Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/45/20241266

Tolerance analysis of wrong labels based on transformer for sentiment analysis

Jianing Liu

Ulster College, Shaanxi University of Science and Technology, Xi'an, 710021, China

202115030413@sust.edu.cn

Abstract. In recent years, with the development of deep learning, the field of sentiment analysis has become more and more widely used. This increases the complexity and diversity of the data, which makes people hard to classify sentiment completely correctly. This paper proposes to use the Transformer model to analyze the tolerance for wrong labels in the process of sentiment analysis tasks. This paper tests the accurate values at different proportions by changing the labels proportionally. Since this paper selects a binary classification text dataset, the specific implementation method is to change 1 into 0 and 0 into 1. The results of this study show that as the proportion of changed labels increases, the overall accuracy of the results shows a downward trend, from 0.888 without changing the label to 0.831 with a 30% change to 0.671 with a 35% change. This paper verifies the stability when changing the 30% label through ten iterations, and the results fluctuate between 0.81 and 0.86. It can be concluded that the Transformer model can accept up to 30% wrong labels when handling sentiment analysis tasks. This study can provide effective help for future model training. Through this study, people can reasonably control the number of wrong labels in future research.

Keywords: Tolerance, Wrong Label, Transformer, Sentiment Analysis.

1. Introduction

With the ongoing advancements in the Internet, social media has seamlessly integrated into the everyday lives of internet users, enabling them to comment and express their opinions on various matters, as well as share their shopping experiences, among other things. The dissemination of this information has a huge impact on society. It may have a positive supervisory role, or it may have a negative incitement. Therefore, it is important to effectively analyze and predict this information. This operation is the increasingly mature sentiment analysis now being developed. An essential step in the process of sentiment analysis is sentiment classification. The remarkable evolution of deep learning in the past few years has prompted its broad application across various areas [1]. The increase in applications generates more data, and the complexity and diversity of data continue to increase, resulting in the inability to exactly correct sentiment classification. Therefore, clarifying the tolerance scale for erroneous data during model learning is crucial. Drawing upon the foundation of deep learning, this paper uses the Transformer model to analyze its tolerance for wrong labels in the sentiment classification task.

Nowadays, more and more research is being done on sentiment analysis, and this technology is becoming more mature. At present, there are three main ways to study sentiment analysis. The first is a sentiment dictionary-based approach. In recent years, Cai et al. successfully addressed the issue of

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/45/20241266

polysemy in sentiment words through the construction of a sentiment dictionary that focused on a specific domain [2]. Similarly, Xu et al. achieved effective text sentiment classification by developing an extended sentiment dictionary that incorporated not only primary sentiment terms but also contextual and multi-meaning sentiment terms [3]. Utilizing traditional machine learning algorithms, the second approach employs Naive Bayes (NB) and support vector machine (SVM) for sentiment analysis. Baid et al. compared the effects of different methods including Bernoulli NB (BNB), SVM, decision tree, and Multinomial NB (MNB) in the sentiment classification, MNB got the best 88.5% [4]. Ahmad et. al proposed an Optimized Sentiment Analysis Framework (OSAF), which uses SVM grid search technology and cross-validation [5]. Sentiment analysis on two Spanish-language datasets has been carried out in recent years, utilizing Bayesian network classifiers by several researchers. By training example, compared with other machine learning methods, it is found that it can predict sentiment polarity more effectively [6]. The third method is based on deep learning, which uses a deep neural network to conduct sentiment classification. Combining pre-training models, attention mechanisms, and other techniques, deep learning methods have gained popularity for sentiment analysis in recent years. Yadav et al. utilized a Gated Recurrent Unit (GRU) along with a bidirectional attention mechanism to develop a unique approach for emotion prediction. Impressively, this approach achieved an accuracy of 89.3% on the Restaurant 16 dataset [7]. In October 2018, a new method was proposed by Google that utilizes Bidirectional Encoder Representation from Transformers (BERT). This approach incorporates semantic word information through a two-way Transformer mechanism [8]. Using the BERT model, researchers have successfully employed sentiment analysis for tasks such as the Stanford Sentiment Treebank (SST). Through experiments on this dataset, an impressive accuracy of 94.7% has been achieved specifically on the SST-2 dataset (binary labels) [9]. Despite various attempts to improve the accuracy of sentiment analysis, previous research suggests that achieving perfect classification of data is a challenging task. The primary focus of this study is to analyze the Transformer model's tolerance for wrong labels during the learning process. Firstly, select a binary classification text data set pass the training test, and record the original test results. Secondly, changing the label proportionally, which means reversing the label (1 becomes 0, 0 becomes 1), and recording the test results at different proportions. The main purpose of this step is to analyze what proportion of tolerance for wrong labels can be achieved in neural network learning. As the proportion of changed labels increases, the overall accuracy of the results shows a downward trend, from 0.888 without changing the label to 0.831 with a 30% change to 0.671 with a 35% change. The experimental results demonstrate that within the range of 30% error labels, there is almost no impact on the learning process of the Transformer model. This study provides effective help for the training of models by discussing the extent to which redundancy accepts wrong labels under the sentiment classification task. Through the test results of this paper, people can reasonably control the number of wrong labels in the future process.

2. Methodology

2.1. Dataset description and preprocessing

This paper selects a take-out dataset from AI studio, and the content is the sentiment classification [10]. It is a binary classification text dataset where 1 represents a positive review and 0 represents a negative review. This dataset contains 11987 pieces of data, in which the ratio of positive reviews to negative reviews is approximately 1:2. The purpose of selecting this dataset is to facilitate subsequent label change processing, that is, to reverse the label by changing 1 to 0 and 0 to 1. Before the train, this paper performs some preprocessing on the dataset. Firstly, to make the dataset easier to divide, the number of data is supplemented to 12000 and partitioned into a training set and a test set, following an 8:2 ratio. Secondly, delete meaningless symbolic words and words that are not Chinese in the dataset to reduce the interference in the learning process of the neural network.

DOI: 10.54254/2755-2721/45/20241266

2.2. Proposed approach

The purpose of this paper is to study the tolerance of wrong labels during neural network learning for sentiment classification tasks. Figure 1 below summarizes the research process. Firstly, select a binary dataset, the reason for choosing binary classification text is to facilitate subsequent changes to labels. Then, do some simple preprocessing on this dataset. Secondly, build the BERT model based on the Transformer to allow the neural network to learn. After that, the comparison method is mainly used during the testing process. At first, through training and testing to get accurate results without changing the label. Afterward, test and record the results of accuracy at different scales by changing the labels in proportion, make statistics on the test results after changing different proportions, and make a histogram to facilitate observation and comparison of the differences with the original test results. Then, verify the stability of the test results. Finally, based on the test results, determine what proportion of wrong labels the neural network can tolerate during the learning process.

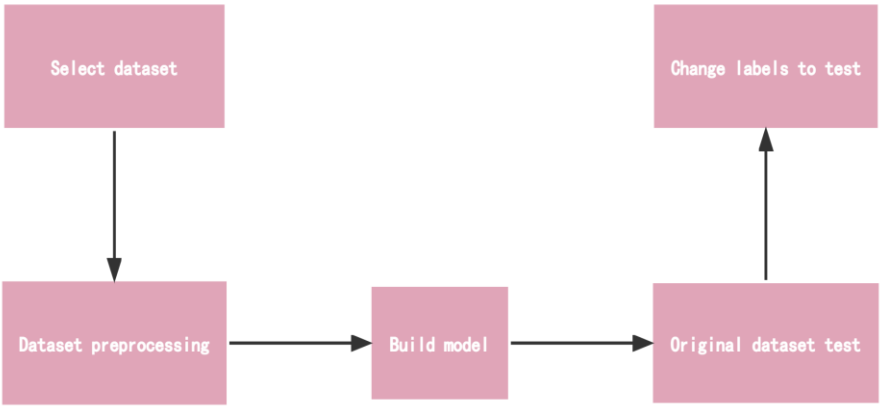


Figure 1. Overall process.

2.2.1. BERT. BERT is an innovative and groundbreaking pre-trained language model that has revolutionized the field. It has made breakthroughs in the field of Natural Language Processing (NLP) and has become one of the most popular language models today. The core of the BERT model is to learn rich language representations through unsupervised large-scale pre-training tasks, and then fine-tune it in various downstream tasks. The BERT model adopts a bidirectional encoding strategy, which considers both contextual information in the self-attention mechanism. Figure 2 illustrates the structure of the BERT model, which is based on the Transformer framework. It consists of variable encoder layers and self-attention heads, serving as the core of the model.

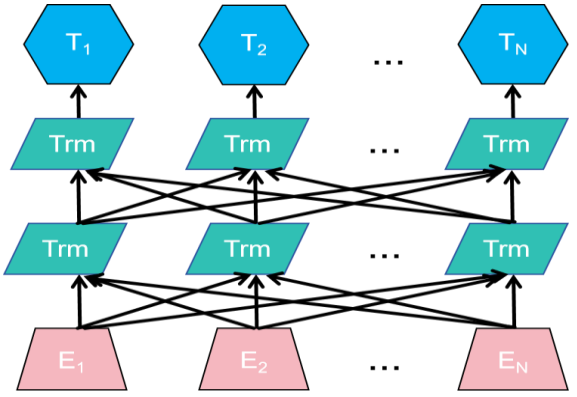


Figure 2. Framework of BERT.

DOI: 10.54254/2755-2721/45/20241266

Firstly, this paper defines a basic Bert classifier model, in which Bert serves as the base model and adds a linear layer for the classification task. Bert's model output enables text classification through linear mapping and probabilistic processing. Afterward, the customized classification data is loaded through the customized Dataset class, and Bert Tokenizer is used to segment and encode the text. The loaded data is converted into the input format required by the Bert model. With these settings, custom classification data can be combined with the BERT model for training and fine-tuning to solve natural language processing tasks. Finally, the information is normalized through the SoftMax function to complete the task.

2.2.2. Transformer. Transformer is a sequence model based on the self-attention mechanism for processing sequence-to-sequence tasks. It models sequence data through a self-attention mechanism and an encoder-decoder structure. This network structure is widely popular in natural language processing because it can handle variable-length sequence data and can capture long-distance dependencies in sequences. This paper uses the Hugging Face transformer library, we built a custom Bert Classifier class with multiple encoder layers and used it for text classification tasks. The core principle of a Transformer is to process sequence data through a self-attention mechanism and encoding and decoding structure. The self-attention mechanism, in sequence-to-sequence tasks, enables the model to effectively capture contextual information and model dependencies between various positions within the input sequence. This is achieved by allowing the model to consider each position's relationship with all other positions. Moreover, the encoder-decoder architecture can be utilized to handle sequence-to-sequence tasks. In the architecture of the Transformer model, an encoder and a decoder play crucial roles, and this paper only uses the encoder part in the text classification task. The encoder layer consists of multiple self-attention mechanism layers and feedforward neural network layers. Stacking encoder layers can extract different levels of semantic information. Preparing the input data and performing feature processing is necessary when using the Transformer model. This paper uses the BERT model as the implementation of the Transformer, obtains the input features through word segmentation, and then passes them to the Transformer model for encoding.

2.2.3. Establishment of wrong labels. The establishment of wrong labels is a very important part of this article because the research of this article is based on this data. The wrong label refers to the reverse label, which means 1 becomes 0, and 0 becomes 1. First, this paper partitioned the dataset into a training set and a test set with an 8:2 ratio. Label changes are applied solely to the training set, which means the wrong label only occurs in the training set. Change in 5% increments, gradually increase, and end at 50%.

2.2.4. Loss function. The importance of the loss function lies in its significant role within deep learning. Since the research in this article is based on sentiment classification text, the Categorical Cross-entropy loss function is a good choice. Cross entropy is mainly used to determine how close the actual output is to the expected output. The cross-entropy primarily explains the discrepancy between the expected output (probability) and the actual output (probability). Consequently, as the cross-entropy value diminishes, it signifies a closer alignment between the two probability distributions. During binary classification, where positive and negative instances are present and their probabilities sum up to 1, the task only requires predicting a single probability, and the definition of the loss function is as follows:

$$\boldsymbol{Loss} = -(\boldsymbol{y} * \mathbf{log}(\hat{\boldsymbol{y}}) + (\mathbf{1} - \boldsymbol{y}) * \mathbf{log}(\mathbf{1} - \hat{\boldsymbol{y}})) \quad (1)$$

The model's positive prediction probability is denoted as $\hat{y}$, while y represents the sample label. If the sample is positive, its value is 1; otherwise, it is 0.

2.3. Implemented details

This article is conducted using Python, and the configuration is as follows: PyTorch: 1.11.0 Python: 3.8 (ubuntu20.04) Cuda: 11.3. The GPU uses RTX 3090 (24GB). During the execution of the model, the

DOI: 10.54254/2755-2721/45/20241266

following settings were made for the hyperparameters: the learning rate was set to 5e-6. The learning rate was not set very high to avoid failure to converge. The batch size used was 8, and the epoch was set to 10. The optimizer chose to use SGD, whose gradient descent algorithm is more efficient. Besides that, the activation function chosen for the model in this study is Gaussian Error Linear Units (GELUs). This function is considered a high-performance neural network activation function, in contrast to other activation functions. It effectively addresses issues such as gradient disappearance, slow convergence speed, and lack of randomness encountered in existing functions. By aligning the function with the natural cognitive process, it enhances consistency and overall performance.

3. Result and discussion

This paper mainly analyzes the tolerance of neural networks to wrong labels in sentiment analysis tasks. This paper selects a binary classification text dataset, changes the labels proportionally, and records the changes in accuracy to analyze the range within which incorrect labels are acceptable.

As shown in Figure 3, the highest accuracy can reach 0.888 when training and testing the original dataset without changing the labels. Overall, the accuracy is relatively flat. Except for the first epoch, it fluctuates between 0.86 and 0.89. It is important to test the original dataset. By using this value, it can be clear to see how much the accuracy drops after changing the labels.

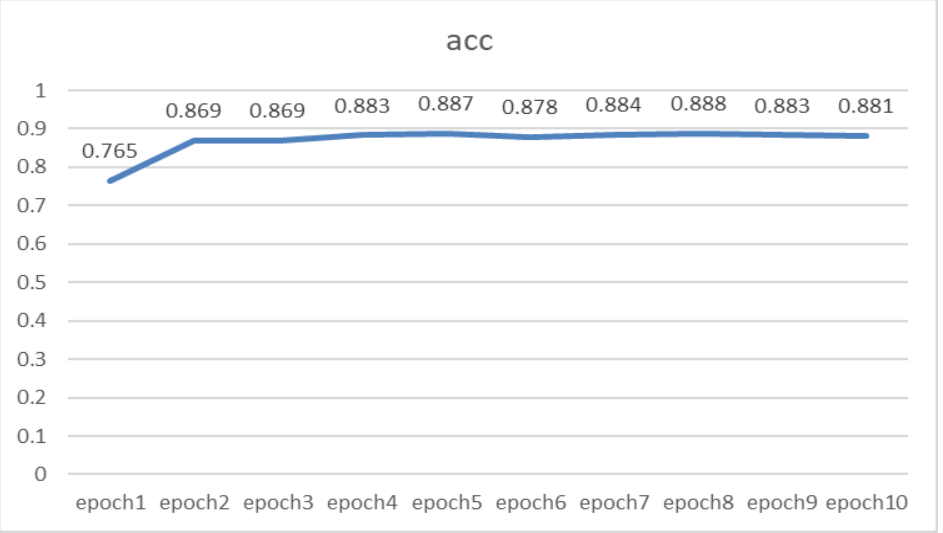


Figure 3. Accuracy values for 10 epochs without changing labels.

After testing the original accuracy value, the most important part is to change the label of the dataset. During this process, select the unit of 5% and gradually change it to 50%. Through the accuracy of these 10 tests, it is easy to observe its changes and use it as a basis to judge the tolerance of neural networks to the wrong labels in the sentiment classification task.

As shown in Figure 4, the accuracy values within 30% tend to be flat. Although they are in a continuous downward trend, the magnitude of the change is not large and can be stable above 0.8. However, after changing 35% of the labels, the accuracy of the test has changed dramatically. When 30% of labels are changed, the accuracy can still reach 0.831, but when 35% of the labels are changed, it drops directly to 0.671. What follows is a continued decline. The accuracy dropped to 0.546 when 40% of the labels were changed, to 0.337 when 45% were changed, and stabilized at 0.337 when 50% were changed. It can be seen from the above data that the data after 30% shows a rapid downward trend. From this data, it can be reasonably concluded that during the task of sentiment classification, the tolerance for wrong labels accounts for up to 30% of all labels.

In this histogram, a key node is the accurate value at 30%. The accuracy before this value can be smoothly above 0.8, and the accuracy after this value continues to drop to 0.337. Therefore, it is crucial

DOI: 10.54254/2755-2721/45/20241266

to confirm the stability of the accuracy when changing 30% of the labels. The paper uses an iterative method to determine the changes in the accurate values when 30% of the labels are changed.

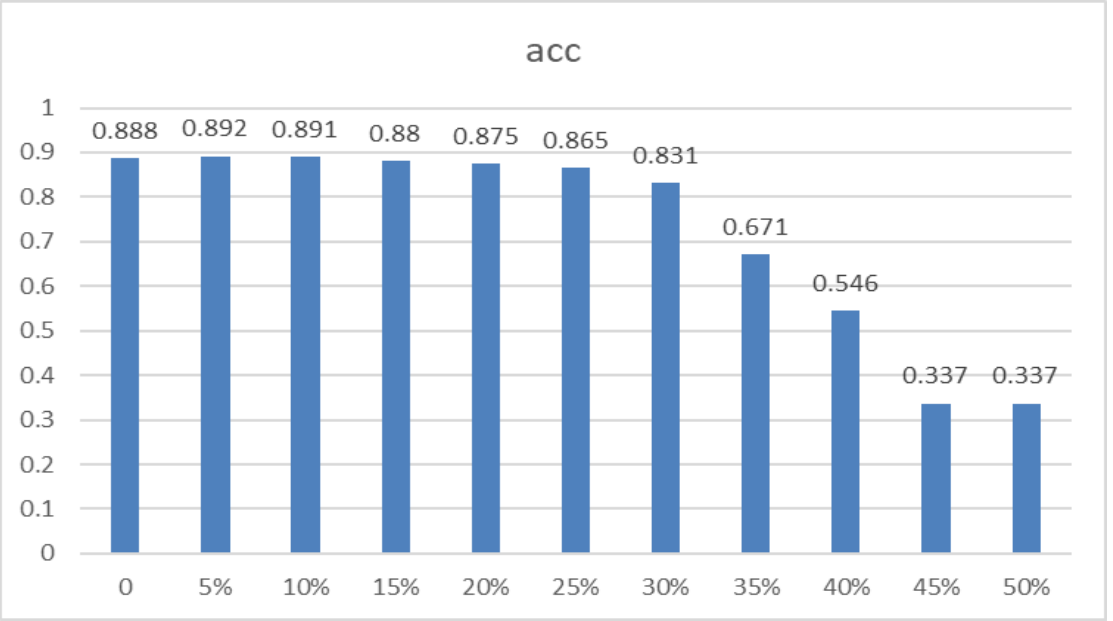


Figure 4. Accuracy values after changing labels proportionally.

Figure 5 is the test result after ten iterations on the dataset with a 30% proportion label change, and its changing trend is shown in the form of a line chart. From Figure 5, it is evident that the test results exhibit a relatively stable and flat change trend over 10 iterations, consistently fluctuating between 0.81 and 0.86. This shows that the test results have good stability and do not fluctuate significantly after changing 30% of the labels. Based on the test results of the above experiments, it can be concluded that the neural network can accept up to 30% of wrong labels in handling sentient analysis tasks, which means the wrong labels within this range do not have a major impact on the test results.

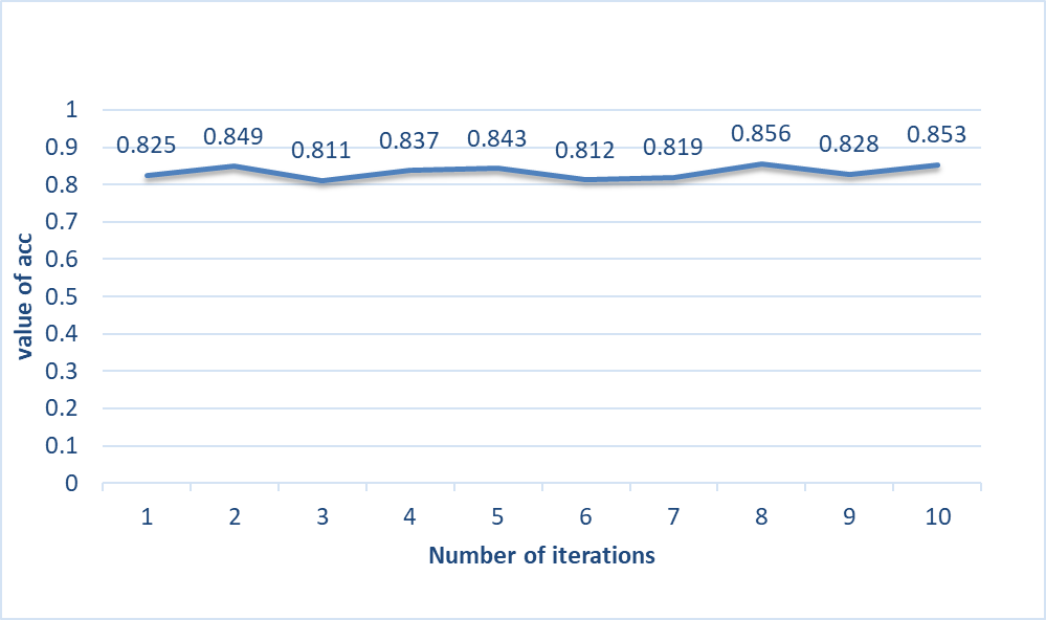


Figure 5. The changing trend of acc when changing 30% of labels under iteration.

DOI: 10.54254/2755-2721/45/20241266

4. Conclusion

The objective of this study is to explore and analyze the tolerance of the Transformer model to wrong labels during sentiment analysis tasks. The study can offer valuable assistance in model training. This study can provide effective help for model training, and at the same time, people can reasonably control the number of wrong labels in future studies. This paper selects a binary classification text dataset to conduct research and analysis. Firstly, do some simple processing on the data to reduce the interference of the learning process. Secondly, train and test the dataset through the built Transformer model, and record the original results. After that, by changing the labels proportionally, test and record the test results under different proportions. Finally, by comparison, the test results after changing the different ratio labels are compared with the original test results, and find the maximum scale that the Transformer model can tolerate. The proposed method is evaluated through comprehensive experiments. According to the experimental findings, the Transformer model can tolerate up to 30% wrong labels when handling sentiment analysis tasks. The result means the wrong labels within this range do not have a major impact on the test results. In the future, further testing of wrong labels will be considered as the research objective for the next stage. The focus of the research lies in examining the tolerance of different models and datasets towards wrong labels. With more comprehensive test results, it will hopefully contribute to a more accurate tolerance result.

References

[1] Mehta P Pandya S 2020 A review on sentiment analysis methodologies, practices and applications International Journal of Scientific and Technology Research 9(2): pp 601-609

[2] Cai Y Yang K Huang D et al. 2019 A hybrid model for opinion mining based on domain sentiment dictionary International Journal of Machine Learning and Cybernetics 10: pp 2131-2142

[3] Xu G Yu Z Yao H et al. 2019 Chinese text sentiment analysis based on extended sentiment dictionary IEEE Access 7: pp 43749-43762

[4] Baid P Gupta A Chaplot N 2017 Sentiment analysis of movie reviews using machine learning techniques International Journal of Computer Applications 179(7): pp 45-49

[5] Ahmad M Aftab S Bashir M S et al. 2018 SVM optimization for sentiment analysis International Journal of Advanced Computer Science and Applications 9(4)

[6] Ruz G A Henríquez P A Mascareño A 2020 Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers Future Generation Computer Systems 106: pp 92-104

[7] Yadav R K Jiao L Goodwin M et al. 2021 Positionless aspect-based sentiment analysis using attention mechanism Knowledge-Based Systems 226: p 107136

[8] Devlin J Chang M W Lee K et al. 2018 Bert: Pre-training of deep bidirectional transformers for language understanding arXiv preprint arXiv:1810.04805, 2018

[9] Munikar M Shakya S Shrestha A 2019 Fine-grained sentiment classification using BERT 2019 Artificial Intelligence for Transforming Business and Society (AITB). 1: pp 1-5

[10] Dataset https://aistudio.baidu.com/datasetdetail/132223