Text Classification Based on Bert

Written by Junjie Fei
College of Electronic Science and Technology, Xiamen University,

Abstract

In order to protect the privacy of individual and the company, we need to classify sensitive data. Text classification belongs to natural language processing(NLP). In NLP, text information is preprocessed firstly. In this process, we use the Bert - pre-training model from google. Then, we use this model to complete the task of senstive data classification by fine-tuning Bert.

In the era of big data, data sharing and opening to enterprise development is increasingly prominent. Data has become one of the most important production factors.

A large number of business data, which may include business secrets and employee privacy information, are involved in business management activities such as industry and service, marketing support, business operation, risk control, information disclosure, analysis and decision-making. If these data is leaked due to improper use, it may cause huge economic losses, even damage enterprise's credit.

Around data security, the state has promulgated a number of laws in recent years. Our country attaches great importance to the protection of sensitive data, especially in key infrastructure and various mobile applications. In order to effecting and standardizing the protection of enterprise sensitive data, the first problem is to classify the data to identify sensitive data. We get the sensitive data so as to further carry out open and dynamic data security governance around the protected object, and solve the contradiction and unity of data between open sharing and privacy protection.

The semantic recognition technology based on natural language processing has been widely used in the existing sensitive data recognition and classification, but there are some problems as follows:

- It needs a large number of high-quality annotation data, which costs a lot of manpower and time, and has high construction cost.
- The ability of generalization is insufficient; the adaptability to new business data is weak; the false positive rate and false negative rate of sensitive data are high.
- It is difficult to carry out self-optimization and self-learning. It also requires manual intervention from experts in the field of business and AI.

Introduction

NLP(Natural Language Processing) is one of the most difficult problems in artificial intelligence, because it is very hard to make AI understand underlying meaning of human language. At the beginning, Rumelhart, Hinton and Williams(1986) use word to represent dates. This idea has since been applied to statistical language model with considerable success by Bengio et al.(2003). After that, Mikolov et al.(2013) introduced the Skip-gram model, an efficient method for learning high-quality vector representations of words from large amounts of unstructured text data. Based on this, Mikolov et al.(2013) proposed Skip-gram model to train word vectors.

Vaswani et al.(2017) gave a paper - Attention is all you need, causing a sensation in the field of NLP. Vaswani abandoned RNN or CNN architecture, and only used self-attention and feed forward neural network to model contextual information. Dai et al.(2019) proposed a novel neural architecture Transformer-XL that enables learning dependency beyond a fixed length without disrupting temporal coherence. Wang et al.(2018) also proposed GLUE Benchmark for collection of diverse natural language understanding tasks.

For NLP, language model pre-training is a good strategy. Clark et al.(2020) proposed pre-training text encoders. Brown et al.(2005) also proposed GPT-3. There are two existing strategies for applying pre-training language: feature-based and fine-tuning. ELMo is a typical feature-based approach(Peters et al. 2018). It combines pre-training word token vectors or contextual word vectors, and learns a deep bidirectional language model(biLM) and uses all its layers in prediction. In addition, it also learns word token vectors using long contexts rather than context windows. Radford et al.(2018) proposed a fine-tuning approach - the Generative Pre-training Transformer(OpenAI GPT), which introduces minimal task-specific parameters. Bert - Bidirectional Encoder Representations from Transformers(Devlin et al. 2018) is also a fine-tuning approach which is a pre-training model using deep bidirectional transformers for language understanding.

Bert, the most popular language pre-training model, is chosen in this text classification task as the following rea-

sons:

- It combines pre-training model and downstream tasks. In other words, when doing downstream tasks, we still use the model - Bert.
- It supports text classification tasks naturally, there is no need to modify the model when doing text classification task.
- Google provides several pre-training model files, we just need fine-tune the pre-training model.
- The major limitation in today's field of NLP is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. Bert uses masked language models to enable pre-training deep bidirectional representations.

Related Work

Since 2016, most studies have focused on the role of long-term context semantics in word embedding and language model pre-training on large-scale corpus. Dai and Le(2015) use language model and sequence self coding to improve the sequence learning of recurrent neural network(RNN), which can be regarded as the beginning of modern PTMs. It systematically expounds the epoch-making idea that the upstream pre-training language model can be used for downstream specific tasks. This viewpoint is supported by a series of experiments on classification tasks. Since then, PTMs has gradually stepped into people's vision.

Subsequently, Ramachandran, Liu and Le(2016) extended above methods and proposed that the accuracy of the sequence to sequence(seq2seq) model(Sutskever, Vinyals and Le 2014) could be improved by using the pre-training method. He proves that the idea of pre-training model on a large number of unsupervised data and fine-tuning model on a small amount of supervised data are also effective for the seq2seq model. Ramachandran proposes the idea of joint-training of seq2seq objectives and language model objectives to improve the generalization ability. The PTMs technology is further developed, which shows the versatility of the method in NLP field.

With the development of computing power, the deep model is constantly improved, and the architecture of PTMs is advancing from the shallow to the deep. Dai and Le use LSTM(Hochreiter and Schmidhuber 1997), which solves the problem of back propagation through time when RNN processes timing models. However, Unidirectional LSTM can only learn the above semantic information. Therefore, scholars from the University of Bologna pioneered the Bidirectional LSTM(Melamud, Goldberger and Dagan 2016). The semantic information of context is integrated into the word embedding, and the relationship between the popular word embedding and language model at that time is carried on. It shows that the vector representation containing context information can be trained with a large number of unlabelled text data, which is significantly better than the traditional word embedding.

In 2018, ELMo proposed a context sensitive text representation method, which performed amazingly on several typical tasks, and could effectively deal with polysemy. After that, GPT(Radford et al. 2018), Bert and other pre-training language models were proposed. PTMs technology began to shine in the field of NLP.

With the SOTA(start of the art) results obtained by ELMo, GPT, Bert and other pre-training models in NLP tasks, a series of improved models based on Bert have been proposed one after another. The pre-training models have been widely used in various downstream tasks. These models have greatly promoted the progress of NLP.

It is worth mentioning that the performance of Bert is a milestone. It has achieved remarkable improvement in 11 basic tasks in the field of NLP. Bert's emergence is based on many important work in the early stage, and it is a master of many important tasks. At the same time, the emergence of Bert has greatly promoted the development of NLP. Many follow-up studies are generally based on the Bert model. It is generally believed that starting from the Bert model, the field of NLP has finally found a way to carry out transfer learning like computer vision.

The emergence of Bert has ushered a new era. After that, a large number of pre-training language models have emerged. These new pre-training language models can be divided into several categories from the model architecture: the improved model based on the Bert model, XLNet and generation model represented by MASS.

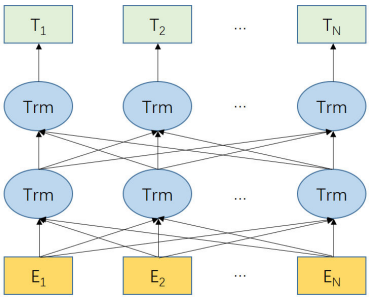

Figure 1: The structure of Bert

.

Proposed Solution

Pre-training

In this text classification task, we choose pre-training model, which is a better set of weights at the end of the training sharing from researchers for others to use. The emergence of pre-training model brings NLP into a new era. Here are the advantages of it:

- It avoids being unable to train due to backward equipments of researchers, especially GPU.
- It can be used to express the large-scale language corpus and complete the subsequent language training.
- Pre-training provides a better model initialization, which usually leads to a better generalization performance and accelerates the convergence of the target task.

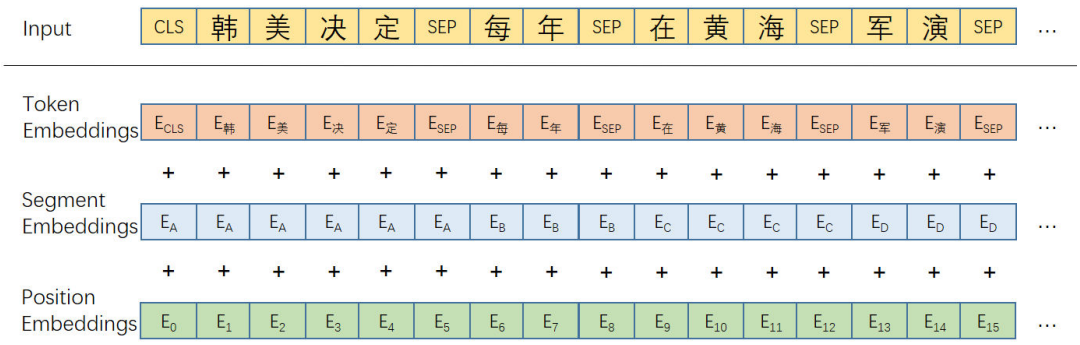

Figure 2: Representation of Input

- Pre-training can be regarded as a regularization to avoid over fitting of small scale data.

Usually, there are two methods for pre-training. feature-based is the process of using the trained network to extract features from new samples. Then, we can input these features into a new classifier to train from scratch. The principle of fine-tuning is to use the known network structure and known network parameters to modify the output layer to our own layer. We fine-tune the parameters of several layers before the last layer. In this way, the powerful generalization ability of deep neural network is effectively utilized, and complex model design and long-time training are avoided. Fine-tuning is a suitable choice when the amount of data is insufficient. Bert's approach is a typical fine-tuning.

The Architecture of Bert

BERT is a pre-training model using deep bidirectional transformers for language understanding. It uses the idea of self-supervised learning, rather than training on any specific NLP task. In this task, we obtain the Bert pre-training model first. Then, we can fine-tune its output layer to adopt our task. The model architecture of Bert(Figuer 1) is a multi-layer biderectional transformer encoder. This thought is similar to the model ELMo, but ELMo is a task-specific model rather than a pre-training model.

The Input of Bert Although Bert is inspired by Transformer, instead of positional embeddings fixed magic number in transformer, Bert's positional embeddings are learnable. Figure 2 shows that the input of Bert is the sum of three embedding features. Among them, position embeddings is to encode the location information of a word into a feature vector. Location embeddings is an important part of introducing the location relationship of words into the model. Segment embedding is used to distinguish two sentences, such as whether B is the following part of A(dialogue scene, question and answer scene, etc.). For sentence pairs, the eigenvalue of the first sentence is 0 and the second sentence is 1. Each input sequence is a pair of sentences, separated by the token [SEP]. It adopted two learnable embeddings to each sentence. [CLS] is a special classification embedding for the first token of every sequence.

Masked Language Model(MLM) The task 1 of pre-training is MLM, which masks some percentage of the input tokens at random, and then predicting only those masked tokens. MLM use token to replace 15% tokens randomly, and use the real token as the label to make it predict. Of course, during fine-tuning, we can never see any [MASK] token. Devlin proposed the following strategy:

- 80% of the time: Replacing the word with the [MASK] token.
- 10% of the time: Replacing the word with a random word
- 10% of the time: Keeping the word unchanged. The purpose of this is to bias the representation towards the actual observed word.

Next Sentence Prediction Next Sentence Prediction makes the model understand the relationship between two text sentences. If we choose the sentences A and B for each pre-training example. There are 50% of the time B is the actual next sentence that follows A, and another 50% of the time it is a random sentence from the corpus.

The Output of Bert For text classification task, we simply plug the task specific inputs and outputs into Bert and fine-tune all the parameters end-to-end. The output of Bert is connected by a Multi-Class Neural Networks and softmax layers(figure3). The output of the whole network is divided into ten categories, which is the target of this text classification. This task will be described in detail in the experiment section.

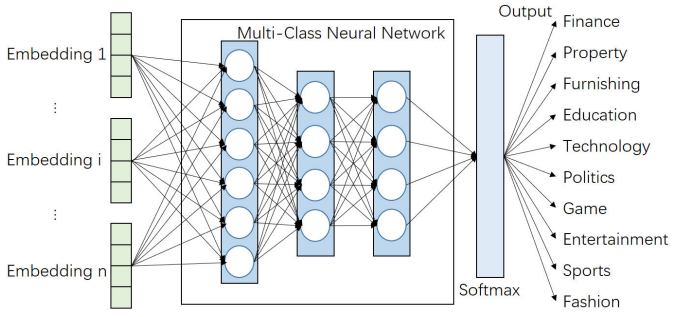

Figure 3: Representation of Output

Automatic Data Annotation

Rule Based Approach Three rules are used here. Based on single category words, the text consists of 10 categories - finance, property, furnishing, education, technology, politics, game, entertainment, sports and fashion. If a paragraph of text contains only one category word, it is considered as belonging to this category. For example, if a paragraph contains only the category word "education", then it is regarded as belonging to the category of education. With this method, 10475 samples can be labelled. Obviously, there are some errors in this method, which will be corrected in the later machine learning method.

Based on category words frequency, if a paragraph contains more than one category words at the same time. For example, if there are two category words of "finance" and "furnishing", the number of times the category words appear in the text will be counted, and the category words with the largest number of times will be used as the label of the text. 3149 samples can be labelled with this method.

Based on custom rules: Except the labelled samples, the remaining 19376 samples don't contain category words. Keyword matching method is used to determine the text category. User-defined keyword matching rules are as follows:

Finance:	基金, 投资, 股票, 分红
Property:	房价, 房贷, 物业, 楼市
Furnishing:	家具, 建材, 厨卫, 涂料
Education:	考试, 学生, 英语, 四六级
Technology:	数码, 手机, 相机, 像素
Politics:	外交, 政治, 时事, 草案
Game:	玩家, 网游, 手游, 技能
Entertainment:	电影, 影视, 奥斯卡, 导演
Sports:	比赛,NBA, 体育讯, 得分
Fashion:	时髦, 潮流, 穿搭, 性感

Figure 4: User-defined keyword matching

If a keyword of a certain category appears in the text, it is considered as belonging to this category. Through this rule, 12228 samples are labelled, and the number of each category is shown in Table 1.

Finance	Property	Furnishing	Education	Technology
1724	583	249	544	1722
Politics	Game	Entertainment	Sports	Fashion
582	1810	2748	1899	367

Table 1: Text annotation results based on custom rules

Data Annotation Based on Word Embeddings Similarity Firstly, the samples' keywords are extracted to train the word embeddings model. The word embeddings' mean of 20 keywords is used to represent the sentence embeddings of samples. Then, the sentence embeddings' mean of each sample is calculated as the category center embeddings. Next, the cosine similarity between unlabelled samples and each category center embeddings is calculated, and the one with high confidence is selected to label. The steps are as follows:

- Sample cleaning, in order to avoid the poor quality of feature extraction due to the large number of symbols appearing in corpus, the space and punctuation in text are deleted.
- Word segmentation is used to segment the text after symbols are deleted. The cut function in 'Jieba' toolkit can accurately segment the word.
- It is helpful to better grasp the content of the text by removing the words that have nothing to do with the theme of the text, such as "le" and "zhi hu zhe ye". Call the Chinese stop glossary to remove the stop words in the text. The common Chinese stop words are in CN_stopwords.txt File.
- The CBOW model, which is trained with the corpus after word segmentation, is used to obtain the word2vec word embeddings model.
- The first 20 keywords are extracted from all 40000 samples by 'Jieba' toolkit, and the embeddings representation of keywords is obtained by word embeddings model.
- 10 categories' center embeddings are obtained by computing category center embeddings of each category
- The cosine similarity between the sentence embeddings of each sample, which is unlabelled, and the center embeddings of 10 categories is calculated. And the one with the largest similarity is selected as the sample category.

Method Based on Self-learning (Data Driven) The idea of data labelled by self-learning is to train a model with a small number of high-quality samples, and this model is used to predict the unlabelled data and labels the samples with high prediction confidence. The steps are as follows:

- Original 7000 given annotation data(7 categories in total) + 1342 samples labelled based on Rules(1000 sports + 553 entertainment + 788 games). The data marked by rules is to expand three categories of data: sports, entertainment and games, which are not included in the original 7000 samples. Above data are used to train the text classification model based on the Bert pre-training model.
- Using the model trained in step 1, 33000 unlabelled data were labelled with pseudo labels. The results with confidence greater than the threshold value of 0.9 were labelled.
- The number of each category is limited to 3000, so as to keep the balance of all kinds of samples and prevent the imbalance of classifier discrimination due to the excessive number of certain data. Finally, the data set is used for self-learning.

Supervised Learning Based on Pre-training Model

Data Preprocessing Data loading and cleaning, after reading these data, it is found that there is a "group picture:" string at the beginning of each sample of fashion category.

It is deleted to avoid the word frequency is too high, which will affect the calculation of word frequency. The results of one-hot coding for 10 category tags are as follows:

- Finance: 1,0,0,...,0
- Property: 0,1,0,...,0
- ...
- Fashion: 0,0,0,...,1

Bert Transfer Learning Loading the Bert pre-training model. Because the project needs to classify ten categories of text, the number of neurons in the output layer needs to be adjusted to 10. The network is initialized with pre-training parameters, then the network parameters are fine-tuned by supervised learning with the data set constructed by ourselves.

Experiments

Experimental Results Based on Word Embeddings Similarity Data Annotation

Large Scale Data Annotation Based on Word Embeddings Similarity When the word embeddings similarity method is used to label samples, a certain threshold is set for the annotation results. When the confidence level is greater than the threshold value, the annotation is considered to be effective. The number of samples labelled under different thresholds is shown in the second column of Table 2. This part of data is combined with the given high-quality labelled data(7000 samples), the data based on single category words method(10475 samples) and the data based on category words frequency method(3149 samples) to form a large-scale annotation data set, which is trained based on the Bert pre-training model. The experimental results are shown in Table 2.

threshold	high confidence scale	dataset scale	F1
0	19376	40000	75.00%
0.65	14858	35482	74.02%
0.70	11920	32544	74.89%
0.75	8040	28664	73.55%

Table 2: Experimental results based on word embeddings similarity data annotation

Direct Labelled Test Set Based on Word Embeddings Similarity This processing steps of this part are the same as above, The difference is that 19376 unlabelled samples are ignored. Using the given high-quality labelled data(7000 samples), data based on single category words method(10475 samples), and data based on category words frequency method(3149 samples). This part of the high-quality annotation data is used as training set, and the labels of the test set samples are obtained directly according to the cosine similarity between the test set and the training set. This direct matching method can achieve 83.39% F1 on the test set, which is much higher than the neural network method. It can be seen that the low quality of large-scale data annotation is still the biggest reason for the performance of deep learning.

Experimental Results Based on Self-learning Data Annotation

1000 sports, 554 entertainment and 788 games is selected to add to the original 7000 true labelled data to package the classification training based on the Bert pre-training model. The training result is 81.80%. This model is used for self-learning training.

The first iteration of self-learning: The threshold value is set to 0.9, and the test result is 82.48%. The reason why the performance is not improved is that the training data categories are not balanced. According to the statistics of training set data distribution, it is found that the entertainment category samples are the least, only 1940. Therefore, the first 1940 data in each category are selected to make the training set. The network training with this data set can achieve the 83.02% F1 in the 9th round.

The second iteration of self-learning: Using the model of the first round of iteration, a data set of self-training is obtained again. However, the self-training method is to sort all 33000 classified output scores. We select the top n high score data for training. When n = 2000, the test set F1 is 84.7%6, and when n = 3000, the test set F1 is 84.23%.

The third iteration of self-learning: Using the model of the second iteration and iterate again. Again, we train a data set. The self-training method is still to sort all 33000 classified output scores. We select the top n high score data for training. When n = 2000, test set F1 is 84.67%; when n = 3000, test set F1 is 84.58%; when n = 4000, test set F1 is 85.42%.

When we obtained the category of text, we can grade this text according to the category by fellowing rule:

- High risk: Finance, politics.
- Medium risk: Property, technology.
- Low risk: Education, fashion, game.
- Open to the public: Furnishing, sports, entertainment.

Conclusion

Generally speaking, the generalization ability of the project model is not good enough, which mainly has two reasons: On the one hand, there are 10 categories of 33000 unlabelled data, while there are only 7 categories of training sets provided by network. Therefore, 3 categories of test data are not marked. In semi supervised learning, the accuracy of data set annotation is not high enough, so the quality of data set in supervised learning is not high enough, which leads to the failure of training to obtain excellent neural network model. On the other hand, in the process of data set construction, there are various kinds of sample imbalance, which also brings some difficulties to the learning of classification model.

The advantage of Bert for This Task

- We choose Bert pre-training model for its multi-layers self-attention mechanism and bidirectional function.
- Bert proposes MASK and Next Sentence Prediction machanism, which have a excellent performance in NLP task
- Bert has small cost for its pre-training and fine-tuning machanism.

Reference

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *Nature* 323(6088): 533–536.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3(Feb): 1137–1155.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information processing systems* 26: 3111–3119.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30: 5998–6008.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Clark, K.; Luong ,M. T.; Le, Q. V.; and Manning, C. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:10555.2020*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; and Amodei, D. 2005. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, Z. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. 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*.

Dai, A. M.; and Le, Q. V. 2015. Semi-supervised sequence learning. *Advances in Neural Information Processing Systems* 3079–3087.

Ramachandran, P.; Liu, P. J.; and Le, Q. V. 2016. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems* 27: 3104–3112.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8): 1735–1780.

Melamud, O.; Goldberger, J.; and Dagan, I. 2016. context2vec: Learning generic context embedding with bidirectional lstm. *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning* 51–61.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training.