, while ignoring different information. $sub_i \in \mathbb{R}^r$ retains the different information of the two and removes similar information. We think that similarities and differences should be complementary, the two and their proportion in each dimension can constitute all the information describing the relationship between two objects. Formula (12) and (13) is similar to the update gate in GRU (Wang, Yang, Wei, Chang, & Zhou, 2017), which determines how much information of similarities and differences should be reserved for the next layer. We call $c_{p,i} \in \mathbb{R}^r$ as a local matching feature.

$c_h \in \mathbb{R}^{n\times r}$ can be obtained by the same operation for $\overline{h}$ and $\tilde{h}$, and parameters are shared with premises.

3.4. Inference layer

3.4.1. Intra inference

The purpose of this layer is to further process local matching feature sequence c_p and c_h by re-encoding them, which is based on the opinion that a feature may have contextual association with another feature in a sequence (Chen et al., 2016; Ghaeini et al., 2018).

Here we keep using masked multi-head self-attention as follow:

$$u_p = MaskedSelfAttention(c_p) \tag{14}$$

After obtained the context-dependent matching feature sequence $u_p \in \mathbb{R}^{m\times r}$, all the feature vectors are compressed into a single feature vector, the motivation is to get a most effective feature for classification. Here we can choose maximum pooling to get a most salient feature (Zhou & Chellappa, 1988), average pooling to get an average feature (Shen, Zhou, Long, Jiang, & Zhang, 2018b), or attentive pooling, weighted summation of all dimensions of features (Lin et al., 2017):

$$G_2 = \sigma(W^{g2}u_p + b^{g2}) \tag{15}$$

$$s_p = \sum_{i=1}^{m} G_{2,i} \odot u_{p,i}, \quad \forall i \in [1, \ldots, m] \tag{16}$$

$W^{g2} \in \mathbb{R}^{r\times r}$ and $b^{g2} \in \mathbb{R}^{m\times r}$ are weight parameters to be learned. $G_2 \in \mathbb{R}^{m\times r}$ determines the importance of each row in the matrix u_p, so the obtained $s_p \in \mathbb{R}^r$ can be regarded as a single vector with all the sentence feature information.

$s_h \in \mathbb{R}^r$ can be obtained by the same operation for u_h. We call s_p and s_h as local inference features.

3.4.2. Inter inference

In this layer, we will aggregate all the local inference features in order to obtain the inter-sentence relationship of multi-premise. In the above, four premise local inference features s_p^1, s_p^2, s_p^3, s_p^4 and four hypothesis local inference features s_h^1, s_h^2, s_h^3, s_h^4 were obtained in the way of solving SPE task. Some of them may be very important for classification, some may be useless, and some may need to be combined with each other to be useful. We have to infer the relationship between them. From this part, we encode them to capture the inter-sentence relationship:

$$t_p = \left[s_p^1; s_p^2; s_p^3; s_p^4\right] \tag{17}$$

$$v_p = SelfAttention(t_p) \tag{18}$$

Where $t_p, v_p \in \mathbb{R}^{4\times r}$. In Intra part, the granularity of self-attention calculation is word in a sentence, which allows the word to have contextual semantics and syntactic information. While the computational granularity in Eq. (18) is sentence, more precisely, is the local inference feature of each premise. Through this operation, each local inference feature can be given global relationship information. We deliberately removed the mask matrix, because the second characteristic of multi-premise is that there is no order information. The same operation for s_h^1, s_h^2, s_h^3 and s_h^4 can get $v_h \in \mathbb{R}^{4\times r}$. We call v_p and v_h are global inference features.

In order to combine the local inference features and the global inference features, we use a fusion gate mechanism to dynamically combine s and v (Gong & Bowman, 2017):

$$G_3 = \sigma\left(W^{o1}s_p^1 + W^{o2}v_p^1 + b^o\right) \tag{19}$$

$$o_p^1 = G_3 \odot s_p^1 + (1 - G_3) \odot v_p^1 \tag{20}$$

$$o_p = \left[o_p^1; o_p^2; o_p^3; o_p^4\right] \tag{21}$$

Where $W^{o1}, W^{o2} \in \mathbb{R}^{r\times r}$, $b^o \in \mathbb{R}^r$ are weight parameters to be learned. The idea of this step is if there are local inference features that are directly beneficial to classification, we hope it can smoothly reach the final classification layer. Therefore, $G_3 \in \mathbb{R}^r$ is a gating device that controls the ratio of s_p to v_p, and $o_p \in \mathbb{R}^{4\times r}$ is a feature that combines local and global information. $o_h \in \mathbb{R}^{4\times r}$ can be obtained by the same operation from s_h and v_h.

Finally, the obtained o_p and o_h are put into an attentive pooling, respectively, i.e. formula (15,16), and a single vector with all premise (hypothesis) features is obtained.

$$P = AttentivePooling(o_p), H = AttentivePooling(o_h) \tag{22}$$

Where $P, H \in \mathbb{R}^r$.

3.5. Classification layer

In this layer, P and H are concatenated into a full-connected network containing at least one hidden layer, which is activated by a relu function. Finally, a softmax layer is used to get the final classification result (LeCun et al., 2015).

4. Fine-tune

The weight parameters of neural network need enough data to optimize, otherwise the model will fall into serious overfitting of training set. In other words, the model will perform well on the training set, but it will perform poorly on the test set. In view of the shortcoming of insufficient training data in MPE corpus, we propose to use pre-training to remedy this neglect.

In our proposal, we pre-train the Intra part of AGNet. Firstly, s_p and s_h are connected to a full connection layer to classify, and their weight parameters are learned with SNLI data set for SPE task. After several epochs of training, we remove the full connection layer, the Inter part is connected, then all weight parameters are learned on MPE training set. In this method, the weight parameters of the Inter part mainly learn the unique multi-premise characteristics.

In terms of the rationality of fine-tune in AGNet, we believe that the Intra part of our model is functionally similar to the convolution process of CNN for image classification. In image classification models (such as ImageNet (Huh, Agrawal, & Efros, 2016)), convolutional kernels trained with a large amount of image data, can adequately extract graphical features, so it is sufficient to fine-tune the final fully connected layer for other particular tasks. The Intra part of AGNet extracts local features as well. In addition, for the attention mechanism that requires a large amount of data to learn reasonable weight parameters, it is also necessary to use pre-training (Lai, Bisk, & Hockenmaier, 2017).

5. Experiments

5.1. Datasets

The premises of Stanford Natural Language Inference (SNLI) (Dagan & Glickman, 2004) come from the image annotation of Flickr30r, the hypotheses are generated artificially, the training set has 549,367 pairs of sentences, the validation set and the test set have 10,000 pairs of sentences respectively, and each pair of sentences is labeled as one of (Entailment, Neutral, Contradiction, -). "-" means that the reviewers cannot make a consistent judgment on the relationship between sentences, in data preprocessing, we filter such sentences.

The Multiple Premise Entailment NLI Corpus (MPE) (Lai, Bisk, & Hockenmaier, 2017) has premises and hypotheses both from Flickr30k image labeling, its training set has 8,000 sets of sentences, the validation set and the test set have 1,000 sets of sentences respectively, each set of sentences is labeled as one of (Entailment, Neutral, Contradiction). The unique characteristic of this corpus is that each set includes 4 premise sentences and 1 hypothesis sentence.

5.2. Experimental setup

We use pre-trained 300D Glove 840B word vector (Pennington, Socher, & Manning, 2014) to initialize our word embedding vectors, the out-of-the-vocabulary words is randomly initialized. There are 6 heads in self-attention of the Intra part, 50D for each head; In the Inter part, there are 3 heads, each 100D. L2 regularization with a coefficient of 0.0003 is added to all weights in the optimization process, and Dropout ratio of 0.25 is applied to all full connection layers (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Layer Normalization (Ba, Kiros, & Hinton, 2016) is used behind each self-attention layer. The batch size is set to 32, the optimizer is Adam (Adams, 2006), and the learning rate is set to 0.0003. The above values are the optimal values we choose after debugging.

5.3. Models for comparison

We experimented with several related models as a comparison of our models. In term of how to deal with multiple-premise, conditional LSTM, Word-to-word Attention (Rocktäschel, Grefenstette, Hermann, Kočiskỳ, & Blunsom, 2015), DR-BiLSTM (Ghaeini et al., 2018) and MIMN (Liu et al., 2018) concatenate the four premise sentences into one sentence, that is, by "Concatenate-Matching" strategy. SE uses conditional LSTM to match each premise and hypothesis separately, then sums logit prediction to obtain the final judgment, belongs to "Local Matching-Integration" strategy.

About whether to encode a sentence to an embedding or not, conditional LSTM and SE encoded the sentence as a sentence vector and judge the relationship between these vectors. Instead, Word-to-word Attention, DR-BiLSTM and MIMN aligns the words in corresponding sentences to get various features then classify them.

In order to verify the effectiveness of large-scale data pre-training, we also pre-train the comparison models: LSTM+snli, Attention+snli, SE+snli, DR-BiLSTM+snli and MIMN+snli. They are firstly trained on SNLI data set to do SPE task, and then fine-tuned by MPE training set.

To verify the effectiveness of the major components in AGNet, we design the following variants for comparison:

AGNet is the model described in Section 3, only trained on MPE training set.

AGNet-gate1 directly concatenates similarity features and difference features then through an activation function, removes the matching gate mechanism:

$$c_{p,i} = relu(W^c[mul_i; sub_i] + b^c), \quad \forall i \in [1, \ldots, m], W^c \in \mathbb{R}^{2r \times r}, b^c \in \mathbb{R}^r \tag{23}$$

AGNet-gate2 omits the fusion gate mechanism in the Inter part:

$$P = AttentivePooling(v_p), H = AttentivePooling(v_h) \tag{24}$$

AGNet-inter removes the whole Inter part of AGNet and directly adds the local inference feature vectors together into a classification layer:

$$P = s_p^1 + s_p^2 + s_p^3 + s_p^4, H = s_h^1 + s_h^2 + s_h^3 + s_h^4 \tag{25}$$

AGNet+snli is the model described in Section 3, uses the SNLI corpus for pre-training in the Intra part.

Table 1
Total accuracy and breakdown of accuracy with respect to classes on MPE test set. N=neutral, E=entailment, C=contradiction.

Models	Test(%acc)	N	E	C
LSTM	53.5	39.2	63.1	53.5
Attention	53.9	30.2	61.3	66.5
SE	56.3	30.6	48.3	71.2
DR-BiLSTM	63.3	37.3	73.6	69.1
MIMN	**66.0**	35.3	77.9	**73.1**
AGNet	65.5	**40.2**	**80.1**	67.5
AGNet-gate1	64.6	39.2	80.0	66.0
AGNet-gate2	63.4	39.1	69.1	72.1
AGNet-inter	59.9	34.5	64.0	70.5

5.4. Results

As shown in Table 1, in these models only use MPE corpus as the training data, "Matching-Aggregation" framework generally achieves better result, which may indicate that for sentence pairs relationship judgment task, matching with multiple granularities (words, phrases and sentences) is better than matching with sentence granularity alone. Among them, MIMN achieves the highest total accuracy. In the judgment of neutral and entailment, AGNet, including its variants, are optimal. Particularly, AGNet is good at judging sentence pairs with entailment. It is worth noting that, compared to the results of AGNet-inter, if there is no Inter part in AGNet, the accuracy on entailment decreased by 16.1%. Let us observe another case: SE and LSTM by the same structure for matching two sentences, LSTM uses the "Concatenate-Matching" strategy, SE uses the "Local Matching-Integration" strategy but only votes to determine the classification results in integration step. SE achieved better total accuracy than LSTM, but poorly in entailment. These facts seem to indicate two points: (i) The "Local Matching-Integration" strategy works better on total accuracy under the same matching method. (ii) If the local inference features are not properly handled, the model will be very bad at judging the entailment. We think the reason could be found in the characteristics of the data. Due to the asymmetry of premises, once the relationship of any one premise and the hypothesis is entailment, the semantic scene composed of all the premises must be related to the hypothesis as entailment. In SE, if not half of the premises and hypothesis are entailment, then the model will not make an entailment judgment. In the Inter part of AGNet, we encode the local inference features with self-attention, then integrate local inference features and global inference features through a fusion gate. This module will give more "rewards" to the local inference features if they containing entailment elements. All of these phenomena indicate that in the "Local Matching-Integration" strategy, it is necessary to reasonably integrate local inference features.

Premises:
1. A teenager is sitting behind a cardboard box smiling at the camera.
2. A young man is sitting and posing for the camera.
3. Man selling goods in a poor country.
4. Man giving thumbs up to the camera.

Hypothesis:
A man makes a hand gesture. ENTAILMENT

Fig. 3. A set of sentences in MPE test set.

As can be seen from the performance of other variants, the matching gate mechanism and the fusion gate mechanism all played a positive role in AGNet. Compared with AGNet-gate1 and AGNet, we can see that the matching gate mechanism which fuses similarity features and difference features decreases the ability of model to identify contradiction, but significantly improves the ability to recognize neutral. It could be explained that the matching gate mechanism can dynamically weigh the entailment and contradiction, similar to the exclusion process used by human brain when judging neutral. Compared with AGNet-gate2 and AGNet, the accuracy of the model with fusion gate mechanism in judging entailment is improved by 11.0%. As mentioned above, self-attention gives the Inter part a "reward" for judging entailment, and the fusion gate also gives the feature a "highway" from Intra part through the Inter part. The two complement each other.

We employ a concrete example in Fig. 3 to confirm the above viewpoint. In this case, the most critical information "thumbs up" from premise 4 and "makes a hand gesture" from hypothesis, the latter's semantics include the former's, that is, the former can infer the latter, so the relationship between the two is entailment. There is no doubt that hypothesis is true in the scenario described by the four premises. In the matching process, the useless information brought by the other three premises needs to be filtered. In models such as MIMN that use the "Concatenate-Matching" strategy, the interference from invalid information is reflected in the encoding phase. In the "Local Matching-Integration" strategy, it is mainly reflected in the selection of local features in the integration part. The results of Table 2 show that the Inter module of AGNet is vital to process such data.

Table 3 shows the accuracy of models on the test set of MPE after pre-training on SNLI. Among the three models with outstanding performance in Table 1, AGNet+snli achieves new state-of-the-art, while MIMN and DR-BiLSTM did not improve after pre-training, but decreased. This may indicate that for complex models, their strong fitting ability leads to over-fitting on pre-training data, and also shows that our pre-training with fine-tuning strategy for AGNet is reasonable.

In addition, we can see the giant improvement in total accuracy between word-to-word Attention before and after, the reason may be that the attention mechanism requires a large amount of data to learn reasonable weight parameters (Lai, Bisk, & Hockenmaier, 2017). But we can't make a conclusion, because DR-BiLSTM, MIMN, and AGNet, which also use the attention mechanism, did not achieve the same effect after receiving pre-training.

SE+snli achieved the best results on neutral, nevertheless, the accuracy of all models on neutral is not commensurate with the other two, which seems to indicate that the judgment of neutral is more difficult in MPE task, we did not see this situation in SPE task (Parikh, Täckström, Das, & Uszkoreit, 2016). We try to explore this issue from a semantic perspective: There is a semantic phenomenon between sentences with relationship of neutral, called compatibility. Compatibility means that two actions can be performed simultaneously by the same agent (e.g. "A boy flying a red and white kite" vs. "A boy is smiling"). The semantic phenomenon existing between sentences with contradiction is called mutual exclusion, is the opposite of compatibility, that is, two actions cannot be performed simultaneously by the same agent (e.g. "Two doctors perform surgery" vs. "Two surgeons are having lunch"). In theory, both in SPE and in MPE, these two phenomena are almost the criteria for determining the relationship between sentences. But the situation is more complicated in MPE, because the model needs to determine whether multiple premises and hypothesis are compatible or mutual exclusion. It is easy to know that as long as one of the premises is mutually exclusive with the hypothesis, the whole set of sentences is contradictory. This means that in actual matching, if the model misjudges in a local matching (misjudges compatibility as mutually exclusive), then a set of sentences with relationship of neutral will be misclassified. SE+snli can achieve good results on neutral, perhaps because pre-training greatly improves the ability of LSTM to identify compatible actions. Similarly, LSTM+snli also achieves a good result on neutral.

In terms of time complexity (Table 4). As a special neural network for sequence processing, LSTM has to input and calculate words word by word, which reflects the ability of capturing sentence order information, but also exposes its poor parallel computing ability. Therefore, once LSTM is used in a model, the computational time of this model mainly depends on the sentence length n, the time complexity will be $O(n)$. This means that the longer the sentences are, the more time the model takes. What's more, in MPE task, it is not only influenced by sentence length, but also by the number of premise sentences m (we assume the length of each premise is n). If a model under the "Concatenate-Matching" strategy uses LSTM, the time complexity will reach $O(n \times m)$. By contrast, AGNet, no RNN-like time series module is used in the whole model, base the self-attention structure to encode a sentence that calculated with matrix can be parallelized. Besides, because of the "Local Matching-Integration" strategy, each pair of sentences can be matched simultaneously. That is to say, neither the length of sentences nor the number of premise sentences in these steps will

Table 2
Classification of the examples in Fig. 3 for each model.

Models	Classification
LSTM	N
Attention	C
SE	C
MIMN	N
AGNet	**E**
AGNet-gate1	**E**
AGNet-gate2	N
AGNet-inter	C

Table 3
Accuracy after pre-training on SNLI data set.

Models	Test(%acc)	N	E	C
LSTM+snli	60.4, ↑ 6.9	40.9, ↑ 1.7	65.1, ↑ 2.0	67.2, ↑13.1
Attention+snli	64.0, ↑10.1	32.8, ↑ 2.6	75.9, ↑14.6	71.5, ↑ 5.0
SE+snli	60.0, ↑ 3.7	**42.7**, ↑12.1	65.4, ↑17.1	65.1, ↓ 6.1
DR-BiLSTM+snli	62.8, ↓ 0.5	34.5, ↓ 2.8	77.1, ↑ 3.5	66.7, ↓ 2.4
MIMN+snli	64.5, ↓ 1.5	32.0, ↓ 3.3	72.5, ↓ 5.4	**75.8**, ↑ 2.7
AGNet+snli	**66.3**, ↑ 0.8	40.2, - 0.0	**80.3**, ↑ 0.2	69.2, ↑ 1.7

Table 4
The time complexity of models.

Model	Time Complexity
LSTM	O(n)
Attention	O(n)
SE	O(n)
DR-BiLSTM	O(n)
MIMN	O(n)
AGNet	**O(1)**

affect the computational time complexity. Therefore, the matching complexity reaches $O(1)$, the number of premise sentences and the length of sentences can be basically ignored.

We assume that other models with good accuracy, such as MIMN, give up LSTM and use self-attention to encode, the time complexity can indeed reach $O(1)$. But because MIMN uses the "Concatenate-Matching" strategy, a long sentence composed of multi-premise will require a matrix of $(m \times n) \times (m \times n)$ at the attention calculation, which will take up very large memory. However, our model only needs $m \times n \times n$ matrices when performing local matching calculation, the computational cost is greatly reduced.

6. Conclusion

In this paper, we propose AGNet for multiple-premise inference and achieve state-of-the-art on MPE test set. The experiments prove that the matching gate mechanism can fully describe the alignment relationship. The inter-sentence self-attention and fusion gate mechanism can fuse the features of multiple-premises. A pre-training and fine-tuning method for a part of the model can alleviate the problem of insufficient MPE training data. In computation, our model achieves a high degree of parallelism. We believe that future work on MPE task will continue on the "Local Matching-Integration" strategy, and the construction of models based on the characteristics of multi-premise sentences will become a hot topic. In addition, it may be another important approach to properly transfer solutions from a large number of existing SPE tasks to MPE tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Pin Wu: Conceptualization. **Zhidan Lei:** Methodology, Writing - original draft. **Quan Zhou:** Investigation. **Rukang Zhu:** Investigation. **Xuting Chang:** Writing - review & editing. **Junwu Sun:** Writing - review & editing. **Wenjie Zhang:** Writing - review & editing. **Yike Guo:** Validation.

References

Adams, R. (2006). Textual entailment through extended lexical overlap. In *Proceedings of the second pascal challenges workshop on recognising textual entailment* (pp. 128–133).

Akhmatova, E. (2005). Textual entailment resolution via atomic propositions. In *Proceedings of the pascal challenges workshop on recognising textual entailment*. Citeseer.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *CoRR*, abs/1607.06450.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR-15),* San Diego, California, USA.

Bayer, S., Burger, J., Ferro, L., Henderson, J., & Yeh, A. (2005). Mitre??s submissions to the eu pascal rte challenge. *Proceedings of the pattern analysis, statistical modelling, and computational learning (pascal) challenges workshop on recognising textual entailment*. Citeseer.

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the EMNLP Conference* (pp. 632–642).

Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., & Inkpen, D. (2016). Enhanced lstm for natural language inference. In *Proceedings of the ACL Conference* (pp. 1657–1668).

Dagan, I., & Glickman, O. (2004). Probabilistic textual entailment: Generic applied modeling of language variability,.

Ghaeini, R., Hasan, S. A., Datla, V., Liu, J., Lee, K., Qadir, A., et al. (2018). Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*.

Giles, C. L., Kuhn, G. M., & Williams, R. J. (1994). Dynamic recurrent neural networks: Theory and applications. *IEEE Transactions on Neural Networks*, (2), 153–156.

Gong, Y., & Bowman, S. R. (2017). Ruminating reader: Reasoning with gated multi-hop attention. *CoRR*. abs/1704.07415

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, (8), 1735–1780.

Huh, M., Agrawal, P., & Efros, A. A. (2016). What makes imagenet good for transfer learning? *CoRR*, abs/1608.08614.

Jijkoun, V., & de Rijke, M. (2005, April). Recognizing textual entailment using lexical similarity. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment* (pp. 73–76).

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1700–1709).

Lai, A., Bisk, Y., & Hockenmaier, J. (2017). Natural language inference from multiple premises. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)* (pp. 100–109).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, (7553), 436.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, (4), 541–551.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., et al. (2017). A structured self-attentive sentence embedding. In *Proceedings of the 5th International Conference on Learning Representations (ICLR-17)*.

Liu, C., Jiang, S., Yu, H., & Yu, D. (2018). Multi-turn inference matching network for natural language inference. In *Ccf international conference on natural language processing and chinese computing* (pp. 131–143). Springer.

Liu, Y., Sun, C., Lin, L., & Wang, X. (2016). Learning natural language inference using bidirectional lstm model and inner-attention. CoRR, *abs/1605.09090*.

Marneffe, M.-C., Rafferty, A. N., & Manning, C. D. (2008). *Proceedings of ACL-08: HLT*, 1039–1047.

Moldovan, D., Clark, C., Harabagiu, S., & Maiorano, S. (2003). Cogex: A logic prover for question answering. In *Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology-volume 1* (pp. 87–93). Association for Computational Linguistics.

Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., et al. (2015). Recognizing entailment and contradiction by tree-based convolution. *CoRR, abs/1512.08422*.

Noh, T.-G., Padó, S., Shwartz, V., Dagan, I., Nastase, V., Eichler, K., et al. (2015). Multi-level alignments as an extensible representation basis for textual entailment algorithms. In *Proceedings of the fourth joint conference on lexical and computational semantics* (pp. 193–198).

Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In *Proceedings of the EMNLP Conference* (pp. 2249–2255).

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., & Blunsom, P. (2015). Reasoning about entailment with neural attention. In *Proceedings of the 3th International Conference on Learning Representations (ICLR-15),* San Diego, California.

Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., & Zhang, C. (2018a). Disan: Directional self-attention network for rnn/cnn-free language understanding. *Thirty-second aaai conference on artificial intelligence*.

Shen, T., Zhou, T., Long, G., Jiang, J., & Zhang, C. (2018b). Bi-directional block self-attention for fast and memory-efficient sequence modeling. In *Proceedings of the 6th International Conference on Learning Representations (ICLR-18),* Vancouver, BC, Canada.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, (1), 1929–1958.

Tang, G., Müller, M., Rios, A., & Sennrich, R. (2018). Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4263–4272).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wang, S., & Jiang, J. (2016). A compare-aggregate model for matching text sequences. In *Proceedings of the 4-th International Conference on Learning Representations (ICLR-16),* San Juan, Puerto Rico.

Wang, W., Yang, N., Wei, F., Chang, B., & Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 189–198).

Zhou, Y.-T., & Chellappa, R. (1988). Computation of optical flow using a neural network. In *Ieee international conference on neural networks* (pp. 71–78).