ChineseBERT: Chinese Pretraining Enhanced by Glyph and Pinyin Information

http://arxiv.org/abs/2106.16038

摘要

考虑到汉语字形和拼音的独特性，本文提出了将汉字字形(*glyph*)和拼音(*pinyin*)信息整合到语言预训练模型的ChineseBERT。根据汉字不同字形获得字形嵌入，在视觉特征中获得汉字语义；拼音嵌入表征汉字发音，处理汉语中“同音异义”现象。本文提出的chineseBERT模型在大规模无标记中文语料库上进行了预训练，训练步骤比baseline少，性能得到提高。该模型在多种中午NLP任务(*machine reading comprehension,natural language inference,text classifition, sentense pair matching,competive performances in named entity recognition, word segmentation*)中取得新的SOTA性能。

1.背景介绍

中文字形，例如“液(liquid)”,“河(river)”，“湖(lake)”，都含有“氵(water)”，拼音中的“乐(le)”或者“乐(yue)”，同字不同音，或者同音不同字。

2.相关工作

NLP大尺度预训练

BERT: Masked Language Model MLM/Next Sentence Prediction NSP
RoBERTa

Chinese: ERNIE (char-level masking, phrase-level masking, entity-level masking)
the Whole Word Masking

字形信息学习

从汉字的图像中学习字形信息(CNNs)

3.模型

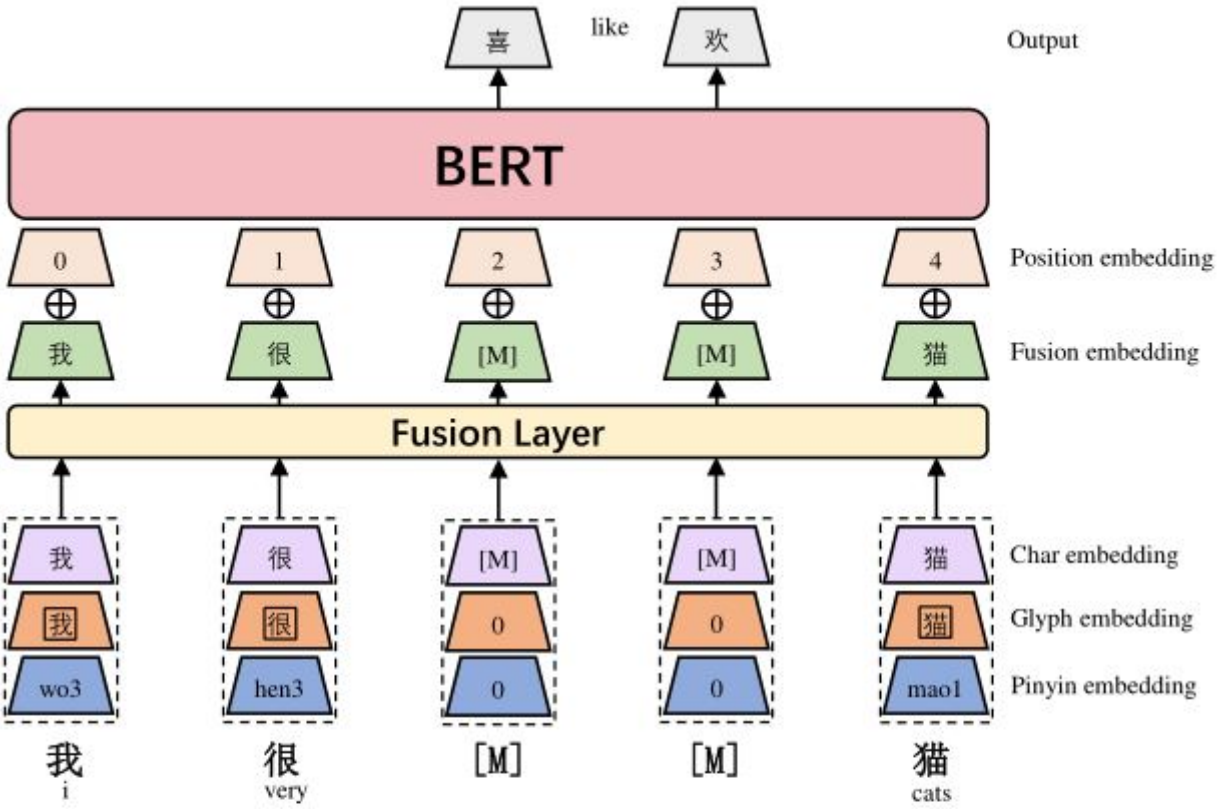

Figure 1: An overview of ChineseBERT. The fusion layer consumes three D-dimensional embeddings – char embedding, glyph embedding and pinyin embedding. The three embeddings are first concatenated, and then mapped to a D-dimensional embedding through a fully connected layer to form the fusion embedding.

3.1概述

Fig.1：ChineseBERT整体模型。char embedding+glyph embedding+pinyin embedding=D-dimensional embedding >> 全连接层 >> fusion embedding。
fusion embedding + position embedding >> BERT Model(Whole Word Masking/Char Masking)

3.2输入

Glyph Embedding

3 types of chinese fonts-FangSong-XingKai-LiShu.24x24 image, Different from CNNs to convert image to representation, use FC layer.
converted the 24x24x3 vector to a 2,352 vector, fed to an FC layer to obtain the output glyph vector

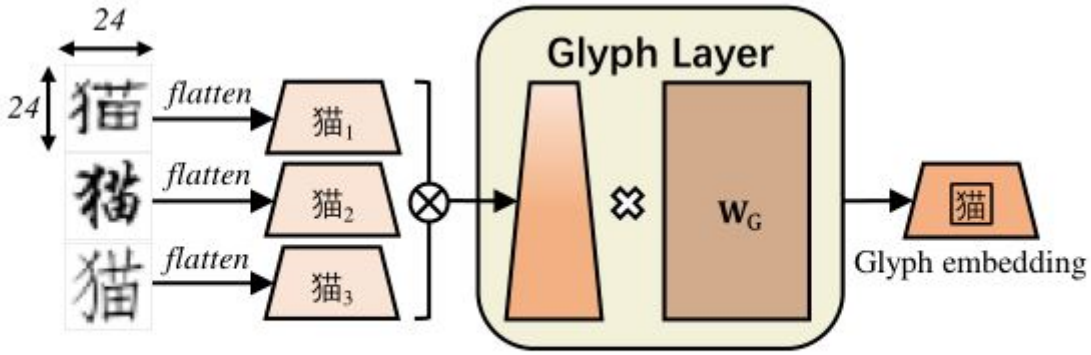

Figure 2: An overview of inducing the *glyph* embedding. ⊗ denotes vector concatenation. For each Chinese character, we use three types of fonts – *FangSong*, *XingKai* and *LiShu*, each of which is a 24×24 image with pixel value ranging $0 \sim 255$. Images are concatenated into a tensor of size $24 \times 24 \times 3$. The tensor is flattened and passed to an FC to obtain the glyph embedding.

Pinyin Embedding

Fig.3所示，使用开源的pypinyin package来生成拼音序列，并且将字的音阶放在拼音之后。使用width为2的CNN模型，最大池化后得到pinyin embedding，将输入的拼音序列长度固定为8，不足8位使用"-"补齐。

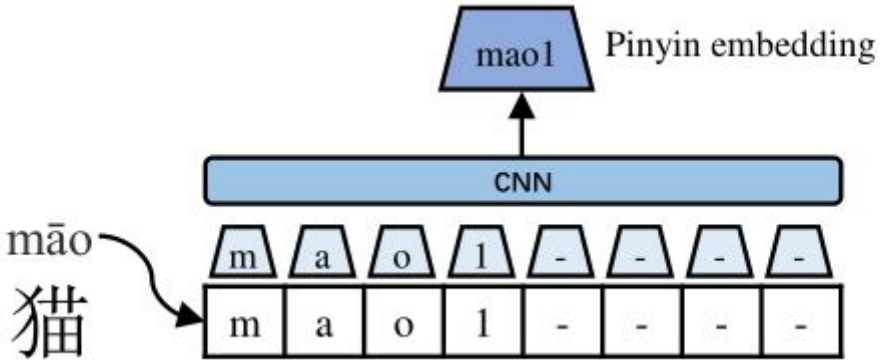

Figure 3: An overview of inducing the *pinyin* embedding. For any Chinese character, e.g. 猫(cat) in this case, a CNN with width 2 is applied to the sequence of Romanized pinyin letters, followed by max-pooling to derive the final pinyin embedding.

Fusion Embedding

获得 char-embedding，glyph-embedding，pinyin-embedding，三层连接起来形成3d-dimensional vector，经过全连接层得到fusion-embedding，fusion-embedding加上position-embedding，输出到BERT层，如Fig.4所示。

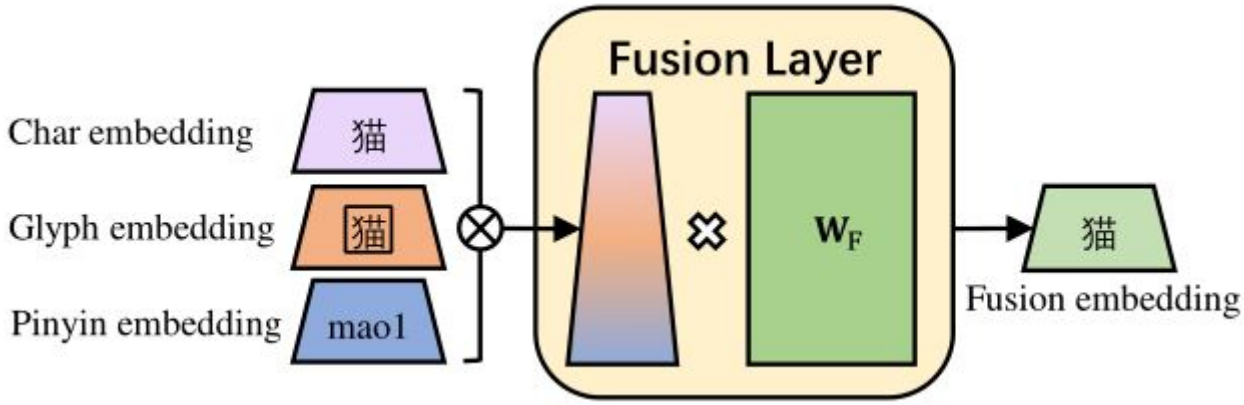

Figure 4: An overview of the fusion layer. ⊗ denotes vector concatenation, and × is vector-matrix multiplication. We concatenate the char embedding, the glyph embedding and the pinyin embedding, and use an FC layer with a learnable matrix $\mathbf{W}_F$ to induce the fusion embedding.

3.3 Output

4.预训练设置

4.1Data

CommonCrwal获取预训练数据，预处理后(删除英文文本过多数据，过滤html标签等)，保留10%的高质量数据进行预处理训练，共包含约4B个汉字。使用LTP toolkit来识别汉语词边界。

4.2 Masking Strategies

Whole Word Masking(WWM)/Char Masking(CM)

4.3 Pretraining Details

概率为0.9的分组输入(512个句子串联)和0.1的单个句子输入。
90%时间内使用WWM，10%时间内CM。
单词/字符 Mask概率15%，掩盖后，80%时间对其掩盖，10%时间对其替换，10%时间保持。
2 model setup: base/large，12/24 Transformer layer，768/1024 Dimension，12/16 heads per layer

base model: 500K setps, learning rate=1e-4, warmup of 20K steps, batch size of 3.2K
large model: 280K steps, learning rate=3e-4, warmup of 90K steps, batch size of 8K
预训练完成后，本模型可以像BERT用域下游任务微调。

5.实验

- Machine Reading Comprehension(MRC)
- Natural Language Inference(NLI)
- Text Classification(TC)
- Sentence Pair Maching(SPM)
- Named Entity Recognition(NER)
- Chinese Word Segmentation(CWS)

Compare: ERNIE/BERT-wwm/MacBERT/

ERNIE: token-level/phrase-level/entity-level masking

	ERNIE	BERT-wwm	MacBERT	ChineseBERT
Data Source	Heterogeneous	Wikipedia	Heterogeneous	CommonCrawl
Vocab Size	18K	21K	21K	21K
Input Unit	Char	Char	Char	Char
Masking	T/P/E	WWM	WWM/N	WWM/CM
Task	MLM/NSP	MLM	MAC/SOP	MLM
Training Steps	-	2M	1M	1M
Init Checkpoint		BERT	BERT	random
# Token	–	0.4B	5.4B	5B

Table 1: Comparison of data statistics between ERNIE (Sun et al., 2019), BERT-wwm (Cui et al., 2019a), MacBERT (Cui et al., 2020) and our proposed ChineseBERT. T: Token, P: Phrase, E: Entity, WWM: Whole Word Masking, N: N-gram, CM: Char Masking, MLM: Masked Language Model, NSP: Next Sentence Prediction, MAC: MLM-As-Correlation. SOP: Sentence Order Prediction.

5.1 Machine Reading Comprehension

机器阅读理解，基于给定上下文回答问题。CMRC 2018数据集和CJRC数据集，CMRC2018/CJRC分别包含10K/3.2K/4.9K和39K/6K/6K数据实例，分别用于train/dev/test。
Result：

Model	*CMRC* Dev	Test
	Base	
ERNIE	66.89	74.70
BERT	66.77	71.60
BERT°	66.96	73.95
RoBERTa°	67.89	75.20
MacBERT	–	–
ChineseBERT	**67.95**	**75.35**
	Large	
RoBERTa°	70.59	77.95
MacBERT	–	–
ChineseBERT	**70.70**	**78.05**

Table 2: Performances of different models on CMRC. EM is reported for comparison. ∘ represents models pretrained on extended data.

Model	*CJRC* Dev EM	Dev F1	Test EM	Test F1
		Base		
BERT	59.8	73.0	60.2	73.0
BERT°	60.8	74.0	61.4	73.9
RoBERTa°	62.9	76.6	63.8	76.6
ChineseBERT	**65.2**	**77.8**	**66.2**	**77.9**
		Large		
RoBERTa°	65.6	77.5	66.4	77.6
ChineseBERT	**66.5**	**77.9**	**67.0**	**78.3**

Table 3: Performances of different models on the MRC dataset CJRC. We report results for baseline models based on their released models. ∘ represents models pretrained on extended data.

ChineseBERT均有提高

5.2 Natural Language Inference

NLI确定假设和前提之间的隐含关系。Cross-lingual Natural Language Inference(XNLI)数据集。5k train/2.5K dev。

Model	*XNLI* Dev	Test
	Base	
ERNIE	79.7	78.6
BERT	79.0	78.2
BERT∘	79.4	78.7
RoBERTa∘	80.0	78.8
MacBERT	80.3	79.3
ChineseBERT	**80.5**	**79.6**
	Large	
RoBERTa∘	82.1	81.2
MacBERT	82.4	81.3
ChineseBERT	**82.7**	**81.6**

Table 4: Performances of different models on XNLI. Accuracy is reported for comparison. ∘ represents models pretrained on extended data.

ChineseBERT均有提高

5.3 Text Classification

文本分类，THUCNews数据集，是THUCTC子集，50K/5K/10K for training/dev/test, 分为10类。
ChnSentiCorp数据集，9.6K/1.2K/1.2K for training/dev/test。
with vanilla BERT, accuracy 95%.
以上数据集太简单，添加了TNEWS数据集，CLUE benchmark更困难的数据集，15类，53K/10K/10K for training/dev/test

Result:
THUCNews/ChunSentiCorp, ChineseBERT改进微不足道，baseline就有95%了，TNEWS数据集，ChineseBERT有所改进，以及额外数据网络训练的ERNIE表现略逊一筹。

Model	*ChnSentiCorp* Dev	*ChnSentiCorp* Test	*THUCNews* Dev	*THUCNews* Test	*TNEWS* Dev	*TNEWS* Test
			Base			
ERNIE	95.4	95.5	97.6	97.5	58.24	58.33
BERT	95.1	95.4	98.0	97.8	56.09	56.58
BERT°	95.4	95.3	97.7	97.7	56.77	56.86
RoBERTa°	95.0	95.6	**98.3**	97.8	57.51	56.94
MacBERT	95.2	95.6	98.2	97.7	–	–
ChineseBERT	**95.6**	**95.7**	98.1	**97.9**	**58.64**	**58.95**
			Large			
RoBERTa°	**95.8**	95.8	**98.3**	97.8	58.32	58.61
MacBERT	95.7	**95.9**	98.1	**97.9**	–	–
ChineseBERT	**95.8**	**95.9**	**98.3**	**97.9**	**59.06**	**59.47**

Table 5: Performances of different models on TC datasets ChnSentiCorp, THUCNews and TNEWS. The results of TNEWS are taken from the CLUE paper (Xu et al., 2020). Accuracy is reported for comparison. ◦ represents models pretrained on extended data.

Sentence Pair Matching

SPM判断给定的足迹是否有相同的语义。LCQMC/BQCorpus数据集。LCQMC，23.9K/8.8K/12.5K for training/dev/test。BQCorpus, 100K/10K/10K for training/dev/test.
Result:
ChineseBERT在LCQMC表现优于MacBERT，略低于BERT-wwm。

Model	*LCQMC* Dev	*LCQMC* Test	*BQ Corpus* Dev	*BQ Corpus* Test
		Base		
ERNIE	89.8	87.2	86.3	85.0
BERT	89.4	87.0	86.1	85.2
BERT°	89.6	87.1	**86.4**	**85.3**
RoBERTa°	89.0	86.4	86.0	85.0
MacBERT	89.5	87.0	86.0	85.2
ChineseBERT	**89.8**	**87.4**	**86.4**	85.2
		Large		
RoBERTa°	90.4	87.0	86.3	85.8
MacBERT	**90.6**	87.6	86.2	85.6
ChineseBERT	90.5	**87.8**	**86.5**	**86.0**

Table 6: Performances of different models on SPM datasets LCQMC and BQ Corpus. We report accuracy for comparison. ○ represents models pretrained on extended data.

5.5 Named Entity Recognition

模型识别文本片段中的命名实体，将其序列标注。OntoNotes4.0/Weibo。OntoNotes4.0有18种命名实体类型，15K/4K/4K for training/dev/test。Weibo有4种命名实体类型，1350/270/270 for training/dev/test。

Result：

ChineseBERT F1 高于BERT/RoBERTa，尽管base精度略有损失，但是recall收益高，使得F1高。

Model	*OntoNotes 4.0* P	R	F	*Weibo* P	R	F
			Base			
BERT	79.69	82.09	80.87	67.12	66.88	67.33
RoBERTa°	**80.43**	80.30	80.37	**68.49**	67.81	68.15
ChineseBERT	80.03	**83.33**	**81.65**	68.27	**69.78**	**69.02**
			Large			
RoBERTa°	80.72	82.07	81.39	66.74	70.02	68.35
ChineseBERT	**80.77**	**83.65**	**82.18**	**68.75**	**72.97**	**70.80**

Table 7: Performances of different models on NER datasets OntoNotes 4.0 and Weibo. Results of precision (P), recall (R) and F1 (F) on test sets are reported for comparison.

Chinese Word Segmentation

将文本分成单词，并将其形式化基于字符的序列标注任务。PKU/MSRA进行中文分词。PKU，19K/2K for training/test。MSRA，87K/4K for training/test。输出字符经过softmax获得最终预测结果。

Result：

ChineseBERT优于其他。

Model	*MSRA* F1	*MSRA* Acc	*PKU* F1	*PKU* Acc
		Base		
BERT°	98.42	99.04	96.82	97.70
RoBERTa°	98.46	99.10	96.88	97.72
ChineseBERT	**98.60**	**99.14**	**97.02**	**97.81**
		Large		
RoBERTa°	98.49	99.13	96.95	97.80
ChineseBERT	**98.67**	**99.26**	**97.16**	**98.01**

Table 8: Performances of different models on CWS datasets MSRA and PKU. We report F1 and accuracy (Acc) for comparison. ∘ represents models pretrained on extended data.

6.Ablation Studies

to understand behaviors of ChineseBERT, OntoNotes4.0数据集，base version。

6.1 The Effect of Glyph Embeddings and Pinyin Embedding

控制变量法，only glyph+char/only pinyin+char/only char(退化为RoBERT)，数据集NER-OntoNotes。

Model	*OntoNotes 4.0* Precision	Recall	F1
RoBERTa°	80.43	80.30	80.37
ChineseBERT	80.03	83.33	81.65
– Glyph	77.67	82.75	80.13 (-1.52)
– Pinyin	77.54	83.65	80.48 (-1.17)
– Glyph – Pinyin	78.22	81.37	79.76 (-1.89)

Table 9: Performances for different models without considering glyph or pinyin information.

去除glyph/pinyin均会影响性能，对F1-score影响最大。only char表现比RoBERT差的原因是使用的模型在更小的数据规模和更少的训练步骤上训练的。

6.2 The Effect of Training Data Size

本文假设glyph/pinyin对文本语义有较强的正则化作用，这表示ChineseBERT能在较少的训练数据下表现更好。

随机抽取10%-90%的训练数据，同时保留Entities和不保留Entities samples，5次实验，报告平均F1。

可以看出，ChineseBERT所有设置表现较好，训练数据占比小于30%时，ChineseBERT提高幅度较小，训练数据占比超过30%时，性能提升幅度较大。表明ChineseBERT仍需要足够的训练数据来训练glyph/pinyin，训练数据不足会导致训练不足。

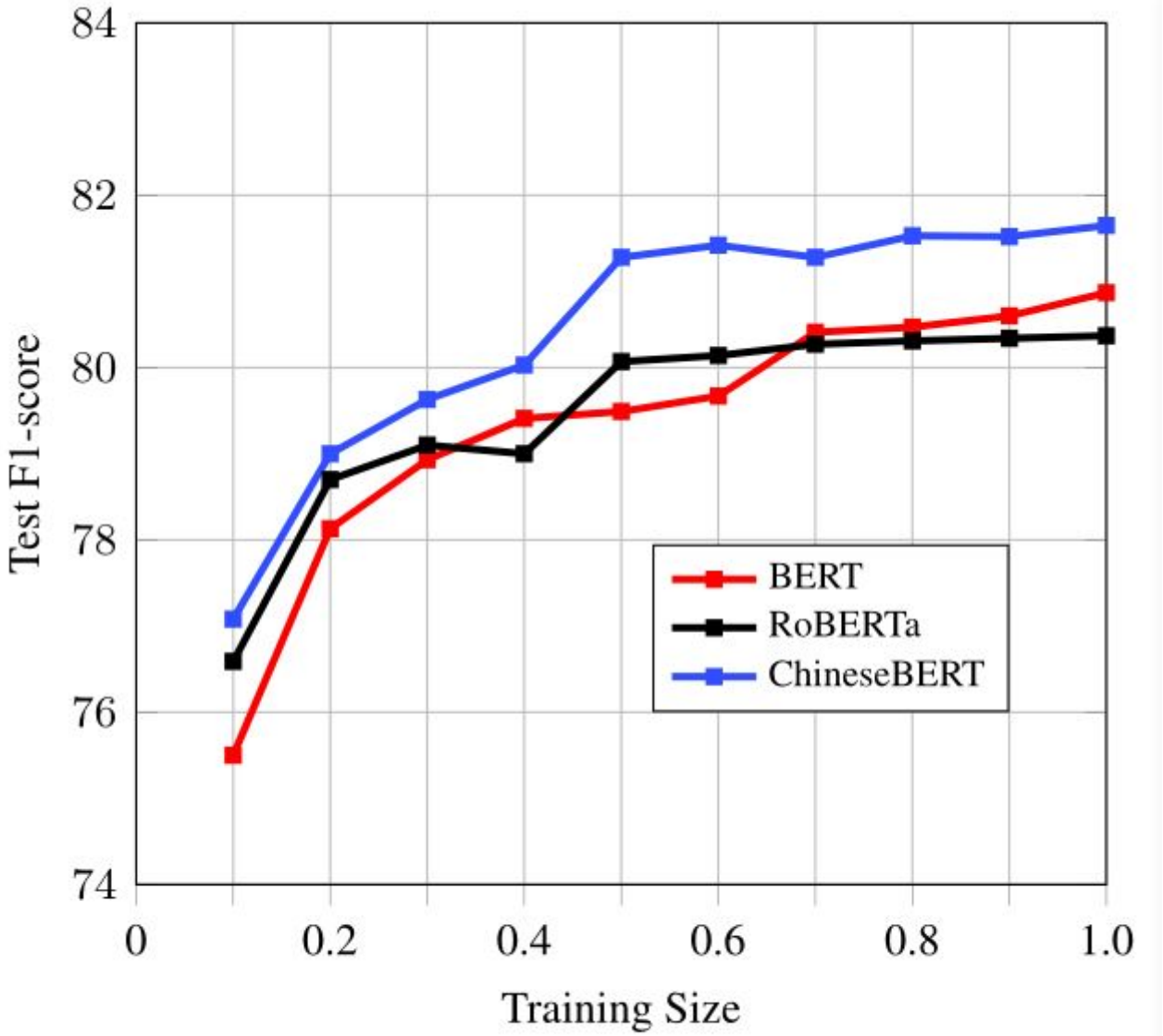

Figure 5: Performances when varying the training size.

7.结论

利用glyph/pinyin embedding，增强了模型对于汉字获取上下文语义和消除汉字复音字歧义的能力。ChineseBERT在NLP任务中取得了提升。