HORNET: Enriching Pre-trained Language Representations with Heterogeneous Knowledge Sources

Taolin Zhang[1], Zerui Cai[1], Chengyu Wang[2], Peng Li[2], Yang Li[2], Minghui Qiu[2*], Chengguang Tang[2], Xiaofeng He[1*], Jun Huang[2]
[1] East China Normal University [2] Alibaba Group, China
zhangtl0519@gmail.com,13081020719@163.com,chengyu.wcy@alibaba-inc.com
jerryli1981@gmail.com,{ly200170,minghui.qmh,chengguang.tcg}@alibaba-inc.com
hexf@cs.ecnu.edu.cn,huangjun.hj@alibaba-inc.com

ABSTRACT

Knowledge-Enhanced Pre-trained Language Models (KEPLMs) improve the language understanding abilities of deep language models by leveraging the rich semantic knowledge from knowledge graphs, other than plain pre-training texts. However, previous efforts mostly use homogeneous knowledge (especially structured relation triples in knowledge graphs) to enhance the context-aware representations of entity mentions, whose performance may be limited by the coverage of knowledge graphs. Also, it is unclear whether these KEPLMs truly understand the injected semantic knowledge due to the "black-box" training mechanism. In this paper, we propose a novel KEPLM named **HORNET**, which integrates **H**eterogeneous kn**O**wledge from various structured and unstructured sources into the **R**oberta **NET**work and hence takes full advantage of both linguistic and factual knowledge simultaneously. Specifically, we design a hybrid attention heterogeneous graph convolution network (**HaHGCN**) to learn heterogeneous knowledge representations based on the structured relation triplets from knowledge graphs and the unstructured entity description texts. Meanwhile, we propose the explicit dual knowledge understanding tasks to help induce a more effective infusion of the heterogeneous knowledge, promoting our model for learning the complicated mappings from the knowledge graph embedding space to the deep context-aware embedding space and vice versa. Experiments show that our HORNET model outperforms various KEPLM baselines on knowledge-aware tasks including knowledge probing, entity typing and relation extraction. Our model also achieves substantial improvement over several GLUE benchmark datasets, compared to other KEPLMs.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Natural Language Processing, Pre-trained Language Model, Knowledge Graph, Heterogeneous Graph Attention Network

ACM Reference Format:
T. Zhang, Z. Cai, C. Wang, P. Li, M. Qiu, Y. Li, C. Tang, X. He, J. Huang. 2021. HORNET: Enriching Pre-trained Language Representations with Heterogeneous Knowledge Sources. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482436

1 INTRODUCTION

Pre-trained Language Models (PLMs) leverage large-scale unstructured pre-training corpora and well-designed self-supervised pre-training tasks to learn effective context-aware token representations, achieving the state-of-the-art performance on a wide range of NLP downstream tasks, such as question answering, relation extraction and natural language inference [13, 16, 20]. Although previous works have shown that deep PLMs pose some degrees of language understanding abilities [27, 43], this advantage induces memorizing facts observed from the pre-training corpus. As Knowledge Graphs (KGs) contain rich structured knowledge in the form of relation triples, Knowledge-Enhanced Pre-trained Language Models (KEPLMs) [17, 22, 42] can further benefit language understanding by grounding these PLMs with the high-quality, human-curated knowledge facts in KGs, which are difficult to learn from raw texts.

In the literature, popular KEPLMs [10, 30, 39] can be mostly divided into two categories: i) PLMs with structured knowledge and ii) PLMs with unstructured knowledge. PLMs with structured knowledge learn context-aware representations of entity mentions from both pre-training corpora and relation triples via entity linking networks and knowledge-aware self-supervised tasks [42]. The relation triples in KGs are usually embedded into a continuous feature space by KG embedding algorithms (e.g., TransE [2]). PLMs with unstructured knowledge specifically learn knowledge from entity description texts [34] or natural-language sentences automatically converted from relation triples [30].

Although previous studies have proved the effectiveness of incorporating external knowledge into PLMs, two critical issues remain to be addressed. i) Previous works only focus on a single knowledge source, such as structured relation triples or unstructured entity-related text information [30, 42]. However, the relations triples and the corresponding description texts are both critical to help the PLMs understand the entity mentions in pre-training texts. As shown in Figure 1, we find that "HORNET' and "ParaNorman"

* Corresponding authors.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482436

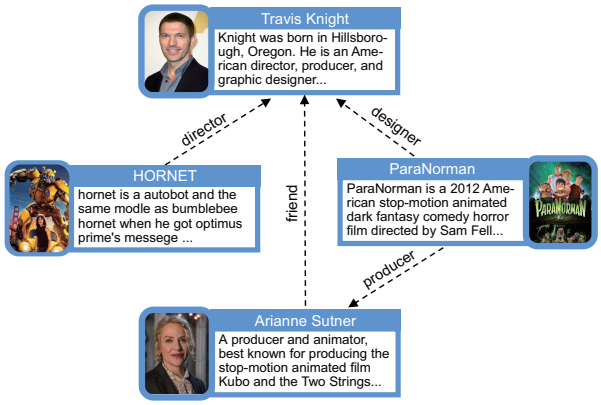


Figure 1: Example of structured relation triples in the KG and the corresponding entity description texts. To fully understand the sentence, it is necessary to obtain the knowledge from both structured relations and unstructured entity descriptions.

are names of movies based on the entity description tests. Hence, the underlying PLM can further understand the relations between "Travis Knight" and "HORNET", together with "Travis Knight" and "animated characters" in the pre-training text. ii) Current KEPLMs pay little attention to whether the KEPLMs pre-trained by the existing self-supervised tasks truly understand the injected knowledge, which is crucial for the realistic practice of the injected knowledge to gain the advantage in the downstream tasks.

To overcome the two challenges mentioned above, we propose a KEPLM based on heterogeneous knowledge information and the RoBERTa model [19] (**HORNET**), which is continuously pre-trained on large-scale corpora and heterogeneous knowledge sources to retrofit the context-aware token representations:

- We use two types of homogeneous knowledge for pre-training HORNET, including structured relation triples and unstructured entity description texts. The structured relation triples are extracted from KGs such as WiKiData [1]. The unstructured knowledge contains entity descriptions and natural-language sentences constructed by entity mention lists in WikiData5M [2]. A heterogeneous graph is constructed based on entities and relations in KGs, together with descriptive sentences. Corresponding to the heterogeneous graph, we propose a hybrid attention heterogeneous graph convolution network (**HaHGCN**) to learn the knowledge-aware entity representations based on two types of attention mechanisms, namely semantic-level attention and node-level attention.
- To alleviate the "black-box" training problem, we propose the dual mapping pre-training tasks to promote the model to learn the complicated mappings between the KEPLM representation space and the knowledge graph embedding space. These tasks intuitively help KEPLMs to understand the injected heterogeneous knowledge information more explicitly, thus the models can easily make use of the knowledge in the downstream tasks. Specifically, we devise two essential learning objects: i) mapping the PLM embedding space to the KG embedding space, and ii) translating the KG embeddings back to context-aware representations generated by KEPLMs.

In the experiments, we compare our HORNET model against various strong baselines, including mainstream KEPLMs pre-trained over the same pre-training resources. The underlying NLP tasks include: *Knowledge Probing*, *Relation Extraction*, *Entity Typing* and the open domain benchmark *GLUE* [33]. The results show that HORNET consistently outperforms all the baselines on these knowledge-aware tasks. Our model also achieves improvements on the GLUE tasks compared to other KEPLMs.

To sum up, the contributions of our work mainly includes the following threefolds:

- We propose a new KEPLM named HORNET to help the PLMs better integrate heterogeneous knowledge. To our knowledge, it is the first work to explicitly integrate structured and unstructured knowledge to enhance the knowledge understanding abilities of KEPLMs.
- We devise the HaHGCN network to fuse the heterogeneous knowledge from different sources into knowledge-aware representations. In addition, we propose dual mapping pre-training tasks to explicitly decode the injected knowledge in the pre-training process.
- We conduct comprehensive experiments to show that our model achieves the state-of-the-art performance compared to various baselines. A detailed analysis is also provided.

2 RELATED WORK

2.1 PLMs

PLMs boost the performance of various downsteam NLP tasks [16, 24] via trained on the large-scale corpus and special self-supervised pre-training tasks. Based on the two-stage training paradigm, namely pre-training and fine-tuning, more PLMs are proposed to train models on large-scale corpora to learn general syntactic and semantic knowledge. Then, the pre-trained models are fine-tuned on downstream tasks with the specific datasets to improve the performance of NLP tasks. BERT [7] (as well as its robustly optimized version RoBERTa [19]) is trained based on the bidirectional transformer architecture by two novel self-supervised tasks. Following the BERT model, a large number of PLMs have been proposed to further improve performance in various NLP tasks, leveraging the following three techniques: i) Self-supervised pre-training tasks: this approach improves the model's semantic understanding ability by modeling large-scale unlabeled corpus in the token level and the sentence level such as Baidu-ERNIE [31], and spanBERT [11]. ii) Encoder architectures: these models boost the performance by changing the internal encoder architecture based on general PLMs. XLNet [38] utilizes the Transformer-XL [6] to encoder long sequence text. Sparse self-attention [4] replaces the self-attention mechanism in transformers to get a more interpretable representation for the

[1] https://www.wikidata.org
[2] https://deepgraphlearning.github.io/project/wikidata5m

whole input. iii) Supervised pre-training tasks: MT-DNN [18] combines pre-training learning in self-supervised tasks and multi-task learning in supervised tasks to improve performance on GLUE tasks [33].

2.2 KEPLMs

The plain PLMs train deep language models only on the large-scale unstructured corpora (such as Wikipedia [3] and BookCorpus [7]), lacking the language understanding abilities of important entity mentions that occurred in pre-training corpora. In contrast, the structured KG data entails the rich semantic knowledge in the form of relation triples.We summary the recent KEPLMs into the following two types: i) Knowledge-enhancement by entity embeddings: ERNIE-THU [42] injects the entity embeddings into the context-aware representations via the knowledge-encoder stacked by the information fusion module and the denoising entity auto-encoder learning objective. KnowBERT [22] proposes the knowledge attention and recontextualization (KAR) and entity-linking mechanisms to inject the knowledge embeddings to PLMs. ii) Knowledge-enhancement by knowledge descriptions: these works replace the entity knowledge embeddings via encoding knowledge description texts. For example, the pre-training corpora and entity descriptions in KEPLER [34] are encoded into a unified semantic space with the same PLM. The model is jointly optimized by the knowledge triple loss and the masked language modeling objectives. K-BERT [17] and CoLAKE [30] convert relation triplets into nodes and insert them into the training samples without pre-trained embeddings.

Previous studies on KG embeddings and knowledge description texts have shown significant improvement on various NLP tasks. We argue that injecting the structured and unstructured knowledge simultaneously (namely heterogeneous knowledge in this paper) into the large-scale unsupervised training corpora can further benefit the context-aware representations, which is the focus of this work.

3 THE HORNET MODEL

3.1 Notation and Model Overview

We state some basic notations as follows. The hidden representations of input tokens $(w_1, w_2, \ldots, w_n)$ in training corpora are denoted as $(h_1, h_2, \ldots, h_n)$ and $h_i \in \mathcal{R}^{d_1}$, where n is the length of the input sequence in PLMs and d_1 is dimension of the PLM's output. Each entity mention span s_m recognized in pre-training corpora is composed by continued token(s) $(h_i, h_{i+1}, \ldots, h_j)$ and $i \leq j$. d_2 is the dimension of the entity embeddings generated by knowledge embedding algorithms.

The overall model architecture of our **HORNET** is shown in Figure 3. HORNET mainly contains three modules: (1) **Knowledge Encoding** aims to recall the knowledge subgraph corresponding to the entity mentions in the training samples and then aggregates the knowledge subgraph representations via our HaHGCN model. (2) **Knowledge Infusion** modules inject the knowledge subgraph representations into the context-aware token hidden features generated by PLMs. (3) **Dual Mapping Pre-training Tasks** attempt to promote the model to utilize the injected heterogeneous knowledge

[3] https://www.wikipedia.org

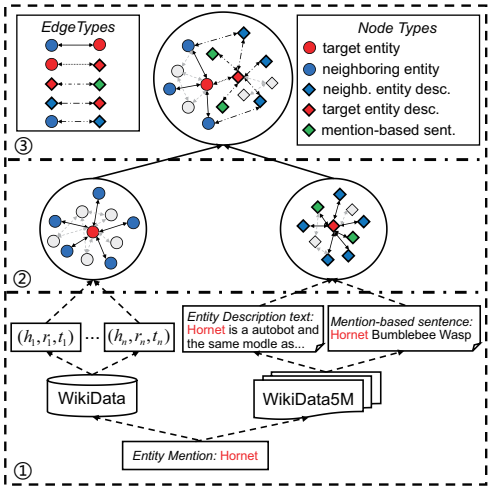


Figure 2: The process of our heterogeneous graph construction, including (1) entity linking and retrieving related knowledge (2) constructing two homogeneous graphs and (3) constructing the heterogeneous graph (Best viewed in color).

by learning the complex mapping functions between the knowledge graph embedding space and the PLM's embedding space.

3.2 Knowledge Encoding

The knowledge encoding processes include two parts: (1) constructing the heterogeneous graph based on two different knowledge sources and (2) performing the graph learning algorithm to obtain the knowledge representations.

3.2.1 Constructing Heterogeneous Graph. The high-level process of constructing the heterogeneous graph is shown in Figure 2, including the following three steps.

- **Entity Linking and Retrieving the Related Knowledge:** In this paper, we use the off-the-shelf tool TAGME [42] to link the mentions (i.e. "Hornet" in Figure 2) to KGs. Our retrieving method of structured homogeneous knowledge in WiKiData utilizes Personalized PageRank (PPR) [21], obtaining the most relevant K neighboring nodes of the target entity via the top-K scores. Unstructured knowledge are extracted via the corresponding entity in WiKiData5M [34].
- **Constructing the Homogeneous Graph:** As for the structured homogeneous graph, we use the original triplet structure constructed by WiKiData. We design the following heuristic rules to construct the unstructured homogeneous graph: (1) The entity description text nodes are connected based on the corresponding triplet structure. (2) We add edges between mention-based sentence nodes and the description text node of the target entity.
- **Constructing the Heterogeneous Graph:** We construct the heterogeneous graph based on the above two homogeneous graphs and add two types of extra special edges. i)

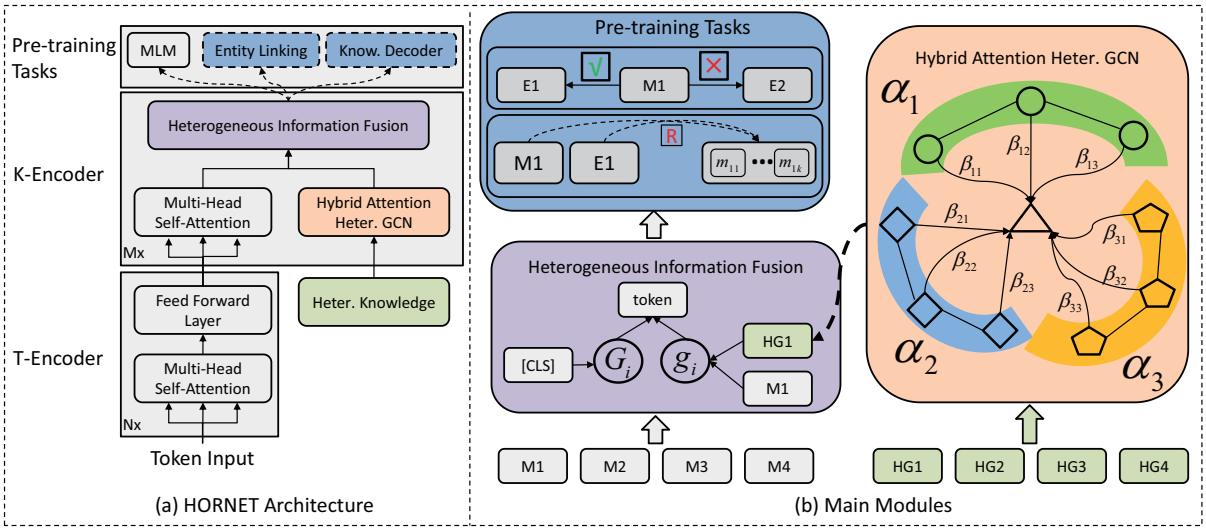


Figure 3: Model overview of HORNET. The left part is our HORNET model architecture. The right part includes three main components: (1) HaHGCN. (2) Heterogeneous information aggregator between context-aware representations and heterogeneous knowledge representations. (3) Pre-training tasks include our designed entity linking and knowledge decoder tasks. (Best viewed in color).

We connect each target entity node with the corresponding description text node. ii) For each neighboring entity node and its description text node, we add an edge to connect them. Formally, a heterogeneous graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ with $v_i \in \mathcal{V}$ and the edges $(v_i, r, v_j) \in \mathcal{E}$. The node set $\mathcal{T}$ and the relation set $\mathcal{R}$ illustrated on the top of Figure 2 contain five types, respectively.

3.2.2 Hybrid Heterogeneous Graph Attention. Existing heterogeneous convolution networks such as R-GCN [26] are proposed to deal with the highly multi-relational data from realistic knowledge bases. The propagation process for calculating the forward-pass update of a node v_i in R-GCN is defined as follows:

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right) \tag{1}$$

where $\mathcal{N}_i^r$ denotes the set of neighbor indices of node i under relation $r \in \mathcal{R}$. $c_{i,r}$ is a problem-specific normalization constant that can either be learned or chosen in advance (such as $c_{i,r} = |\mathcal{N}_i^r|$). l is the layer index of R-GCN model and σ is the action function. $W_i^{(l)}$ is the model parameters of R-GCN. $h_i^{(l+1)}$ is the hidden representation of the node v_i in the $(l+1)$-th layer.

Although relation-based GCN models consider various relation types in the heterogeneous graph, it ignores the heterogeneity of different information types. A straightforward way to adapt relation-based GCN to the heterogeneous nodes types $\mathcal{T}$ is as follows:

$$H^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{\tau\in\mathcal{T}}\frac{1}{c_r}\tilde{A}_{r\tau}\cdot H^{(l)}\cdot W_r^{(l)}\cdot W_\tau^{(l)} + H^{(l)}\cdot W_0^{(l)}\right) \tag{2}$$

where $\tilde{A}_{r\tau} \in \mathcal{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the adjacent matrix with node type τ, relation type r and self-connections. $W_r^{(l)}$ and $W_\tau^{(l)}$ is a layer-specific trainable transformation matrix with for the different relation type r and node type τ. $H^{(l+1)}$ is obtained by aggregating information from the features of their neighboring nodes $H^{(l)}$.

Considering target node with different neighboring nodes types could have different importance, our HaHGCN learns adjacent matrix $\tilde{A}_{r\tau}$ dynamically with type-level graph attention and node-level graph attention.

- **Semantic-level Attention:** Given the target node v, the semantic-level graph attention mechanism aims to capture the importance of different neighboring node types. Specifically, we firstly get the different node type representation as initialized embedding $h_\tau = \sum_{v'}\tilde{A}_{vv'}h_{v'}$, where $\tilde{A}_{vv'}$ is the adjacent matrix with self-connections between node v and neighboring node $v' \in \mathcal{N}_v$ with node type τ. Then, we can calculate the weight of neighboring node types based on the target node representation h_v and each node types representation h_τ.

$$\alpha_\tau = \sigma\left(\mu_\tau^T\left[h_vW_v \parallel h_\tau W_\tau\right]\right)W_1 + b_1 \tag{3}$$

  where μ_τ^T is the attention vector for type τ and "$\parallel$" means the concatenation of two vectors. Finally, we obtain the each neighboring node types normalized weight with Softmax operation:

$$\alpha_\tau = \frac{\exp(a_\tau)}{\sum_{\tau'\in T}\exp(a_{\tau'})} \tag{4}$$

- **Node-level Attention:** Given a specific target node v, different neighboring nodes $v' \in \mathcal{N}_v$ have also different importance with different types τ. Hence, we calculate the node-level attention weight based on different types weight

α_τ.

$$\beta'_{vv'} = \sigma\left(v^T \cdot \alpha_\tau \left[W_v h_v \| W_{v'} h_{v'}\right]\right) \tag{5}$$

where the v^T is the node-level attention vector and α_τ is corresponding node type weight. Then, the node-level weight $\beta_{vv'}$ is normalized with the Softmax function as follows:

$$\beta_{vv'} = \frac{\exp\left(\beta'_{vv'}\right)}{\sum_{v' \in \mathcal{N}_v} \exp\left(\beta'_{vv'}\right)} \tag{6}$$

Finally, we use our hybrid attention graph mechanism to replace the relation-based adjacent matrix $\tilde{A}_{r\tau}$ in Equation 2. Each element in $\tilde{A}_{r\tau}$ is replaced by the node v row and node v' column in $\beta_{vv'}$. For each heterogeneous graph corresponding to the entity mention, we encode the subgraph representation utilizing the graph average pooling operation on the all nodes $N_\mathcal{V}$. $h_g = \frac{1}{|N_\mathcal{V}|} \sum_{v' \in \mathcal{N}_\mathcal{V}} h_{v'}$ denotes the injected knowledge information of each mention.

3.3 Knowledge Infusion

As the knowledge-injected representations may divert the texts from its original meanings, we inject h_g into the context-aware mention representation h_{s_m} calculated by self-attentive pooling [14] of the PLMs' token output to further reduce knowledge noises. In this paper, we utilize the gated position infusion mechanism including local gate and global gate to refine the injected knowledge representations.

$$h'_{s_{mf}} = \sigma\left(\left[h_g \parallel h_{s_m} W_{poj}\right] W_{mf} + b_{mf}\right) \tag{7}$$

$$\widetilde{h}'_{s_{mf}} = LayerNorm(h'_{s_{mf}} W_{bp} + b_{bp}) \tag{8}$$

where $W_{poj} \in \mathcal{R}^{d_1 \times d_2}$,$W_{mf} \in \mathcal{R}^{2d_2 \times 2d_2}$, $W_{bp} \in \mathcal{R}^{2d_2 \times d_1}$, $b_{mf} \in \mathcal{R}^{2d_2}$, $b_{bp} \in \mathcal{R}^{d_1}$. $h'_{s_{mf}} \in \mathcal{R}^{2d_2}$ is the span-level infusion representation. $\widetilde{h}'_{s_{mf}} \in \mathcal{R}^{d_1}$ is the final heterogeneous knowledge-injected representation for mention s_m. For each final token representation h_{if} [4] is generated by local gate and global gate mechanisms.

$$g_i = \tanh\left(\left(\left[h_i \parallel \widetilde{h}'_{s_{mf}}\right]\right) W_{ug} + b_{ug}\right) \tag{9}$$

$$G_i = \tanh\left(h_s W_{sg} + b_{sg}\right) \tag{10}$$

where g_i denotes the local gated score to control the degree of knowledge injection generated by mention-related tokens. The global gated score G_i is calculated by the first token representation h_s of the input sequence, which is at the position of special token $< s >$ in case of RoBERTa [19]. $W_{ug} \in \mathcal{R}^{2d_1 \times d_1}$,$W_{sg} \in \mathcal{R}^{d_1 \times d_1}$ and $b_{ug}, b_{sg} \in \mathcal{R}^{d_1}$ are learnable parameters. $tanh$ is the action function. Hence, the final token representation h_{if} is calculated by above local score g_i and global score G_i simultaneously:

$$h_{if} = \alpha_1 \cdot \sigma\left(\left(\left[h_i \parallel G_i * g_i * \widetilde{h}'_{s_{mf}}\right]\right) W_{ex} + b_{ex}\right) + h_i \tag{11}$$

where $W_{ex} \in \mathcal{R}^{2d_1 \times d_1}$, α_1 is the warm-up factor and "$*$" is element-wise multiplication.

[4] We find that restricting the knowledge infusion position to tokens within the mention span is helpful to improve the model performance.

3.4 Dual Mapping Pre-training

In this section, we elaborate the two well-designed knowledge-aware pre-training tasks, forcing our model to understand the injected knowledge explicitly. Since the parameters of the pre-training tasks are not reused in fine-tuning, we use the simple network module for the dual tasks.

3.4.1 Text to Entity Pre-training. In this task, we train the model to learn the function that mapping the PLMs' embedding space to knowledge graph embedding space makes the knowledge injection more accurate. To predict the corresponding linked-entity e_m in KG from the mention-span s_m, we first employ a self-attentive pooling [14] f_{sp} over $(h_i \dots h_j)$ inside s_m to get the representation of the mention-span:

$$h_{s_m} = f_{sp}\left(h_i, \dots, h_j\right) \tag{12}$$

Then, we get KG embedding of e_m by a single neural layer, and define the aligned entity distribution as follow:

$$h_{e_m} = tanh(h_{s_m} \cdot W_s) \tag{13}$$

$$p(e_m|s_m) = \frac{exp(cs(h_{e_m}, e_m))}{exp(cs(h_{e_m}, e_m)) + \sum_{k=1}^{N} exp(cs(h_{e_m}, e_k))} \tag{14}$$

where cs function represents the cosine similarity. $W_s \in \mathcal{R}^{d_1 \times d_2}$ is a trainable matrix and N is the number of negative samples. The strategies of sampling the negative entity e_k are described in section 4.2. We use the $\mathcal{L}_{EL}$ to represent the cross-entropy criterion between the $p(e_m|s_m)$ and the ground-truth entity label.

3.4.2 Entity to Text Pre-training. In this task, we transfer the predicted entity embedding to the target entity embedding given the relation, and then decode the corresponding mention text from it.

We use DisMult [37] in this paper to obtain the target entity embedding based on h_{e_m} and h_{s_m} encoded in the above modules. The score function is as follows:

$$f_r(h, t) = h^T diag(r) t = (h * r)^T t \tag{15}$$

where h, t are the head and tail entity vectors, respectively. r is the relation vector. In DisMult, the norm of h and t are both 1, and the norm of r is constrained to be less than 1. Hence, we have:

$$\| (h * r)^T t \| \leq \| h \| \times \| r \| \times \| t \| = \| r \| \leq 1 \tag{16}$$

Let $r' = r/||r||$, it's easy to show when $t = h * r'$, the score function is optimal. In practice, we also assign a scalar δ_t for each entity t and apply additional non-linear mapping as the score function may not be close to 1. Finally, we denote the representation t as:

$$h_t = tanh(\delta_t \times r' * h_{e_m} \cdot W_m) \tag{17}$$

In order to generate the mentions of the target entity, we extract the information from h_{s_m} under the guidance of h_t with the gating mechanism:

$$g_t = tanh(h_t \cdot W_g) \tag{18}$$

$$g_{mt} = GeLU(h_{s_m} \cdot W_p) * g_t \tag{19}$$

where $W_m \in \mathcal{R}^{d_2 \times d_2}$,$W_g \in \mathcal{R}^{d_2 \times d_1}$,$W_p \in \mathcal{R}^{d_1 \times d_1}$ are trainable parameters, g_{mt} is the representation of the target entity mention for decoding. As for the decoder, we employ a two-layers transformer encoder initialized and share the last layer weight from the last layer of RoBERTa to decode the g_{mt}, reducing the information left

Datasets	PLMs				KEPLMs				
	ELMo	ELMo5.5B	BERT	RoBERTa	CoLAKE	K-Adapter	KEPLER	HORNET	+△
LAMA-Google-RE	2.2%	3.1%	11.4%	5.3%	9.5%	7.0%	7.3%	**11.28**%	1.78%
LAMA-UHN-Google-RE	2.3%	2.7%	5.7%	2.2%	4.9%	3.7%	4.1%	**6.76**%	1.86%
LAMA-T-REx	0.2%	0.3%	32.5%	24.7%	28.8%	29.1%	24.6%	**32.95**%	3.85%
LAMA-UHN-T-REx	0.2%	0.2%	23.3%	17.0%	20.4%	23.0%	17.1%	**24.21**%	1.21%

Table 1: The performance of PLMs and KEPLMs on knowledge probing datasets. △ represents an improvement over the best results of existing KEPLMs compared to our model.

in pre-training-specific modules. Specifically, we firstly reuse the embedding layer of RoBERTa model with input sequence as:

$$h_s, h_1, ..., h_L, h_e = RoBERTaEmb(< s >, < mask >, ..., < mask >, < e >) \tag{20}$$

where L is the length of mentions of target entity. Then we replace h_s with g_{mt}, and get decoding result as:

$$y_s, y_1, .., y_L, y_e = RoBERTaLayers(g_{mt}, h_1, .., h_L, h_e) \tag{21}$$

A naive loss function for optimizing the masked language model (MLM) task is to perform Cross-Entropy over the whole vocabulary. To alleviate the training cost, we apply Sampled Softmax to the i-th token of the mention-span as the loss function:

$$\mathcal{L}_{DE_i} = \frac{exp(f_s(y_i, t_i))}{exp(f_s(y_i, t_i)) + N \times \mathcal{E}_{t_n \sim Q(t_n|t_i)}[\exp(f_s(y_i, t_n))]} \tag{22}$$

$$f_s(y, t) = y^T \cdot t - \log(Q(t|t_i)) \tag{23}$$

where t_i is the ground-truth token and t_n is the negative token sampled in $Q(t_n|t_i)$. $Q(\cdot|\cdot)$ is the negative sampling function (described in section 4.2). N is the number of negative sampling.

3.5 Training Objective

In the HORNET model, the pre-training tasks include three parts: (1) masked language model loss $\mathcal{L}_{\text{MLM}}$, (2) entity linking loss $\mathcal{L}_{\text{EL}}$ described in Section 3.4.1 and (3) mention decode loss $\mathcal{L}_{\text{DE}}$ derived from Equation 22. The total loss is as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MLM}} + \alpha_2\lambda_1\mathcal{L}_{\text{EL}} + \alpha_3\lambda_2\mathcal{L}_{\text{DE}} \tag{24}$$

where the λ_1 and λ_2 are the hyperparameters, which in our paper are set to 0.5. α_2 and α_3 are the warmup factors.

4 EXPERIMENTS

4.1 Pre-training Data

We use the English Wikipedia data (2020/03/01) [5] as pre-training corpora and use WikiExtractor [6] to process the downloaded Wikipedia dump. The Wikipedia anchors are used to aligned mentions of entities to Wikidata5M [34]. The additional pre-processing and filtration steps of invalid pre-training texts are kept the same as CoLAKE [30]. Finally, there are 3,085,345 entities and 822 relations in KGs and 26M training samples in our pre-training corpus. We train a transformer-style auto-encoder to transfer unstructured texts to the corresponding embeddings. Specifically, we use two transformer-encoder layers as the encoder and another two layers as the decoder. For each entity descriptions or alias, we feed them into the encoder, take the first token representations of the output as the input of the decoder to generate the embeddings (with the method mentioned in Section 3.4.2).

4.2 Model Settings and Training Details

In this work, we use the RoBERTa-base model [19] as our backbone encoder. The hidden dimension of token embeddings is $d_1 = 768$. The dimensions of all types of entity and relation embeddings are $d_2 = 100$. The entity embeddings are fixed during training while the relation embeddings are allowed to be updated. HORNET works as an additional layer inserted after the ninth layer of Roberta. The single-layer graph network has already performed well for the knowledge encoding module, and more layers do not bring considerable improvement.

As for the negative sampling during the text to entity pre-training, we find 16 mentions in all aliases can obtain the smallest Levenshtein distance given the mention. For the negative sampling of the entity to text pre-training, we need to select negative tokens for each decoding position. We first find all possible entities under the given relation and retrieve all mentions of those entities as negative mentions. For each decoding position, we fill it with tokens in negative mentions at the same position. We randomly select tokens in RoBERTa vocabulary when there are too few negative samples. The overall vocabulary size for each decoding position is 2500.

We pre-train the HORNET with weights initialized by the RoBERTa base model [19] released by Hugging Face[7]. In the pre-training process, we use a linear learning-rate schedule with warm-up and the warm-up proportion is 0.1 with a peak value of 5e-4. The AdamW optimizer is used with $\beta_1 = 0.9$, $\beta_2 = 0.998$ and $\epsilon = 1e-6$. The batch size is set to 2048. For a fair comparison, we only pre-train our model by one epoch with the max sequence length as 512. All the warm-up factors α_1, α_2, α_3 are linearly increased from 0 to 1 in the model warm-up stage and then are kept unchanged during the rest of the pre-training process. Other pre-training hyper-parameters and unmentioned details could be referred to CoLAKE [30]. The pre-training process is conducted with 8 V100-16G GPUs for about five days. For the hyperparameters of the downstream tasks, we find the following ranges of possible values work well, i.e., the batch size: 8,16, the learning rate: 5e-5, 4e-5, 3e-5, 2e-5, and the number of epochs ranging from 3 to 5.

[5] https://dumps.wikimedia.org/enwiki/
[6] https://github.com/attardi/wikiextractor
[7] https://huggingface.co/roberta-base

Model	Accuracy	Macro F1	Micro F1
BERT	52.04	75.16	71.63
RoBERTa	56.3	76.9	74.2
NFGEC (Attentive) [28]	54.53	74.76	71.58
NFGEC (LSTM) [28]	55.60	75.15	71.73
ERNIE	57.19	76.51	73.39
HORNET	**60.28**	**83.99**	**79.65**

Table 2: The performance of various models on FIGER (%).

Model	Precision	Recall	F1
UFET [3]	77.4	60.6	68.0
BERT	76.4	71.0	73.6
RoBERTa	77.4	73.6	75.4
ERNIE_{BERT}	78.4	72.9	75.6
$\text{ERNIE}_{RoBERTa}$	80.3	70.2	74.9
KnowBERT_{BERT}	77.9	71.2	74.4
$\text{KnowBERT}_{RoBERTa}$	78.7	72.7	75.6
KEPLER_{WiKi}	77.8	74.6	76.2
CoLAKE	77.0	75.7	76.4
HORNET	**80.54**	**75.8**	**76.9**

Table 3: The performance of models on Open Entity (%).

4.3 Knowledge Probing

LAMA [23] (LAnguage Model Analysis) aims to recall factual knowledge without any fine-tuning. This task demonstrates the language models' knowledge understanding abilities via the cloze-style statement like "The official language of Mauritius is <mask>". In addition, LAMA-UHN, the more complex "factual" subset of LAMA, is proposed to alleviate the problem of overly relying on the surface form of entity names. We report the mean precision at one (P@1) macro-averaged. For fair comparison, we use the intersection of the vocabularies for all considered models and construct a common vocabulary of 18K case-sensitive tokens.

The results of LAMA and LAMA-UHN performance are shown in Table 1. From the results, we can observe that: (1) the performance of the BERT model has a large gap over the KEPLMs continually trained on RoBERTa. This may be due to the difference in the size of the vocabulary loaded by the BERT and RoBERTa models, which makes the semantic reasoning of the LAMA-style probing tasks simple. (2) Although our model is based on RoBERTa, it still outperforms all the baseline KEPLMs significantly over four datasets. From this phenomenon, we believe that the dual mapping pre-training enhances the ability of our model to transform the heterogeneous knowledge into the representation of the PLMs' tokens.

4.4 Knowledge-Driven Tasks

In this section, we fine-tune and evaluate HORNET on knowledge-driven tasks, including entity typing and relation extraction.

Model	FewRel			TACRED		
	P	R	F1	P	R	F1
CNN	69.51	69.64	69.35	70.30	54.20	61.20
PA-LSTM [41]	-	-	-	65.70	64.50	65.10
C-GCN [40]	-	-	-	69.90	63.30	66.40
BERT	85.05	85.11	84.89	67.23	64.81	66.00
RoBERTa	86.3	86.3	86.3	70.80	69.60	70.20
ERNIE_{BERT}	88.5	88.40	88.30	70.01	66.14	68.09
KnowBERT	-	-	-	71.62	**71.49**	71.53
CoLAKE	90.60	90.60	90.50	-	-	-
HORNET	**91.24**	**91.08**	**91.16**	**73.51**	71.07	**72.26**

Table 4: The performance of various models on FewRel and TACRED datasets (%).

4.4.1 Entity Typing. This task requires the model to predict the fine-grained types of a set of free-form phrases given their context. We fine-tune the HORNET model on two well-established datasets FIGER [15] and Open Entity [3] to evaluate its performance on this task. To identify the entity mentions of interest, we add two special tokens before the first and after the last token of the entity spans, and use the first special tokens representation of the last layer output for classification.

Table 2 shows the performance of various models on FIGER dataset. It can be saw that the KEPLMs have superior performance than corresponding vanilla PLMs. In addition, our HORNET model with heterogeneous knowledge achieves a large gap performance compared to baselines (+3.09% Acc., +7.48% macro F1 and +6.26% micro F1). We believe that the heterogeneous knowledge, especially that from unstructured texts, plays an important role as described later in the ablation study (See Section 4.7).

The Open Entity results of our model are in Table 3. Note that the dataset is in a relatively small size, thus the model usually easily overfits the training set and the performance of the model varies greatly when the hyper-parameters are set differently. Compared with the improvement of the SOTA results (+0.2%, CoLAKE vs. KEPLER), we achieve a relatively large step forward (+0.5%).

4.4.2 Relation Extraction. The relation extraction task (RE) aims to determine the fine-grained semantic relation between two entities in the given sentence. We fine-tune and evaluate our HORNET model in two public authoritative RE datasets including FewRel [9] and TACRED [41]. Following the ERNIE-THU [42] settings, we sample 100 instances from each class for the training set, and sample 200 instances for the development and test, respectively. The relation types in the two datasets are 80 and 42 (including a special relation "no relation"), respectively. We adopt micro averaged metrics and macro averaged metrics for TACRED and FewRel following the previous work [41], respectively.

From the Table 4, we can observe that the knowledge-injected models comparing to the PLMs have further improved in these datasets, while our model achieves the new SOTA performance, which implies injecting the heterogeneous knowledge into the PLMs for the RE task is also very effective.

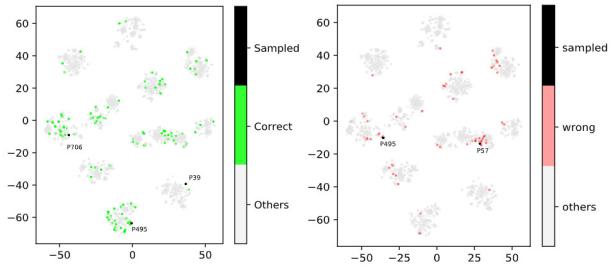
Model	MNLI (m/mm) 392K	QQP 363K	QNLI 104K	SST-2 67K	CoLA 8.5K	STS-B 5.7K	MRPC 3.5K	RTE 2.5K	Average
$BERT_{BASE}$	84.6 / 83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
RoBERTa	87.5 / 87.3	91.9	**92.8**	94.8	63.6	91.2	90.2	78.7	86.4
ERNIE	84.0 / 83.2	71.2	91.3	93.5	52.3	83.2	88.2	68.8	79.5
KEPLER	87.2 / 86.5	91.5	92.4	94.4	62.3	89.4	89.3	70.8	84.9
CoLAKE	87.4 / 87.2	92.0	92.4	94.6	63.4	90.8	**90.9**	77.9	86.3
HORNET	**87.6 / 87.5**	**92.4**	92.2	**94.9**	**64.6**	**91.3**	90.5	**79.7**	**86.7**

Table 5: The results of PLMs and KEPLMs on GLUE tasks (%). We use F1 to evaluate QQP and MRPC, Spearman correlations for STS-B, and accuracy scores for the other tasks. The "m/mm" means matched/mismatched evaluation sets for MNLI [36].

4.5 Open Domain Benchmark: GLUE

The General Language Understanding Evaluation (GLUE) benchmark [33] aims to verify the language understanding abilities of the proposed model, including 8 datasets [1, 5, 8, 12, 25, 29, 35, 36]. Those tasks are selected so as to favor models that share information across tasks using transfer learning techniques like PLMs.

Table 5 shows the overall results on GLUE datasets. From the table, we can observe that in the large-scale downstream tasks including MNLI, QQP, QNLI, and SST-2, our HORNET model achieves the improvement steadily. However, we find that KEPLMs including our model do not significantly improve the gap on these datasets compared to the PLMs (i.e. BERT and RoBERTa) as the size of the datasets are small and the injected knowledge may become noisy features when the model can already make prediction based on context.

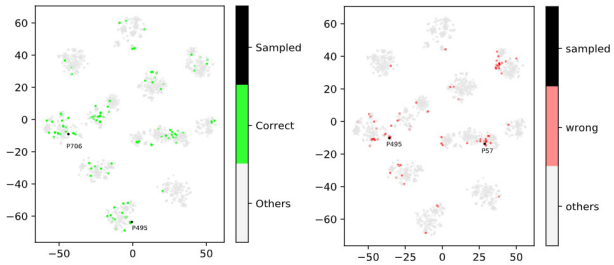


(a) Improvement distribution (b) Degeneration distribution

Figure 4: Improvement/degeneration distributions from RoBERTa.

4.6 Case Studies on Representative Samples

In this section, we perform case studies on representative samples to explain why our model would be successful or fail in certain cases. Specifically, we employ t-SNE [32] to cluster the samples and select representative ones for evaluation. We consider the two representative categories of samples, where (1) the predictions of the other models are wrong while the predictions of our model are right and (2) the opposite cases. We choose RoBERTa and CoLAKE as baselines to compare with our model. The last hidden states on FewRel test set before the final classifier are fed to t-SNE [32] to generate distributions, highlighting the strengths and weaknesses



(a) Improvement distribution (b) Degeneration distribution

Figure 5: Improvement/degeneration distribution from CoLAKE.

of our model in figures. Note that the FewRel testset contains 16,000 samples and 80 types of relations, which are too many to plot clearly. Thus, we select ten categories of samples that appear firstly in the test set.

Figure 4 shows the improvement and degeneration between our model and RoBERTa. We elaborate those samples in Table 6 and 7, where the red phrases are the head entities while the blue phrases are the tail entities. In Table 6, it can be seen that our model does figure out a real relation between the two entities that can not be inferred from the contexts. We hypothesize that the injected knowledge disturbs the prediction, causing our model to ignore the text contexts. For the first sample in Table 7, the knowledge of two entities indicate both of them are locations, then there should be a terrain relation rather than the component relation. For the second sample, we can see that "Tuditanus" was a politician. The knowledge helps us to know the censor is a political office to be held. However, we can hardly tell it when we know nothing about the person.

In summary, comparing our model to RoBERTa, we can conjecture that the injected knowledge helps our model realize properties of entities when the inputs contain uncommon words (entities), and thus leading to the correct predictions. On the other hand, when the plain PLMs can make the right judgment based on the contexts, our model may fail because the predictions can be dominated by injected knowledge rather than the contexts.

Truth Relation	Predicted Relation	Sample
P495: country of origin	P364: original language of film	Constance last acted in a run ofminor films made in Italy between 1955 and 1959, including a role as Lucretia Borgia in "La congiura dei Borgia" (1959).
P57: director(s) of film.	P58: screenwriter, scriptwriter	In 2008 he produced , alongside Christian Colson , the critically acclaimed feature film "Eden Lake" (2008), directed by James Watkins.

Table 6: Representative samples where HORNET degenerates from RoBERTa.

Truth Relation	Predicted Relation	Sample
P706: located on terrain feature	P361: parts of subject	Heliaster Solaris is a possibly extinct sea star which was known from the waters near Española in the Galapagos.
P39: political office held.	P106: profession job work	The censor Tuditanus among possible candidates for Princeps Senatus chose instead his kinsman Quintus Fabius Maximus Verrucosus.

Table 7: Representative samples where HORNET improves from RoBERTa.

Figure 5 shows the comparison between our model and CoLAKE [30]. We can see that both the cases of degeneration and improvement are similar to RoBERTa, while CoLAKE is more competitive than RoBERTa. Base on this observation, we believe that although our model may be easy to discount the information in text contexts than CoLAKE [30], it can make better use of injected knowledge, thus achieving higher overall performance.

4.7 Ablation Study

In this subsection, we evaluate the effectiveness of three important model components of HORNET on entity typing and relation extraction tasks. Specifically, We introduce several variants of HORNET removing certain components. HORNET-Desc replaces all injected knowledge graph embeddings with <pad> embeddings while HORNET-Triple does the replacement for all the text embeddings. HORNET-NK replaces all knowledge graph embeddings and text embeddings with <pad> embeddings. HORNET-SI uses a simple infusion strategy that fixes the local gate g_i and the global gate G_i to be 1 at all the time. HORNET-MLM removes all pre-training tasks except the masked language model(MLM) task. The performance of those variants and HORNET on the testset of FIGER and fewRel are shown in Table 8.

From the result, we can observe that: (1) Comparing HORNET-Desc to HORNET-Triple, we could tell that the text information of entities is more useful than triples in the entity typing task, while the case is vice-verse for the relation extraction task. This phenomenon reveals that different types of knowledge spotlight different tasks, thus integrating knowledge from heterogeneous knowledge sources would be useful. (2) Comparing HORNET-SI to HORNET-NK, the former performs only a little better than the latter, which indicates that the simple infusion strategy is inefficient at utilizing injected knowledge. (3) We can see that HORNET-MLM performs worst in all variants, we hypothesize that the heavy-parameter encoding and infusion modules of HORNET could not get effective training with the plain MLM task, and the small downstream dataset may not train it sufficiently, thus fail to further help the whole model to achieve a better result.

Model	FIGER (Micro F1)	FewRel (F1)
HORNET	79.65	91.16
HORNET-Desc	78.94	89.89
HORNET-Triple	77.42	90.74
HORNET-NK	76.13	89.45
HORNET-SI	77.10	89.41
HORNET-MLM	75.91	87.42

Table 8: The performance of models for ablation study (%).

5 CONCLUSION AND FUTURE WORK

In this work, we propose a novel KEPLM named HORNET to address the knowledge-intensive language understanding tasks with heterogeneous knowledge sources. Accordingly, we encode heterogeneous knowledge with hybrid attention heterogeneous graph convolution network (HaHGCN) into a unified representation, infusing it into PLMs with local and global level gating mechanisms that are reducing the knowledge noise. Moreover, we propose the dual mapping pre-training tasks that enhance the model's ability of language understanding explicitly, and thus our model could better utilize the injected knowledge for downstream tasks. The experimental results show the significant improvement of our model on knowledge-driven and knowledge-probing takes while achieving competitive results on the open domain benchmark. We also provide intuitive evaluation for why and where our model would succeed or fail. There are two research directions that can be further explored: (1) More effective models or training methods are designed to further reduce the knowledge noise. (2) Conducting a deeper theoretical analysis of the model's understanding of knowledge.

6 ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their valuable comments. This work is supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904, and Alibaba Group through Alibaba Research Intern Program.

REFERENCES

[1] Eneko Agirre, Lluís Màrquez, and Richard Wicentowski (Eds.). 2007. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic.

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[3] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-Fine Entity Typing. In *ACL*. 87–96.

[4] Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2019. Fine-tune BERT with Sparse Self-Attention Mechanism. In *EMNLP*. 3539–3544.

[5] Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *PASCAL*, Vol. 3944. 177–190.

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* (2019).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.

[8] William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *IWP*.

[9] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A Large-Scale Supervised Few-shot Relation Classification Dataset with State-of-the-Art Evaluation. In *EMNLP*. 4803–4809.

[10] Hiroaki Hayashi, Zecong Hu, Chenyan Xiong, and Graham Neubig. 2020. Latent Relation Language Models. In *AAAI*. 7911–7918.

[11] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Trans. Assoc. Comput. Linguistics* 8 (2020), 64–77.

[12] Hector J. Levesque. 2011. The Winograd Schema Challenge. In *AAAI*.

[13] Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: Chinese NER Using Flat-Lattice Transformer. In *ACL*. 6836–6842.

[14] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-Attentive Sentence Embedding. In *ICLR*.

[15] Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design Challenges for Entity Linking. *Trans. Assoc. Comput. Linguistics* 3 (2015), 315–328.

[16] Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, and Nan Duan. [n.d.]. RikiNet: Reading Wikipedia Pages for Natural Question Answering. In *ACL*. 6762–6771.

[17] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling Language Representation with Knowledge Graph. In *AAAI*. 2901–2908.

[18] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *ACL*. 4487–4496.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

[20] Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with Latent Structure Refinement for Document-Level Relation Extraction. In *ACL*. 1546–1557.

[21] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab.

[22] Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *EMNLP*. 43–54.

[23] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language Models as Knowledge Bases?. In *EMNLP*. 2463–2473.

[24] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained Models for Natural Language Processing: A Survey. *CoRR* abs/2003.08271 (2020).

[25] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP*. 2383–2392.

[26] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*. 593–607.

[27] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. AAAI. 3776–3784.

[28] Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An Attentive Neural Architecture for Fine-grained Entity Type Classification. In *AKBC*. 69–74.

[29] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *EMNLP*. 1631–1642.

[30] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized Language and Knowledge Embedding. In *COLING*. 3660–3670.

[31] Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced Representation through Knowledge Integration. *CoRR* abs/1904.09223 (2019).

[32] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[33] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *ICLR*.

[34] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2019. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *CoRR* abs/1911.06136 (2019).

[35] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *Trans. Assoc. Comput. Linguistics* 7 (2019), 625–641.

[36] Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *NAACL*. 1112–1122.

[37] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*.

[38] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NIPS*. 5754–5764.

[39] Denghui Zhang, Zixuan Yuan, Yanchi Liu, Zuohui Fu, Fuzhen Zhuang, Pengyang Wang, Haifeng Chen, and Hui Xiong. 2020. E-BERT: A Phrase and Product Knowledge Enhanced Language Model for E-commerce. *CoRR* abs/2009.02835 (2020).

[40] Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *EMNLP*. 2205–2215. https://doi.org/10.18653/v1/d18-1244

[41] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *EMNLP*. 35–45.

[42] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *ACL*. 1441–1451.

[43] Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *AAAI*. 9733–9740.