A Hybrid Siamese Neural Network for Natural Language Inference in Cyber-Physical Systems

PIN NI, University of Liverpool, UK
YUMING LI, University of Liverpool, UK
GANGMIN LI, Xi'an Jiaotong-Liverpool University, China
VICTOR CHANG, Teesside University, UK

Cyber-Physical Systems (CPS), as a multi-dimensional complex system that connects the physical world and the cyber world, has a strong demand for processing large amounts of heterogeneous data. These tasks also include Natural Language Inference (NLI) tasks based on text from different sources. However, the current research on natural language processing in CPS does not involve exploration in this field. Therefore, this study proposes a Siamese Network structure that combines Stacked Residual LSTM (bidirectional) with the Attention mechanism and Capsule Network for the NLI module in CPS, which is used to infer the relationship between text/language data from different sources. This model is mainly used to implement Natural Language Inference tasks and conduct a detailed evaluation in 3 main NLI benchmarks as the basic semantic understanding module in Cyber-Physical Systems. Comparative experiments prove that the proposed method achieves competitive performance, has a certain generalization ability, and can balance the size and performance of trained parameters.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Machine learning**.

Additional Key Words and Phrases: Cyber-Physical Systems, Natural Language Inference, Siamese Neural Networks

ACM Reference Format:
Pin Ni, Yuming Li, Gangmin Li, and Victor Chang. 2020. A Hybrid Siamese Neural Network for Natural Language Inference in Cyber-Physical Systems. 1, 1 (June 2020), 25 pages. https://doi.org/pending

1 INTRODUCTION

Cyber-Physical Systems (CPS) are multiple dimensional complex systems that connect the physical and virtual worlds, relying on the deep collaboration and organic integration of multiple types of modules. Through the interaction between the physical process and human-computer interaction interface, this next-generation intelligent system, which integrates computing, communication, and control, can realize dynamic control, real-time perception, and information service. CPS is inseparable from key technologies related to processing and fusing information flowing from physical space to information space as a combination of perception and cognitive intelligence. This involves many processing and recognition technologies ranging from speech (analog quantity) to semantic (discrete quantity). As we know, voice-related technologies have been widely innovated and applied in the Internet of Things in recent years. However, CPS is a comprehensive system that expands and contains more dimensional information (e.g., Internet) based on IoT, a

Authors' addresses: Pin Ni, University of Liverpool, Liverpool L69 3BX, UK, P.Ni2@liverpool.ac.uk; Yuming Li, University of Liverpool, Liverpool L69 3BX, UK; Gangmin Li, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China; Victor Chang, Teesside University, Middlesbrough TS1 3BA, UK.

Manuscript submitted to ACM

system based on perception intelligence. Therefore, it also focuses more on cognitive intelligence than traditional IoT applications. Semantic understanding plays an important role in CPS's cognitive module system. At the same time, as the core module for implementing the interactive system in CPS, it is a key technology for processing and analyzing the deep meaning in the speech signal from the real-world, and it is also one of the main ways to make CPS realize a more natural human-computer interaction. Natural language inference, as difficulty in semantic understanding, is mainly used to discover the semantic logic and implication relationship in natural language. This task provides a semantic representation of natural language by defining a semantic relationship (Entailment, milling, Neutral), resulting in a more precise output of text vectors that can be easily moved or deployed to other real-world tasks (e.g., dialogue, qa, semantic matching in the cyber-physical world, etc.). Therefore, this task also provides a more effective channel or mechanism for communication and integration between the Cyber and Physical worlds. This more natural way of interaction is also more readable, convenient, and free than conventional virtual-reality communication based on structured data (e.g. metadata). At the same time, this form of free-text data is also a promising direction for the future iterative development of metadata [19].

NLI is mainly used to determine whether the given two randomly language fragments (e.g. paragraphs, sentences, phrases, etc.) are statements of the same fact and whether the main logic of their content is consistent. The main objective of this task is to judge the promise (P) and hypothesis (H) and determine whether it has an entailment relationship. Therefore, NLI can be transformed into a classification problem in the form. This task has been widely used in question answering systems. Much of the work is based on textual inclusion techniques to generate candidate answers or to sort candidate answers generated by other methods [1, 2, 40, 58, 77, 79]. For example, if a user issues the question like "Who is the founder of Google?" and the corresponding knowledge in the knowledge base happens to be similar (e.g. "Larry Page and Sergey Brin founded Google in 1998"). At this time, if the question and answer system can infer the result of "Larry Page and Sergey Brin is the founder of Google, " then the system can answer such questions directly. In fact, there is generally an inference relationship between questions and candidate answers, and between candidate answers and supporting documents. Moreover, the research [31] shows that the application of textual entailment technology can improve the correct answer rate by about 20%. At the same time, in Machine Translation Evaluation [24, 45, 59], Relationship Extraction [6, 48, 67], Multi-document Summary [32, 43, 49], Student Answers Assessment [22, 23, 57], and Syntactic Analysis Evaluation [21, 26, 96] and other tasks, textual entailment task also plays a significant role. As the core technical part of semantic understanding, the level of textual entailment also represents its reasoning ability to a certain extent, which also limits the interactive reasoning ability of the current human-computer interaction systems (e.g., Siri, Alexa, Microsoft Xiaoice, etc.) to a certain extent. Therefore, the ultimate goal of textual entailment related research is to provide a general text-based inference engine to support other semantic-related NLP tasks and other daily applications. And the performance improvement of this part also undoubtedly plays a vital role in the semantic understanding in CPS and the enhancement of the interactive system.

This study proposed a Siamese Network structure that integrates Stacked Residual LSTM with Attention mechanism and Capsule Network for the NLI module in Cyber-Physical Systems. The basic semantic understanding module in CPS is mainly used to implement Natural Language Inference task. And a detailed evaluation based on three classic NLI datasets to test and valid the performance of our proposed model.

The main contributions of this study are as follows:

- This work is currently the first exploration of NLI in the field of Cyber-Physical Systems and has conducted technical empirical analysis and discussion. And this is a valuable exploration for creating effective artificial intelligence models for CPS.
- We propose a Siamese Network based on Stacked Residual LSTM with Attention mechanism and Capsule Network for the task of NLI, which is used for the semantic understanding module in CPS. This study conducted experiments on three classic NLI datasets (SNLI, MultiNLI-m/-mm, MRPC), and obtained competitive results, proving the effectiveness of the proposed model.
- This study designed a dual feature representation layer method combining LSTM-based (Stacked Residual LSTM with Attention) and CNN alternative-based (Capsule Network) for NLI tasks and embedded them into the Siamese Network with weight sharing mechanism. This model can reduce the number of training parameters and achieve more ideal context modeling in the case of multi-input-based parallel computing to obtain better performance of the actual task.

In the remaining sections, we will discuss in detail the background, technical solutions, experimental results, and analysis of the research. In the second section (Related Work), we mainly reviewed some previous major explorations on Natural Language Processing (NLP) in Cyber-Physical Systems and investigated some technological innovations in the field of NLI in recent years. In the third section (Methodology), we introduced the details of the designed network structure in a hierarchical form. The fourth part (Experiments) introduces the experimental environment and datasets. In the fifth section (Results and Analysis), we conducted a specific analysis and discussion of the experiment results. The last section (Limitation and Conclusion) is mainly about the limitation and summary discussion of the study.

2 RELATED WORK

2.1 NLP in Cyber-Physical Systems

The unique challenge in CPS integration comes from the heterogeneity of components and interactions. This heterogeneity drives the need for modeling cross-domain interactions in the virtual world (information domain) and the real world (physical domain). And by understanding the deeper features and attributes of heterogeneous sources, to better serve the needs of specific scenarios [75]. One of CPS's main goals is how to integrate the voice signals received from the physical world and the linguistic information circulating in the information world to achieve barrier-free cross-domain communication. Related research on natural language processing in CPS is currently quite rare. Wiesner et al. [87], based on the survey, explored the challenges, potential applications, and preliminary suggestions about CPS in Requirements Engineering. The study pointed out that the shared informal requirements specifications can be converted into formal domain-specific models of related disciplines through NLP to integrate the collaboratively described requirements of multiple disciplines (e.g. mechanical engineering, software, and systems engineering) and map them to the system element. At the same time, the research also suggests adopting the form of establishing a dialogue system to meet the needs of ambiguity and semi-automatic conversion. Vogelsang et al. [82] reported on their work in automatic knowledge extraction, query answer retrieval based on an expert system, and automatic requirement classification in building CPS. This work envisions that NLP will become a key component that connects requirements and simulation models and can explain tool-based decision-making processes, and provides some inspiration for the development of related fields. At present, the research related to NLP in CPS is mainly concentrated in the field of requirements engineering, but there are also works for Web of Things (WoT) facilitates (CPS system) that involve semantic reasoning in the field of knowledge engineering. Wu et al. [90] design the semantic Web of Things (SWoT) framework for Cyber-Physical

Systems (SWoT4CPS), which provides a hybrid ontology engineering method by extending the semantic sensor network and the deep learning method based on the entity connection solution. The feasibility of the framework is verified through experiments that realize temperature abnormality diagnosis and automatic control use cases. In addition, the evaluation of physical connections shows that it has a high accuracy of linking domain knowledge to DBpedia and an acceptable level of time-consuming. In summary, previous research mainly regarded NLP as "tools" that provide local support related to text data processing for integrated systems at the development level but did not raise its Attention to the methodology system used to process multi-source language information. Therefore, the current progress of NLP in CPS, whether at the application level or technological innovation level, remains at the mere imagination level or embryonic stage, which also limits CPS's ability to process multi-source language information.

2.2 Natural Language Inference

The current NLI has evolved from traditional logical expressions to deep learning methods. Raina et al. [65] express the premise and hypothesis as a conjunction of two sets of sub-propositions using the dependency relationship. Then use the abductive reasoning mechanism to infer the hypothesis from the premise and calculate its cost, to determine whether there is an entailment relationship between the premise and the assumption. In short, the method is to express the text as a mathematical, logical expression to form a set of facts and whether the hypothesis can be inferred according to the premise is determined by logical inference rules. Bar-Haim et al. [8] uses a "calculus idea" similar to [65]. They turned to language analysis techniques (e.g., syntactic analysis, semantic role labeling) instead of strict mathematical logic expressions. Then, Bowman et al. [11] published a data set on textual entailment recognition, SNLI [11]. SNLI has a total of 570K textual pairs, a total of three types (Entailment, Contradiction, Neutral) relationship, and completely by manual annotation. The label integrates the opinions of five experts and is based on the principle of minority obeying the majority. And both promise and hypothesis are based on the same specific scene. At the same time, SNLI is also a benchmark test that enables neural network-based models to demonstrate high competitiveness for the first time in related NLI tasks. The NLI model based on neural networks has developed rapidly with high-quality and rich sources of training data.

2.3 RNN-based methods for NLI

Some of these works are based on deep learning-based models of Recurrent Neural Networks (RNN) and its variants. Rocktäschel et al. [66] proposed the generic end-to-end differentiable system, which uses two LSTMs (Long Short-Term Memory, variants of RNN) to encode the promise and hypothesis to determine the delivery relationship. The method achieved better performance on the SNLI dataset compared to the previous SOTA classifier (Lexicalized classifier (LSTM) [11]. And the study also qualitatively analyzed the Attention weight generated by the model. However, in the study of Wang et al. [85], the approach proposed by [66] has certain limitations: only a single vector of promise is used to match hypothesis; Neither increases the weight of the more important matching part between promise and hypothesis nor decreases the weight of non-critical parts. Therefore, for these two defects, they matched the Attention vector and the hypothesis vector based on the original model. Then input to match-LSTM. In addition, the special word NULL is also introduced in the premise. When the word in hypothesis does not form any match with the word in premise, match-LSTM will match the word with NULL, which is equivalent to increasing the dangling alignment and improving the alignment model. Kim et al. [41] designed a densely-connected co-attentive RNN similar to DenseNet (a densely connected convolutional network) [35]. The model performs concatenated information processing on attentive features and all hidden features of preceding recurrent layers. In addition, the model also uses AutoEncoder to solve the problem of the rapid growth of parameters after concatenation. Chen et al. [15] proved that the sequential reasoning model based on the chain LSTM could be superior to the model of the complex network structure used before. On this basis, it further shows that by adding local reasoning and reasoning combination, it can well improve the sequential reasoning model. In addition, the study introduced the ESIM model (Enhanced Sequential Inference Model) and introduced a tree-type LSTM network incorporating syntactic parsing information. A majority of previous research projects on text matching only considered unidirectional semantic matching, but did not consider the importance of bidirectional semantics. Generally speaking, only consider a single granularity semantic matching (word by word or sentence by sentence). In view of the above defects, Wang et al. [86] designed the BiMPM (Bilateral Multi-perspective Matching) model to solve the above problems. In this work, we draw on the idea of bidirectional semantic modeling to build a Stacked LSTM based on bidirectional to capture richer features.

2.4 CNN-based methods for NLI

The deep learning model based on CNN (Convolutional Neural Networks) and its variants also play a significant role in the NLI. Mou et al. [53] use a tree-structured CNN to model a single sentence. They use the syntactic dependency tree as the operation object of convolution to form a subtree feature extractor, which can extract the dependency relationship between the parent node and its child nodes at once. This method uses two Tree-based CNNs to model promise and hypothesis, respectively. It uses vectors constructed by heuristic features construction methods such as concatenation, difference, and multiplication to represent the semantic information of a pair of premise and hypothesis, and finally, use softmax for classification. Zhang et al. [98] emphasized that in many cases, it is necessary to recognize the association between words before recognizing the association between sentences, and vocabulary entailment largely depends on the contextual representation of words. Therefore, this study proposes a recognition method based on Context-Enriched Neural Network (CENN). Specifically, the representation of the input word pairs uses multiple embedding vectors, from different contexts, to integrate and optimize these vectors through multiple combination methods and Attention weighting, and then output the relationship between the predicted word pairs. Yin et al. [94] adds Attention Matrix to the convolution layer. This method allows the model to consider the correlation between sentences (the hypothesis and premise sentences are originally independent) and the context relationship of sentences through the Attention mechanism. This method fully considers the situation that the information of promise and hypothesis needs to be considered in the entailment relationship (the pair of texts needs to be modeled simultaneously). Chen et al. [13] proposed a GCNN (Gated CNN) for sentence matching, which solves the problem of time-consuming and other problems that the recursive architecture cannot perform parallel computing within sequences. The convolution stacked by this method encodes the hierarchical context-aware representations of the sentence, where the gating mechanism optionally controls and stores the convolutional context information. Gong et al. [27] designed an IIN (Interactive Inference Network), to realize the hierarchical extraction of high-dimensional semantic features of sentence pairs from interactive space. The research mainly proves that an interactive tensor matrix (Attention weight) contains the semantic information in NLI, while a denser interactive tensor matrix often potentially contains richer semantic information. So they built Densely Interactive Inference Network (DIIN), and achieved a performance improvement of more than 5% over the previous methods [56] on the MultiNLI [88] dataset.

3 METHODOLOGY

The inference relationship between natural languages, also known as Textual Entailment, refers to the basic semantic connection between texts (or refers to the pointing relationship between two text fragments). In simple terms, the

Table 1. An example of three conditions for Textual Entailment

A positive Textual Entailment example (hypothesis is entailed in the given text):	Given text: If you help those in need, God will reward you.
	Hypothesis: Giving money to the poor can get a good return.
A negative Textual Entailment example (the meaning of the given text is about refuting the hypothesis):	Given text: If you help those in need, God will reward you.
	Hypothesis: There is no reward for giving money to the poor.
An example without Textual Entailment (the given text neither entails hypothesis nor refute it):	Given text: If you help those in need, God will reward you.
	Hypothesis: Giving money to the poor will make you better.

textual entailment relationship describes the reasoning relationship between two texts. One of the texts is used as a premise, and the hypothesis is the other text. If premise P can infer hypothesis H, then P can be regarded as entailing H and annotated as $P \longrightarrow H$. For example, we can find from Table 1 that the textual entailment relationship is not purely logical reasoning, and its conditions are looser, which can be defined as follows: if a person can infer from h that h has a high probability of being true, then t is entailed h $(t => h)$.

Textual entailment is different from similar tasks like Paraphrasing. Two text fragments from different sources do not contain the same semantics with high probability. Strictly speaking, Paraphrasing can be regarded as a "Textual Equivalence" relationship, or it can be called as a "Bidirectional Textual Entailment" relationship. But Textual Entailment is a unidirectional inference relationship (H cannot infer P backward) [5]. In addition, Textual Entailment and Text Similarity are also quite different. If a pair of texts contain similar semantics, they can be evaluated using edit distance or other similar measures, but they may not constitute an entailment relationship [3, 29, 33, 39, 51]. The main idea of Textual Entailment is more like the human decision-making process of judging the authenticity of the semantic propositions given by hypothesis after understanding the semantic propositions of premises and combining their own common sense. In addition, the decision-making process differs from logical reasoning to a certain extent. Although some textual entailment relationship methods refer to the basic idea of logical reasoning [4, 9, 52, 65, 68], it does not strictly abide by the rules of mathematical, logical reasoning, and its discriminating process is also different from the process of mathematical logic. In summary, compared to the other tasks mentioned above, the Textual Entailment task has: focusing on the inclusion or reasoning attributes between the semantics of the text, using human common sense as the basis for reasoning, having the characteristics of directionality, etc.

3.1 Overall Structure

This study proposes a novel multi-dimensional data simultaneous input structure for NLI tasks. The structure is based on the Siamese Network with a binary branch. Each branch net composed of the Embedding Layer, the Encoder Layer, the Contextual Representation Layer, the Aggregation, and the Predict Layer (Fig. 1). Each layer mentioned above is composed of the following models: Embedding Layer: BERT [18] is used as word embedding to obtain more accurate dynamic word vectors than static word embedding (e.g., Word2Vec, GloVe, etc.). Encoder Layer: Stacked Residual LSTM as a semantic modeling method, using a multi-layer and different directional LSTM network with Residual layer and Attention mechanism to model the language. Contextual Representation Layer: Capsule Network as a feature extraction mechanism for extracting local context Features, this part can also be regarded as a semantic modeling layer together with the Encoder Layer. Aggregation and Prediction Layer: the text representation vectors output by the Contextual Representation Layer of two branch nets are concatenated, and softmax is used as the activate function of the prediction layer to output the inferred results. Since the Embedding Layer directly uses BERT as word embedding in this study,

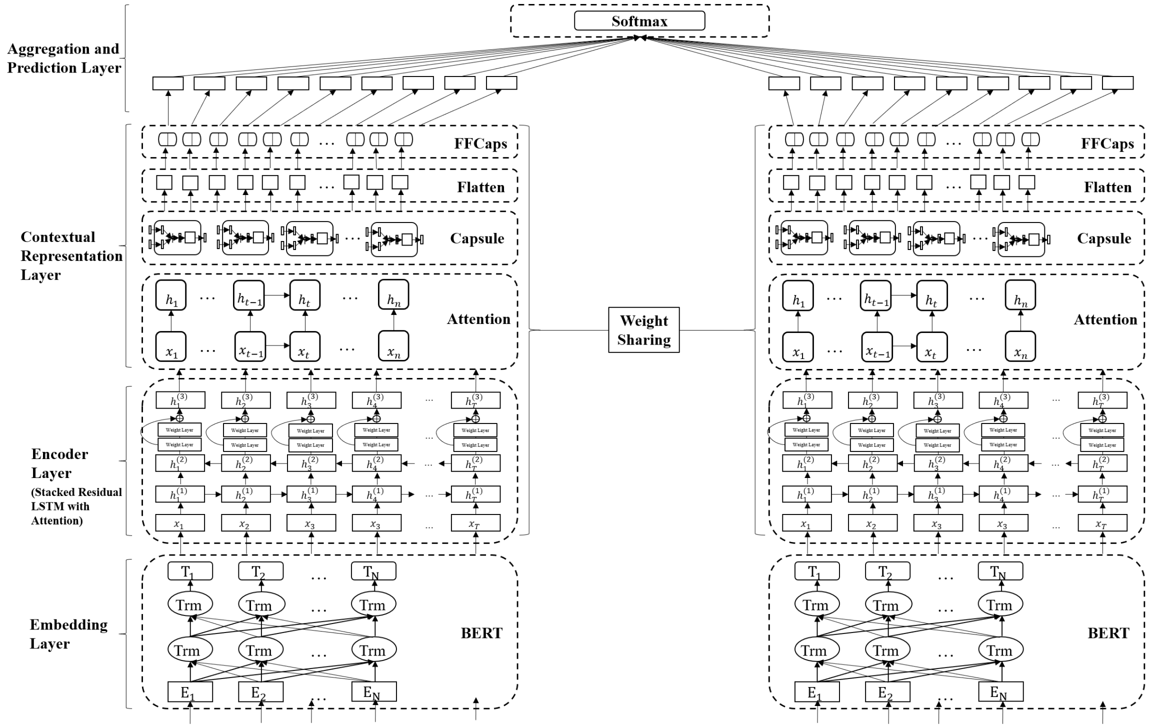

Fig. 1. The overall structure of the proposed model

the other three layers will be mainly introduced in the subsequent content. For an overview of the structure of the model, please refer to Fig. 1.

3.2 Siamese Network

The Siamese Network was originally designed to determine the similarity of any two images [12]. The two photos are simultaneously modeled by convolutional neural networks to obtain separate vector representations, and then the edit distance of the two vector representations is solved to determine whether they are similar. It also provides a new direction for Textual Entailment research. One of its core ideas is parameter sharing among multiple encoders. Therefore, Siamese Network as the main component of the proposed structure to achieve weight sharing and matching mechanism, its weight sharing has two purposes: (1) reduce the number of parameters, reduce the complexity of the model; map the vectors of two different spatial dimensions to the same spatial dimension, to keep their distribution consistent, and to encode different vectors in the same spatial dimension. Therefore, we built a shared weight layer (share embedding layer) in the two branches of the Siamese Network to process these inputs from similar spatial dimensions for encoding (e.g. two texts with different meanings with similar vocabularies). Through this layer, information can be shared between different inputs, and the model can be trained on fewer data.

In summary, the role of this part is to input two different feature vectors, and finally judge and output the relationship between the two feature vectors. Its specific structure is shown in Fig. 2.

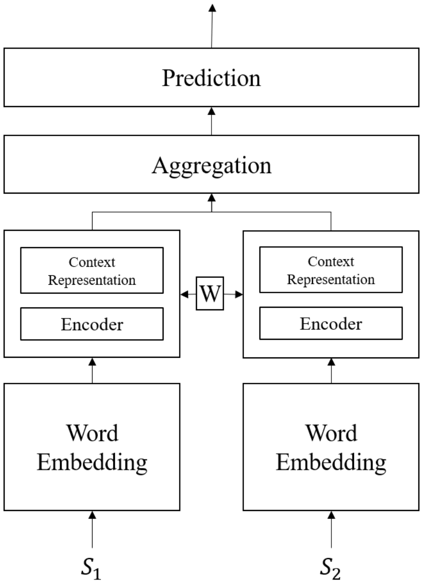

Fig. 2. The overall structure of the Siamese Neural Network

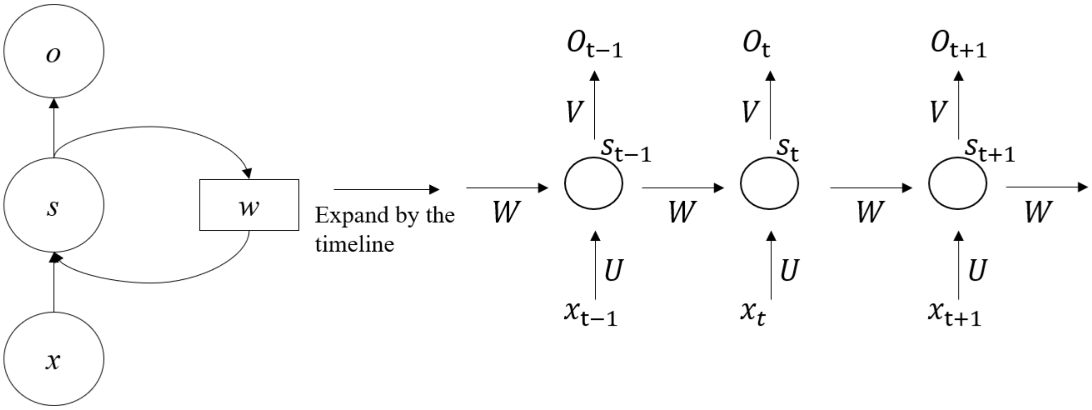

Fig. 3. The structure of the RNN

3.3 Encoder Layer

The convolutional layer of the capsule network is mainly used to extract local features. It uses capsule vectors to represent the spatial position relationship between local features [89]. To make up for the defect that the capsule network does not consider the context relationship of features, we have added an LSTM network before the capsule network to obtain context information of local features. Additionally, Encoder Layer is composed of multiple stacked LSTM networks, which has better semantic representation capabilities than shallow layers.

RNN is a recursive neural network based on sequence input, which recurs in the direction of sequence evolution and all nodes (recurrent units) are connected in a chain. The neurons of the hidden layer are calculated by inputting x_t at the time of t, and s_{t-1}, the activation value of the hidden layer neurons at the time of $t-1$ (Fig. 3).

The neurons can be calculated as follows (where U and W are the parameters of the network, $f(.)$ represents the activation function, and x_t is the input at the time of t):

$$s_t = f\left(x_t U + s_{t-1} W\right) \tag{1}$$

The hidden layer h at the moment of t is calculated as (b is bias):

$$h_t = \tanh\left(W x_t + U h_{t-1} + b\right) \tag{2}$$

Although RNN solves the problem of information preservation that appeared in previous neural networks, it is still limited to the issue about long-term dependence (i.e. the memory capacity is limited, and it is unable to capture the key information of the remote context), so it derives its variant LSTM [97]. LSTM solves this problem by introducing a memory cell so that the information that needs to be memorized can always be transmitted, ensuring that the state will not disappear with time (Fig. 4). The unit of LSTM at each moment is defined as a set of vectors in the d dimension space: forget gate f_t, memory unit c_t, input gate i_t and output gate o_t. In addition, σ means the sigmoid activation function and $\odot$ represents the symbol multiplied by the element, and h_t represents the hidden state (different from RNN's h_t). It is worth noting that the range of vectors i_t, o_t, f_t is $[0, 1]$.

1. Gates

$$i_t = \sigma\left(W_{xi} x_t + W_{hi} h_{t-1} + b_i\right) \tag{3}$$

$$o_t = \sigma\left(W_{xo} x_t + W_{ho} h_{t-1} + b_o\right) \tag{4}$$

$$f_t = \sigma\left(W_{xf} x_t + W_{hf} h_{t-1} + b_f\right) \tag{5}$$

2. Input transform

$$c\_in_t = \tanh\left(W_{xc} x_t + W_{hc} h_{t-1} + b_{c\_in}\right) \tag{6}$$

3. State update

$$c_t = f_t \odot c_{t-1} + i_t \odot c\_in_t \tag{7}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{8}$$

Essentially, the input gate controls the update level of each unit, the output gate controls the output level of the internal unit, the forget gate is responsible for controlling to what extent the memory unit at the previous moment is forgotten. Hence, the hidden state vector in the LSTM unit is a gated local representation of the state of the memory in the unit. The model can learn information representation on multiple time scales because the value of the gated variable changes according to each vector element.

In view of unidirectional LSTM modeling the sentence, there is a problem that the backward information (from right to left) cannot be encoded. Especially in more fine-grained classification tasks (e.g., the effect of Sentiment Analysis is influenced by a large number of vocabulary such as emotion word, degree of adverb, negator, etc.), it has higher requirements for the model's context representation ability.

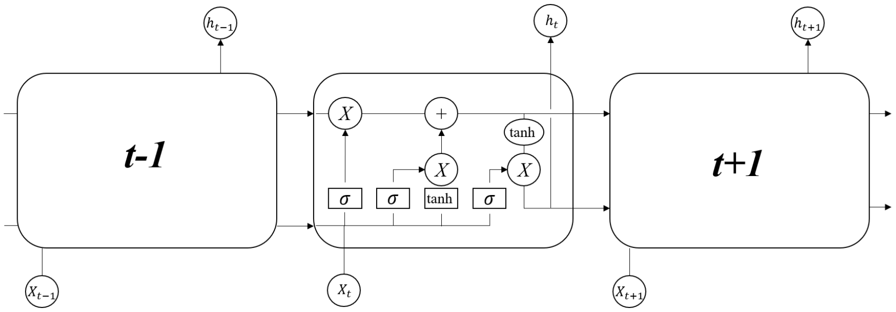

Fig. 4. The structure of the LSTM

Graves et al. [28] found that a deeper network and more memory cells played a positive role in the overall effect of the method. Therefore, deepening the neural network may be considered as a potentially feasible solution. In our study, we build Stacked LSTM to capture the bidirectional semantic dependence and learn the comprehensive representation from previous layers by increasing the depth of the neural network, to create new representations at a higher level of abstraction.

Each LSTM memory cell requires 3D input. When the LSTM processes each time step sequence, each memory cell will output a single value of the entire sequence in the form of a 2D array. However, in order to stack multiple LSTM layers, the output of each LSTM needs to be adjusted to the form of a 3D array as the input of the subsequent layer (i.e. adjust the original sequence return mode of "only output the hidden state of the last time step" to the form of "output the hidden state of each time step").

We refer to the design of Stacked Residual LSTM in the work of Prakash et al. [62]. In our work, the stacked LSTM is consist of a group of BiLSTM and a forward unidirectional LSTM, which are successively stacked into three layers of the network. In the first two layers of LSTM, when entering a sentence (eg "I like it."), its forward $LSTM_L$ enters "I", "like", "it" in turn, and three corresponding vectors h_{L0}, h_{L1}, h_{L2} can be obtained from the network. And backward $LSTM_R$ input "it", "like", "I" in turn, the corresponding vectors h_{R0}, h_{R1}, h_{R2} can be obtained from the network. The last layer is the forward LSTM to perform deeper unified modeling of the previous two layers of LSTM. The hidden states of all LSTM layers are fully connected. Except for the first layer, the input of all other layers is passed from the hidden state of the previous layer l at time step t. Therefore, the activation of the l layer can be expressed as:

$$h_t^{(l)} = f_h^l\left(h_t^{(l-1)}, h_{t-1}^{(l)}\right) \tag{9}$$

In addition, the $h\hat{}$ of the l layer is updated with the residual value x_{l-n}, and the xi represents the input of the layer $i+1$, and the residual connection is added after a single LSTM layer. According to the experience gained in the study of Prakash et al. [62], when the number of layers exceeds 3, it will cause expensive computational overhead. Therefore, we only add after the second layer when adding the residual layer. In the first layer, the learned function is a standard LSTM with deviation, which depends on the input x, so not every layer of LSTM needs to add a residual connection, which does not increase any trainable parameters and computational complexity. At the same time, this will also reduce the probability of the gradient disappearing during training.

As an effective mechanism to capture global semantic dependency, Attention is widely used by Transformer, BERT, XLNET, and other models. Here, we add the Attention mechanism [81] after Stacked Residual LSTM to weight the

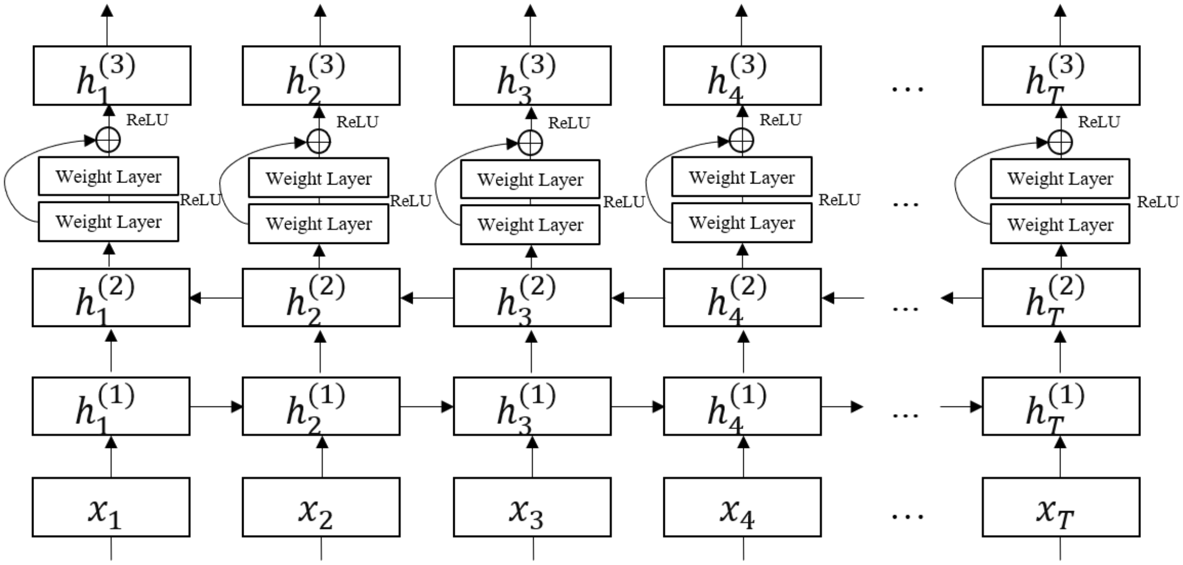

Fig. 5. The structure of the LSTM

output, which is used to give higher weight to the key semantic features to improve the overall semantic modeling performance of the model.

Therefore, the Attention Value needs to be calculated to weight the output of the Stacked Residual LSTM. Attention function generates the Attention distribution coefficient, which is used to get Attention Value. The specific details are: Q is a Query element in the given target, and K is a part of the $(Key, Value)$ of the constituent elements in the source, which is the Key. Then by calculating the correlation between Q and each K, the weight of each K corresponding value can be obtained, and calculate the product of each value and its weight and then sum them (i.e. $Value$). The calculation method of the Attention function can be roughly divided into three stages as follows: 1. Calculate the similarity between Q and K to obtain the weight; 2. Normalize the weight calculated in the previous stage; 3. Multiply the normalized weight with V and then weighted sum them. At this time, the result of weighted summation is Attention Value. The specific calculation process is presented in Fig. 6.

Therefore, the specific method of Attention is to use the h_i output by the Stacked Residual LSTM as the input of the Attention mechanism, by expressing the set of vectors output by the LSTM layer at each moment as $H : h_1, h_2, ..., h_t$, the weight matrix obtained by the Attention layer can be obtained by the following formula:

$$M = tanh(H) \tag{10}$$

$$\alpha = softmax(w^T M) \tag{11}$$

$$r = H\alpha^T \tag{12}$$

$H^{d_w T}$, d_w means the dimension of the word vector, and w^T represents a transposition of the parameter matrix obtained by training. The semantic representation of the final output can be annotated as follows:

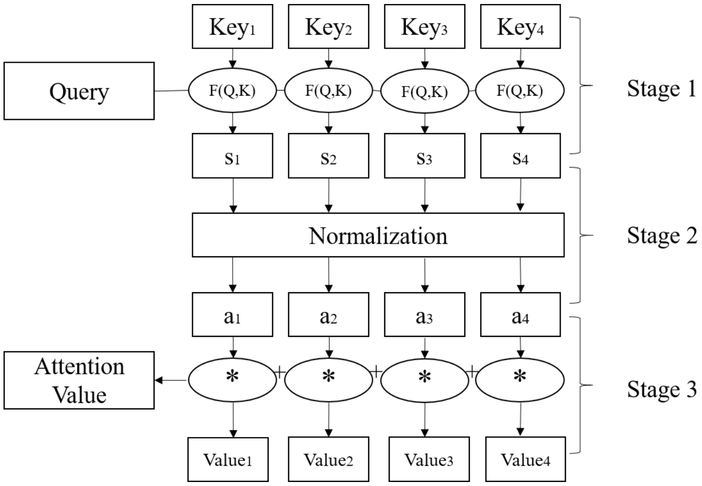

Fig. 6. The calculation process of the Attention mechanism

$$h^{*} = tanh(r) \tag{13}$$

3.4 Contextual Representation Layer

In traditional CNN, the underlying features construct higher-order features through the network to represent higher-level meanings. However, the CNN model will repeatedly use many feature detectors, and it cannot model spatial information while maintaining the feature representation. Unlike the traditional neural network in the past, each neuron in the Capsule Network is a vector instead of a scalar and uses vectors to represent attributes. Therefore, it adopts a new "vector in vector out" network transmission scheme, which is more suitable for NLP tasks based on word vectors as input and output. This scheme is interpretable to a certain extent. Here we adopted the Capsule Network variant adopted by Zhao et al. [101] as the scheme of this part. The specific method consists of 5 layers: N-gram Convolutional Layer (Conv), Primary Capsule Layer (PrimaryCaps), Convolutional Capsule Layer (ConvCaps), Capsule Flatten Layer (Flatten), and Fully Connected Capsule Layer (FCCaps). And its specific structure is shown in Fig. 7.

In the N-gram Convolutional Layer, the vector output by the previous Encoder Layer is first extracted the N-gram features by a convolution layer to obtain the output M. This part is the N-gram features extracted by multiple different convolution kernels at different positions of the sentence.

At the first capsule network layer, the Primary Capsule Layer, the capsule replaces the scalar output of the convolution operation with a vector output, thereby retaining the instantiated parameters (e.g., the semantic representation of words and the local order of words).

In the Dynamic Routing part, the basic thought is to design a nonlinear mapping. The nonlinear mapping iteratively ensures that the output of each capsule can be fed into the appropriate parent capsule in the next layer. The nonlinear mapping iteratively ensures that each capsule's output can be fed into the appropriate parent capsule in the next layer. For each potential parent capsule, the capsule network can increase or decrease the connection strength between the child capsule and each parent capsule through the dynamic routing process, which is more effective than the original Downsampling Strategy (e.g., max-pooling, etc.) and other pooling operations.

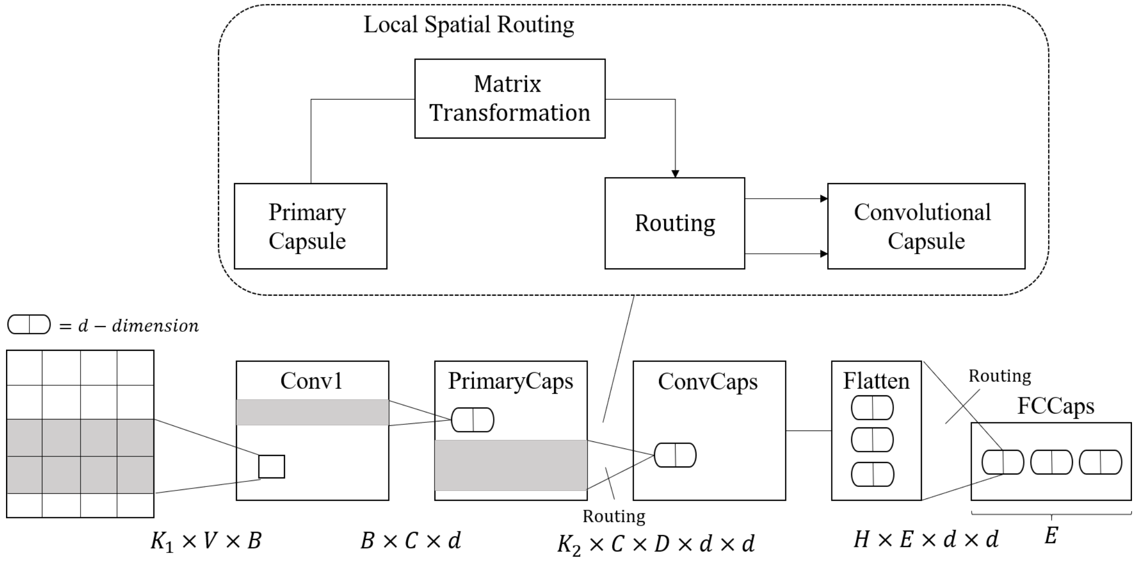

Fig. 7. The structure of capsule network

Although the pooling operation can detect features that appear anywhere in the text, it also brings about problems such as shift-invariant of representation, which causes a lot of spatial location information to be lost. Therefore, the method of dynamic routing can learn more advanced and flexible representation; the specific methods are:

Use $u_{j|i}$ to represent the prediction vector output from the i primary capsule to the j digit capsule, which can be obtained by multiplying the weight matrix $W_{i|j}$ by the vector u_i output from the primary capsule.

$$\hat{u}_{j|i} = W_{ij} u_i \tag{14}$$

$b_{i|j}$ is logits of coupling coefficients, which can be considered as the logarithmic prior probability between the i capsule and the j capsule, initialized to 0. Secondly, normalize $b_{i|j}$ through Leaky-softmax [69] to obtain the coupling coefficient $c_{j|i}$ of connection strength of the connection from the i-th capsule to the j-th capsule.

$$c_{ij} = \text{Leaky-softmax}\left(b_{ij}\right) \tag{15}$$

Then by calculating the sum of the products of all prediction vectors and their corresponding connection probabilities, the digital capsule input s_j can be obtained.

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{16}$$

Similar to the activation function sigmoid in the neuron model, the input is mapped to the interval of 0 1. $g(.)$ is a nonlinear activation function called "squashing". It is used to map shorter vectors to vectors with a length close to 0, and to map longer vectors to vectors with a length close to 1. And it is applied to the digital capsule input s_j for calculation, so the digit capsule output v_j can be obtained.

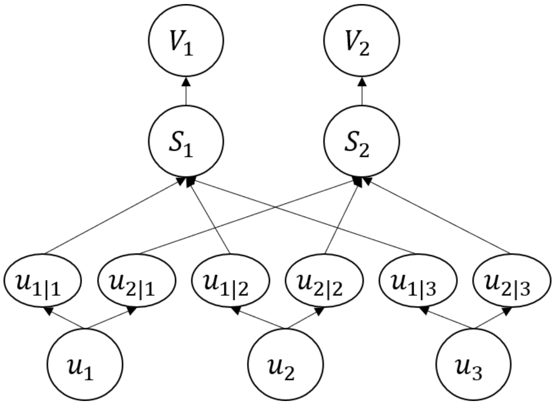

Fig. 8. The calculation process of Dynamic Routing

$$g\left(s_j\right) = \frac{\left\|s_j\right\|^2}{1+\left\|s_j\right\|^2} \cdot \frac{s_j}{\left\|s_j\right\|} \tag{17}$$

The corresponding weight $b_{i|j}$ is updated by the following equation, i.e. $b_{i|j}$ is updated by the result calculated by the inner product of two vectors (the prediction vector u_i output from the primary capsule, the vector v_j of output from the digit capsule). Repeat this process until convergence.

$$b_{j|i} = b_{j|i} + \hat{u}_{j|i} \cdot v_j \tag{18}$$

The visualization of the above process is shown in Fig. 8.

In Convolutional Capsule Layer, each capsule is only connected to a local area in the next layer. The relationship between child capsules (low-layer capsules) and parent capsules (high-level capsules) is calculated by multiplying these capsules and the transformation matrix, and then the upper-level parent capsules are calculated according to routing-by-agreement.

In Capsule Flatten Layer and FCCaps layer, flatten the capsules in the next layer into a list of capsules and fed into the FCCaps layer. In the FCCaps layer, the capsule is multiplied by a transformation matrix, and then the final capsule and its probability for each category are generated by routing-by-agreement.

3.5 Aggregation and Prediction Layer

The purpose of Aggregation and Prediction Layer is to concatenate the matching vectors, which output from the branch network of the Siamese Network on both sides into a matching vector (fixed-length). And then, feed it into the softmax layer through the last Dense layer to predict the final relationship judgment result of the Textual Entailment: Entailment, Contradiction, or Neutral.

4 EXPERIMENTS

4.1 Environment

The experiment is based on Dual Nvidia GeForce GTX 1080 Ti GPU (11GB), Dual 2.50GHz RAM, Intel Xeon E5-2678 v3 CPU environment. Due to GPU memory limitations, we use BERT-base (110M parameters, 12-layer, 768-hidden) [18] as the word embedding in our model's Embedding Layer.

4.2 Datasets

Here, we used three classic natural language inference datasets (SNLI, MultiNLI-m/-mm, MRPC) as tests to evaluate the effectiveness of the proposed method in different applications compared to other previous baseline models. These datasets are as follows:

- Stanford Natural Language Inference (SNLI) Corpus [11]: The SNLI corpus has about 550k hypothesis/premise pairs whose measurement is based on accuracy.
- Multi-Genre Natural Language Inference (MultiNLI) corpus [88]: This series of the corpus contains about 433k hypothesis and promise pairs, which covers a large number of genres of the spoken and written text, and the creation of this corpus refers to SNLI. In addition, in the work of Gururangan et al. [30], the test set is divided into two parts, matched and mismatched. Therefore, in this experiment, we use MultiNLI-m (matched) and MultiNLI-mm (mismatched) to evaluate the performance of the models.
- Microsoft Research Paraphrase Corpus (MRPC) [20]: The sentence pairs are derived from comments on the same news. It is used by the model to determine whether each of the 3600 pairs of annotated sentences is semantically the same.

5 RESULTS AND ANALYSIS

5.1 SNLI

From the comparative experiments (Tab. 2, Fig. 10, 11), we can find that our method has great advantages over other hybrid methods on SNLI datasets. Among them, our model exceeds 300D Full tree matching NTI-SLSTM-LSTM global attention [54] improved by 2.4% on Test Accuracy and 7.8% by Train Accuracy; compared to BiMPM [86] has 2.2% and 5.6% improvement on Test Accuracy and Train Accuracy respectively; compared with ESIM+ELMo [61] has 1% and 4.7% improvement on Test Accuracy and Train Accuracy respectively. In addition, our method also has a slightly different performance compared to fine-tuning the current SOTA language model, such as: compared to BERT-Base [99], our method is higher on Test Accuracy 0.5%. However, compared with BERT-Large+SRL [99], it is 1.6% lower on Test Accuracy and 0.6% higher on Train Accuracy, and the parameter amount is only one-third of BERT-Large+SRL (mainly using BERT-base as word embedding, the parameter quantity is much less than Large version). This also applies to comparisons with SemBERT [100]. In addition, there is a certain correlation between the parameter quantity and the performance of the model (Fig. 10), and the model with a larger parameter quantity will perform better than the model with a smaller parameter quantity. To some extent, this has something to do with the size of the pre-training corpus, the training duration, and computing power.

In summary, although the method proposed by this study is not the best solution on SNLI, it has a better balance in parameter quantity and effect than other large language models.

Table 2. The comparative experimental results on the SNLI dataset

METHOD	TEST ACCURACY (%)	TRAIN ACCURACY (%)	PARAMETERS	YEAR
SemBERT [100]	91.9	94.4	339m	2020
MT-DNN [47]	91.6	97.2	330m	2019
BERT-Large+SRL [99]	91.3	95.7	308m	2019
Fine-Tuned LM-Pretrained Transformer [63]	89.9	96.6	85m	2018
Our Method	**89.7**	**96.3**	**108m**	**2020**
BERT-Base [99]	89.2	-	-	2019
300D DMAN Ensemble [60]	89.6	96.1	79m	2018
150D Multiway Attention Network Ensemble [76]	89.4	95.5	58m	2018
450D DR-BiLSTM Ensemble [25]	89.3	94.8	45m	2018
300D CAFE Ensemble [78]	89.3	92.5	17.5m	2018
ESIM+ELMo Ensemble [61]	89.3	92.1	40m	2018
KIM Ensemble [14]	89.1	93.6	43m	2018
SLRC [99]	89.1	89.1	6.1m	2019
RE2 [93]	88.9	94.0	2.8m	2019
Densely-Connected Recurrent and Co-Attentive Network [41]	88.9	93.1	6.7m	2019
448D Densely Interactive Inference Network (DIIN) [27]	88.9	92.3	17m	2018
300D DMAN [60]	88.8	95.4	9.2m	2018
BiMPM Ensemble [86]	88.8	93.2	6.4m	2017
ESIM+ELMo [61]	88.7	91.6	8.0m	2018
KIM [14]	88.6	94.1	4.3m	2018
600D ESIM+300D Syntactic TreeLSTM [15]	88.6	93.5	7.7m	2017
450D DR-BiLSTM [25]	88.5	94.1	7.5m	2018
Stochastic Answer Network [46]	88.5	93.3	3.5m	2018
300D CAFE [78]	88.5	89.8	4.7m	2018
150D Multiway Attention Network [76]	88.3	94.5	14m	2018
Biattentive Classification Network+CoVe+Char [50]	88.1	88.5	22m	2017
aESIM [44]	88.1	-	-	2018
BiMPM [86]	87.5	90.7	1.6m	2017
2400D Multiple-Dynamic Self-Attention Model [95]	87.4	89.0	7.0m	2018
300D Full tree matching NTI-SLSTM-LSTM global attention [54]	87.3	88.5	3.2m	2017

5.2 MultiNLI

In this part of the comparative experiment, our proposed method has some advantages over other Attention-based models on the accuracy of Matched and Mismatched (Tab. 3, Fig. 11), including: compared with BiLSTM+ELMo+Attn [84], our method is improved by 10.7% and 9.8% respectively; compared with Multi-task BiLSTM+Attention [84], our method is improved by 12.2% and 12.1%, respectively; compared to BiGRU sentence encoder+ Attention [10], our method

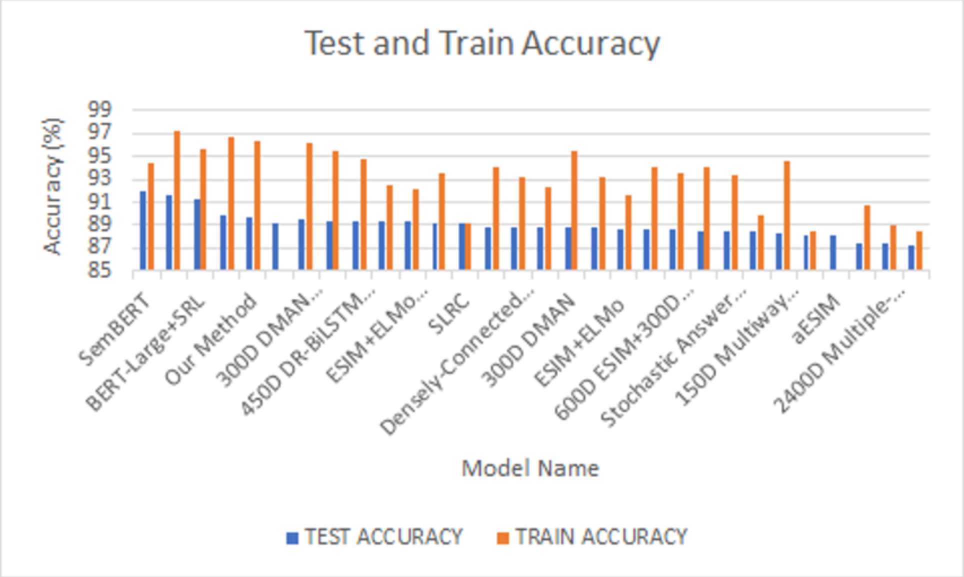

Fig. 9. Visualization of Test and Train Accuracy on SNLI dataset

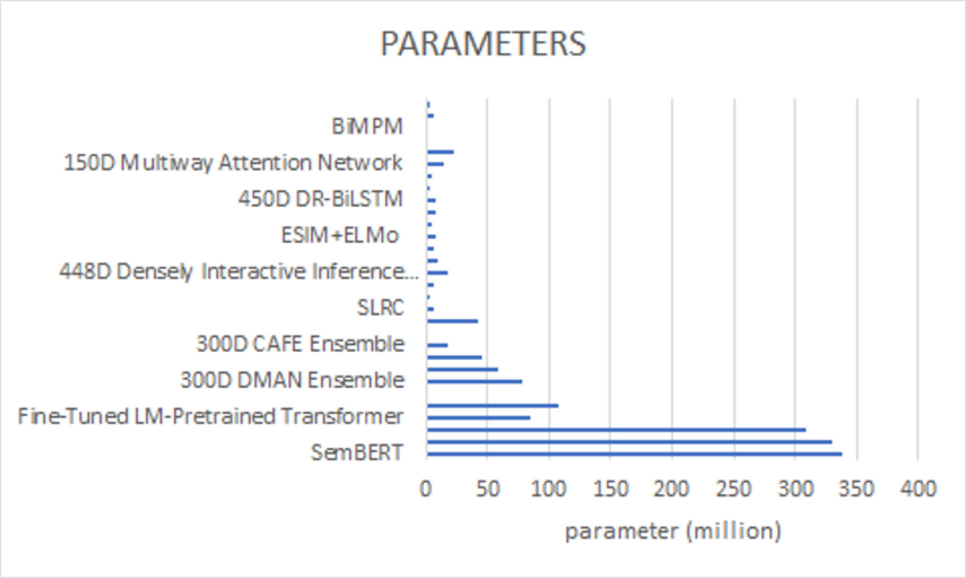

Fig. 10. Visualization of parameter quantity of models on SNLI dataset

is improved by 15.4% and 14.8%, respectively; compared with Character-level Intra-attention BiLSTM encoders [92], our method is improved by 16.9% and 16.1%, respectively, etc. At the same time, the proposed method also has some advantages for other stack types: compared with Stacked Bi-LSTMs (shortcut connections, max-pooling, Attention) [10], our method is 14.1% higher on Matched accuracy and 13.8% higher on Mismatched accuracy. In addition, compared

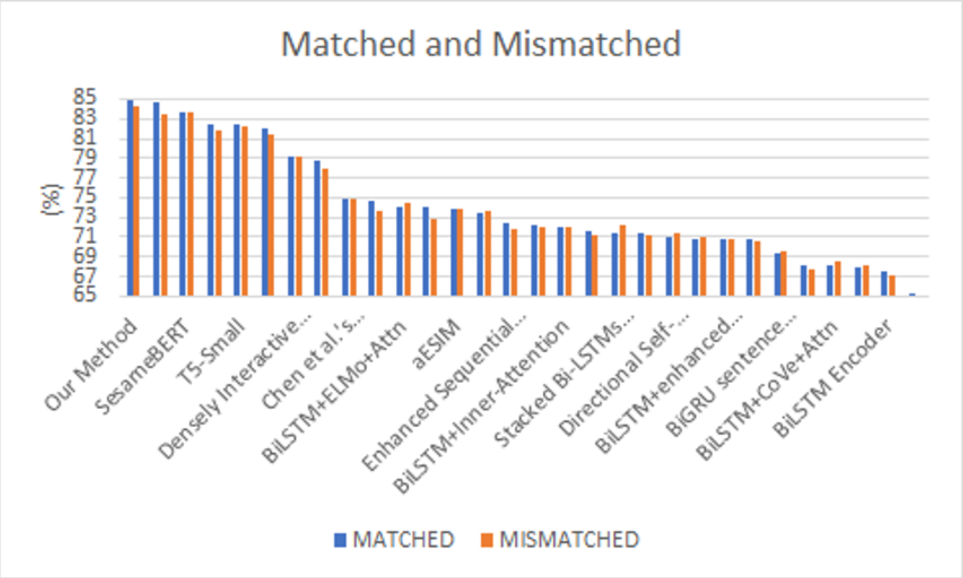

Fig. 11. Visualization of accuracy of Matched and Mismatched on the MultiNLI dataset

with the fine-tuning of some language models, there is also a slight improvement: 0.2% and 0.9% higher than BERT-Base [18]; 1.1% and 0.7% higher than SesameBERT [72]; 2.3% and 2.5% higher than TinyBERT [38]; 2.4% and 2% higher than T5-Small [64]; 2.7% and 2.9% higher than OpenAI GPT [63], etc. Therefore, based on the evaluation of the comparative models on the MultiNLI dataset, it can be proved that our method has a certain generalization ability and is relatively effective to some extent.

5.3 MRPC

On the MRPC dataset, compared with other BiLSTM-based models or fine-tuning language models, the proposed model can still guarantee the effect of Accuracy and F1-Score to a certain extent (Tab. 4, Fig. 12). In particular, the BiLSTM+ELMo, BiLSTM+ELMo+Attn, BiLSTM+Attn, BiLSTM, BiLSTM+CoVe+Attn, BiLSTM+CoVe model on GLUE benchmark [84] are lower than our proposed methods on the above two measurements: 3.1% and 6.1%; 3.4% and 6.1%; 7.7% and 7.9%; 6% and 7.9%; 7.8% and 12.3%; 9.1% and 9.6%, respectively. At the same time, compared to BERT-of-Theseus (6-layer; single model) [91], TinyBERT [38], T5-Small [64], OpenAI GPT [72], our method has improved on the above two measurements: 0.2% and 0.9%; 0.5% and 1.5%; 1.2% and -5.6%; 5.5% (only accuracy), respectively. This high probability may have something to do with the level of the language model selected by the word embedding layer.

6 LIMITATION AND CONCLUSION

Compared with the unsupervised method (e.g., distance measurement, etc.), the supervised natural language inference models (SNLINMs) generally have higher computational complexity. In actual application scenarios (e.g., question answering system, information retrieval, etc.), SNLINMs have problems, such as extended response time and high computing resource overhead, which undoubtedly have negative impacts on user experience and actual deployment costs. Therefore, in the future natural language inference task, the combination of unsupervised and supervised methods

Table 3. The comparative experimental results on the MultiNLI-m and MultiNLI-mm datasets

METHOD	**MATCHED**	**MISMATCHED**	**YEAR**
Our Method	**84.8**	**84.3**	**2020**
BERT-Base [18]	84.6	83.4	2019
SesameBERT [72]	83.7	83.6	2019
TinyBERT [38]	82.5	81.8	2019
T5-Small [64]	82.4	82.3	2019
OpenAI GPT [63]	82.1	81.4	2018
Densely Interactive Inference Network [27]	79.2	79.1	2018
Compare-Propagate Alignment-Factorized Encoders [78]	78.7	77.9	2018
Chen et al.'s Ensembled ESIM [16]	74.9	74.9	2017
Shortcut-Stacked Encoder [56]	74.6	73.6	2017
BiLSTM+ELMo+Attn [84]	74.1	74.5	2018
Distance-Based Self-Attention Network [36]	74.1	72.9	2017
aESIM [44]	73.9	73.9	2018
Deep Gated Attention BiLSTM encoders [16]	73.5	73.6	2017
Enhanced Sequential Inference Model [15]	72.4	71.9	2017
Multi-task BiLSTM+Attention [84]	72.2	72.1	2018
BiLSTM+Inner-Attention [7]	72.1	72.1	2017
BiLSTM-Max Encoder [56]	71.7	71.2	2017
Stacked Bi-LSTMs (shortcut connections, max-pooling) [10]	71.4	72.2	2019
GenSen [73]	71.4	71.3	2018
Directional Self-Attention Encoders [71]	71.0	71.4	2018
BiLSTM sentence encoder (max-pooling) [10]	70.7	71.1	2019
BiLSTM+enhanced embedding+max-pooling [83]	70.7	70.8	2017
Stacked Bi-LSTMs (shortcut connections, max-pooling, Attention) [10]	70.7	70.5	2019
BiGRU sentence encoder+Attention [10]	69.4	69.5	2019
SWEM-max [70]	68.2	67.7	2018
BiLSTM+CoVe+Attn [84]	68.1	68.6	2018
Character-level Intra-attention BiLSTM encoders [92]	67.9	68.2	2017
BiLSTM Encoder [88]	67.5	67.1	2018
Continuous BOW (Averaging Word Embeddings) [88]	65.2	64.6	2018

Table 4. The comparative experimental results on the MRPC dataset

METHOD	ACCURACY (%)	F1 (%)	YEAR
Our Method	**87.8**	**84.1**	**2020**
BERT-of-Theseus (6-layer; single model) [91]	87.6	83.2	2020
TinyBERT [38]	87.3	82.6	2019
PD-BERT [80]	86.8	81.7	2019
T5-Small [64]	86.6	89.7	2019
Vanilla KD [34]	86.2	80.6	2015
BERT-PKD [74]	85.0	79.9	2019
BiLSTM+ELMo [84]	84.7	78.0	2018
BiLSTM+ELMo+Attn [84]	84.4	78.0	2019
BiLSTM+Attn [84]	83.9	76.2	2018
OpenAI GPT [72]	82.3	-	2019
BiLSTM [84]	81.8	74.3	2018
DisSent [55]	81.7	74.1	2017
Continuous BOW (Averaging Word Embeddings) [88]	81.5	73.4	2018
Skip-thought vectors [42]	80.8	71.7	2015
TF-KLD [37]	80.4	85.9	2013
BiLSTM+CoVe+Attn [84]	80.0	71.8	2018
BiLSTM+CoVe [84]	78.7	71.5	2018
GenSen [73]	78.6	84.4	2018
InferSent [17]	76.2	83.1	2017

can be regarded as a promising solution. One of the potential ideas is to refer to transfer learning: pre-trained the corpus in the knowledge base through the supervised model in advance, and store it by converting it into a vector form. Once new text data is entered and a match or inference is requested, unsupervised similarity measures (e.g., distance calculation, etc.) can be performed directly with the candidate text stored in the repository. Thus, the computation and time overhead at the execution terminal level could be greatly reduced.

In conclusion, in this study, we propose a Siamese Network structure that combines Stacked Residual LSTM with Attention mechanism and Capsule Network for the NLI module in Cyber-Physical Systems. In this network structure, the BERT language model is used as the word embedding to input the text, and then the language is modeled bidirectionally by Stacked Residual LSTM, the important semantic features are captured and weighted by Attention mechanism, and the local features of the context are extracted by Capsule Network. Finally, the vectors represented by the text output of the previous layers are concatenated and then transmitted to the prediction layer, and the final inference result is predicted by softmax. On the tests on SNLI, MultiNLI, and MRPC datasets, our model achieved results of 91.9% and 94.4% (Test and Train accuracy, SNLI), 84.8% and 84.3% (accuracy of Matched and Mismatched, MultiNLI), 87.8% and 84.1% (accuracy and F1 score, MRPC), respectively. Among them, the performance, generalization ability, balance ability between parameter quantity, and performance of the proposed method tested on each data set have also been proven to some extent. This also lays a good foundation for the subsequent exploration of CPS systems with more dimensional

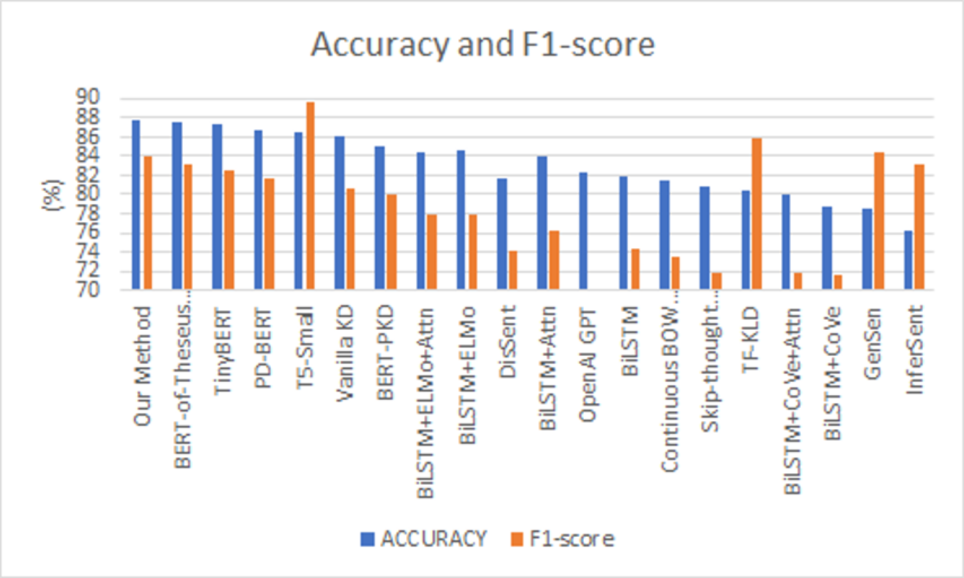

Fig. 12. Visualization of Accuracy and F1-Score on the MRPC dataset

processing capabilities, so that textual data from different sources can be better detected with the relationships entailed in them. In other words, this work is also a beneficial practice about CPS for heterogeneous text data.

7 ACKNOWLEDGEMENT

This research is partly funded by VC Research (VCR 0000059).

REFERENCES

[1] Abacha, A. B., and Demner-Fushman, D. A question-entailment approach to question answering. *BMC bioinformatics 20*, 1 (2019), 511.

[2] Abacha, A. B., Shivade, C., and Demner-Fushman, D. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task* (2019), pp. 370–379.

[3] Adams, R. Textual entailment through extended lexical overlap. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment* (2006), pp. 128–133.

[4] Akhmatova, E. Textual entailment resolution via atomic propositions. In *in Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment* (2005).

[5] Androutsopoulos, I., and Malakasiotis, P. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research 38* (2010), 135–187.

[6] Angeli, G., Premkumar, M. J. J., and Manning, C. D. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (2015), pp. 344–354.

[7] Balazs, J., Marrese-Taylor, E., Loyola, P., and Matsuo, Y. Refining raw sentence representations for textual entailment recognition via attention. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP* (2017), pp. 51–55.

[8] Bar-Haim, R., Dagan, I., Greental, I., and Shnarch, E. Semantic inference at the lexical-syntactic level. In *Proceedings of the National Conference on Artificial Intelligence* (2007), vol. 22, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 871.

[9] Bayer, S., Burger, J., Ferro, L., Henderson, J., and Yeh, E. Mitre's submission to the eu pascal rte challenge. In *In PASCAL. Proc. of the First Challenge Workshop. Recognizing Textual Entailment* (2005).

[10] Borges, L., Martins, B., and Calado, P. Combining similarity features and deep representation learning for stance detection in the context of checking fake news. *Journal of Data and Information Quality (JDIQ) 11*, 3 (2019), 1–26.

[11] Bowman, S., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015), pp. 632–642.

[12] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems* (1994), pp. 737–744.

[13] Chen, P., Guo, W., Chen, Z., Sun, J., and You, L. Gated convolutional neural network for sentence matching. *Proc. Interspeech 2018* (2018), 2853–2857.

[14] Chen, Q., Zhu, X., Ling, Z.-H., Inkpen, D., and Wei, S. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018), pp. 2406–2417.

[15] Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017), pp. 1657–1668.

[16] Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. Recurrent neural network-based sentence encoder with gated attention for natural language inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP* (2017), pp. 36–40.

[17] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017), pp. 670–680.

[18] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (2019), pp. 4171–4186.

[19] Ding, L., Finin, T., Joshi, A., Pan, R., Scott, R., Peng, C. Y., Reddivari, P., Doshi, V., and Sachs, J. Swoogle: A semantic web search and metadata engine. In *Acm Conference on Information Knowledge Management Acm* (2004).

[20] Dolan, B., Brockett, C., and Quirk, C. Microsoft research paraphrase corpus. Microsoft, 2005. Accessed: May 03, 2020.

[21] Du, Q., Zong, C., and Su, K.-Y. Conducting natural language inference with word-pair-dependency and local context. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 19*, 3 (2020), 1–23.

[22] Dzikovska, M. O., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., and Dang, H. T. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (2013), pp. 263–274.

[23] Dzikovska, M. O., Nielsen, R. D., and Leacock, C. The joint student response analysis and recognizing textual entailment challenge: making sense of student responses in educational applications. *Language Resources and Evaluation 50*, 1 (2016), 67–93.

[24] Gao, J., Galley, M., and Li, L. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018), pp. 1371–1374.

[25] Ghaeini, R., Hasan, S. A., Datla, V., Liu, J., Lee, K., Qadir, A., Ling, Y., Prakash, A., Fern, X., and Farri, O. 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, Volume 1 (Long Papers)* (2018), pp. 1460–1469.

[26] Gómez-Rodríguez, C., Alonso-Alonso, I., and Vilares, D. How important is syntactic parsing accuracy? an empirical evaluation on rule-based sentiment analysis. *Artificial Intelligence Review 52*, 3 (2019), 2081–2097.

[27] Gong, Y., Luo, H., and Zhang, J. Natural language inference over interaction space. In *International Conference on Learning Representations* (2018).

[28] Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (2013), IEEE, pp. 6645–6649.

[29] Gupta, A., Kaur, M., Garg, D., and Saini, K. Using variant directional dis (similarity) measures for the task of textual entailment. In *International Conference on Recent Developments in Science, Engineering and Technology* (2017), Springer, pp. 287–297.

[30] Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (2018), pp. 107–112.

[31] Harabagiu, S., and Hickl, A. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (Sydney, Australia, July 2006), Association for Computational Linguistics, pp. 905–912.

[32] Harabagiu, S., Hickl, A., and Lacatusu, F. Satisfying information needs with multi-document summaries. *Information Processing & Management 43*, 6 (2007), 1619–1642.

[33] Heilman, M., and Smith, N. A. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2010), Association for Computational Linguistics, pp. 1011–1019.

[34] Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[35] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 4700–4708.

[36] Im, J., and Cho, S. Distance-based self-attention network for natural language inference. *arXiv preprint arXiv:1712.02047* (2017).

[37] Ji, Y., and Eisenstein, J. Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (2013), pp. 891–896.

[38] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351* (2019).

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

[40] Khot, T., Sabharwal, A., and Clark, P. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[41] Kim, S., Kang, I., and Kwak, N. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 6586–6593.

[42] Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2* (2015), MIT Press, pp. 3294–3302.

[43] Korman, D. Z., Mack, E., Jett, J., and Renear, A. H. Defining textual entailment. *Journal of the Association for Information Science and Technology 69*, 6 (2018), 763–772.

[44] Li, G., Zhang, P., and Jia, C. Attention boosted sequential inference model. *arXiv preprint arXiv:1812.01840* (2018).

[45] Liu, C., Dahlmeier, D., and Ng, H. T. Better evaluation metrics lead to better machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2011), Association for Computational Linguistics, pp. 375–384.

[46] Liu, X., Duh, K., and Gao, J. Stochastic answer networks for natural language inference. *arXiv preprint arXiv:1804.07888* (2018).

[47] Liu, X., He, P., Chen, W., and Gao, J. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 4487–4496.

[48] MacCartney, B., and Manning, C. D. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing* (2007), Association for Computational Linguistics, pp. 193–200.

[49] Matsuyoshi, S. Identification of event and topic for multi-document summarization. In *Human Language Technology* (2016), Springer, p. 304.

[50] McCann, B., Bradbury, J., Xiong, C., and Socher, R. Learned in translation: contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (2017), Curran Associates Inc., pp. 6297–6308.

[51] Mehdad, Y., Negri, M., Cabrio, E., Kouylekov, M. O., and Magnini, B. Edits: An open source framework for recognizing textual entailment. In *Text Analysis Conference (TAC 2009)* (2009).

[52] Moldovan, D., Clark, C., Harabagiu, S., and Maiorano, S. 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* (2003), Association for Computational Linguistics, pp. 87–93.

[53] Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., and Jin, Z. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2016), pp. 130–136.

[54] Munkhdalai, T., and Yu, H. Neural tree indexers for text understanding. In *Proceedings of the conference. Association for Computational Linguistics. Meeting* (2017), vol. 1, NIH Public Access, p. 11.

[55] Nie, A., Bennett, E. D., and Goodman, N. D. Dissent: Sentence representation learning from explicit discourse relations. *arXiv preprint arXiv:1710.04334* (2017).

[56] Nie, Y., and Bansal, M. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP* (2017), pp. 41–45.

[57] Nielsen, R. D., Ward, W., and Martin, J. H. Recognizing entailment in intelligent tutoring systems. *Natural Language Engineering 15*, 4 (2009), 479–501.

[58] Ojokoh, B., and Adebisi, E. A review of question answering systems. *Journal of Web Engineering 17*, 8 (2018), 717–758.

[59] Padó, S., Cer, D., Galley, M., Jurafsky, D., and Manning, C. D. Measuring machine translation quality as semantic equivalence: A metric based on entailment features. *Machine Translation 23*, 2-3 (2009), 181–193.

[60] Pan, B., Yang, Y., Zhao, Z., Zhuang, Y., Cai, D., and He, X. Discourse marker augmented network with reinforcement learning for natural language inference. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018), pp. 989–999.

[61] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018), pp. 2227–2237.

[62] Prakash, A., Hasan, S. A., Lee, K., Datla, V., Qadir, A., Liu, J., and Farri, O. Neural paraphrase generation with stacked residual lstm networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (2016), pp. 2923–2934.

[63] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf* (2018).

[64] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[65] Raina, R., Ng, A. Y., and Manning, C. D. Robust textual inference via learning and abductive reasoning. In *AAAI* (2005), pp. 1099–1105.

[66] Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., and Blunsom, P. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)* (2016).

[67] Romano, L., Kouylekov, M., Szpektor, I., Dagan, I., and Lavelli, A. Investigating a generic paraphrase-based approach for relation extraction. In *11th Conference of the European Chapter of the Association for Computational Linguistics* (2006).

[68] Roy, S., Vieira, T., and Roth, D. Reasoning about quantities in natural language. *Transactions of the Association for Computational Linguistics 3* (2015), 1–13.

[69] Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules. In *Advances in neural information processing systems* (2017), pp. 3856–3866.

[70] Shen, D., Wang, G., Wang, W., Min, M. R., Su, Q., Zhang, Y., Li, C., Henao, R., and Carin, L. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018), pp. 440–450.

[71] Shen, T., Jiang, J., Zhou, T., Pan, S., Long, G., and Zhang, C. Disan: directional self-attention network for rnn/cnn-free language understanding. In *AAAI Conference on Artificial Intelligence 2018* (2018), Association for the Advancement of Artificial Intelligence (AAAI), pp. 5446–5455.

[72] Su, T.-C., and Cheng, H.-C. Sesamebert: Attention for anywhere. *arXiv preprint arXiv:1910.03176* (2019).

[73] Subramanian, S., Trischler, A., Bengio, Y., and Pal, C. J. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations* (2018).

[74] Sun, S., Cheng, Y., Gan, Z., and Liu, J. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019), pp. 4314–4323.

[75] Sztipanovits, J., Koutsoukos, X., Karsai, G., Kottenstette, N., Antsaklis, P., Gupta, V., Goodwine, B., Baras, J., and Wang, S. Toward a science of cyber–physical system integration. *Proceedings of the IEEE 100*, 1 (2011), 29–44.

[76] Tan, C., Wei, F., Wang, W., Lv, W., and Zhou, M. Multiway attention networks for modeling sentence pairs. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (2018), pp. 4411–4417.

[77] Tan, M., Dos Santos, C., Xiang, B., and Zhou, B. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2016), pp. 464–473.

[78] Tay, Y., Luu, A. T., and Hui, S. C. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (2018), pp. 1565–1575.

[79] Tran, N. K., and Niedereée, C. Multihop attention networks for question answer matching. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018), pp. 325–334.

[80] Turc, I., Chang, M.-W., Lee, K., and Toutanova, K. Well-read students learn better: The impact of student initialization on knowledge distillation. *arXiv preprint arXiv:1908.08962* (2019).

[81] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems* (2017), pp. 5998–6008.

[82] Vogelsang, A., Hartig, K., Pudlitz, F., Schlutter, A., and Winkler, J. Supporting the development of cyber-physical systems with natural language processing: A report. In *NLP4RE 2019: 2nd Workshop on Natural Language Processing for Requirements Engineering* (2019). Available Open Access at https://doi.org/10.14279/depositonce-8276.

[83] Vu, H. T., Pham, T.-H., Bai, X., Tanti, M., van der Plas, L., and Gatt, A. LCT-MALTA's submission to RepEval 2017 shared task. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP* (Copenhagen, Denmark, Sept. 2017), Association for Computational Linguistics, pp. 56–60.

[84] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (2018), pp. 353–355.

[85] Wang, S., and Jiang, J. Learning natural language inference with lstm. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2016), pp. 1442–1451.

[86] Wang, Z., Hamza, W., and Florian, R. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (2017), pp. 4144–4150.

[87] Wiesner, S., Gorldt, C., Soeken, M., Thoben, K.-D., and Drechsler, R. Requirements engineering for cyber-physical systems. In *IFIP International Conference on Advances in Production Management Systems* (2014), Springer, pp. 281–288.

[88] Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018), pp. 1112–1122.

[89] Wu, Y., Li, J., Wu, J., and Chang, J. Siamese capsule networks with global and local features for text classification. *Neurocomputing*, 88–98.

[90] Wu, Z., Xu, Y., Yang, Y., Zhang, C., Zhu, X., and Ji, Y. Towards a semantic web of things: a hybrid semantic annotation, extraction, and reasoning framework for cyber-physical system. *Sensors 17*, 2 (2017), 403.

[91] Xu, C., Zhou, W., Ge, T., Wei, F., and Zhou, M. Bert-of-theseus: Compressing bert by progressive module replacing, 2020.

[92] Yang, H., Costa-jussà, M. R., and Fonollosa, J. A. Character-level intra attention network for natural language inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP* (2017), pp. 46–50.

[93] Yang, R., Zhang, J., Gao, X., Ji, F., and Chen, H. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), pp. 4699–4709.

[94] Yin, W., Schütze, H., Xiang, B., and Zhou, B. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics 4* (2016), 259–272.

[95] Yoon, D., Lee, D., and Lee, S. Dynamic self-attention: Computing attention over words dynamically for sentence embedding. *arXiv preprint arXiv:1808.07383* (2018).

[96] Yuret, D., Han, A., and Turgut, Z. Semeval-2010 task 12: Parser evaluation using textual entailments. In *Proceedings of the 5th International Workshop on Semantic Evaluation* (2010), pp. 51–56.

[97] Zaremba, W., and Sutskever, I. Learning to execute. *arXiv preprint arXiv:1410.4615* (2014).

[98] Zhang, K., Chen, E., Liu, Q., Liu, C., and Lv, G. A context-enriched neural network method for recognizing lexical entailment. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

[99] Zhang, Z., Wu, Y., Li, Z., and Zhao, H. Explicit contextual semantics for text comprehension. In *33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33)* (2019), pp. 298–308.

[100] Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., and Zhou, X. Semantics-aware BERT for language understanding. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)* (2020).

[101] Zhao, W., Ye, J., Yang, M., Lei, Z., Zhang, S., and Zhao, Z. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (September 2018).