Semi-supervised Bert for question answering matching

Zhang Liang[1], **Ye Peigen**[2], **Wang Haojie**[3], **Lu Xinyu**[4], **Fu Biao**[5]

[1]31520201153906, [2]31520201153093, [3]23020201153802, [4]20520201151962, [5]31520201153868

[1,2,5]Department of Artificial Intelligence, Xiamen University

[3]Department of Computer Science, Xiamen University

[4]Department of Chemistry, Xiamen University

Abstract

This paper presents Semi-Bert which combined semi-supervised learning and pre-trained language models. We use Semi-Bert in chat Q&A matching to help many companies judge whether the service provider provides the correct answer to the customer. Semi-Bert consists of two parts: supervised learning and unsupervised learning. The supervised learning part mainly fine-tunes Bert to adapt to downstream tasks. The unsupervised learning part mainly improves the robustness of the model and makes the Bert perceive the data distribution of the unlabeled data to improve the generalization ability of the Semi-Bert. We also use data augmentation to generate a large amount of data for unsupervised learning. The final loss of our model is the sum of the cross-entropy loss of supervised learning and the consistency loss of unsupervised learning. Our experiments show that our model achieves inspiring performance on the dataset of the Real Estate Industry Chat Quiz Matching Competition in DataFountain.

Introduction

Many companies will assign a service provider, such as real estate agent and insurance account manager, to answer customer questions to better serve customers. With the rapid development of the Internet, Instant Messaging (IM) has become the most popular communication method on the Internet. At present, most service providers and customers communicate on the IM APP. IM communication is a necessary link for both parties to build trust. Customers need to frequently ask the server many questions in this scenario, and it is very important whether the server provides the customer with a good feeling and professional service. Therefore, there is a crucial issue how to judge whether the service provider has answered the customer's question, so we developed a chat Q&A matching system to solve this problem. Unlike common Q&A systems, our system does not answer customer questions, but judges whether the questions answered by the server are the questions asked by the customer. So our task can be regarded as a classification task, in other words, whether the answer answers the question.

Due to the rapid development of deep learning, previous works is based on neural network modeling of sentence pairs through supervised learning. Kalchbrenner, Grefenstette, and Blunsom (2014) describe a convolutional architecture dubbed the Dynamic Convolutional Neural Network (DCNN) that they adopt for the semantic modelling of sentences. Liu et al. (2015) propose a multi-timescale long short-term memory (MT-LSTM) neural network to model sentences. Yin et al. (2016) presents a general Attention Based Convolutional Neural Network (ABCNN) for modeling a pair of sentences. Recently, the pre-trained language model, such as Bert(Devlin et al. 2018) and RoBerta(Liu et al. 2019), achieved state-of-the-art results in most NLP tasks, the method based on the pre-trained language model has also been applied to Q&A matching and has achieved significant performance. However, the Q&A matching task of chat in IM apps is different from other Q&A matching tasks. The contents of chat in IM apps are random and fragmented. Questions and answers are composed of irregular and informal sentences, and the text length is short, so the previous method can't solve this problem well. therefore, we combined semi-supervised learning and pre-trained language models to solve this problem.

Related work

Pre-training models Pre-training models and fine-tuning methods have achieved great success in NLP applications in recent years, and have been applied to various NLP tasks. As we all know, Bert model is a language representation model released by Google in October 2018, and Bert has swept the optimal results of 11 tasks in the field of NLP, which is arguably the most important breakthrough in NLP nowadays. Bert model is a model obtained by training Masked Language Model and predicting the next sentence task. Our task is to automatically detect whether a question answer matches, which is a text classification problem. Considering the excellent performance of BERT on NLP tasks, we choose BERT as our baseline.

Semi-supervised learning of text data Supervised learning often requires a large amount of labeled data, and the cost of labeled data is relatively high, so it is of great importance how to use a large amount of unlabeled data to improve the effect of supervised learning. This way of learning using small amount of labeled data and large amount of unlabeled data is called Semi-Supervised Learning (SSL). Semi-supervised learning has received a lot of attention in the field of NLP because unlabeled data is often rich compared to labeled data. Lee (2013) predicts the unlabeled data by the network, and then sharpen the predictions to obtains pseudo-labels. Xie et al. (2019) proposes a data augmentation method based on unsupervised data, UDA (Unsupervised Data Augmentation). The UDA method generates unsupervised data that is consistent with the original unsupervised data in terms of distribution.

Text data augmentation Since the training data is insufficient, we choose to use data augmentation methods to expand the dataset. Data augmentation methods in the field of deep learning vision can largely improve the performance of the model and reduce the data dependency, while doing data augmentation on NLP is not as convenient as on images, but there are still some methods. Unlike data augmentation using images in computer vision, text data augmentation in NLP is very rare. This is because some simple operations on an image, such as rotating it or converting it to grayscale, do not change its semantics. The presence of semantic invariant transformations makes augmentation an important tool in computer vision research. Wei and Zou (2019) expands text data using synonym replacement, random insertion, random swapping, and random deletion. In many previous studies on data augmentation for supervised or semi-supervised learning, the data augmentation method of reverse translation has been used. Reverse translation not only increases linguistic diversity but also preserves the semantic information of the sentences, which greatly improves the robustness of the model.

Methodology

In this part, we first describe our architecture, and then introduce each key part of the model in detail.Our Semi-Bert is shown Figure 1.

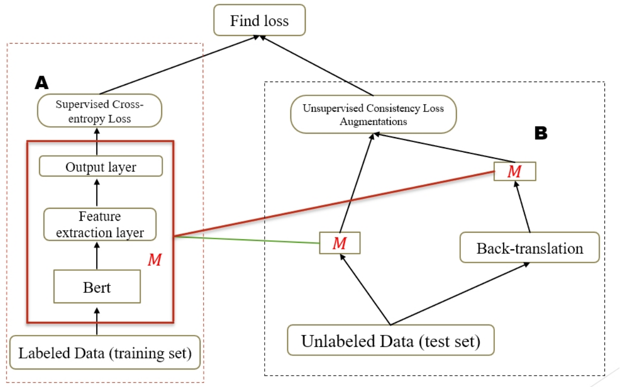

Find loss
A
Supervised Cross-entropy Loss
Output layer
Feature extraction layer
M
Bert
Labeled Data (training set)
Unsupervised Consistency Loss Augmentations
B
M
M
Back-translation
Unlabeled Data (test set)

Figure 1: Semi-Bert consists of two parts: (1)Supervised learning (A); (2)Unsupervised learning(B); parameters are shared by three M modules in the middle. The blue line indicates shared parameters without gradient return, and the red line indicates those parameters whose gradients can be returned.

Semi-Bert consists of two parts: 1.Supervised learning (A); 2.Unsupervised learning(B). Supervised learning is responsible for slightly adjusting Bert to adapt it to downstream tasks.Unsupervised learning is responsible for: (1) improving the robustness of the model, (2) making the Bert perceive the distribution of unlabeled dataset, thereby improving the generalization ability.The loss of the final model is the sum of the cross-entropy loss of supervised learning and unsupervised consistency loss.

Supervised Learning (A)

This part is mainly composed of input layer, **Bert layer**, **feature extraction layer** and **output layer**. **Input layer:**We follow the design of Bert's input layer and stitch a question and its corresponding answer into a sentence pair form, $[cls]question[sep]answer1[sep]$. We add the word embedding, segment embeding and position embedding, then input it into the Bert. As shown Figure 2.

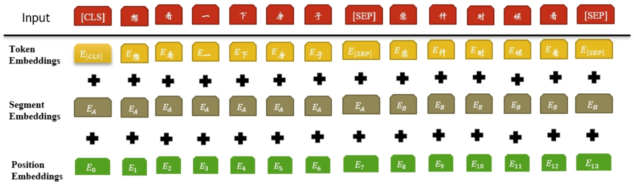

Input
[CLS] 想 看 一 下 房 子 [SEP] 您 什 时 候 看 [SEP]
Token Embeddings
E[CLS] E想 E看 E一 E下 E房 E子 E[SEP] E您 E什 E时 E候 E看 E[SEP]
Segment Embeddings
EA EA EA EA EA EA EA EA EB EB EB EB EB EB
Position Embeddings
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13

Figure 2: Input layer

Bert layer: The Bert model is our backbone. The standard Bert released by Google we use is mainly used to encode sentence pairs in the input layer to obtain context-sensitive hidden layers' representations.

Feature extraction layer: This layer extracts a feature from the 12 hidden layers representations generated by the Bert. We mainly selected two features: pooling features and CNN features.(1)For the pooling feature, we first average pool the output of the 12 hidden layers generated by Bert, then maximize the pooling of the 12 average pooling results. Maximum level will be used to extract the useful features of each layer.(2)According to Jawahar, Sagot, and Seddah (2019), we know that each hidden layer's output in the Bert model contains different language information, some contain superficial language information, and some contain deep grammatical and semantic information. We select the 7, 9, and 12 layers which contain grammatical and semantic information that are most helpful for text classification for CNN feature extraction. Specifically, we first sum the 7, 9, and 12 layers, and then perform CNN operations to extract CNN features. (3)Finally, we spliced the two features into the output layer.The whole process is shown in the Figure 3.

Output layer: Output layer is a full link layer plus a softmax layer, and finally get the final classification probability.We use cross-entropy loss function in supervised learning.

Unsupervised Learning (B)

In the design of the unsupervised learning, we followed Xie et al. (2019), this module is composed of three parts: **input layer**, **reverse translation data enhancement layer** and **Bert model**.

The design of the input layer is the same as the input layer of the supervised learning. The only difference is their

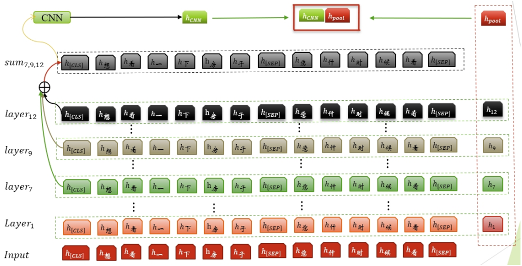

CNN
h_{CNN}
h_{CNN} h_{pool}
h_{pool}
$sum_{7,9,12}$
$layer_{12}$
$layer_9$
$layer_7$
$Layer_1$
$Input$
$h_{[CLS]}$ $h_{想}$ $h_{看}$ $h_{一}$ $h_{下}$ $h_{房}$ $h_{子}$ $h_{[SEP]}$ $h_{您}$ $h_{什}$ $h_{时}$ $h_{候}$ $h_{看}$ $h_{[SEP]}$
h_{12}
h_9
h_7
h_1

Figure 3: Feature extraction layer

data source. One is unsupervised data without labels, and the other is supervised data with labels.

We take a batch sample from the unsupervised dataset each time, and send these samples to the reverse-translated data enhancement model to obtain a batch enhancement sample, and then pass the original source and the enhanced sample into the Bert model separately , And then sharpen its output by controlling the temperature of the source sample output layer softmax, and finally calculate the KL divergence of the source sample and the enhanced sample output as the consistency loss function.

It should be noted that the ***M*** module in the unsupervised module is a copy of ***M*** module in the supervised learning. And in the unsupervised learning, ***M*** on the left does not perform the back propagation of the gradient, and the gradient of the unsupervised loss function can only be passed back from ***M*** on the right.The detailed diagram of the calculation process of this module is as Figure 4.

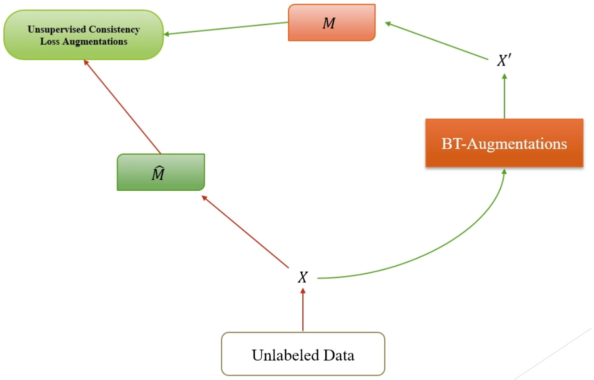


Figure 4: Unsupervised Learning module:$\hat{\text{M}}$ indicates that the model does not back-propagate the gradient.The final loss: the sum of the supervised learning loss and the unsupervised loss.

Data enhancement

We use reverse translation and random perturbation operations for data enhancement. This method has been used in many previous studies on data enhancement of supervised learning or semi-supervised learning. Reverse translation can not only increase the linguistic diversity but also retain the semantic information of the sentence, which can greatly improve the robustness of the model.

In order to increase the diversity of languages, we followed Wei and Zou (2019), before reverse translation of the sentence, a series of disturbance operations are performed on the sentence, including: unified word replacement, random Insert and random swap. However, we did not perform random deletion. The main reason is that random deletion may make a big change in the semantics of the sentence.

Optimization Strategy

Based on intuition and previous pre-trained models(Sun et al. 2019; Howard and Ruder 2018). The lower layer of the Bert model contains more general semantic information, so we hope that the lower layer has a small learning rate when fine-tuning. Therefore, we use a layer-by-layer decreasing learning rate and parameter freezing method to ensure that the Bert model will not be catastrophically forgotten during fine-tuning. Specifically, we freeze the entire Bert model at the beginning of the model iteration training, and as the iteration increases, we gradually unfreeze the Bert model from top to bottom. At the same time, we use the following formula to set the learning rate of different layers of Bert(1):

$$\zeta^{k-1} = \gamma \zeta^{k} \tag{1}$$

Where ζ represents the learning rate of the k-1th layer, krepresents the learning rate of the kth layer, and γ is a coefficient less than or equal to 1

We all know that in the early stage of training, the model usually perform unstably. At this time, the gradient will fluctuate relatively large, causing parameters to oscillate violently. This is very unfavorable for our fine-tuned pretraining model, and may even cause the forgetful disaster of our pre-trained model. In order to solve this problem, we use the oblique triangle learning rate. In the early stage of model training, we use a lower learning rate. As the number of model iterations increases and the model performance improves, we gradually increase the learning rate. In the later stage of training, in order to ensure the model stability We gradually reduce the learning rate of convergence. This changing learning rate is like the oblique triangle in the Figure 5.

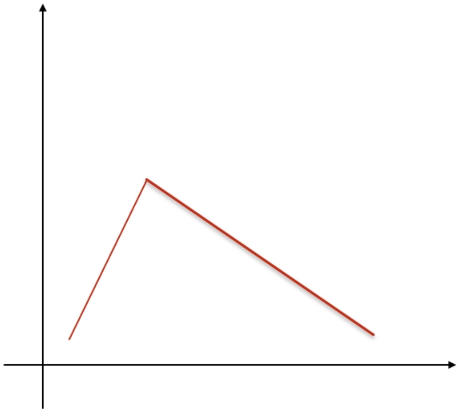
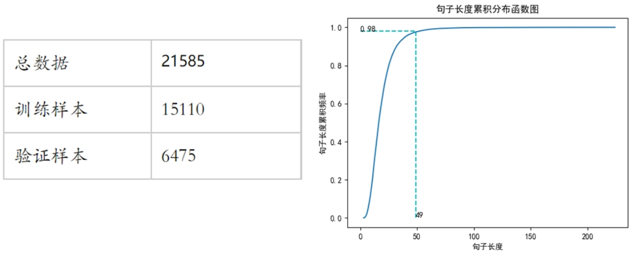


Figure 5: Slanted triangular learning rates

Experiments

Dataset

We used the dataset of the Real Estate Industry Chat Quiz Matching Competition in DataFountain to verify our model. Each piece of data in the dataset includes three fields: question field, answer field and label field. The statistics of the dataset are as shown Figure 6.



总数据	21585
训练样本	15110
验证样本	6475

Figure 6: The left half of the figure describes the number of samples in the dataset, which are the total number of samples in the data, the number of samples in the training set, and the number of samples in the validation set. The figure on the right shows the statistical information of the sample length in the dataset.

From the figure, it can be seen that 98% of the sample lengths are less than 50, so we set the longest sample length to 90 to cover more long samples.We use the training set as the dataset for supervised learning, and we remove the labels from the entire sample set as the data for unsupervised learning.

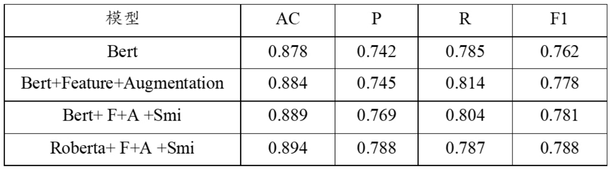
Data Augmentation

We take use of a Google translation crawler to achieve our reverse translation. We first perform random synonym replacement, random insertion and random exchange operations from the source Chinese sentence, and then translate it into English and then translate English back to Chinese.

We perform this data enhancement process on the entire dataset, so that more than 20,000 samples will be expanded to more than 40,000 samples. Then we use the training set and its corresponding enhancement samples as the training set to achieve supervised data enhancement. We remove the labels of the entire sample set and the enhanced samples of the entire sample set as unsupervised training data.

Parameter setting

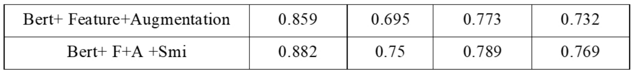
We use the BERT-base model with 12 layers and a hidden layer dimension of 769; the dropout rate used throughout the model training process remains unchanged at 0.1. Using the adam optimizer where β_1=0.9 and β_1=0.999, the basic learning rate is 2×10-5; the warm-up probability is 0.1; and the maximum sentence length we use is 90.For the convolutional layer, we use three filters, and the window size of each filter is 2, 3, 4. The filter depth is 256.In the final stage of training, we replaced the Bert model with the Roberta-large model to obtain high accuracy. Our test results are as shown Figure 7.



模型	AC	P	R	F1
Bert	0.878	0.742	0.785	0.762
Bert+Feature+Augmentation	0.884	0.745	0.814	0.778
Bert+ F+A +Smi	0.889	0.769	0.804	0.781
Roberta+ F+A +Smi	0.894	0.788	0.787	0.788

Figure 7: Where Feature represents the feature extraction module, Augmentation represents the data enhancement module, and Smi represents the unsupervised learning module on the left.

In order to further verify the effectiveness of unsupervised learning, we exchange the training set with the verification set, that is, use 30% of the data for training and 70% of the data for testing. The test results are as shown Figure 8.



Bert+ Feature+Augmentation	0.859	0.695	0.773	0.732
Bert+ F+A +Smi	0.882	0.75	0.789	0.769

Figure 8: The effectiveness of unsupervised learning

Conclusion

In this paper, we propose a new Smi-Bert model, which is a semi-supervised model composed of a supervised learning module and an unsupervised learning module. We mainly use the Smi-Bert model to solve the question and answer matching problem in the real estate industry. Our experiments further verified: (1) The pre-training model can greatly improve the performance of downstream tasks; (2) The information contained in each layer of the pre-training model(Bert) is different, and some layers are very helpful for classification tasks. (3) Fine-tuning strategies such as Slanted triangular learning rates and Discriminative finetunin can further improve model performance and can effectively avoid forgetting disasters; (4) Unsupervised consistency training can help the model in unlabeled datasets to further improve the model performance, and the performance improvement is more obvious when the amount of training data is very small. (5) The data enhancement method of reverse translation and random disturbance can effectively expand the dataset to improve the performance and robustness of the model.

References

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328–339.

Jawahar, G.; Sagot, B.; and Seddah, D. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3651–3657.

Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* .

Lee, D.-H. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

Liu, P.; Qiu, X.; Chen, X.; Wu, S.; and Huang, X.-J. 2015. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2326–2335.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Sun, C.; Qiu, X.; Xu, Y.; and Huang, X. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, 194–206. Springer.

Wei, J.; and Zou, K. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* .

Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.-T.; and Le, Q. V. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848* .

Yin, W.; Schütze, H.; Xiang, B.; and Zhou, B. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics* 4: 259–272.