Neural Graph Matching Networks for Chinese Short Text Matching

Lu Chen, Yanbin Zhao, Boer Lv, Lesheng Jin, Zhi Chen, Su Zhu, Kai Yu*
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{chenlusz, zhaoyb, boerlv, King18817550378}@sjtu.edu.cn
{zhenchi713, paul2204, kai.yu}@sjtu.edu.cn

Abstract

Chinese short text matching usually employs word sequences rather than character sequences to get better performance. However, Chinese word segmentation can be erroneous, ambiguous or inconsistent, which consequently hurts the final matching performance. To address this problem, we propose neural graph matching networks, a novel sentence matching framework capable of dealing with multi-granular input information. Instead of a character sequence or a single word sequence, paired word lattices formed from multiple word segmentation hypotheses are used as input and the model learns a graph representation according to an attentive graph matching mechanism. Experiments on two Chinese datasets show that our models outperform the state-of-the-art short text matching models.

1 Introduction

Short text matching (STM) is a fundamental task of natural language processing (NLP). It is usually recognized as a paraphrase identification task or a sentence semantic matching task. Given a pair of sentences, a matching model is to predict their semantic similarity. It is widely used in question answer systems and dialogue systems (Gao et al., 2019; Yu et al., 2014).

The recent years have seen advances in deep learning methods for text matching (Mueller and Thyagarajan, 2016; Gong et al., 2017; Chen et al., 2017; Lan and Xu, 2018). However, almost all of these models are initially proposed for English text matching. Applying them for Chinese text matching, we have two choices. One is to take Chinese characters as the input of models. Another is first to segment each sentence into words, and then to 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 of these models is that explicit word information is not fully exploited, which can be potentially useful for semantic matching. However, word-based models often suffer some potential issues caused by word segmentation. As shown in Figure 1, the character sequence "南京市长江大桥(South Capital City Long River Big Bridge)" has two different meanings with different word segmentation. The first one refers to a bridge (Segment-1, Segment-2), and the other refers to a person (Segment-3). The ambiguity may be eliminated with more context. Additionally, the segmentation granularity of different tools is different. For example, "长江大桥(Yangtze River Bridge)" in Segment-1 is divided into two words "长江(Yangtze River)" and "大桥(Bridge)" in Segment-2. It has been shown that multi-granularity information is important for text matching (Lai et al., 2019).

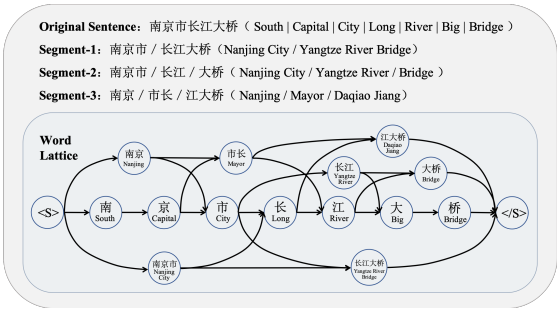

Figure 1: An example of the word segmentation and the corresponding word lattice

Here we propose a neural graph matching method (GMN) for Chinese short text matching. Instead of segmenting each sentence into a word sequence, we keep all possible segmentation paths to form a word lattice graph, as shown in Figure 1. GMN takes a pair of word lattice graphs as input and updates the representations of nodes according to the graph matching attention mechanism. Also,

*Kai Yu is the corresponding author.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6152–6158
July 5 - 10, 2020.

GMN 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 information in these pre-trained language models during the fine-tuning phase. The experiments on two Chinese Datasets show that our model outperforms not only previous state-of-the-art models but also the pre-trained model BERT as well as some variants of BERT.

2 Problem Statement

First, we define the Chinese short text matching task in a formal way. Given two Chinese sentences $S^a = \{c_1^a, c_2^a, \cdots, c_{t_a}^a\}$ and $S^b = \{c_1^b, c_2^b, \cdots, c_{t_b}^b\}$, the goal of a text matching model $f(S^a, S^b)$ is to predict whether the semantic meaning of S^a and S^b is equal. Here, c_i^a and c_j^b represent the i-th and j-th Chinese character in the sentences respectively, and t_a and t_b denote the number of characters in the sentences.

In this paper, we propose a graph-based matching model. Instead of segmenting each sentence into a word sequence, we keep all possible segmentation paths to form a word lattice graph $G = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is the set of nodes and includes all character subsequences that match words in a lexicon $\mathcal{D}$. $\mathcal{E}$ is the set of edges. If a node $v_i \in \mathcal{V}$ is adjacent to another node $v_j \in \mathcal{V}$ in the original sentence, then there is an edge e_{ij} between them. $\mathcal{N}_{fw}(v_i)$ denotes the set of all reachable nodes of node v_i in its forward direction, while $\mathcal{N}_{bw}(v_i)$ denotes the set of all reachable nodes of node v_i in its backward direction.

With two graphs $G^a = (\mathcal{V}^a, \mathcal{E}^a)$ and $G^b = (\mathcal{V}^b, \mathcal{E}^b)$, our graph matching model is to predict their similarity, which indicates whether the original sentences S^a and S^b have the same meaning or not.

3 Proposed Framework

As shown in Figure 2, our model consists of three components: a contextual node embedding module (BERT), a graph matching module, and a relation classifier.

3.1 Contextual Node Embedding

For each node v_i in graphs, its initial node embedding is the attentive pooling of contextual character representations. Concretely, we first concat the original character-level sentences to form a new sequence $S = \{[\text{CLS}], c_1^a, \cdots, c_{t_a}^a, [\text{SEP}], c_1^b, \cdots, c_{t_b}^b, [\text{SEP}]\}$, and then feed them to the BERT model to obtain the contextual representations for each charater $\mathbf{c}^{\text{CLS}}, \mathbf{c}_1^a, \cdots, \mathbf{c}_{t_a}^a, \mathbf{c}^{\text{SEP}}, \mathbf{c}_1^b, \cdots, \mathbf{c}_{t_b}^b, \mathbf{c}^{\text{SEP}}$. Assuming that the node v_i consists of n_i consecutive character tokens $\{c_{s_i}, c_{s_i+1}, \cdots, c_{s_i+n_i-1}\}$[1], a feature-wised score vector $\hat{\mathbf{u}}_{s_i+k}$ is calculated with a feed forward network (FNN) with two layers for each character c_{s_i+k}, i.e. $\hat{\mathbf{u}}_{s_i+k} = \text{FFN}(\mathbf{c}_{s_i+k})$, and then normalized with feature-wised multi-dimensional softmax. The corresponding character embedding $\mathbf{c}_{s_i+k}$ is weighted with the normalised scores $\mathbf{u}_{s_i+k}$ to obtain the initial node embedding $\mathbf{v}_i = \sum_{k=0}^{n-1} \mathbf{u}_{s_i+k} \odot \mathbf{c}_{s_i+k}$, where $\odot$ represents element-wise product of two vectors.

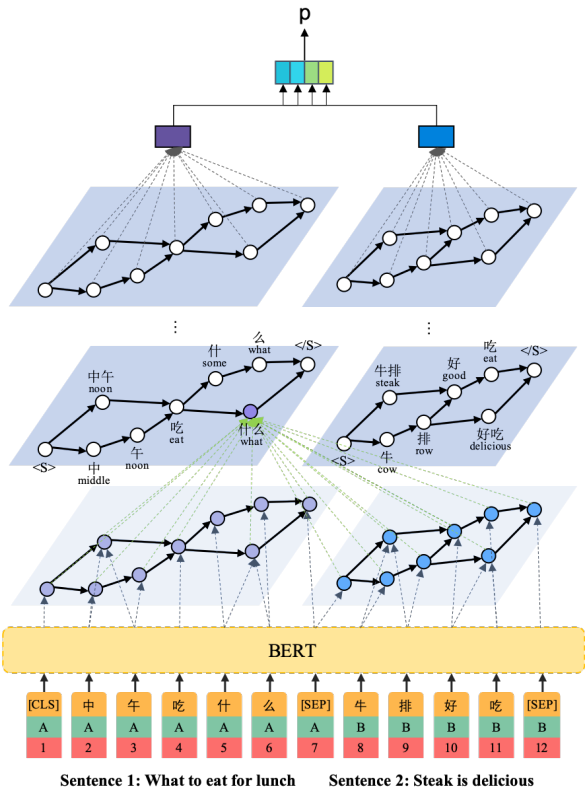

Figure 2: Overview of our proposed framework

3.2 Neural Graph Matching Module

Our proposed neural graph matching module is based on graph neural networks (GNNs) (Scarselli et al., 2009). GNNs are widely applied in various NLP tasks, such as text classification (Yao et al., 2019), machine translation (Marcheggiani et al., 2018), Chinese word segmentation (Yang et al., 2019), Chinese named entity recognition (Zhang

[1]Here s_i denotes the index of the first character of v_i in the sentence S^a or S^b. For brevity, the superscript of c_{s_i+k} is omitted.

and Yang, 2018), dialogue policy optimization (Chen et al., 2018c, 2019, 2018b), and dialogue state tracking (Chen et al., 2020; Zhu et al., 2020), etc. To the best of our knowledge, we are the first to introduce GNN in Chinese shot text matching.

The neural graph matching module takes the contextual node embedding $\mathbf{v}_i$ as the initial representation $\mathbf{h}_i^0$ for the node v_i, then updates its representation from one step (or layer) to the next with two sub-steps: message propagation and representation updating.

Without loss of generality, we will use nodes in G^a to describe the update process of node representations, and the update process for nodes in G^b is similar.

Message Propagation At l-th step, each node v_i in G^a will not only aggregate messages $\mathbf{m}_i^{fw}$ and $\mathbf{m}_i^{bw}$ from its reachable nodes in two directions:

$$\begin{aligned}\mathbf{m}_i^{fw} &= \sum_{v_j \in \mathcal{N}_{fw}(v_i)} \alpha_{ij} \left(\mathbf{W}^{fw}\mathbf{h}_j^{l-1}\right), \\ \mathbf{m}_i^{bw} &= \sum_{v_k \in \mathcal{N}_{bw}(v_i)} \alpha_{ik} \left(\mathbf{W}^{bw}\mathbf{h}_k^{l-1}\right),\end{aligned} \tag{1}$$

but also aggregate messages $\mathbf{m}_i^{b1}$ and $\mathbf{m}_i^{b2}$ from all nodes in graph G^b,

$$\begin{aligned}\mathbf{m}_i^{b1} &= \sum_{v_m \in \mathcal{V}^b} \alpha_{im} \left(\mathbf{W}^{fw}\mathbf{h}_m^{l-1}\right), \\ \mathbf{m}_i^{b2} &= \sum_{v_q \in \mathcal{V}^b} \alpha_{iq} \left(\mathbf{W}^{bw}\mathbf{h}_q^{l-1}\right).\end{aligned} \tag{2}$$

Here α_{ij}, α_{ik}, α_{im} and α_{iq} are attention coefficients (Vaswani et al., 2017). The parameters $\mathbf{W}^{fw}$ and $\mathbf{W}^{bw}$ as well as the parameters for attention coefficients are shared in Eq. (1) and Eq. (2). We define $\mathbf{m}_i^{self} \triangleq [\mathbf{m}_i^{fw}, \mathbf{m}_i^{bw}]$ and $\mathbf{m}_i^{cross} \triangleq [\mathbf{m}_i^{b1}, \mathbf{m}_i^{b2}]$. With this sharing mechanism, the model has a nice property that, when the two graphs are perfectly matched, we have $\mathbf{m}_i^{self} \approx \mathbf{m}_i^{cross}$. The reason why they are not exactly equal is that the node v_i can only aggregate messages from its reachable nodes in graph G^a, while it can aggregate messages from all nodes in G^b.

Representation Updating After aggregating messages, each node v_i will update its representation from $\mathbf{h}_i^{l-1}$ to $\mathbf{h}_i^l$. Here we first compare two messages $\mathbf{m}_i^{self}$ and $\mathbf{m}_i^{cross}$ with multi-perspective cosine distance (Wang et al., 2017),

$$d_k = \text{cosine}\left(\mathbf{w}_k^{cos} \odot \mathbf{m}_i^{self}, \mathbf{w}_k^{cos} \odot \mathbf{m}_i^{cross}\right), \tag{3}$$

Dataset	Size	Pos:Neg	Domain
BQ	120,000	1:1	bank
LCQMC	260,068	1.3:1	open-domain

Table 1: Features of two datasets BQ and LCQMC

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 update the representation of v_i,

$$\mathbf{h}_i^l = \text{FFN}\left(\left[\mathbf{m}_i^{self}, \mathbf{d}_i\right]\right), \tag{4}$$

where $[\cdot, \cdot]$ denotes the concatation of two vectors, $\mathbf{d}_i \triangleq [d_1, d_2, \cdots, d_P]$. FFN is a feed forward network with two layers.

After updating node representation L steps, we will obtain the *graph-aware* representation $\mathbf{h}_i^L$ for each node v_i. $\mathbf{h}_i^L$ includes not only the information from its reachable nodes but also information of pairwise comparison with all nodes in another graph. The graph level representations $\mathbf{g}^a$ and $\mathbf{g}^b$ for two graphs G^a and G^b are computed by attentive pooling of representations of all nodes in each graph.

3.3 Relation Classifier

With two graph level representations $\mathbf{g}^a$ and $\mathbf{g}^b$, we can predict the similarity of two graphs or sentences,

$$p = \text{FFN}\left(\left[\mathbf{g}^a, \mathbf{g}^b, \mathbf{g}^a \odot \mathbf{g}^b, |\mathbf{g}^a - \mathbf{g}^b|\right]\right), \tag{5}$$

where $p \in [0, 1]$. During the training phase, the training object is to minimize the binary cross-entropy loss.

4 Experiments

4.1 Experimental Setup

Dataset We conduct experiments on two Chinese datasets for semantic textual similarity: LCQMC (Liu et al., 2018) and BQ (Chen et al., 2018a). LCQMC is a large-scale open-domain corpus for question matching, while BQ is a domain-specific corpus for bank question matching. The sample in both datasets contains a pair of sentences and a binary label indicating whether the two sentences have the same meaning or share the same intention. All features of the two datasets are summarized in Table 1. For each dataset, the accuracy (ACC) and F1 score are used as the evaluation metrics.

Models	BQ		LCQMC	
	ACC.	F1	ACC.	F1
Text-CNN	68.5	69.2	72.8	75.7
BiLSTM	73.5	72.7	76.1	78.9
Lattice-CNN	78.2	78.3	82.1	82.4
BiMPM	81.9	81.7	83.3	84.9
ESIM-char	79.2	79.3	82.0	84.0
ESIM-word	81.9	81.9	82.6	84.5
GMN (Ours)	**84.2**	**84.1**	**84.6**	**86.0**
BERT	84.5	84.0	85.7	86.8
BERT-wwm	84.9	-	86.8	-
BERT-wwm-ext	84.8	-	86.6	-
ERNIE	84.6	-	87.0	-
GMN-BERT (Ours)	**85.6**	**85.5**	**87.3**	**88.0**

Table 2: Performance of various models on LCQMC and BQ test datasets

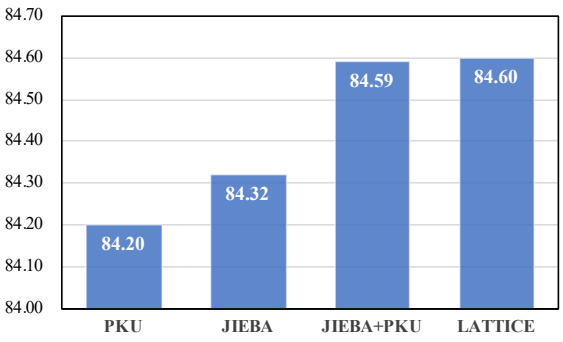

Figure 3: Performance (ACC) of GMN with different inputs on LCQMC dataset

Hyper-parameters The number of graph updating steps/layers L is 2 on both datasets. The dimension of node representation is 128. The dropout rate for all hidden layers is 0.2. The number of matching perspectives P is 20. Each model is trained by RMSProp with an initial learning rate of 0.0001 and a batch size of 32. We use the vocabulary provided by Song et al. (2018) to build the lattice.

4.2 Main Results

We compare our models with two types of baselines: basic neural models without pre-training and BERT-based models pre-trained on large-scale corpora. The basic neural approaches also can be divided into two groups: representation-based models and interaction-based models. The representation-based models calculate the sentence representations independently and use the distance as the similarity score. Such models include Text-CNN (Kim, 2014), BiLSTM (Graves and Schmidhuber, 2005) and Lattice-CNN (Lai et al., 2019). Note that Lattice-CNN also takes word lattices as input. The interaction-based models consider the interaction between two sentences when calculating sentence representations, which include BiMPM (Wang et al., 2017) and ESIM (Chen et al., 2017). ESIM has achieved state-of-the-art results on various matching tasks (Bowman et al., 2015; Chen and Wang, 2019; Williams et al., 2018). For pre-trained models, we consider BERT and its several variants such as BERT-wmm (Cui et al., 2019), BERT-wmm-ext (Cui et al., 2019) and ERNIE (Sun et al., 2019; Cui et al., 2019). One common feature of these variants of BERT is that they all use word information during the pre-trained phase. We use GMN-BERT to denote our proposed model. We also employ a character-level transformer encoder instead of BERT as the contextual node embedding module described in Section 3.1, which is denoted as GMN. The comparison results are reported in Table 2.

From the first part of the results, we can find that our GMN performs better than five baselines on both datasets. Also, the interaction-based models in general outperform the representation based models. Although Lattice-CNN [2] also utilizes word lattices, it has no node-level comparison due to the limits of its structure, which causes significant performance degradation. As for interaction-based models, although they both use the multi-perspective matching mechanism, GMN outperforms BiMPM and ESIM (char and word) [3], which indicates that the utilization of word lattice with our neural graph matching networks is powerful.

From the second part of Table 2, we can find that the three variants of BERT (BERT-wwm, BERT-wwn-ext, ERNIE) [4] all outperform the original BERT, which indicates using word-level information during pre-training is important for Chinese matching tasks. Our model GMN-BERT performs better than all these BERT-based models. It shows that utilizing word information during the fine-tuning phase with GMN is an effective way to boost the performance of BERT for Chinese semantic matching.

[2]The results of Lattice-CNN is produced by the open source code https://github.com/Erutan-pku/LCN-for-Chinese-QA.

[3]The results of ESIM is produced by the open source code https://github.com/lanwuwei/SPM_toolkit.

[4]The results of BERT-wwm, BERT-wwm-ext and ERNIE are taken from the paper (Cui et al., 2019).

Example ID	Sentence	Segmentation		Label	Prediction	
					Jieba	Lattice
1	重庆哪里有做滑雪装的厂 In Chongqing where can I find a ski equipment factory	Jieba	重庆|哪里|有|做|**滑雪**|**装**|的|厂 In Chongqing | where | can I | find a | **ski** | **equipment** | factory	0 (not similar)	1	0
		Human	重庆|哪里|有|做|**滑雪装**|的|厂 In Chongqing | where | can I | find a | **ski equipment** | factory			
	重庆哪里可以滑雪 In Chongqing where can I ski	Jieba	重庆|哪里|可以|滑雪 In Chongqing | where | can I | ski			
		Human	重庆|哪里|可以|滑雪 In Chongqing | where | can I | ski			
2	怎么织宝宝背心 How to knit baby vest	Jieba	怎么 | 织 | 宝宝 | 背心 How to | knit | baby | vest	1 (similar)	0	1
		Human	怎么 | 织 | 宝宝 | 背心 How to | knit | baby | vest			
	宝宝背心裙怎么打 How to knit baby vest skirt	Jieba	宝宝|**背心**|**裙**|怎么|打 How to | knit | baby | **vest** | **skirt**			
		Human	宝宝|**背心裙**|怎么|打 How to | knit | baby | **vest skirt**			

Figure 4: Examples of different prediction of Jieba and Lattice

4.3 Analysis

In this section, we investigate the effect of word segmentation on our model GMN. A word sequence can be regarded as a thin graph. Therefore, it can be used to replace the word lattice as the input of GMN. As shown in Figure 3, we compare four models: Lattice is our GMN with word lattice as the input. PKU and JIEBA are similar to Lattice except that their input is word sequence produced by two word segmentation tools: Jieba [5] and pkuseg (Luo et al., 2019), while the input of JIEBA+PKU is a small lattice graph generated by merging two word segmentation results. We can find that lattice-based models (Lattice and JIEBA+PKU) performs much better then word-based models (PKU and JIEBA). We can also find that the performance of PKU+JIEBA is very close to the performance of Lattice. The union of different word segmentation results can be regarded as a tiny lattice, which is usually the sub-graph of the overall lattice. Compared with the tiny graph, the overall lattice has more noisy nodes (i.e. invalid words in the corresponding sentence). Therefore We think it is reasonable that the performance of tiny lattice (PKU+JIEBA) is comparable to the performance of the overall lattice (Lattice). On the other hand, this indicates that our model has the ability to deal with the introduced noisy information in the lattice graph. In Figure 4, we give two examples to show that word segmentation errors result in incorrect prediction of JIEBA, while Lattice can give the right answers.

[5]https://github.com/fxsjy/jieba

5 Conclusion

In this paper, we propose a neural graph matching model for Chinese short text matching. It takes a pair of word lattices as input instead of word or character sequences. The utilization of word lattice can provide more multi-granularity information and avoid the error propagation issue of word segmentation. Additionally, our model and the pre-training model are complementary. It can be regarded as a flexible method to introduce word information into BERT during the fine-tuning phase. The experimental results show that our model outperforms the state-of-the-art text matching models as well as some BERT-based models.

Acknowledgments

This work has been supported by the National Key Research and Development Program of China (Grant No. 2017YFB1002102) and Shanghai Jiao Tong University Scientific and Technological Innovation Funds (YG2020YQ01).

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 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)*, pages 4946–4951.

Lu Chen, Cheng Chang, Zhi Chen, Bowen Tan, Milica Gašić, and Kai Yu. 2018b. Policy adaptation for deep reinforcement learning-based dialogue management. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 6074–6078. IEEE.

Lu Chen, Zhi Chen, Bowen Tan, Sishan Long, Milica Gasic, and Kai Yu. 2019. Agentgraph: Towards universal dialogue management with structured deep reinforcement learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(9):1378–1391.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Lu Chen, Bowen Tan, Sishan Long, and Kai Yu. 2018c. Structured dialogue policy with graph neural networks. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1257–1268.

Qian Chen and Wen Wang. 2019. Sequential matching model for end-to-end multi-turn response selection. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7350–7354. IEEE.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186.

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

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Yuxuan Lai, Yansong Feng, Xiaohan Yu, Zheng Wang, Kun Xu, and Dongyan Zhao. 2019. Lattice cnns for matching based chinese question answering. *arXiv preprint arXiv:1902.09087*.

Wuwei Lan and Wei Xu. 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*, pages 3890–3902.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of chinese representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3242–3252.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455.

Diego Marcheggiani, Joost Bastings, and Ivan Titov. 2018. Exploiting semantics in neural machine translation with graph convolutional networks. *arXiv preprint arXiv:1804.08313*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 175–180.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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)*, pages 1112–1122. Association for Computational Linguistics.

Jie Yang, Yue Zhang, and Shuailong Liang. 2019. Subword encoding in lattice lstm for chinese word segmentation. 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)*, pages 2720–2725.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of AAAI*, pages 7370–7377.

Kai Yu, Lu Chen, Bo Chen, Kai Sun, and Su Zhu. 2014. Cognitive technology in task-oriented dialogue systems: Concepts, advances and future. *Chinese Journal of Computers*, 37(18):1–17.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564.

Su Zhu, Jieyu Li, Lu Chen, and Kai Yu. 2020. Efficient context and schema fusion networks for multi-domain dialogue state tracking. *arXiv preprint arXiv:2004.03386*.