Glyph Enhanced Chinese Character Pre-Training for Lexical Sememe Prediction

Boer Lyu, Lu Chen[*], Kai Yu[*]
X-LANCE Lab, Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
Shanghai Jiao Tong University, Shanghai, China
State Key Lab of Media Convergence Production Technology and Systems, Beijing, China
{boerlv, chenlusz, kai.yu}@sjtu.edu.cn

Abstract

Sememes are defined as the atomic units to describe the semantic meaning of concepts. Due to the difficulty of manually annotating sememes and the inconsistency of annotations between experts, the lexical sememe prediction task has been proposed. However, previous methods heavily rely on word or character embeddings, and ignore the fine-grained information. In this paper, we propose a novel pre-training method which is designed to better incorporate the internal information of Chinese character. The **G**lyph enhanced **C**hinese **C**haracter representation (**GCC**) is used to assist sememe prediction. We experiment and evaluate our model on HowNet, which is a famous sememe knowledge base. The experimental results show that our method outperforms existing non-external information models.

1 Introduction

In linguistics, sememes are defined as the minimum semantic units for human language (Bloomfield, 1926), which describe the semantic meaning of concepts. HowNet (Dong and Dong, 2003) is one of the most well-known sememe knowledge bases (KB), which has been widely used in many NLP tasks (Qi et al., 2021), such as semantic similarity computation (Liu, 2002), sentiment analysis (Fu et al., 2013; Huang et al., 2014), language modeling (Gu et al., 2018), word representation learning (Niu et al., 2017) and short text matching (Lyu et al., 2021).

In order to free human experts from the laborious sememe annotating job, Xie et al. (2017) propose the task of sememe prediction, which intends to automatically select related sememes from a closed sememe set for each word. They propose two frameworks based on word embedding and matrix factorization. But these methods usually fail to deal with the prediction problem of low-frequency words.

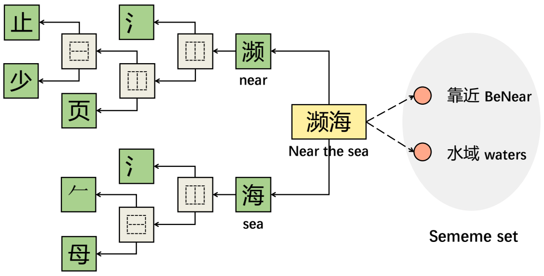


Figure 1: Glyphs of Chinese character which are beneficial to lexical sememe prediction.

Motivated by this, Jin et al. (2018) present character-enhanced sememe prediction (CSP), taking advantage of both internal character information and external context information of words. However, CSP is an ensemble model which still relies on word and character representation, and ignores the fine-grained information.

For internal structural information of words, many researchers believe that only using characters is not sufficient for capturing the semantic information (Yu et al., 2017; Cao et al., 2018; Sun et al., 2019; Meng et al., 2019). For instance, the words “森林(forest)” and “木头(wood)” are semantically related. But these two words share no information since they consist of different characters. To address this problem, we split each Chinese character into several components, and regard component as the minimum unit to express the meaning of the character. We believe that fine-grained units can share more information between semantically related words, which helps model prediction. Take Figure 1 for example, the characters of word “濒海(near the sea)” have components “步(step)” and “氵(water)”, which are related to its sememes, namely “靠近(BeNear)” and “水域(waters)”, respectively.

In order to better incorporate the internal information of Chinese character, we pre-train a **G**lyph

[*]The corresponding authors are Lu Chen and Kai Yu.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4549–4555
November 7–11, 2021. ©2021 Association for Computational Linguistics

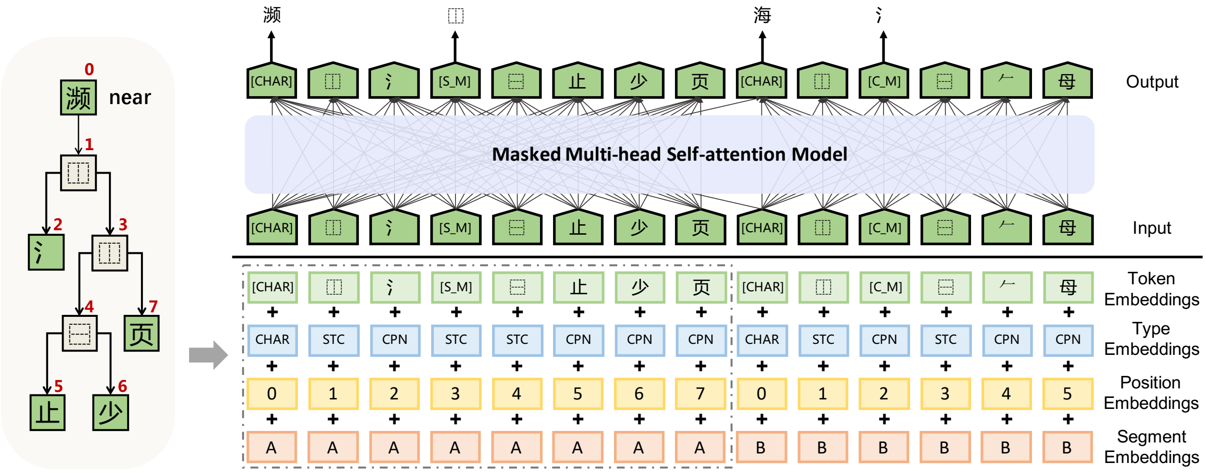


Figure 2: Architecture of glyph enhanced pre-training model for Chinese characters.

enhanced **C**hinese **C**haracter embedding (**GCC**) for sememe prediction task. More specifically, we use the same model structure as BERT (Kenton and Toutanova, 2019), but change the input unit and the masking scheme. First, we regard Chinese words as our training samples and take components of each character in the word to form the input sequence. Second, we mask random tokens and predict the modified tokens as well as all characters in the sample.

We evaluate our model on HowNet sememe KB. Experimental results demonstrate that our model outperforms the baseline model. In summary, our contributions include:

- To the best of our knowledge, we are the first to use masked language model (MLM) objective to force the model to learn the internal information of characters.
- We propose a novel sememe prediction framework considering both internal and contextual character information.
- Our method is particularly useful for low-frequency words and shows the effectiveness and robustness on the dataset.

2 Methodology

In this section, we first introduce the architecture of pre-training model. Then, we describe how to incorporate pre-trained representation into sememe prediction task.

2.1 Pre-Training Model Architecture

As shown in Figure 2, the framework of our pre-training model includes an embedding layer and a masked transformer encoder layer.

First, we use the file[1] about structures of Han Ideographs and refer to Ke and Hagiwara[2] to get all the Chinese character trees. Then, we use the depth-first algorithm to convert each character tree into the format of a sequence (Nguyen et al., 2019). Note that, there are two types of tokens in the input sequence. As shown in the left block in Figure 2, the leaf nodes (position 2, 5, 6, 7) are components of Chinese character, and the inner nodes (position 1, 3, 4) are structural composition operators (such as vertical stacking) applied to children nodes. The character “濒 (near)” can be serialized as $\{char, x_1^{\mathcal{T}}, x_2^{\mathcal{C}}, x_3^{\mathcal{T}}, x_4^{\mathcal{T}}, x_5^{\mathcal{C}}, x_6^{\mathcal{C}}, x_7^{\mathcal{C}}\}$, where $\mathcal{C}$ is the set of components, $\mathcal{T}$ is the structural composition operator set.

2.1.1 Embedding Layer

The input embedding of the model is the sum of token embedding, type embedding, position embedding and character segmentation embedding.

For **token embedding**, we maintain two lookup tables (Sun et al., 2020) and use `[CHAR]` as the character tag which represents the entire character information, `[S_M]` to mask the structure type token and `[C_M]` to mask the component token. To distinguish them, we simply use **type embedding** to indicate the token types, i.e. `CHAR` for character tag, `STC` for structure type token and `CPN` for component token. As for **position embedding**, we assign a number starting from 0 to each token belonging to the same character. Finally, our model use **segmentation embedding** to identify different characters. For instance, the input sequence in Fig-

[1] https://github.com/tomcumming/chise-ids

[2] https://github.com/yuanzhiKe/Radical_CR_Encoder

ure 2 is marked with a sequence of segment tags, i.e. {A, ..., A, B, ..., B}. All the embeddings have the same dimension d.

2.1.2 Masked Transformer Encoder

We use the multi-head self-attention network as the basic structure. Given the representation of sequence tokens $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of tokens in the sequence and d is the dimension of each token. The process of masked self-attention can be formulated by

$$\begin{aligned} \mathbf{A} &= \frac{(\mathbf{X}\mathbf{W}^Q)(\mathbf{X}\mathbf{W}^K)^\top}{\sqrt{d_k}}, \\ \widetilde{\mathbf{X}} &= \mathrm{Softmax}\left(\mathbf{A} + \mathbf{M}\right)(\mathbf{X}\mathbf{W}^V), \end{aligned} \tag{1}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d_k}$ are learnable parameters, and $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the attention mask matrix (Liu et al., 2020). We obtain $\mathbf{M}$ by setting $\mathbf{M}_{ij}$ to 0 when x_j is visible to x_i while setting $\mathbf{M}_{ij}$ to $-\infty$ when x_j is invisible to x_i. More specifically, all tokens belonging to the same character are visible to each other; and the special tags `[CHAR]` are also visible to each other. Thus, the output representation of `[CHAR]` not only contains internal component information of the character itself, but also other character information in the word.

2.1.3 Pre-Training Objective

MLM objective is used in our model. Generally, we mask 15% of the input sequence at random; of those, 80% are replaced by their mask token (`[C_M]` for component tokens, `[S_M]` for structure type tokens), 10% are replaced by a random token which belongs to the same token type, and 10% are kept unchanged. We train a model to predict the original tokens from the modified input. *Masking component tokens* helps model to learn the fine-grained information from the contextual component sequence. *Masking structure type tokens* helps model to learn the structural information of components.

We also predict the character of tag `[CHAR]`. This objective forces model to gather all useful multi-granularity information to the token `[CHAR]`. The advantage is that we can easily use the hidden output of `[CHAR]` as the character representation $\mathbf{u}$ for downstream tasks, such as sememe prediction task.

2.2 Sememe Prediction Model

Given a word $w \in \mathcal{W}$, the goal of our model is to predict the corresponding $P(s|w)$ for each sememe $s \in \mathcal{S}$, where $\mathcal{W}$ is the word set and $\mathcal{S}$ is the set of sememes existing in HowNet. Then, we recommend sememes with high scores to w.

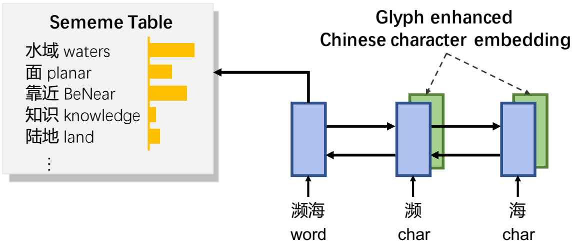


Figure 3: The framework of GCC for sememe prediction task.

Our sememe prediction model GCC (Figure 3) has two parts, one is an encoder which encodes the word-related information into a vector and the other is a multi-label classifier, which uses the vector to compute scores for each sememe.

We use Bidirectional LSTM (Bi-LSTM) (Schuster and Paliwal, 1997) as the encoder. For each word w, we concatenate the word and the characters c_i in the word as $\{w, c_1, ..., c_n\}$, and then convert it to $\{\mathbf{w}, \mathbf{c_1}, ..., \mathbf{c_n}\}$ with the embedding trained on SogouT corpus[3] using Skip-gram (Mikolov et al., 2013). We incorporate our pre-trained character embedding by addition operation:

$$\hat{\mathbf{c_i}} = \mathbf{c_i} + \mathbf{W}^U \mathbf{u_i}, \tag{2}$$

where $\mathbf{W}^U$ is a projection matrix and $\mathbf{u_i}$ is the character representation mentioned in Section 2.1.3.

Then, we pass it to Bi-LSTM. The concatenation of the last hidden states in both directions, denoted as $\mathbf{h}$, is fed to the multi-label classifier:

$$\mathbf{h} = \text{Bi-LSTM}(\mathbf{w}, \hat{\mathbf{c_1}}, ..., \hat{\mathbf{c_n}}), \tag{3}$$

$$\mathbf{x} = \mathbf{W}\mathbf{h} + \mathbf{b}, \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{|\mathcal{S}| \times 2l}$, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^{|\mathcal{S}|}$, l represents the dimension of hidden states in a single direction. Each element of $\mathbf{x}$ is a score related to the sememe in $\mathcal{S}$. For training, we use the multi-label one-versus-all cross-entropy loss, where σ is a sigmoid function and $\mathbf{y}_j \in \{0, 1\}$ represents whether the j-th sememe is in the sememe set of word w:

$$L = -\frac{1}{|S|}\sum_{j=1}^{|S|} \mathbf{y}_j \sigma\left(\mathbf{x}_j\right) + \left(1 - \mathbf{y}_j\right) \sigma\left(-\mathbf{x}_j\right). \tag{5}$$

[3] https://www.sogou.com/labs/resource/t.php

3 Experiments

3.1 Experimental Setup

Pre-Training Data We adopt Tencent embedding corpus (Song et al., 2018) which covers over 8 million Chinese words and phrases. We remove non-Chinese characters such as punctuation and pure digits, and finally get 7,291,828 words as our pre-training samples.

Sememe Prediction Dataset To make the results comparable, we follow Du et al. who proposed the previous state-of-the-art model. This dataset is constructed from HowNet sememe KB, where they disregard the hierarchical structures of sememes and filter out the low-frequency sememes which appear less than 5 times in HowNet. The final number of sememes we use is $1,400$. The total number of words in the dataset is 48,383, which are divided into non-overlapping training, validation, and test sets in the ratio of 8:1:1.

Hyper-parameters Both pre-training and the sememe prediction models are trained by Adam with a learning rate of 0.0001 (Kingma and Ba, 2014). For pre-training, we use the structure of BERT-base and the batch size is 1024. For sememe prediction, the dimension of word embedding is 200, the dimension of Bi-LSTM hidden states is 512×2, and the batch size is 128. Our code is available at `https://github.com/lbe0613/GCC`.

3.2 Evaluation Metrics

Following Xie et al., we use mean average precision (MAP) as evaluation metrics. We rank all sememes according to the model output. For a word with K sememes, we get MAP by

$$\text{MAP} = \sum_{k=1}^{K} \frac{k}{r_k}, \tag{6}$$

where the rankings of the K sememes are $r_1 \leq r_2 \leq ... \leq r_K$.

3.3 Results

In Table 1, we report **average** results of 5 runs to ensure the reliability of results.

We compare our model with two types of baselines: representation-based models and definition-based models. Traditional representation-based models include SPWE and CSP, which is an ensemble model relying on word and character embedding. Definition-based models utilize dictionary definitions as the external information. Such models include LD+Seq2Seq, MC and SCorP. Our GCC models belong to representation-based models. We also compare GCC with other models utilizing glyph information. Here, we simply replace our GCC embedding in Figure 3 with character embedding in JWE and Glyce.

Models	MAP
SPWE (Xie et al., 2017)	55.04
CSP (Jin et al., 2018)	58.93
LD+Seq2Seq† (Li et al., 2018)	30.49
MC† (Du et al., 2020)	60.55
SCorP† (Du et al., 2020)	64.65
GCC w/o pre-train (Ours)	58.18
GCC$^{\clubsuit}$ (Ours)	**60.23**
JWE$^{\clubsuit}$ (Yu et al., 2017)	59.03
Glyce$^{\clubsuit}$ (Meng et al., 2019)	59.10

Table 1: Sememe prediction results of all models. The second part models with † utilize external dictionary definition information, and the third part models with $^{\clubsuit}$ consider glyph information.

As shown in Table 1, the models considering glyph information perform better than all traditional representation-based models, which demonstrates that glyph can enhance Chinese character embedding for sememe prediction task. Especially, GCC has an absolute improvement of 2.05% compared to GCC baseline without pre-training and significantly outperforms JWE and Glyce. The reason is that firstly Chinese characters are pictographic characters, and glyphs express the meaning of the word to a certain extent, which is related to the sememes of the word. Secondly, pre-training enables GCC to better integrate fine-grained information into Chinese character representation.

In addition, since experts refer to dictionary definitions when annotating sememes (Dong and Dong, 2003), it is very powerful semantic information for sememe prediction. Even though, our model is still comparable to MC and even better than LD+Seq2Seq when only using the information in words.

3.4 Influence of Word Frequency

Figure 4 shows the evaluation results of different frequencies on four strong models. We can see that GCC is superior to other models in all word frequency ranges. In addition, word frequency has great impacts on sememe prediction. Since low-frequency words are usually unrelated to each other and contain fewer and simpler sememes, the per-

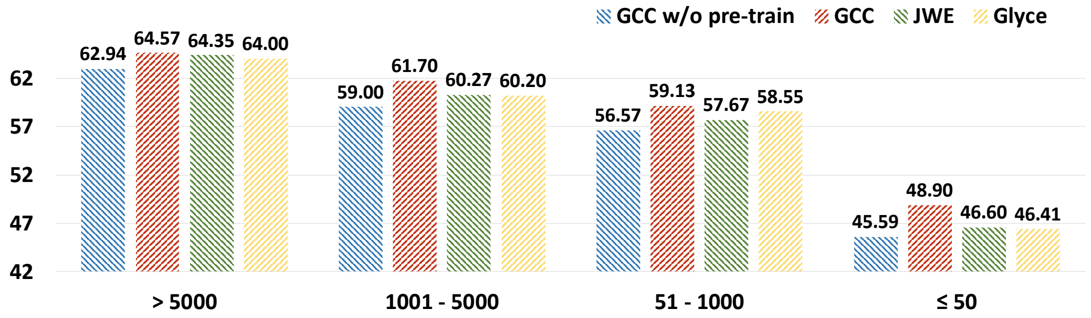
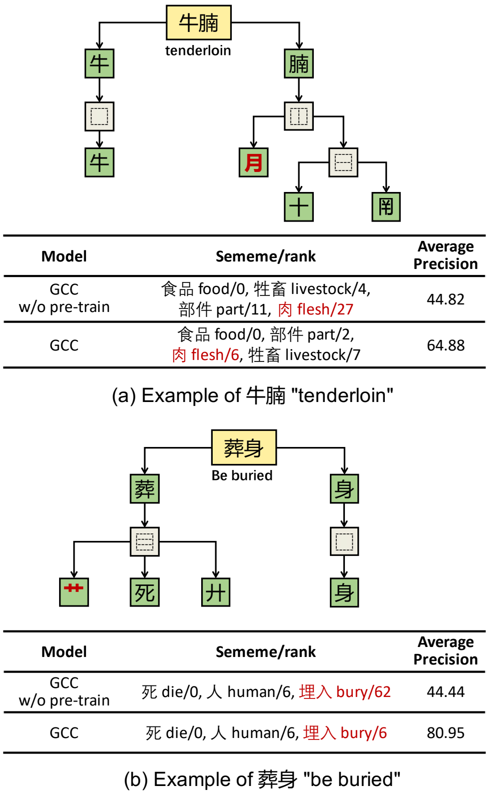


Figure 4: Results of different word frequencies on sememe prediction. The numbers of words in the four ranges are 3316, 2407, 2874 and 839 respectively.

formance of the model is drastically reduced when facing low-frequency words. However, our model GCC is particularly helpful in improving the performance of them. When the word frequency is less than 50, the MAP increases by 3.31% after utilizing glyph enhanced character embedding. Compared with other models using glyph information (JWE and Glyce), it has an increase of at least 2.3%, which is greater than that of all other word frequency ranges.

3.5 Case Study



Model	Sememe/rank	Average Precision
GCC w/o pre-train	食品 food/0, 牲畜 livestock/4, 部件 part/11, 肉 flesh/27	44.82
GCC	食品 food/0, 部件 part/2, 肉 flesh/6, 牲畜 livestock/7	64.88

(a) Example of 牛腩 "tenderloin"

Model	Sememe/rank	Average Precision
GCC w/o pre-train	死 die/0, 人 human/6, 埋入 bury/62	44.44
GCC	死 die/0, 人 human/6, 埋入 bury/6	80.95

(b) Example of 葬身 "be buried"

Figure 5: Examples of using glyphs to assist sememe prediction. The lower the rank, the better.

The examples in Figure 5 show how glyph information assist sememe prediction. We present the sememe labels with their corresponding ranks, and average precision score of each model. Average precision refers to the accuracy of a single sample. The model recommends low-rank sememes to words. In Figure (a), the meaning of component "月(moon)" in Chinese is related to "肉 (flesh)". Thus, the rank of sememe `flesh` is raised from 27 to 6 when incorporating glyph information. And the average precision score increases from 44.82 to 64.88.

In Figure (b), the component "艹(grass)" is the same as grass, which is related to `bury`, because objects can be buried by grass. And the sememe `die` is also the component of the character "葬(burial)", which demonstrates the glyphs are related to the semantics of the word. The result is also convincing. The rank of sememe `bury` is raised from 62 to 6 while the average precision score increases from 44.44 to 80.95.

4 Conclusion

In this work, we pre-train a **G**lyph enhanced **C**hinese **C**haracter embedding (**GCC**) for sememe prediction. The model is evaluated on HowNet sememe KB and outperforms existing non-external information models. Our experiments show that glyph information can enhance the semantic expression of words, and has a better performance on low-frequency words.

Acknowledgments

We thank the anonymous reviewers for their thoughtful comments. This work has been supported by the China NSFC Projects (No. 62120106006 and No. 62106142), CCF-Tencent Open Fund and Startup Fund for Youngman Research at SJTU (SFYR at SJTU).

References

Leonard Bloomfield. 1926. A set of postulates for the science of language. *Language*, 2(3):153–164.

Shaosheng Cao, Wei Lu, Jun Zhou, and Xiaolong Li. 2018. cw2vec: Learning chinese word embeddings with stroke n-gram information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zhendong Dong and Qiang Dong. 2003. Hownet-a hybrid language and knowledge resource. In *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, pages 820–824. IEEE.

Jiaju Du, Fanchao Qi, Maosong Sun, and Zhiyuan Liu. 2020. Lexical sememe prediction using dictionary definitions by capturing local semantic correspondence. *arXiv preprint arXiv:2001.05954*.

Xianghua Fu, Guo Liu, Yanyan Guo, and Zhiqiang Wang. 2013. Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon. *Knowledge-Based Systems*, 37:186–195.

Yihong Gu, Jun Yan, Hao Zhu, Zhiyuan Liu, Ruobing Xie, Maosong Sun, Fen Lin, and Leyu Lin. 2018. Language modeling with sparse product of sememe experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4642–4651.

Minlie Huang, Borui Ye, Yichen Wang, Haiqiang Chen, Junjun Cheng, and Xiaoyan Zhu. 2014. New word detection for sentiment analysis. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 531–541.

Huiming Jin, Hao Zhu, Zhiyuan Liu, Ruobing Xie, Maosong Sun, Fen Lin, and Leyu Lin. 2018. Incorporating chinese characters of words for lexical sememe prediction. *arXiv preprint arXiv:1806.06349*.

Yuanzhi Ke and Masafumi Hagiwara. 2017. Radical-level ideograph encoder for rnn-based sentiment analysis of chinese and japanese. In *Asian Conference on Machine Learning*, pages 561–573. PMLR.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wei Li, Xuancheng Ren, Damai Dai, Yunfang Wu, Houfeng Wang, and Xu Sun. 2018. Sememe prediction: Learning semantic knowledge from unstructured textual wiki descriptions. *arXiv preprint arXiv:1808.05437*.

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

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Boer Lyu, Lu Chen, Su Zhu, and Kai Yu. 2021. Let: Linguistic knowledge enhanced graph transformer for chinese short text matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13498–13506.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: glyph-vectors for chinese character representations. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 2746–2757.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Minh Nguyen, Gia H Ngo, and Nancy F Chen. 2019. Hierarchical character embeddings: Learning phonological and semantic representations in languages of logographic origin using recursive neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:461–473.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Improved word representation learning with sememes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2049–2058.

Fanchao Qi, Ruobing Xie, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. Sememe knowledge computation: a review of recent advances in application and expansion of sememe knowledge bases. *Frontiers of Computer Science*, 15(5):1–11.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

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.

Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. 2019. Vcwe: Visual character-enhanced word embeddings. 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 2710–2719.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670.

Ruobing Xie, Xingchi Yuan, Zhiyuan Liu, and Maosong Sun. 2017. Lexical sememe prediction via word embeddings and matrix factorization. In *IJCAI*, pages 4200–4206.

Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. 2017. Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 286–291.