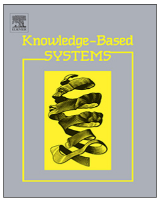
Contents lists available at ScienceDirect

Knowledge-Based Systems

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



DGI: Recognition of Textual Entailment via dynamic gate Matching[☆]

Xi Xiong [a,d,1], Yuanyuan Li [b,1], Rui Zhang [c], Zhan Bu [e], Guiqing Li [f], Shenggen Ju [c,*]

[a] *School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China*
[b] *West China School of Medicine, Sichuan University, Chengdu 610041, China*
[c] *College of Computer Science, Sichuan University, Chengdu 610065, China*
[d] *School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore*
[e] *College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China*
[f] *School of Management, Chengdu University of Information Technology, Chengdu 610103, China*

ARTICLE INFO

Article history:
Received 19 July 2019
Received in revised form 11 December 2019
Accepted 20 January 2020
Available online 1 February 2020

Keywords:
Textual entailment
Dynamic gate inference
Attention mechanism
Natural language processing
Long short-term memory

ABSTRACT

Recognizing Textual Entailment (RTE) is an integral part of intelligent machines which is able to understand and reason with natural languages. Some special embedding methods such as the attention mechanism exploit the semantic information without considering the features of sentence interaction, which also affect the word-level attention weight when a word appears at multiple positions of a sentence. In this study, we propose a **D**ynamic **G**ate **I**nference model (DGI) to fulfill the RTE task. In the DGI model, different aspects of semantic information are extracted from a premise sentence and a hypothesis sentence by a proposed dynamic *g*ate *Match*ing LSTM structure (*gMatch*), which combines the word-level fine-grained reasoning mechanism with the sentence-level gating structure to capture the global semantics. The textual relationship between the premise and the hypothesis is inferred by the three categories of attention including direct concatenation, similarity and difference. Extensive experiments were conducted to evaluate the performance of the proposed DGI model in two popular corpus by the metric of accuracy, and the results demonstrate that our approach outperforms the state-of-the-art baseline models in textual entailment in an effective manner.

1. Introduction

As a fundamentally important task of Natural Language Processing (NLP), Recognizing Textual Entailment (RTE) is an integral part of intelligent machines which is able to understand and reason with natural languages and applied to address many problems including question answering, semantic search and automatic text summarization. RTE is the task of determining the semantic relationship between two sentences, that is, a premise sentence S and a hypothesis sentence T [1]. The relationship is divided into three categories: (i) the premise entails the hypothesis, (ii) contradicting each other, and (iii) not related, which are illustrated by the following examples of sentence pairs with different relationship labels in Table 1. The first hypothesis can be inferred from the premise and their relationship is labeled as entailment. A phrase "*A cat*" is mentioned in the second hypothesis which is viewed as a contradiction. The third hypothesis is neutral to the premise because the phrase "with his owner" cannot be inferred from the premise.

1.1. Motivation and challenges

Early studies determined the textual entailment between two sentences by constructing different features manually. When Bowman et al. [2] released the Stanford Natural Language Inference (SNLI) corpus in 2015, it is feasible to design deep neural network (DNN) models which require a great amount of data for training. To the best of our knowledge, there are two major challenges that affect the performance of RTE.

(1) **Multiple features of the interaction between sentences.** Textual embedding provides a general sentence representation for a premise and a hypothesis [3], and represents their relationship by concatenating the two embedding vectors. However, for a specific RTE task, no interactive inference is considered between two sentences. Some special embedding methods such as the attention mechanism can exploit the semantic information effectively, but still ignore many features of textual interaction [4].

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105544.

[*] Corresponding author.
E-mail address: jsg@scu.edu.cn (S. Ju).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.knosys.2020.105544
0950-7051/© 2020 Elsevier B.V. All rights reserved.

Table 1
Examples of sentence pairs with different relationship labels.

	ID	Sentence	Label
Premise		A dog jumping for a frisbee in the snow.	
Hypothesis	Example 1	An animal is outside in the cold weather, playing with a plastic toy	Entailment
	Example 2	A cat washed his face and whiskers with his front paw.	Contradiction
	Example 3	A pet is enjoying a game of fetch with his owner.	Neutral

(2) **Dynamic relationship between sentences.** The relationship between a premise and a hypothesis can be inferred based on the sentence embeddings [5–9] and the sentence alignment. The attention-weighted representation of the premise [10] helps to improve the prediction accuracy. The alignment between a premise and a hypothesis happens when the two sentences have identical words. However, the opposite parts of two sentences should also be given a large attention. The word-level attention weight is also affected by different textual interaction when a word appears at multiple positions of a sentence, as shown in Example 1.

Example 1. Consider a premise "*Tony found an apple at the kitchen.*" and a hypothesis "*Bob remembered leaving the apple in the kitchen, but finally found it at the bedroom.*" "*apple*" and "*kitchen*" in the first part of the hypothesis have large attention weights because they also appear in the premise. However, the second part of the hypothesis denotes an opposite meaning and the attention is given to "*bedroom*".

1.2. Contributions

In order to predict textual entailment, it is necessary to extract word-level and sentence-level semantics from the interaction between sentences. As a state-of-the-art method, the *m*LSTM model [11] goes through the tokens of the hypothesis sequentially to predict whether the premise entails the hypothesis. Inspired by *m*LSTM, we propose a model that essentially sequentially aggregates the matching of the attention-weighted premise to each token of the hypothesis and uses the aggregated matching result to make a final prediction. In regard of the dynamic sentence characteristics, we adopt a novel gate-reasoning architecture to combine the sentence-level interaction information by the gating mechanism and finally capture the global semantic information which can infer the dynamic relationship between a premise and a hypothesis.

Textual entailment is mainly inferred by the concatenation of the two sentences [12]. Given the dynamic relationship between the premise and the hypothesis, we can also obtain the complex textual structure by calculating the similarity degree and the difference degree of the two sentences. These integrated semantic features of sentence relationships will improve the performance of RTE.

In this study, we make the following contributions:

(1) We propose a novel model called **D**ynamic **G**ate **I**nference model (DGI) to fulfill the RTE task. DGI extracts different aspects of semantic information from a premise sentence and a hypothesis sentence to infer the textual relationship between the sentence pair based on the three categories of attention including direct concatenation, similarity and difference.

(2) We propose a dynamic **g**ate **Match**ing LSTM structure (*gMatch*) to combine the word-level fine-grained reasoning with the sentence-level gating structure to capture the global semantics. The *gMatch* structure essentially sequentially aggregates the matching of the attention-weighted premise to each token of the hypothesis and uses the aggregated matching result to make a final prediction. In regard of the dynamic sentence characteristics, we adopt a novel gate-reasoning architecture to combine the sentence-level interaction information by the gating mechanism and finally capture the global semantic information which can infer the dynamical relationship between a premise and a hypothesis.

(3) We conduct extensive experiments to evaluate the performance of the proposed DGI model in two popular corpus by the metric of accuracy, and the results demonstrate that our approach outperforms the state-of-the-art baseline models in textual entailment in an effective fashion.

The remainder of this paper is organized as follows: Section 2 surveys the related works. Section 3 lists the notations used throughout this paper and formulate the problem. Section 4 details our DGI model with the *gMatch* structure. The experimental results are presented in Section 5. Lastly, we conclude this study in Section 6.

2. Related works

Much attention have been paid to the RTE task, from the conventional solutions to the popular end-to-end neural network models.

Conventional solutions. Conventional solutions to the RTE task rely on NLP pipelines that involve multiple steps of linguistic analyses and feature engineering, including syntactic parsing, named entity recognition and sentence classification, etc. Glickman et al. [13] formalized the concept of textual entailment by introducing a universal probabilistic setting, then proposed a textual entailment model based on word alignment and document co-occurrence probabilities. Jijkoun [14] proposed an approach based on the new metric of directed sentence similarity, equal to the directed word overlap between the premise and the hypothesis, then combined the metric with other two lexical similarity measurements. Marneffe et al. [15] introduced a strict definition of contradiction for NLP tasks and presented a set of available contradictions, from which a typology of contradictions are created. The approach determines the entailment relationship based on an alignment-based similarity instead of the direct similarity. Bar-Haim et al. [16] proposed a general semantic inference structure based on syntactic trees, which are constructed by entailment rules and provide a unified way to represent various kinds of inferences. Rules are generated manually and automatically based on general linguistic structures and specific lexical inferences.

End-to-end neural network models. RTE has gained much attention in recent years and an increasing number of researchers have built data-drive, end-to-end neural network models for fulfilling the task [17]. Parikh et al. [18] proposed a simple neural network for natural language inference. By means of attention mechanism, the problem is decomposed into multiple subproblems that can be solved separately. The approach is also trivially parallelizable. Tay et al. [19] proposed an architecture in which compressed alignment features are propagated to upper encoders for enhancing representation. Each alignment vector is reduced to a scalar feature to facilitate the propagation of alignment features. CAFE has been proved to be conceptually simple, compact and effective. Khot et al. [20] proposed a textual entailment model that only leverages the structure of the hypothesis instead of extracting the structure of the premise which is much longer and harder to parse. The model aims to find words in the premise which can validate the hypothesis structure. Shen et al. [21] proposed a light-weight neural network that encodes sentence only based on the attention without any recursive or convolutional structure. It only consists of a directional temporal self-attention block followed by a multi-dimensional attention which converts the textual sequence into a vector. Cheng et al. [22] proposed a simulator that reads texts in sequence and conducts shallow reasoning with memory and attention. The simulator extends the LSTM structure with a memory network instead of a single memory cell, which enables adaptive utilization of memory in recurrence with neural attention and capture the subtle relationships between tokens. Choi et al. [23] proposed a new tree-structured LSTM structure that learns to construct task-relevant tree structures just from plain text data effectively.

Multiple features of sentence interaction. The structural information in sentences is also important to RTE. Many approaches were proposed to effectively capture relationships between two sentences, but keep low complexity. Chen et al. [4] proposed a sequential inference model that infers contextual information by explicitly considering recursive structures in both local and global scopes. A distance-sensitive intra-sentence attention can significantly improve the performance of the language inference task. Wang et al. [11] focuses on significant word-level matching of the hypothesis with the premise. Some critical mismatches are stored in memory and applied to the prediction for the relationship label of contradiction or neutral. Kang et al. [24] proposed a knowledge-based adversarial example generator for the RTE purpose by incorporating large lexical resources in textual entailment models with only a small number of rule templates. The generator is learned by a Generative Adversarial Network (GAN) style framework according to the discriminator's performance on the generated examples.

The interaction features between sentences can be represented by an attention matrix. Unreliable embeddings and attention results lead to a significant error of interaction features. The above literatures based on word-level fine-grained features do not integrate global semantic information for textual inference and cannot capture the interaction features from multiple aspects such as the similarity and the difference between sentences.

3. Problem formulation

In this section, we first present important notations, and then formalize the RTE problem. To facilitate understanding, Table 2 describes the important notations used throughout this paper.

Table 2
Notations and their descriptions.

Symbol	Description
$\mathbf{X}_s$, $\mathbf{X}_t$	The premise sentence and the hypothesis sentence
$\mathbf{S}$, $\mathbf{T}$	The embedding vectors of the premise and the hypothesis
$\vec{s}_i$, $\vec{t}_i$	The ith column vector in $\mathbf{H}^s$ ($\mathbf{H}^t$)
$\vec{h}^s_j$	The word-level attention-weighted representation of the premise corresponding to the jth word of the hypothesis
$\vec{h}^{s'}_j$	The sentence-level attention-weighted representation of the premise corresponding to the jth word of the hypothesis
$\vec{h}_j$	The jth hidden vector of *gMatch*
$\vec{t}_a$	The self-attention of the hypothesis
$\mathbf{W}^x$	Matrix parameters where $x \in \{s, t, r, p, h\}$
$\vec{b}^s$, $\vec{b}^v$	Vector parameters
b	A scalar parameter
$\vec{\alpha}_j$	The attention weight vector of the premise corresponding to position j of the hypothesis
α_{kj}	The kth element of $\vec{\alpha}_j$, representing the degree to which the jth word in the hypothesis is aligned with the kth word in the premise.

Problem statement. For the task of recognizing textual entailment, we have a premise sentence $X^s = (x^s_1, x^s_2, \ldots, x^s_{l_s})$ with l_s words and a hypothesis sentence $X^t = (x^t_1, x^t_2, \ldots, x^t_{l_t})$ with l_t words. Here each x is the embedding vector of a corresponding word. The goal is to predict a label y that indicates the relationship between X^s and X^t. In this paper, we assume y is one of *entailment*(E), *contradiction*(C) and *neutral*(N). Then the problem can be formulated as follows:

$$X^s \overset{y}{\longleftrightarrow} X^t \Longrightarrow y \in \{E, C, N\} \tag{1}$$

4. Method

In this section, we propose a **D**ynamic **G**ate **I**nference model (DGI) to fulfill the RTE task. In the DGI model, different aspects of semantic information are extracted from a premise sentence and a hypothesis sentence by a proposed dynamic **g**ate **Match**ing LSTM structure (*gMatch*), which combine the word-level fine-grained reasoning mechanism with the sentence-level gating structure to capture the global semantics. The textual relationship between the premise and the hypothesis is inferred by the three categories of attention including direct concatenation, similarity and difference. Fig. 1 shows an overview of the DGI model, which consists of three layers: (1) the encoding layer represents each word of the two sentences and incorporates the contextual information into the representation of each word token; (2) the gate matching layer sequentially goes through the hypothesis and infers the relationship between the two sentences dynamically based on the *gMatch* structure; (3) the output layer provides a label to indicate the relationship between the pair of sentences.

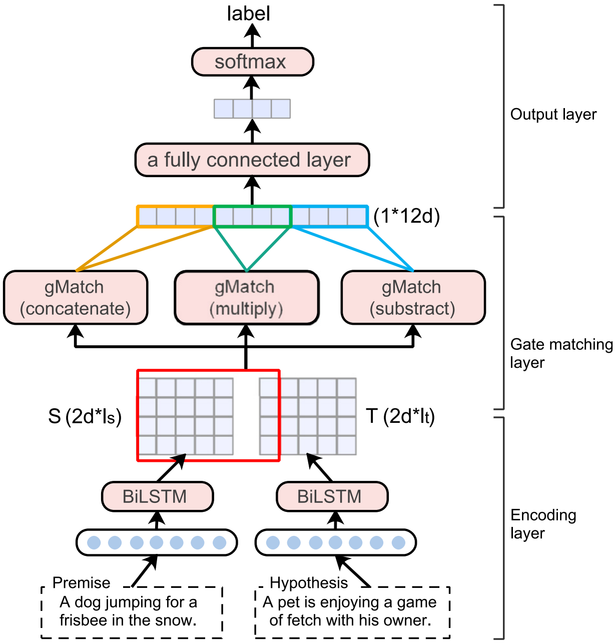


Fig. 1. Overview of the DGI model.

4.1. Encoding layer

The word embeddings of the two sentences are trained by the GloVe method [25] instead of word2vec [26], because GloVe covers more words in the SNLI corpus than word2vec. Words

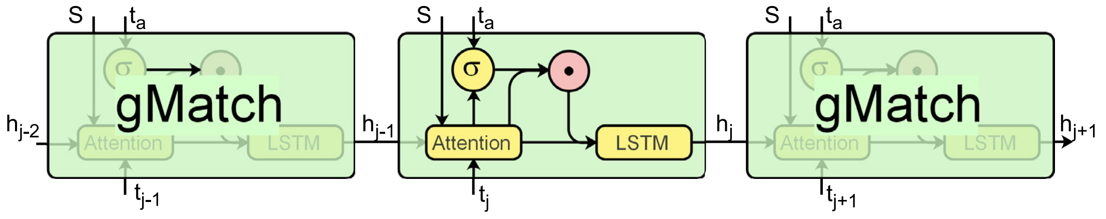


Fig. 2. Architecture of *gMatch*.

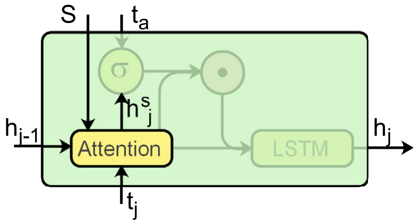


Fig. 3. Attention structure in *gMatch*.

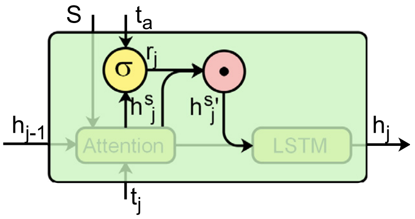


Fig. 4. Sentence-level gating mechanism in *gMatch*.

that are not found in GloVe are initialized as zero vectors. The encoding layer incorporates contextual information into the representation of each token in the premise and the hypothesis by the bi-directional LSTM (BiLSTM), as shown below:

$$\mathbf{S} = \text{BiLSTM}(\mathbf{X}^s) \tag{2}$$

$$\mathbf{T} = \text{BiLSTM}(\mathbf{X}^t) \tag{3}$$

The resulting matrices $\mathbf{S} \in \mathbb{R}^{2d \times l_s}$ and $\mathbf{T} \in \mathbb{R}^{2d \times l_t}$ are hidden representations of the premise and the hypothesis, where $2d$ is the dimension of the hidden vectors, that is, the ith column vector $\vec{s}_i$ (or $\vec{t}_i$) in $\mathbf{S}$ (or $\mathbf{T}$) represents the ith token in the premise (or the hypothesis) accompanied by some contextual information from both directions. Moreover, the self-attention of the hypothesis is denoted by the vector $\vec{t}_a \in \mathbb{R}^{2d}$ which is computed by the dot-product attention structure with both inputs from the hypothesis.

$$\vec{t}_a = \text{Self-Attention}(\mathbf{T}) \tag{4}$$

4.2. Gate matching layer

The gate matching layer sequentially goes through the premise and infers the relationship between premise and hypothesis dynamically based on the *gMatch* (*gateMatch*) structure, which is shown in Fig. 2.

The *gMatch* Structure: The *gMatch* structure adopts a novel gate-reasoning architecture to combine the sentence-level interaction information by means of the gating mechanism and finally capture the global semantic information which can infer the dynamical relationship between a premise and a hypothesis. The *gMatch* structure involves three parts.

(1) The attention weight vector $\vec{\alpha}_j \in \mathbb{R}^{l_s}$ at position j of the hypothesis is obtained by the standard word-by-word attention mechanism, as shown in Fig. 3. Each element $\alpha_{kj} \in \vec{\alpha}_j$ is an attention weight that encodes the degree to which the jth word in the hypothesis is aligned with the kth word in the premise. The attention can be calculated as the aggregated representation of all the words in the premise and a particular word in the hypothesis according to the following equations:

$$\mathbf{G}_j = \tanh(\mathbf{W}^s\mathbf{S} + (\mathbf{W}^t\vec{t}_j + \mathbf{W}^r\vec{h}_{j-1} + \vec{b}^s) \otimes e_{l_s}) \tag{5}$$

$$\vec{\alpha}_j^T = \text{softmax}(\vec{w}^T\mathbf{G}_j + b \otimes e_{l_s}) \tag{6}$$

$$\vec{h}_j^s = \mathbf{S} \cdot \vec{\alpha}_j \tag{7}$$

where $\mathbf{W}^s$, $\mathbf{W}^t$, $\mathbf{W}^r \in \mathbb{R}^{d \times 2d}$, $\vec{b}^s$, $\vec{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are the parameters to be learned. $\vec{h}_{j-1} \in \mathbb{R}^d$ is the (j-1)th hidden vector of *gMatch*, and the outer product ($\otimes e_{l_s}$) produces a matrix or a vector by repeating the vector or scalar on the left for l_s times. $\mathbf{G}_j \in \mathbb{R}^{d \times l_p}$ aggregates all the previous $\vec{h}_{j-1}$. $\vec{h}_j^s \in \mathbb{R}^{2d}$ can be seen as an attention-weighted representation of the premise corresponding to the jth word of the hypothesis.

(2) The premise and the hypothesis are inferred and aligned by word-level semantic information, which captures the local features. The premise is represented by the word-level attention related to the specific word in the hypothesis. However, two words with similar semantics may lead to irrelevant even opposite meanings in the local or global scope, vice versa. Therefore, we introduce the sentence-level gating mechanism to select the contextual information. As shown in Fig. 4, the sentence-level representation of the premise is computed as follows:

$$\vec{r}_j = \sigma(\mathbf{W}^p\vec{h}_j^s + \mathbf{W}^h\vec{t}_a + \vec{b}_v) \tag{8}$$

$$\vec{h}_j^{s'} = \vec{h}_j^s \odot \vec{r}_j \tag{9}$$

where $\mathbf{W}^p$, $\mathbf{W}^h \in \mathbb{R}^{2d \times 2d}$, $\vec{b}_v \in \mathbb{R}^{2d}$ are the parameters to be learned. $\vec{h}_j^s$ is the word-level attention-weighted representation of the premise, and $\vec{t}_a \in \mathbb{R}^{2d}$ is the weighted self-attention of the hypothesis, which captures the sentence-level information. $\odot$ is the element-wise multiplication of two vectors or two matrices. Then $\vec{r}_j \in \mathbb{R}^{2d}$ is the gate output representing the weighted result of the two attention vectors by the sigmoid function according to Eq. (8). $\vec{h}_j^{s'} \in \mathbb{R}^{2d}$ represents the sentence-level hypothesis-related part of the premise which is selected by the gate $\vec{r}_j$.

(3) We define $\vec{m}_j^{conc} \in \mathbb{R}^{4d}$ as the concatenation of $\vec{h}_j^{s'}$ and $\vec{h}_j^s$, which denote the sentence-level and word-level representation of the premise corresponding to the jth word of the hypothesis, respectively.

$$\vec{m}_j^{conc} = [\vec{h}_j^{s'}, \vec{h}_j^s] \tag{10}$$

As shown in Fig. 5, a LSTM structure is adopted to process a sequence of data and capture the entire contextual features of the two texts. Eq. (11) illustrates the LSTM structure in *gMatch*.

$$\vec{h}_j = \text{LSTM}(\vec{m}_j^{conc}, \vec{h}_{j-1}) \tag{11}$$

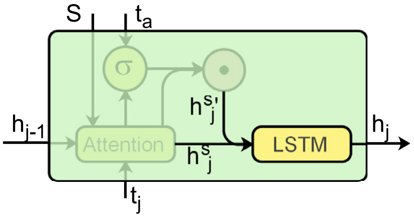


Fig. 5. LSTM structure in *gMatch*.

where $\vec{h}_j \in \mathbb{R}^d$ is the jth hidden state vector and $\vec{m}_j^{conc}$ is the input. The *gMatch* structure is shown as Eqs. (5)–(11). The output $\vec{h}_j$ in Eq. (11) is regarded as the next input of Eq. (5).

Besides the concatenation of the two sentences, their dynamic relationship can be inferred from their similarity and difference, which represent the semantics of sentence interaction via multiple categories of attention. As one of the simplest methods to calculate the similarity between two vectors, the element-wise multiplication captures the similarity information of two sentences, and the subtraction can directly infer their difference.

The similarity attention can be calculated by replacing the plus signs in Eqs. (5) and (8) with element-wise multiplication signs, and we have the following equations:

$$\mathbf{G}_j^{sim} = \tanh(\mathbf{W}^s\mathbf{S} \odot (\mathbf{W}^t\vec{t}_j + \mathbf{W}^r\vec{h}_{j-1} + \vec{b}^s) \otimes e_{l_s}) \tag{12}$$

$$\vec{r}_j^{sim} = \sigma(\mathbf{W}^p\vec{h}_j^s \odot \mathbf{W}^h\vec{t}_a + \vec{b}_v) \tag{13}$$

$$\vec{m}_j^{sim} = \vec{h}_j^{s'} \odot \vec{h}_j^s \tag{14}$$

where $\vec{r}_j^{sim} \in \mathbb{R}^{2d}$ is the gate output representing the weighted result by multiplying the two attention vectors. $\vec{m}_j^{sim} \in \mathbb{R}^{4d}$ denotes the sentence-level and word-level representation of the premise corresponding to the jth word of the hypothesis in the similarity aspect.

The difference attention can be calculated by replacing the plus signs in Eqs. (5) and (8) with subtraction signs, and we have the following equations:

$$\mathbf{G}_j^{dif} = \tanh(\mathbf{W}^s\mathbf{S} - (\mathbf{W}^t\vec{t}_j + \mathbf{W}^r\vec{h}_{j-1} + \vec{b}^s) \otimes e_{l_s}) \tag{15}$$

$$\vec{r}_j^{dif} = \sigma(\mathbf{W}^p\vec{h}_j^s - \mathbf{W}^h\vec{t}_a + \vec{b}_v) \tag{16}$$

$$\vec{m}_j^{dif} = \vec{h}_j^{s'} - \vec{h}_j^s \tag{17}$$

where $\vec{r}_j^{dif} \in \mathbb{R}^{2d}$ is the gate output representing the weighted result by the subtraction of the two attention vectors. $\vec{m}_j^{dif} \in \mathbb{R}^{4d}$ denotes the sentence-level and word-level representation of the premise corresponding to the jth word of the hypothesis in the difference aspect.

As the key component of the DGI model, the *gMatch* structure combines three aspects of the sentence pair and dynamically infers word-level and sentence-level information simultaneously to capture the multiple relationships between sentences.

4.3. Output layer

This layer first concatenates the three integrated outputs as given below:

$$\vec{m}_j = [\vec{m}_j^{conc}, \vec{m}_j^{sim}, \vec{m}_j^{dif}] \tag{18}$$

where $\vec{m}_j \in \mathbb{R}^{12d}$ represents the premise vector corresponding to the jth word of the hypothesis in the multiple aspects. Then a fully connected layer with an activation function of *tanh* is adopted for dimension reduction and outputs a $2d$-dimensional vector. The DGI model finally obtains a classification label by means of *softmax*. The whole model is trained end-to-end with a loss function of cross entropy.

Table 3
Statistical features of the datasets.

	SNLI	SciTail
Number of examples	570k	27k
Size of the training set	549 367	23 596
Size of the validation set	9842	1304
Size of the testing set	9824	2126
Labels	*N*, *E*, *C*	*N*, *E*

4.4. Computational complexity

In this section, we discuss the time complexity of our DGI approach based on the dynamic sentence-level gating mechanism. The complexity of the GloVe embedding is $\mathcal{O}(V^2)$, where V represents the size of the vocabulary. Given the embedding dimension d and the sentence length l, the complexity of an LSTM cell is $\mathcal{O}(d^2)$, resulting in a complexity of $\mathcal{O}(d^2 * l)$ to encode the sentence. Therefore, the encoding layer has the complexity of $\mathcal{O}(d^2 * l)$. In the gate matching layer, each kind of attention requires $\mathcal{O}(d^2)$ operations, and the three kinds of attention can be computed in parallel and the complexity of the gate matching layer is also $\mathcal{O}(d^2)$. Thus the entire complexity of the model is $\mathcal{O}(d^2 * l)$, equal to that of an LSTM with attention.

5. Experiments

5.1. Experimental settings

In this section, a series of experiments are conducted to evaluate the performance of our proposed DGI model on two datasets. We will introduce the datasets, the state-of-the-art baseline approaches and the parameter settings for experiments, then demonstrate its effectiveness and efficiency by comparing it with the baselines. In the experiments, we adopt classification accuracy as the evaluation metric, as in related works.

5.1.1. Datasets

In order to show the performance of the proposed DGI model, the following two datasets are used in the experiments:

SNLI (Stanford Natural Language Inference)[2] [2] is a baseline corpus to evaluate learning-centered methods such as deep neural networks for natural language inference (NLI). The original dataset contains 570,152 sentence pairs, each labeled with one of the following relationships: *entailment*, *contradiction*, *neutral* and –, where – indicates a lack of consensus from the human annotators. We discard the sentence pairs labeled with – and keep the remaining ones for our experiments. We perform three-class classification and use accuracy as our evaluation metric.

SciTail[3] [20] is the first corpus for binary textual entailment task which is constructed only from natural sentences rather than man-made sentences. Unlike other datasets, hypotheses were created from science questions and the corresponding answer candidates, and premises were created from relevant web sentences retrieved from a large corpus. This dataset contains 27k premise-hypothesis pairs, each of which is labeled as support or not.

The statistical features of the datasets are displayed in Table 3, where *N*, *E* and *C* stand for neutral, entailment and contradiction, respectively.

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

[3] http://data.allenai.org/scitail.

5.1.2. Implementation details

We first tokenize all the premises and hypotheses. Word embeddings are initialized with 300-dimensional GloVe embeddings [25]. For the out-of-vocabulary (OOV) words, we initialize the word embeddings randomly following a Gaussian distribution. The word embeddings are not updated during the training of the model. We use ADAMAX [27] with the coefficients $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize the model. Each update is computed through a minibatch of 30 instances. The hidden size is set as 100 for all BiLSTM layers. L2 regularization is set to 10^{-6}. Dropout [28] with a probability of 0.8 is applied after each fully connected or recurrent layer. All parameters are initialized with Xavier initialization [29] which aims to acquire the variances of the gradients on the weights are the same for all layers.

5.1.3. Baselines

We compare our proposed RTE model, i.e., DGI, with the following state-of-the-art natural language inference techniques.

ESIM (Enhanced Sentence Inference Model) [4] is a sequential inference model that infers contextual information by explicitly considering recursive structures in both local and global scopes. A distance-sensitive intra-sentence attention can significantly improve the performance of the language inference task.

Decomposed Attention [18] is a simple neural network for natural language inference. By means of attention mechanism, the problem is decomposed into multiple subproblems that can be solved separately. The approach is also trivially parallelizable.

DGEM (Decomposed Graph Entailment Model) [20] is a textual entailment model that only leverages the linguistic structure of the hypothesis instead of extracting the structure of the premise. The model based on the graph structure of the hypothesis aims to find words in the premise which can validate the hypothesis structure.

AdvEntuRe [24] is a knowledge-based adversarial example generator for the RTE purpose by incorporating large lexical resources in textual entailment models with only a small number of rule templates. The generator is learned by a Generative Adversarial Network (GAN) style framework according to the discriminator's performance on the generated examples.

DiSAN (Directional Self-Attention Network) [21] is a lightweight neural network that encodes sentence only based on the attention without any recursive or convolutional structure. It only consists of a directional temporal self-attention block followed by a multi-dimensional attention which converts the textual sequence into a vector.

LSTMN (Long Short-Term Memory-Network) [22] is a simulator that reads texts in sequence and conducts shallow reasoning with memory and attention. The simulator extends the LSTM structure with a memory network instead of a single memory cell, which enables adaptive utilization of memory in recurrence with neural attention and capture the subtle relationships between tokens.

Gumbel Tree-LSTM [23] is a new tree-structured LSTM structure that learns to construct task-relevant tree structures just from plain text data effectively.

*m*LSTM [11] focuses on significant word-level matching of the hypothesis with the premise. Some critical mismatches are stored in memory and applied to the prediction for the relationship label of contradiction or neutral.

CAFE (ComProp Alignment-Factorized Encoders) [19] adopted an architecture in which compressed alignment features are propagated to upper encoders for enhancing representation. Each alignment vector is reduced to a scalar feature to facilitate the propagation of alignment features. CAFE has been proved to be conceptually simple, compact and effective.

The differences of the above natural language inference methods are shown in Table 4, where the first line refers to several features including: (1) using attention or not (denoted by Attention), (2) the type of LSTM (denoted by LSTM, and chosen from LSTM, BiLSTM and TreeLSTM), (3) analyzing inter-sentence syntax or not (denoted by inter-sent.), (4) embedding method (denoted by embedding) as well as (5) special techniques.

It is not realistic to implement all these methods. We utilize the experimental results extracted from the existing references. The results on the SNLI corpus are listed in the reference [30], and the results on the SciTail corpus are listed in the references [20] and [30]. Therefore, different parts of the baselines in Table 4 are employed for experiments on the two different corpora.

5.2. Accuracy comparison

The experimental results on the SNLI corpus is shown in Table 5. In order to report the performance of our proposed DGI model effectively, we use single models rather than stacked models. As shown in Table 5, the performance of our proposed DGI model is competitive with a 88.9% accuracy in the testing set. Competitive results can also be achieved with a much smaller parameterization. For example, DGI achieves a 88.2% accuracy with only 550k parameters. This outperforms the state-of-the-art CAFE model with only a fraction of the parameter cost. At 300 dimensions, our DGI model has about three times less parameters than CAFE, i.e., 1.9M versus 4.7M. Moreover, our lightweight adaptation achieves 88.2% with only 550k parameters, which makes it well-performing among models with the same number of parameters such as the decomposable attention model (86.8%) and the CAFE model (87.7%).

Table 6 reports our results on the SciTail dataset. Our DGI model slightly outperforms CAFE on both settings. The accuracy gap between DGI and CAFE is 0.5%-1.4% with a fraction of parameter cost for DGI (3.5M versus 550k). Some effective models such as Decomposed Attention, ESIM and CAFE fulfill the RTE task by word-level attention, but their performance on the challenging SciTail dataset is unsatisfactory, which indicates that complicated textual relationships cannot be inferred only by means of word-level local attention. The dynamic inference mechanism in DGI is proved to be effective.

As such, experimental results demonstrate the effectiveness of our proposed DGI model on the two datasets. We believe that the good performance of DGI can be attributed to the following reasons: (1) DGI essentially sequentially aggregates the matching of the attention-weighted premise to each token of the hypothesis and uses the aggregated matching result to make a final prediction. (2) In regard of the dynamic sentence characteristics, a novel gate-reasoning architecture combines the sentence-level interaction information by the gating mechanism. (3) The dynamical inference mechanism integrates three aspects of the sentence pair and dynamically infers word-level and sentence-level information simultaneously to finally capture the global semantic information.

5.3. Ablation analysis

In this section, we make the ablation study on the SciTail dataset and show the results in Table 7, where "2. DGI-Conc", "3. DGI-Sim" and "4. DGI-Dif" represent removing the concatenation attention, the similarity attention and the difference attention, respectively. "5. DGI-Sim-Dif" and "6. DGI-3Att" represent removing the two types of attention and all the three types of attention, respectively. "7. DGI-Gate" represents removing the sentence-level gate. "8. DGI-Gate-3Att" represents removing all the three types of attention and the gate, simultaneously, and the model is converted to the *m*LSTM model. "9. DGI-Gate-3Att-Dyn (LSTM)"

Table 4
Accuracy comparison with state-of-the-art published models on the SNLI corpus.

Model	Attention	LSTM	Inter-sent.	Embedding	Special technique
Gumbel Tree-LSTM [23]	Y	Tree	Y	GloVe	–
DiSAN [21]	Y	N	N	GloVe	Self-attention block
LSTMN+DeepAtt [22]	Y	LSTM	N	GloVe	A memory network
*m*LSTM [11]	Y	Bi	N	GloVe	Word-by-word matching
DGEM [20]	Y	LSTM	N	GloVe	–
ESIM [4]	Y	Bi & Tree	Y	GloVe	Lightweight
Word-by-word Att [1]	Y	LSTM	N	word2vec	Reasoning over words and phrases
Ngram [19]	N	–	–	–	–
DecompAtt [18]	Y	LSTM	N	GloVe	Problem factorization
AdvEntuRe [24]	Y	LSTM	N	Retrofitting	GAN
CAFE [19]	Y	LSTM	Y	GloVe	Problem factorization
DGI (Ours)	**Y**	**Bi**	**Y**	GloVe	**Multiple types of attention**

Table 5
Accuracy comparison with the baselines on the SNLI corpus.

	Dim.	Params	Validat. (%)	Test (%)
Gumbel Tree	300	2.9M	91.2	85.6
	600	10M	93.1	86.0
DiSAN	300	2.4M	91.1	85.6
LSTMN	450	3.4M	88.5	86.3
*m*LSTM	300	1.9M	92.0	86.1
DecompAtt	200	580k	90.5	86.8
ESIM	600	4.3k	92.6	88.0
W-by-W Att	300	3.9M	85.3	83.5
CAFE	150	750k	88.2	87.7
	300	4.7M	89.8	88.5
DGI (Ours)	**150**	**550k**	**90.7**	**88.2**
	300	**1.9M**	**91.5**	**88.9**

Table 6
Accuracy comparison with the baselines on the SciTail corpus.

	Dim.	Params	Validat. (%)	Test (%)
DGEM	300	112k	79.6	77.3
ESIM	450	4.3M	70.5	70.6
Ngram	–	–	65.0	70.6
DecompAtt	200	341k	75.4	72.3
AdvEntuRe	–	–	–	77.8
CAFE	300	3.5M	–	83.3
DGI (Ours)	**150**	**550k**	**84.6**	**83.8**
	300	**1.9M**	**85.7**	**84.7**

Table 7
Ablation study on the SciTail corpus.

No	Model	Accuracy (%)
1	DGI	84.7
2	DGI-Conc	82.7
3	DGI-Sim	83.5
4	DGI-Dif	83.8
5	DGI-Sim-Dif	82.9
6	DGI-3Att	81.6
7	DGI-Gate	81.3
8	DGI-Gate-3Att (*m*LSTM)	80.5
9	DGI-Gate-3Att-Dyn (LSTM)	72.1

represents that without the *gMatch* structure including the dynamic inference, the DGI model is converted to a conventional LSTM model.

We observe from Table 7 that the three types of attention and the sentence-level gating mechanism improve the effectiveness of RTE. According to the significance of these three types of attention (No.2–4), it is interesting to find that these types of attention are ranked as concatenation>similarity>difference. We can conclude that the concatenation attention is more significant for RTE than other types of attention. Moreover, the accuracy of DGI declines from 84.7% to 81.3% when removing the sentence-level gating mechanism which is proved to be the most important factor.

5.4. Matching performance

In order to obtain a better understanding of how our proposed DGI model actually performs the matching between a premise and a hypothesis, we examine the learned word-by-word alignment weight α_{kj} in Eq. (6), which denotes the degree to which the jth word in the hypothesis is aligned with the kth word in the premise. To check whether the alignment makes sense, Figs. 6 and 7 show the values of α_{kj} in *m*LSTM and DGI, where a darker color represents a larger value. From the three subfigures in Fig. 6, the word "animal" is strongly aligned with "dog", and the phrase "cold weather" is aligned with "snow". In a pair of strong contradictory sentences like Fig. 6(b), "cat" is strongly aligned with "dog" and "washes" is aligned with "jumping". Some words in the hypothesis cannot be aligned with any word in the premise and is therefore aligned with the *NULL* token which is inserted at the end of each premise.

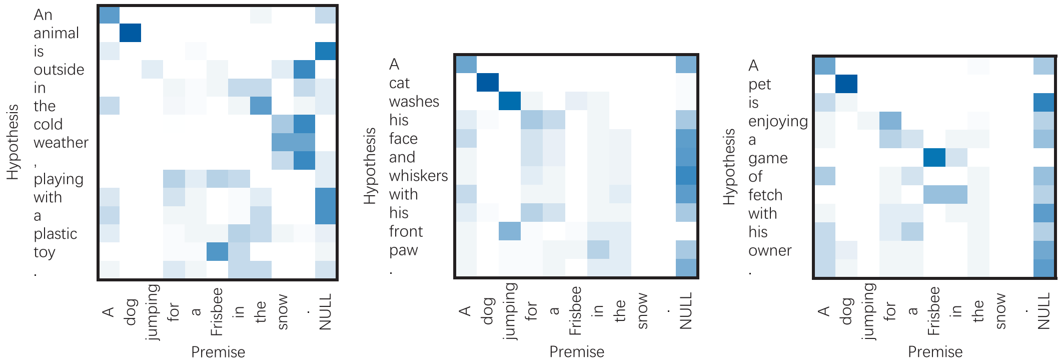
Fig. 7 shows the alignment results of the two sentences by the DGI model. We observe that α_{kj} of DGI is distributed more evenly than that of *m*LSTM. For example, in Fig. 7(a), the word "frisbee" is aligned with "cold". In Fig. 7(c), the word "snow" is aligned with "game". This can be explained by the reason that DGI can capture multiple categories of textual relationships between a premise and a hypothesis including direct concatenation, similarity and difference, which help to incorporate more information between these two sentences. Moreover, the *gMatch* structure combines the word-level fine-grained reasoning with the sentence-level gating structure to capture the global semantics, which is missing in *m*LSTM.

5.5. Example analysis

In this section, we analyze some examples of sentence pairs from the SciTail dataset, as shown in Table 8. *m*LSTM is equivalent to the DGI model without the gating mechanism and the three types of attention. From Example 1 and Example 2 in Table 8, we observe that the word-level inference tends to align the words in the premise and the hypothesis. The relationship of the sentence pair is more probably regarded as entailment instead of neutral when more identical words are found between the two sentences. The sentence pairs in Example 3 and Example 4 are long and have complex semantics. The *m*LSTM model recognize them as the wrong relationship for the reason that only word-level information is involved. The DGI model can infer the relationships correctly by multiple kinds of attention.

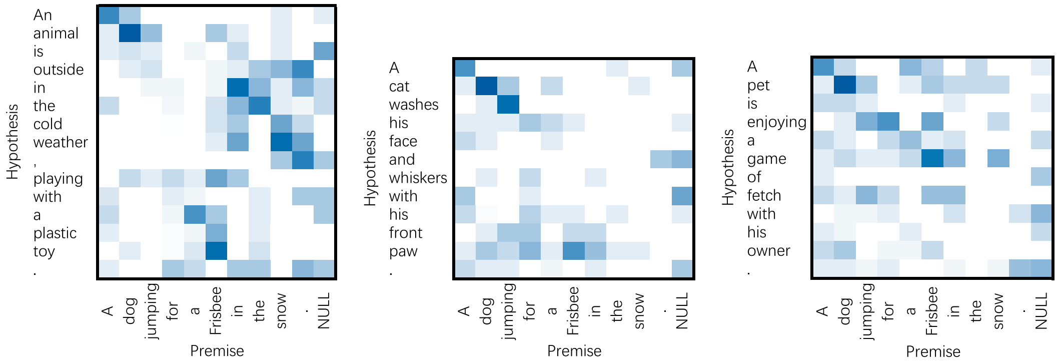
6. Conclusion

In order to overcome two major challenges of the textual entailment task including: (1) multiple features of the interaction between sentences, and (2) dynamic relationship between sentences, we propose a dynamic gate inference model based on the



(a) Entailment (b) Contradiction (c) Neutral

Fig. 6. Alignment weights of the three examples by *m*LSTM.



(a) Entailment (b) Contradiction (c) Neutral

Fig. 7. Alignment weights of the three examples by DGI.

Table 8
Example analysis.

No.	Sentence	Truth	DGI	*m*LSTM
P1 H1	The pupil is the opening through which light enters the eye. The pupil of the eye allows light to enter.	E	E	E
P2 H2	This can be dangerous to both plants and animals. Nematodes can be a parasite of both.	N	N	N
P3 H3	The angiosperms, or flowering plants, are all members of the phylum. Angiosperms are the most successful phylum of plants.	N	N	E
P4 H4	The angiosperms, or flowering plants, are all members of the phylum. A bee will sometimes do a dance to tell other bees in the hive where to find food.	E	E	N

multiple categories of attention involving direct concatenation, similarity and difference. In the proposed model, different aspects of semantic information are extracted from a premise sentence and a hypothesis sentence by a proposed dynamic gate matching LSTM structure, which combines the word-level fine-grained reasoning mechanism with the sentence-level gating structure to capture the global semantics. Compared with state-of-the-art baselines, our proposed DGI model achieves the best performance on the SNLI and SciTail datasets. An inevitable limitation of our model is the unsatisfactory representation of knowledge in different scenarios which may create contradiction. Therefore, refining the scenarios and the contexts can improve the performance in the feature.

CRediT authorship contribution statement

Xi Xiong: Methodology, Investigation, Writing - original draft. **Yuanyuan Li:** Methodology, Writing - review & editing. **Rui Zhang:** Data curation, Software, Validation. **Zhan Bu:** Writing - review & editing. **Guiqing Li:** Funding acquisition. **Shenggen Ju:** Project administration.

Acknowledgments

The work was partially supported by the National Natural Science Foundation of China under Grant No. 81901389; the National Social Science Fund of China under Grant No. 19BGL123; the China Postdoctoral Science Foundation under Grant No. 2019M653400; the Youth Foundation for Humanities and Social Sciences of Ministry of Education of China under Grant No. 17YJCZH202; the Sichuan Science and Technology Program under Grant Nos. 2018GZ0253, 2019YFS0236, 2018GZ0182, 2018GZ0093 and 2018GZDZX0039.

References

[1] T. Rocktaschel, E. Grefenstette, K.M. Hermann, T.K. Isk, P. Blunsom, Reasoning about entailment with neural attention, in: International Conference on Learning Representations, 2016, pp. 1–9.

[2] S.R. Bowman, G. Angeli, C. Potts, C.D. Manning, A large annotated corpus for learning natural language inference, in: Empirical Methods in Natural Language Processing, 2015, pp. 632–642.

[3] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, 2017, pp. 1–12, arXiv preprint arXiv:1705.02364.

[4] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, D. Inkpen, Enhanced LSTM for natural language inference, in: Meeting of Association for Computational Linguistics, Vol. 1, 2017, pp. 1657–1668.

[5] X. Xiong, S. Qiao, Y. Li, H. Zhang, P. Huang, N. Han, R.-H. Li, ADPDF: A hybrid attribute discrimination method for psychometric data with fuzziness, IEEE Trans. Syst. Man Cybern.: Syst. 49 (1) (2019) 265–278.

[6] X. Xiong, S. Qiao, Y. Li, F. Xiong, L. He, N. Han, Affective impression: Sentiment-awareness POI suggestion via embedding in heterogeneous LBSNs, IEEE Trans. Affect. Comput. (2019) 1.

[7] X. Xiong, S. Qiao, Y. Li, N. Han, G. Yuan, Y. Zhang, A point-of-interest suggestion algorithm in multi-source geo-social networks, Eng. Appl. Artif. Intell. 88 (2020) 103374.

[8] X. Xiong, S. Qiao, N. Han, F. Xiong, Z. Bu, R.-H. Li, K. Yue, G. Yuan, Where to go: An effective point-of-interest recommendation framework for heterogeneous social networks, Neurocomputing 373 (2020) 56–69.

[9] Z. Li, F. Xiong, X. Wang, H. Chen, X. Xiong, Topological influence-aware recommendation on social networks, Complexity 2019 (2019) 6325654.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Neural Inf. Process. Syst. (2017) 5998–6008.

[11] S. Wang, J. Jiang, Learning natural language inference with LSTM, in: Proceedings of the Conference on the North American Chapter of the Association for Computational Linguistics, 2016, pp. 1442–1451.

[12] Z. Wang, W. Hamza, R. Florian, Bilateral multi-perspective matching for natural language sentences, in: International Joint Conference on Artificial Intelligence, 2017, pp. 4144–4150.

[13] O. Glickman, I. Dagan, M. Koppel, A lexical alignment model for probabilistic textual entailment, 2005, pp. 287–298.

[14] V. Jijkoun, M. Rijke, Recognizing textual entailment using lexical similarity, in: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, Citeseer, 2005, pp. 73–76.

[15] M.-C. De Marneffe, A.N. Rafferty, C.D. Manning, Finding contradictions in text, in: Meeting of the Association for Computational Linguistics, 2008, pp. 1039–1047.

[16] R. Bar-Haim, I. Dagan, I. Greental, E. Shnarch, Semantic inference at the lexical-syntactic level, in: Proceedings of the National Conference on Artificial Intelligence, Vol. 22, AAAI Press; MIT Press, Menlo Park, CA; Cambridge, MA; London, 2007, p. 871, 1999.

[17] H.A. Mohamed, M. Marwa, A. Mohammed, Recognizing textual entailment based on deep learning approach, Int. J. Comput. Appl. 181 (43) (2019) 36–41.

[18] A.P. Parikh, O. Täckström, D. Das, J. Uszkoreit, A decomposable attention model for natural language inference, in: Empirical Methods in Natural Language Processing, 2016, pp. 2249–2255.

[19] Y. Tay, L.A. Tuan, S.C. Hui, A compare-propagate architecture with alignment factorization for natural language inference., 2018, pp. 1–10, arXiv: Computation and Language.

[20] T. Khot, A. Sabharwal, P. Clark, SciTail: A textual entailment dataset from science question answering, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 5189–5197.

[21] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, C. Zhang, DiSAN: Directional self-attention network for RNN/CNN-free language understanding, in: National Conference on Artificial Intelligence, 2018, pp. 5446–5455.

[22] J. Cheng, L. Dong, M. Lapata, Long short-term memory-networks for machine reading, in: Empirical Methods in Natural Language Processing, 2016, pp. 551–561.

[23] J. Choi, K.M. Yoo, S. Lee, Learning to compose task-specific tree structures, in: National Conference on Artificial Intelligence, 2018, pp. 5094–5101.

[24] D. Kang, T. Khot, A. Sabharwal, E. Hovy, AdvEntuRe: Adversarial training for textual entailment with knowledge-guided examples, in: The 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 2418–2428.

[25] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.

[26] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, pp. 1–12, arXiv preprint arXiv:1301.3781.

[27] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, 2015, pp. 1–10.

[28] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[29] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.

[30] Y. Tay, L.A. Tuan, S.C. Hui, Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference, in: Empirical Methods in Natural Language Processing, 2018, pp. 1565–1575.