# LET: Linguistic Knowledge Enhanced Graph Transformer for Chinese Short Text Matching

**Boer Lyu, Lu Chen*, Su Zhu, Kai Yu***

X-LANCE Lab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
State Key Laboratory of Media Convergence Production Technology and Systems, Beijing, China
{boerlv, chenlusz, paul2204, kai.yu}@sjtu.edu.cn

## Abstract

Chinese short text matching is a fundamental task in natural language processing. Existing approaches usually take Chinese characters or words as input tokens. They have two limitations: 1) Some Chinese words are polysemous, and semantic information is not fully utilized. 2) Some models suffer potential issues caused by word segmentation. Here we introduce HowNet as an external knowledge base and propose a Linguistic knowledge Enhanced graph Transformer (LET) to deal with word ambiguity. Additionally, we adopt the word lattice graph as input to maintain multi-granularity information. Our model is also complementary to pre-trained language models. Experimental results on two Chinese datasets show that our models outperform various typical text matching approaches. Ablation study also indicates that both semantic information and multi-granularity information are important for text matching modeling.

## 1 Introduction

Short text matching (STM) is generally regarded as a task of paraphrase identification or sentence semantic matching. Given a pair of sentences, the goal of matching models is to predict their semantic similarity. It is widely used in question answer systems (Liu, Rong, and Xiong 2018) and dialogue systems (Gao et al. 2019; Yu et al. 2014).

Recent years have seen great progress in deep learning methods for text matching (Mueller and Thyagarajan 2016; Chen et al. 2017; Gong, Luo, and Zhang 2018; Lan and Xu 2018). However, almost all of these models are initially proposed for English text matching. For Chinese language tasks, early work utilizes Chinese characters as input to the model, or first segment each sentence into words, and then take these words as input tokens. Although character-based models can overcome the problem of data sparsity to some degree (Li et al. 2019), the main drawback is that explicit word information is not fully utilized, which has been demonstrated to be useful for semantic matching (Li et al. 2020b).

However, a large number of Chinese words are polysemous, which brings great difficulties to semantic understanding (Xu et al. 2016). Word polysemy in short text is more an issue than that in long text because short text usually has less contextual information, so it is extremely hard for models to capture the correct meaning. As shown in Fig. 1, the word in red in sentence-1 actually has two meanings: one is to describe bragging (*exaggeration*) and another is *moisture*. Intuitively, if other words in the context have similar or related meanings, the probability of them will increase. To integrate semantic information of words, we introduce HowNet (Dong and Dong 2003) as an external knowledge base. In the view of HowNet, words may have multiple senses/meanings and each sense has several sememes to represent it. For instance, the first sense *exaggeration* indicates some boast information in his words. Therefore, it has sememes `information` and `boast`. Similarly, we can also find the sememe `boast` describing the sense *brag* which belongs to the word "ChuiNiu (bragging)" in sentence-2. In this way, model can better determine the sense of words and perceive that two sentences probably have the same meaning.

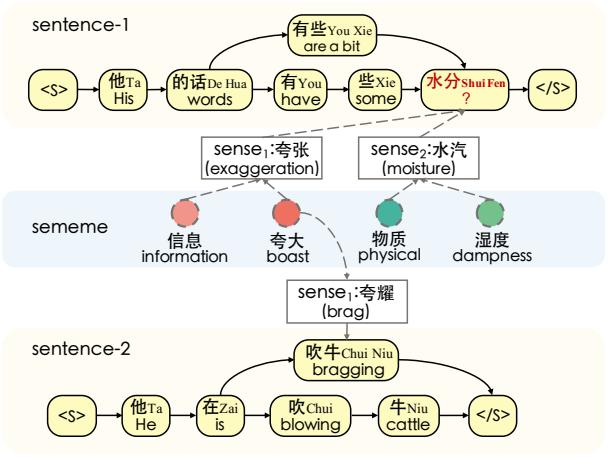


Figure 1: An example of word segmentation and the potential word ambiguity.

Furthermore, word-based models often encounter some potential issues caused by word segmentation. If the word segmentation fails to output "ChuiNiu (bragging)" in sentence-2, we will lose useful sense information. In Chinese, "Chui (blowing)" "Niu (cattle)" is a bad segmentation, which deviates the correct meaning of "ChuiNiu (bragging)". To tackle this problem, many researchers propose word lattice

*The corresponding authors are Lu Chen and Kai Yu.

graphs (Lai et al. 2019; Li et al. 2020a; Chen et al. 2020b), where they retain words existing in the word bank so that various segmentation paths are kept. It has been shown that multi-granularity information is important for text matching.

In this paper, we propose a Linguistic knowledge Enhanced graph Transformer (LET) to consider both semantic information and multi-granularity information. LET takes a pair of word lattice graphs as input. Since keeping all possible words will introduce a lot of noise, we use several segmentation paths to form our lattice graph and construct a set of senses according to the word. Based on HowNet, each sense has several sememes to represent it. In the input module, starting from the pre-trained sememe embeddings provided by OpenHowNet (Qi et al. 2019), we obtain the initial sense representation using a multi-dimensional graph attention transformer (MD-GAT, see Sec. 3.1). Also, we get the initial word representation by aggregating features from the character-level transformer encoder using an Att-Pooling (see Sec. 4.1). Then it is followed by SaGT layers (see Sec. 4.2), which fuse the information between words and semantics. In each layer, we first update the nodes' sense representation and then updates word representation using MD-GAT. As for the sentence matching layer (see Sec. 4.3), we convert word representation to character level and share the message between texts. Moreover, LET can be combined with pre-trained language models, e.g. BERT (Devlin et al. 2019). It can be regarded as a method to integrate word and sense information into pre-trained language models during the fine-tuning phase.

Contributions in this work are summarized as: a) We propose a novel enhanced graph transformer using linguistic knowledge to moderate word ambiguity. b) Empirical study on two Chinese datasets shows that our model outperforms not only typical text matching models but also the pre-trained model BERT as well as some variants of BERT. c) We demonstrate that both semantic information and multi-granularity information are important for text matching modeling, especially on shorter texts.

## 2 Related Work

**Deep Text Matching Models** based on deep learning have been widely adopted for short text matching. They can fall into two categories: representation-based methods (He et al. 2016; Lai et al. 2019) and interaction-based methods (Wang, Hamza, and Florian 2017; Chen et al. 2017). Most representation-based methods are based on Siamese architecture, which has two symmetrical networks (e.g. LSTMs and CNNs) to extract high-level features from two sentences. Then, these features are compared to predict text similarity. Interaction-based models incorporate interactions features between all word pairs in two sentences. They generally perform better than representation-based methods. Our proposed method belongs to interaction-based methods.

**Pre-trained Language Models**, e.g. BERT, have shown its powerful performance on various natural language processing (NLP) tasks including text matching. For Chinese text matching, BERT takes a pair of sentences as input and each Chinese character is a separated input token. It has ignored word information. To tackle this problem, some Chinese variants of original BERT have been proposed, e.g. BERT-wwm (Cui et al. 2019) and ERNIE (Sun et al. 2019). They take the word information into consideration based on the whole word masking mechanism during pre-training. However, the pre-training process of a word-considered BERT requires a lot of time and resources. Thus, Our model takes pre-trained language model as initialization and utilizes word information to fine-tune the model.

## 3 Background

In this section, we introduce graph attention networks (GATs) and HowNet, which are the basis of our proposed models in the next section.

### 3.1 Graph Attention Networks

Graph neural networks (GNNs) (Scarselli et al. 2008) are widely applied in various NLP tasks, such as text classifcation (Yao, Mao, and Luo 2019), text generation (Zhao et al. 2020), dialogue policy optimization (Chen et al. 2018c,b, 2019, 2020c) and dialogue state tracking (Chen et al. 2020a; Zhu et al. 2020), etc. GAT is a special type of GNN that operates on graph-structured data with attention mechanisms. Given a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the set of nodes $x_i$ and the set of edges, respectively. $\mathcal{N}^+(x_i)$ is the set including the node $x_i$ itself and the nodes which are directly connected by $x_i$.

Each node $x_i$ in the graph has an initial feature vector $\mathbf{h}_i^0 \in \mathbb{R}^d$, where $d$ is the feature dimension. The representation of each node is iteratively updated by the graph attention operation. At the $l$-th step, each node $x_i$ aggregates context information by attending over its neighbors and itself. The updated representation $\mathbf{h}_i^l$ is calculated by the weighted average of the connected nodes,

$$\mathbf{h}_i^l = \sigma\left(\sum_{x_j \in \mathcal{N}^+(x_i)} \alpha_{ij}^l \cdot \left(\mathbf{W}^l \mathbf{h}_j^{l-1}\right)\right), \tag{1}$$

where $\mathbf{W}^l \in \mathbb{R}^{d\times d}$ is a learnable parameter, and $\sigma(\cdot)$ is a nonlinear activation function, e.g. ReLU. The attention coefficient $\alpha_{ij}^l$ is the normalized similarity of the embedding between the two nodes $x_i$ and $x_j$ in a unified space, i.e.

$$\begin{aligned}\alpha_{ij}^l &= \text{softmax}_j\, f_{sim}^l\left(\mathbf{h}_i^{l-1},\, \mathbf{h}_j^{l-1}\right) \\ &= \text{softmax}_j\left(\mathbf{W}_q^l \mathbf{h}_i^{l-1}\right)^T \left(\mathbf{W}_k^l \mathbf{h}_j^{l-1}\right),\end{aligned} \tag{2}$$

where $\mathbf{W}_q^l$ and $\mathbf{W}_k^l \in \mathbb{R}^{d\times d}$ are learnable parameters for projections.

Note that, in Eq. (2), $\alpha_{ij}^l$ is a scalar, which means that all dimensions in $\mathbf{h}_j^{l-1}$ are treated equally. This may limit the capacity to model complex dependencies. Following Shen et al. (2018), we replace the vanilla attention with multi-dimensional attention. Instead of computing a single scalar score, for each embedding $\mathbf{h}_j^{l-1}$, it first computes a feature-wise score vector, and then normalizes it with feature-wised multi-dimensional softmax (MD-softmax),

$$\boldsymbol{\alpha}_{ij}^l = \text{MD-softmax}_j\left(\hat{\alpha}_{ij}^l + f_m^l\left(\mathbf{h}_j^{l-1}\right)\right), \tag{3}$$

where $\hat{\alpha}_{ij}^l$ is a scalar calculated by the similarity function $f_{sim}^l(\cdot)$ in Eq. (2), and $f_m^l(\cdot)$ is a vector. The addition in above equation means the scalar will be added to every element of the vector. $\hat{\alpha}_{ij}^l$ is utilized to model the pair-wised dependency of two nodes, while $f_m^l(\cdot)$ is used to estimate the contribution of each feature dimension of $\mathbf{h}_j^{l-1}$,

$$f_m^l(\mathbf{h}_j^{l-1}) = \mathbf{W}_2^l \sigma\left(\mathbf{W}_1^l \mathbf{h}_j^{l-1} + \mathbf{b}_1^l\right) + \mathbf{b}_2^l, \quad (4)$$

where $\mathbf{W}_1^l$, $\mathbf{W}_2^l$, $\mathbf{b}_1^l$ and $\mathbf{b}_2^l$ are learnable parameters. With the score vector $\boldsymbol{\alpha}_{ij}^l$, Eq. (1) will be accordingly revised as

$$\mathbf{h}_i^l = \sigma\left(\sum_{x_j \in \mathcal{N}^+(x_i)} \boldsymbol{\alpha}_{ij}^l \odot \left(\mathbf{W}^l \mathbf{h}_j^{l-1}\right)\right), \quad (5)$$

where $\odot$ represents element-wise product of two vectors. For brevity, we use MD-GAT$(\cdot)$ to denote the updating process using multi-dimensional attention mechanism, and rewrite Eq. (5) as follows,

$$\mathbf{h}_i^l = \text{MD-GAT}\left(\mathbf{h}_i^{l-1}, \left\{\mathbf{h}_j^{l-1} | x_j \in \mathcal{N}^+(x_i)\right\}\right). \quad (6)$$

After $L$ steps of updating, each node will finally have a context-aware representation $\mathbf{h}_i^L$. In order to achieve a stable training process, we also employ a residual connection followed by a layer normalization between two graph attention layers.

### 3.2 HowNet

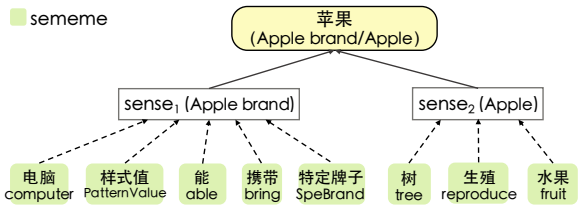


Figure 2: An example of the HowNet structure.

HowNet (Dong and Dong 2003) is an external knowledge base that manually annotates each Chinese word sense with one or more relevant sememes. The philosophy of HowNet regards sememe as an atomic semantic unit. Different from WordNet (Miller 1995), it emphasizes that the parts and attributes of a concept can be well represented by sememes. HowNet has been widely utilized in many NLP tasks such as word similarity computation (Liu 2002), sentiment analysis (Fu et al. 2013), word representation learning (Niu et al. 2017) and language modeling (Gu et al. 2018).

An example is illustrated in Fig. 2. The word "Apple" has two senses including *Apple Brand* and *Apple*. The sense *Apple Brand* has five sememes including `computer`, `PatternValue`, `able`, `bring` and `SpecificBrand`, which describe the exact meaning of sense.

## 4 Proposed Approach

First, we define the Chinese short text matching task in a formal way. Given two Chinese sentences $\mathcal{C}^a = \{c_1^a, c_2^a, \cdots, c_{T_a}^a\}$ and $\mathcal{C}^b = \{c_1^b, c_2^b, \cdots, c_{T_b}^b\}$, the goal of a text matching model $f(\mathcal{C}^a, \mathcal{C}^b)$ is to predict whether the semantic meaning of $\mathcal{C}^a$ and $\mathcal{C}^b$ is equal. Here, $c_t^a$ and $c_{t'}^b$ represent the $t$-th and $t'$-th Chinese character in two sentences respectively, and $T_a$ and $T_b$ denote the number of characters in the sentences.

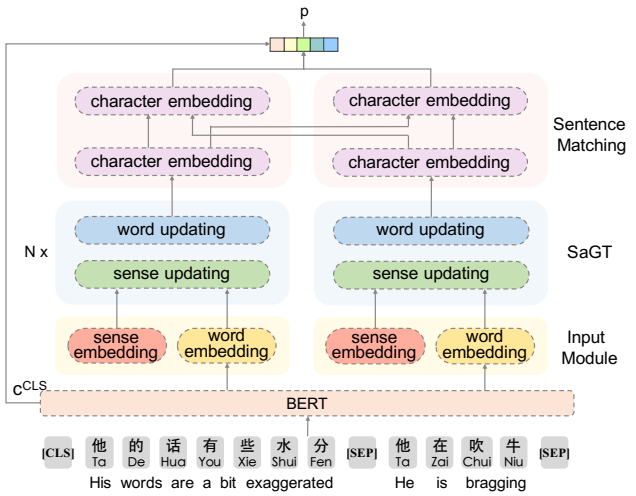


Figure 3: The framework of our proposed LET model.

In this paper, we propose a linguistic knowledge enhanced matching model. Instead of segmenting each sentence into a word sequence, we use three segmentation tools and keep these segmentation paths to form a word lattice graph $G = (\mathcal{V}, \mathcal{E})$ (see Fig. 4 (a)). $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each node $x_i \in \mathcal{V}$ corresponds to a word $w_i$ which is a character subsequence starting from the $t_1$-th character to the $t_2$-th character in the sentence. As introduced in Sec. 1, we can obtain all senses of a word $w_i$ by retrieving the HowNet.

For two nodes $x_i \in \mathcal{V}$ and $x_j \in \mathcal{V}$, if $x_i$ is adjacent to $x_j$ in the original sentence, then there is an edge between them. $\mathcal{N}_{fw}^+(x_i)$ is the set including $x_i$ itself and all its reachable nodes in its forward direction, while $\mathcal{N}_{bw}^+(x_i)$ is the set including $x_i$ itself and all its reachable nodes in its backward direction.

Thus for each sample, we have two graphs $G^a = (\mathcal{V}^a, \mathcal{E}^a)$ and $G^b = (\mathcal{V}^b, \mathcal{E}^b)$, and our graph matching model is to predict their similarity. As shown in Fig. 3, LET consists of four components: input module, semantic-aware graph transformer (SaGT), sentence matching layer and relation classifier. The input module outputs the initial contextual representation for each word $w_i$ and the initial semantic representation for each sense. The semantic-aware graph transformer iteratively updates the word representation and sense representation, and fuses useful information from each other. The sentence matching layer first incorporates word representation into character level, and then matches two character sequences with the bilateral multi-perspective matching mechanism. The relation classifier takes the sentence vectors as input and predicts the relation of two sentences.

### 4.1 Input Module

**Contextual Word Embedding** For each node $x_i$ in graphs, the initial representation of word $w_i$ is the attentive pooling of contextual character representations. Concretely, we first concat the original character-level sentences to form a new sequence $\mathcal{C} = \{[\text{CLS}], c_1^a, \cdots, c_{T_a}^a, [\text{SEP}], c_1^b, \cdots, c_{T_b}^b, [\text{SEP}]\}$, and then feed them to the BERT model to obtain the contextual representations for each character $\{\mathbf{c}^{\text{CLS}}, \mathbf{c}_1^a, \cdots, \mathbf{c}_{T_a}^a, \mathbf{c}^{\text{SEP}}, \mathbf{c}_1^b, \cdots, \mathbf{c}_{T_b}^b, \mathbf{c}^{\text{SEP}}\}$. Assuming that the word $w_i$ consists of some consecutive character tokens $\{c_{t_1}, c_{t_1+1}, \cdots, c_{t_2}\}$[1], a feature-wised score vector is calculated with a feed forward network (FFN) with two layers for each character $c_k$ $(t_1 \le k \le t_2)$, and then normalized with a feature-wised multi-dimensional softmax (MD-softmax),

$$\mathbf{u}_k = \text{MD-softmax}_k\left(\text{FFN}(\mathbf{c}_k)\right), \tag{7}$$

The corresponding character embedding $\mathbf{c}_k$ is weighted with the normalized scores $\mathbf{u}_k$ to obtain the contextual word embedding,

$$\mathbf{v}_i = \sum_{k=t_1}^{t_2} \mathbf{u}_k \odot \mathbf{c}_k, \tag{8}$$

For brevity, we use Att-Pooling$(\cdot)$ to rewrite Eq. (7) and Eq. (8) for short, i.e.

$$\mathbf{v}_i = \text{Att-Pooling}\left(\{\mathbf{c}_k | t_1 \le k \le t_2\}\right). \tag{9}$$

**Sense Embedding** The word embedding $\mathbf{v}_i$ described in Sec. 4.1 contains only contextual character information, which may suffer from the issue of polysemy in Chinese. In this paper, we incorporate HowNet as an external knowledge base to express the semantic information of words.

For each word $w_i$, we denote the set of senses as $\mathcal{S}^{(w_i)} = \{s_{i,1}, s_{i,2}, \cdots, s_{i,K}\}$. $s_{i,k}$ is the $k$-th sense of $w_i$ and we denote its corresponding sememes as $\mathcal{O}^{(s_{i,k})} = \{o_{i,k}^1, o_{i,k}^2, \cdots, o_{i,k}^M\}$. In order to get the embedding $\mathbf{s}_{i,k}$ for each sense $s_{i,k}$, we first obtain the representation $\mathbf{o}_{i,k}^m$ for each sememe $o_{i,k}^m$ with multi-dimensional attention function,

$$\mathbf{o}_{i,k}^m = \text{MD-GAT}\left(\mathbf{e}_{i,k}^m, \left\{\mathbf{e}_{i,k}^{m'} | o_{i,k}^{m'} \in \mathcal{O}^{(s_{i,k})}\right\}\right), \tag{10}$$

where $\mathbf{e}_{i,k}^m$ is the embedding vector for sememe $o_{i,k}^m$ produced through the Sememe Attention over Target model (SAT) (Niu et al. 2017). Then, for each sense $s_{i,k}$, its embedding $\mathbf{s}_{i,k}$ is obtained with attentive pooling of all sememe representations,

$$\mathbf{s}_{i,k} = \text{Att-Pooling}\left(\left\{\mathbf{o}_{i,k}^m | o_{i,k}^m \in \mathcal{O}^{(s_{i,k})}\right\}\right). \tag{11}$$

### 4.2 Semantic-aware Graph Transformer

For each node $x_i$ in the graph, the word embedding $\mathbf{v}_i$ only contains the contextual information while the sense embedding $\mathbf{s}_{i,k}$ only contains linguistic knowledge. In order to harvest useful information from each other, we propose a semantic-aware graph transformer (SaGT). It first takes $\mathbf{v}_i$ and $\mathbf{s}_{i,k}$ as initial word representation $\mathbf{h}_i^0$ for word $w_i$ and initial sense representation $\mathbf{g}_{i,k}^0$ for sense $s_{i,k}$ respectively, and then iteratively updates them with two sub-steps.

[1]For brevity, the superscript of $c_k$ $(t_1 \le k \le t_2)$ is omitted.

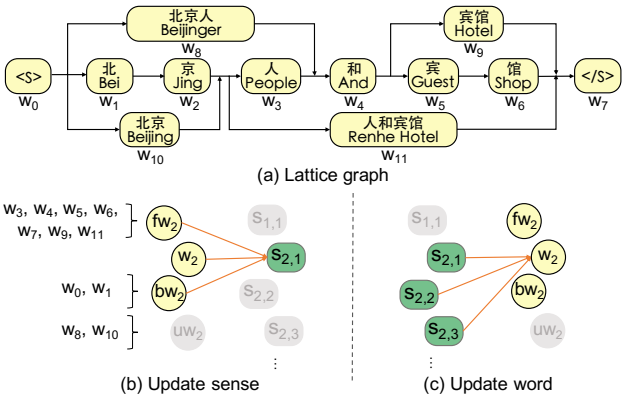


Figure 4: (a) is an example of lattice graph. (b) shows the process of sense updating. $\text{fw}_2$ and $\text{bw}_2$ refer to the words in forward and backward directions of $\text{w}_2$ respectively. $\text{uw}_2$ is the words that $\text{w}_2$ cannot reach. (c) is word updating; we will not update the corresponding word representation if the word is not in HowNet.

**Updating Sense Representation** At $l$-th iteration, the first sub-step is to update sense representation from $\mathbf{g}_{i,k}^{l-1}$ to $\mathbf{g}_{i,k}^l$. For a word with multiple senses, which sense should be used is usually determined by the context in the sentence. Therefore, when updating the representation, each sense will first aggregate useful information from words in forward and backward directions of $x_i$,

$$\begin{aligned}
\mathbf{m}_{i,k}^{l,fw} &= \text{MD-GAT}\left(\mathbf{g}_{i,k}^{l-1}, \left\{\mathbf{h}_j^{l-1} | x_j \in \mathcal{N}_{fw}^+(x_i)\right\}\right), \\
\mathbf{m}_{i,k}^{l,bw} &= \text{MD-GAT}\left(\mathbf{g}_{i,k}^{l-1}, \left\{\mathbf{h}_j^{l-1} | x_j \in \mathcal{N}_{bw}^+(x_i)\right\}\right),
\end{aligned} \tag{12}$$

where two multi-dimensional attention functions MD-GAT$(\cdot)$ have different parameters. Based on $\mathbf{m}_{i,k}^l = [\mathbf{m}_{i,k}^{l,fw}, \mathbf{m}_{i,k}^{l,bw}]$ [2], each sense updates its representation with a gate recurrent unit (GRU) (Cho et al. 2014),

$$\mathbf{g}_{i,k}^l = \text{GRU}\left(\mathbf{g}_{i,k}^{l-1}, \mathbf{m}_{i,k}^l\right). \tag{13}$$

It is notable that we don't directly use $\mathbf{m}_{i,k}^l$ as the new representation $\mathbf{g}_{i,k}^l$ of sense $s_{i,k}$. The reason is that $\mathbf{m}_{i,k}^l$ only contains contextual information, and we need to utilize a gate, e.g. GRU, to control the fusion of contextual information and semantic information.

**Updating Word Representation** The second sub-step is to update the word representation from $\mathbf{h}_i^{l-1}$ to $\mathbf{h}_i^l$ based on the updated sense representations $\mathbf{g}_{i,k}^l$ $(1 \le k \le K)$. The word $w_i$ first obtains semantic information from its sense representations with the multi-dimensional attention,

$$\mathbf{q}_i^l = \text{MD-GAT}\left(\mathbf{h}_i^{l-1}, \left\{\mathbf{g}_{i,k}^l | s_{i,k} \in \mathcal{S}^{(w_i)}\right\}\right), \tag{14}$$

[2]$[\cdot,\cdot]$ denotes the concatenation of vectors.

and then updates its representation with a GRU:

$$\mathbf{h}_i^l = \text{GRU}\left(\mathbf{h}_i^{l-1},\ \mathbf{q}_i^l\right). \tag{15}$$

The above GRU function and the GRU function in Eq. (13) have different parameters.

After multiple iterations, the final word representation $\mathbf{h}_i^L$ contains not only contextual word information but also semantic knowledge. For each sentence, we use $\mathbf{h}_i^a$ and $\mathbf{h}_i^b$ to denote the final word representation respectively.

### 4.3 Sentence Matching Layer

After obtaining the semantic knowledge enhanced word representation $\mathbf{h}_i^a$ and $\mathbf{h}_i^b$ for each sentence, we incorporate this word information into characters. Without loss of generality, we will use characters in sentence $\mathcal{C}^a$ to introduce the process. For each character $c_t^a$, we obtain $\hat{\mathbf{c}}_t^a$ by pooling the useful word information,

$$\hat{\mathbf{c}}_t^a = \text{Att-Pooling}\left(\left\{\mathbf{h}_i^a | w_i^a \in \mathcal{W}^{(c_t^a)}\right\}\right), \tag{16}$$

where $\mathcal{W}^{(c_t^a)}$ is a set including words which contain the character $c_t^a$. The semantic knowledge enhanced character representation $\mathbf{y}_t$ is therefore obtained by

$$\mathbf{y}_t^a = \text{LayerNorm}\left(\mathbf{c}_t^a + \hat{\mathbf{c}}_t^a\right), \tag{17}$$

where LayerNorm$(\cdot)$ denotes layer normalization, and $\mathbf{c}_t^a$ is the contextual character representation obtained using BERT described in Sec. 4.1.

For each character $c_t^a$, it aggregates information from sentence $\mathcal{C}^a$ and $\mathcal{C}^b$ respectively using multi-dimensional attention,

$$\begin{aligned}\mathbf{m}_t^{self} &= \text{MD-GAT}\left(\mathbf{y}_t^a,\ \{\mathbf{y}_{t'}^a | c_{t'}^a \in \mathcal{C}^a\}\right),\\ \mathbf{m}_t^{cross} &= \text{MD-GAT}\left(\mathbf{y}_t^a,\ \{\mathbf{y}_{t'}^b | c_{t'}^b \in \mathcal{C}^b\}\right).\end{aligned} \tag{18}$$

The above multi-dimensional attention functions MD-GAT$(\cdot)$ share same parameters. With this sharing mechanism, the model has a nice property that, when the two sentences are perfectly matched, we have $\mathbf{m}_t^{self} \approx \mathbf{m}_t^{cross}$.

We utilize the multi-perspective cosine distance (Wang, Hamza, and Florian 2017) to compare $\mathbf{m}_t^{self}$ and $\mathbf{m}_t^{cross}$,

$$d_k = \text{cosine}\left(\mathbf{w}_k^{cos} \odot \mathbf{m}_t^{self}, \mathbf{w}_k^{cos} \odot \mathbf{m}_t^{cross}\right), \tag{19}$$

where $k \in \{1, 2, \cdots, P\}$ ($P$ is number of perspectives). $\mathbf{w}_k^{cos}$ is a parameter vector, which assigns different weights to different dimensions of messages. With $P$ distances $d_1, d_2, \cdots, d_P$, we can obtain the final character representation,

$$\hat{\mathbf{y}}_t^a = \text{FFN}\left(\left[\mathbf{m}_t^{self}, \mathbf{d}_t\right]\right), \tag{20}$$

where $\mathbf{d}_t \triangleq [d_1, d_2, \cdots, d_P]$, and FFN$(\cdot)$ is a feed forward network with two layers.

Similarly, we can obtain the final character representation $\hat{\mathbf{y}}_t^b$ for each character $c_t^b$ in sentence $\mathcal{C}^b$. Note that the final character representation contains three kinds of information: contextual information, word and sense knowledge, and character-level similarity. For each sentence $\mathcal{C}^a$ or $\mathcal{C}^b$, the sentence representation vector $\mathbf{r}^a$ or $\mathbf{r}^b$ is obtained using the attentive pooling of all final character representations for the sentence.

### 4.4 Relation Classifier

With two sentence vectors $\mathbf{r}^a$, $\mathbf{r}^b$, and the vector $\mathbf{c}^{\text{CLS}}$ obtained with BERT, our model will predict the similarity of two sentences,

$$p = \text{FFN}\left(\left[\mathbf{c}^{\text{CLS}}, \mathbf{r}^a, \mathbf{r}^b, \mathbf{r}^a \odot \mathbf{r}^b, |\mathbf{r}^a - \mathbf{r}^b|\right]\right), \tag{21}$$

where FFN$(\cdot)$ is a feed forward network with two hidden layers and a sigmoid activation after output layer.

With $N$ training samples $\{\mathcal{C}_i^a, \mathcal{C}_i^b, y_i\}_{i=1}^N$, the training object is to minimize the binary cross-entropy loss,

$$\mathcal{L} = -\sum_{i=1}^{N} \left(y_i \log\left(p_i\right) + \left(1 - y_i\right) \log\left(1 - p_i\right)\right), \tag{22}$$

where $y_i \in \{0, 1\}$ is the label of the $i$-th training sample and $p_i \in [0, 1]$ is the prediction of our model taking the sentence pair $\{\mathcal{C}_i^a, \mathcal{C}_i^b\}$ as input.

## 5 Experiments

### 5.1 Experimental Setup

**Dataset** We conduct experiments on two Chinese short text matching datasets: LCQMC (Liu et al. 2018) and BQ (Chen et al. 2018a).

LCQMC is a question matching corpus with large-scale open domain. It consists of 260068 Chinese sentence pairs including 238766 training samples, 8802 development samples and 12500 test samples. Each pair is associated with a binary label indicating whether two sentences have the same meaning or share the same intention. Positive samples are 30% more than negative samples.

BQ is a domain-specific large-scale corpus for bank question matching. It consists of 120000 Chinese sentence pairs including 100000 training samples, 10000 development samples and 10000 test samples. Each pair is also associated with a binary label indicating whether two sentences have the same meaning. The number of positive and negative samples are the same.

**Evaluation metrics** For each dataset, the accuracy (ACC.) and F1 score are used as the evaluation metrics. ACC. is the percentage of correctly classified examples. F1 score of matching is the harmonic mean of the precision and recall.

**Hyper-parameters** The input word lattice graphs are produced by the combination of three segmentation tools: jieba (Sun 2012), pkuseg (Luo et al. 2019) and thulac (Li and Sun 2009). We use the pre-trained sememe embedding provided by OpenHowNet (Qi et al. 2019) with 200 dimensions. The number of graph updating steps/layers $L$ is 2 on both datasets, and the number of perspectives $P$ is 20. The dimensions of both word and sense representation are 128. The hidden size is also 128. The dropout rate for all hidden layers is 0.2. The model is trained by RMSProp with an initial learning rate of 0.0005 and a warmup rate of 0.1. The learning rate of BERT layer is multiplied by an additional factor of 0.1. As for batch size, we use 32 for LCQMC and 64 for BQ. [3]

[3] Our code is available at https://github.com/lbe0613/LET.

| Models | Pre-Training | Interaction | BQ | | LCQMC | |
|---|---|---|---|---|---|---|
| | | | ACC. | F1 | ACC. | F1 |
| Text-CNN(He et al. 2016) | × | × | 68.52 | 69.17 | 72.80 | 75.70 |
| BiLSTM(Mueller and Thyagarajan 2016) | × | × | 73.51 | 72.68 | 76.10 | 78.90 |
| Lattice-CNN (Lai et al. 2019) | × | × | 78.20 | 78.30 | 82.14 | 82.41 |
| BiMPM (Wang, Hamza, and Florian 2017) | × | √ | 81.85 | 81.73 | 83.30 | 84.90 |
| ESIM (Chen et al. 2017) | × | √ | 81.93 | 81.87 | 82.58 | 84.49 |
| **LET** (Ours) | × | √ | **83.22** | **83.03** | **84.81** | **86.08** |
| BERT-wwm (Cui et al. 2019) | √ | √ | 84.89 | 84.29 | 86.80 | 87.78 |
| BERT-wwm-ext (Cui et al. 2019) | √ | √ | 84.71 | 83.94 | 86.68 | 87.71 |
| ERNIE (Sun et al. 2019) | √ | √ | 84.67 | 84.20 | 87.04 | 88.06 |
| BERT(Devlin et al. 2019) | √ | √ | 84.50 | 84.00 | 85.73 | 86.86 |
| **LET-BERT** (Ours) | √ | √ | **85.30** | **84.98** | **88.38** | **88.85** |

Table 1: Performance of various models on LCQMC and BQ test datasets. The results are average scores using 5 different seeds. All the improvements over baselines are statistically significant ($p < 0.05$).

## 5.2 Main Results

We compare our models with three types of baselines: representation-based models, interaction-based models and BERT-based models. The results are summarized in Table 1. All the experiments in Table 1 and Table 2 are running five times using different seeds and we report the **average** scores to ensure the reliability of results. For the baselines, we run them ourselves using the parameters mentioned in Cui et al. (2019).

**Representation-based models** include three baselines Text-CNN, BiLSTM and Lattice-CNN. Text-CNN (He et al. 2016) is one type of Siamese architecture with Convolutional Neural Networks (CNNs) used for encoding each sentence. BiLSTM (Mueller and Thyagarajan 2016) is another type of Siamese architecture with Bi-directional Long Short Term Memory (BiLSTM) used for encoding each sentence. Lattice-CNN (Lai et al. 2019) is also proposed to deal with the potential issue of Chinese word segmentation. It takes word lattice as input and pooling mechanisms are utilized to merge the feature vectors produced by multiple CNN kernels over different $n$-gram contexts of each node in the lattice graph.

**Interaction-based models** include two baselines: BiMPM and ESIM. BiMPM (Wang, Hamza, and Florian 2017) is a bilateral multi-perspective matching model. It encodes each sentence with BiLSTM, and matches two sentences from multi-perspectives. BiMPM performs very well on some natural language inference (NLI) tasks. There are two BiLSTMs in ESIM (Chen et al. 2017). The first one is to encode sentences, and the other is to fuse the word alignment information between two sentences. ESIM achieves state-of-the-art results on various matching tasks. In order to be comparable with the above models, we also employ a model where BERT in Fig. 3 is replaced by a traditional character-level transformer encoder, which is denoted as LET.

The results of the above models are shown in the first part of Table 1. We can find that our model LET outperforms all baselines on both datasets. More specifically, the performance of LET is better than that of Lattice-CNN. Although they both utilize word lattices, Lattice-CNN only focuses on local information while our model can utilize global information. Besides, our model incorporates semantic messages between sentences, which significantly improves model performance. As for interaction-based models, although they both use the multi-perspective matching mechanism, LET outperforms BiMPM and ESIM. It shows the utilization of word lattice with our graph neural networks is powerful.

**BERT-based models** include four baselines: BERT, BERT-wwm, BERT-wwm-ext and ERNIE. We compare them with our presented model LET-BERT. BERT is the official Chinese BERT model released by Google. BERT-wwm is a Chinese BERT with whole word masking mechanism used during pre-training. BERT-wwm-ext is a variant of BERT-wwm with more training data and training steps. ERNIE is designed to learn language representation enhanced by knowledge masking strategies, which include entity-level masking and phrase-level masking. LET-BERT is our proposed LET model where BERT is used as a character level encoder.

The results are shown in the second part of Table 1. We can find that the three variants of BERT (BERT-wwm, BERT-wwn-ext, ERNIE) all surpass the original BERT, which suggests using word level information during pre-training is important for Chinese matching tasks. Our model LET-BERT performs better than all these BERT-based models. Compared with the baseline BERT which has the same initialization parameters, the ACC. of LET-BERT on BQ and LCQMC is increased by 0.8% and 2.65%, respectively. It shows that utilizing sense information during fine-tuning phrases with LET is an effective way to boost the performance of BERT for Chinese semantic matching.

We also compare results with K-BERT (Liu et al. 2020), which regards information in HowNet as triples {word, contain, sememes} to enhance BERT, introducing soft position and visible matrix during the fine-tuning and inferring phases. The reported ACC. for the LCQMC test set of K-BERT is 86.9%. Our LET-BERT is 1.48% better than that. Different from K-BERT, we focus on fusing useful information between word and sense.

| Seg. | Sense | ACC. | F1 |
|---|---|---|---|
| jieba | √ | 87.84 | 88.47 |
| pkuseg | √ | 87.72 | 88.40 |
| thulac | √ | 87.50 | 88.27 |
| lattice | √ | **88.38** | **88.85** |
| lattice | × | 87.68 | 88.40 |

Table 2: Performance of using different segmentation on LCQMC test dataset.

| text length | number of samples | ACC. | | RER(%) |
|---|---|---|---|---|
| | | w/o sense | sense | |
| $< 16$ | 2793 | 88.99 | 90.05 | **9.63** |
| $16 - 18$ | 3035 | 88.49 | 89.25 | 6.60 |
| $19 - 22$ | 3667 | 88.58 | 89.04 | 4.03 |
| $> 22$ | 3005 | 84.53 | 85.13 | 3.88 |

Table 3: Influences of text length on LCQMC test dataset. Relative error reduction (RER) is calculated by $\frac{\text{sense}-\text{w/o sense}}{100-\text{w/o sense}} \times 100\%$.

### 5.3 Analysis

In our view, both multi-granularity information and semantic information are important for LET. If the segmentation does not contain the correct word, our semantic information will not exert the most significant advantage.

Firstly, to explore the impact of using different segmentation inputs, we carry out experiments using LET-BERT on LCQMC test set. As shown in Table 2, when incorporating sense information, improvement can be observed between lattice-based model (the fourth row) and word-based models: jieba, pkuseg and thulac. The improvements of lattice with sense over other models in Table 2 are all statistically significant ($p < 0.05$). The possible reason is that lattice-based models can reduce word segmentation errors to make predictions more accurate.

Secondly, we design an experiment to demonstrate the effectiveness of incorporating HowNet to express the semantic information of words. In the comparative model without HowNet knowledge, the sense updating module in SaGT is removed, and we update word representation only by a multi-dimensional self-attention. The last two rows in Table 2 list the results of combined segmentation (lattice) with and without sense information. The performance of integrating sense information is better than using only word representation. More specifically, the average absolute improvement in ACC. and F1 scores are 0.7% and 0.45%, respectively, which indicates that LET has the ability to obtain semantic information from HowNet to improve the model's performance. Besides, compared with using a single word segmentation tool, semantic information performs better on lattice-based model. The probable reason is lattice-based model incorporates more possible words so that it can perceive more meanings.

We also study the role of GRU in SaGT. The ACC. of removing GRU in lattice-based model is 87.82% on average, demonstrating that GRU can control historical messages and combine them with current information. Through experiments, we find that the model with 2 layers of SaGT achieves the best. It indicates multiple information fusion will refine the message and make the model more robust.

**Influences of text length on performance** As listed in Table 3, we can observe that text length also has great impacts on text matching prediction. The experimental results show that the shorter the text length, the more obvious the improvement effect of utilizing sense information. The reason is, on the one hand, concise texts usually have rare contextual information, which is difficult for model to understand. However, HowNet brings a lot of useful external information to these weak-context short texts. Therefore, it is easier to perceive the similarity between texts and gain great improvement. On the other hand, longer texts may contain more wrong words caused by insufficient segmentation, leading to incorrect sense information. Too much incorrect sense information may confuse the model and make it unable to obtain the original semantics.

**Case study** We compare LET-BERT between the model with and without sense information (see Fig. 5). The model without sense fails to judge the relationship between sentences which actually have the same intention, but LET-BERT performs well. We observe that both sentences contain the word "yuba", which has only one sense described by sememe `food`. Also, the sense of "cook" has a similar sememe `edible` narrowing the distance between texts. Moreover, the third sense of "fry" shares the same sememe `cook` with the word "cook". It provides a powerful message that makes "fry" attend more to the third sense.

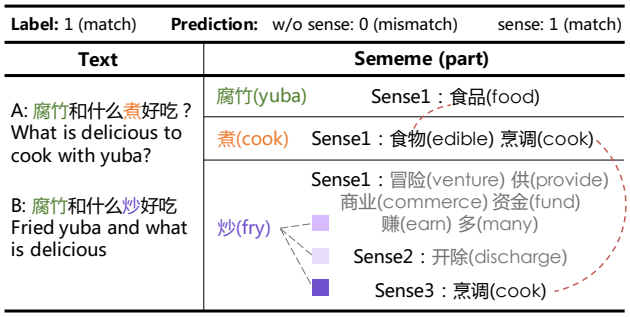


Figure 5: An example of using sense information to get the correct answer.

## 6 Conclusion

In this work, we proposed a novel linguistic knowledge enhanced graph transformer for Chinese short text matching. Our model takes two word lattice graphs as input and integrates sense information from HowNet to moderate word ambiguity. The proposed method is evaluated on two Chinese benchmark datasets and obtains the best performance. The ablation studies also demonstrate that both semantic information and multi-granularity information are important for text matching modeling.

## Acknowledgments

We thank the anonymous reviewers for their thoughtful comments. This work has been supported by No. SKLM-CPTS2020003 Project.

## References

Chen, J.; Chen, Q.; Liu, X.; Yang, H.; Lu, D.; and Tang, B. 2018a. The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4946–4951.

Chen, L.; Chang, C.; Chen, Z.; Tan, B.; Gašić, M.; and Yu, K. 2018b. Policy adaptation for deep reinforcement learning-based dialogue management. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 6074–6078. IEEE.

Chen, L.; Chen, Z.; Tan, B.; Long, S.; Gašić, M.; and Yu, K. 2019. AgentGraph: Toward universal dialogue management with structured deep reinforcement learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27(9): 1378–1391.

Chen, L.; Lyu, B.; Wang, C.; Zhu, S.; Tan, B.; and Yu, K. 2020a. Schema-Guided Multi-Domain Dialogue State Tracking with Graph Attention Neural Networks. In *AAAI*, 7521–7528.

Chen, L.; Tan, B.; Long, S.; and Yu, K. 2018c. Structured Dialogue Policy with Graph Neural Networks. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 1257–1268.

Chen, L.; Zhao, Y.; Lyu, B.; Jin, L.; Chen, Z.; Zhu, S.; and Yu, K. 2020b. Neural Graph Matching Networks for Chinese Short Text Matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6152–6158.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1657–1668.

Chen, Z.; Chen, L.; Liu, X.; and Yu, K. 2020c. Distributed Structured Actor-Critic Reinforcement Learning for Universal Dialogue Management. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28: 2400–2411.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; and Hu, G. 2019. Pre-Training with Whole Word Masking for Chinese BERT. *arXiv preprint arXiv:1906.08101* .

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. 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)*, 4171–4186.

Dong, Z.; and Dong, Q. 2003. HowNet-a hybrid language and knowledge resource. In *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, 820–824. IEEE.

Fu, X.; Liu, G.; Guo, Y.; and Wang, Z. 2013. Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. *Knowledge-Based Systems* 37: 186–195.

Gao, J.; Galley, M.; Li, L.; et al. 2019. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval* 13(2-3): 127–298.

Gong, Y.; Luo, H.; and Zhang, J. 2018. Natural Language Inference over Interaction Space. In *International Conference on Learning Representations*.

Gu, Y.; Yan, J.; Zhu, H.; Liu, Z.; Xie, R.; Sun, M.; Lin, F.; and Lin, L. 2018. Language modeling with sparse product of sememe experts. *arXiv preprint arXiv:1810.12387* .

He, T.; Huang, W.; Qiao, Y.; and Yao, J. 2016. Text-attentional convolutional neural network for scene text detection. *IEEE transactions on image processing* 25(6): 2529–2541.

Lai, Y.; Feng, Y.; Yu, X.; Wang, Z.; Xu, K.; and Zhao, D. 2019. Lattice cnns for matching based chinese question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6634–6641.

Lan, W.; and Xu, W. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, 3890–3902.

Li, X.; Meng, Y.; Sun, X.; Han, Q.; Yuan, A.; and Li, J. 2019. Is Word Segmentation Necessary for Deep Learning of Chinese Representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3242–3252.

Li, X.; Yan, H.; Qiu, X.; and Huang, X.-J. 2020a. FLAT: Chinese NER Using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6836–6842.

Li, Y.; Yu, B.; Mengge, X.; and Liu, T. 2020b. Enhancing Pre-trained Chinese Character Representation with Word-aligned Attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3442–3448.

Li, Z.; and Sun, M. 2009. Punctuation as implicit annotations for Chinese word segmentation. *Computational Linguistics* 35(4): 505–512.

Liu, Q. 2002. Word similarity computing based on HowNet. *Computational linguistics and Chinese language processing* 7(2): 59–76.

Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; and Wang, P. 2020. K-BERT: Enabling Language Representation with Knowledge Graph. In *AAAI*, 2901–2908.

Liu, X.; Chen, Q.; Deng, C.; Zeng, H.; Chen, J.; Li, D.; and Tang, B. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1952–1962.

Liu, Y.; Rong, W.; and Xiong, Z. 2018. Improved text matching by enhancing mutual information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Luo, R.; Xu, J.; Zhang, Y.; Ren, X.; and Sun, X. 2019. PKUSEG: A Toolkit for Multi-Domain Chinese Word Segmentation. *arXiv preprint arXiv:1906.11455* .

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38(11): 39–41.

Mueller, J.; and Thyagarajan, A. 2016. Siamese recurrent architectures for learning sentence similarity. In *thirtieth AAAI conference on artificial intelligence*.

Niu, Y.; Xie, R.; Liu, Z.; and Sun, M. 2017. Improved word representation learning with sememes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2049–2058.

Qi, F.; Yang, C.; Liu, Z.; Dong, Q.; Sun, M.; and Dong, Z. 2019. Openhownet: An open sememe-based lexical knowledge base. *arXiv preprint arXiv:1901.09957* .

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20(1): 61–80.

Shen, T.; Zhou, T.; Long, G.; Jiang, J.; Pan, S.; and Zhang, C. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Sun, J. 2012. Jieba chinese word segmentation tool. *Accessed: Jun* 25: 2018.

Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; and Wu, H. 2019. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv preprint arXiv:1904.09223* .

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

Xu, J.; Liu, J.; Zhang, L.; Li, Z.; and Chen, H. 2016. Improve chinese word embeddings by exploiting internal structure. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1041–1050.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7370–7377.

Yu, K.; Chen, L.; Chen, B.; Sun, K.; and Zhu, S. 2014. Cognitive Technology in Task-Oriented Dialogue Systems: Concepts, Advances and Future. *Chinese Journal of Computers* 37(18): 1–17.

Zhao, Y.; Chen, L.; Chen, Z.; Cao, R.; Zhu, S.; and Yu, K. 2020. Line Graph Enhanced AMR-to-Text Generation with Mix-Order Graph Attention Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 732–741.

Zhu, S.; Li, J.; Chen, L.; and Yu, K. 2020. Efficient Context and Schema Fusion Networks for Multi-Domain Dialogue State Tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 766–781.