Application of Natural Language Processing Technology in Text Classification

First author: Tian Li
SETTING: Geographic Information Science, Shandong University of Science and Technology, Qingdao, Shandong, China
Corresponding author:1257454198@qq.com

Abstract:

With the development of science and technology and the arrival of the era of big data, artificial intelligence is constantly developing, and related theories are maturing. One of the important branches of artificial intelligence, that is, natural language processing technology, is also constantly developing. In the Internet era, there are more and more text information. Therefore, we should seek a good way to effectively obtain and manage these information. Through experiments, this paper tests the feasibility and efficiency of using natural language processing technology for text classification, and draws a conclusion that natural language processing technology is efficiently applied to text classification.

Keywords: natural language processing; Text classification; Convolutional Neural Network

1. Introduction

Text classification refers to the classification of texts into several categories according to a certain classification basis. With the advent of the information age, a large amount of information is constantly emerging, followed by a large number of texts. Therefore, classifying texts to better process texts and obtain effective information has become one of the current problems. From another perspective, text classification also belongs to one of the application directions of natural language processing.

2. Background

Text refers to the expression of written language, which is a sentence or a combination of multiple sentences with complete and systematic meaning. A text can be a sentence, a paragraph or a chapter. For the application of computers, the current role of computers for mathematical computing accounted for only 10%, for process control classes less than 5%, the remaining 85% are used in the current language information processing, the main thing is the computer for our human natural language sound, shape, meaning processing, and then simply put, is the general text input, small to a word a word, as large as a sentence a paragraph, just like a publishing house, the text of the coherent publication[1] .

With the development of information network, more and more information is obtained, and how to deal with a large amount of information has become a huge problem. Simply relying on manpower to deal with this information is not only time-consuming and labor-intensive, but also may lead to mistakes and low efficiency, so tools are needed to better deal with these problems. Text classification uses computer to automatically classify and mark text sets (or other entities or objects) according to a certain classification system or standard. According to a set of labeled training documents, it finds the relationship model between document features and document categories, and then uses the learned relationship model to judge the categories of new documents. Text classification has gradually changed from knowledge-based methods to methods based on statistics and machine learning[2]. Automatic text classification can effectively help people to search, query, filter and use information. Text classification is an important foundation of data mining.

Text classification first originated in foreign countries, especially in 1957. Especially in 1957, H.P.Luhn of IBM in the United States first made a sudden achievement in the field of automatic classification, researched and put forward the idea of word frequency statistics for automatic classification. Besides, In 1960, M.E.Maron published "On Relevance Probabilistic Indexing and Information Retrieval" in Journal of ACM, which was a well-known article of automatic classification. At that time, it was also a sensation, because a new search member-keyword was put forward in this article, and thus a new automatic classification technology came into being, which made automatic classification technology enter a new era . From 1960s to 1980s, text classification still needed the participation of a large number of staff, but there were still some shortcomings[3]. In 1990s, text classification was improved, and automatic text classification appeared with

Third International Symposium on Computer Engineering and Intelligent Communications (ISCEIC 2022), edited by Xianye Ben, Proc. of SPIE Vol. 12462, 124620D · © The Authors. Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2661012

strong adaptability, such as Construe system developed by Carnegie for Reuters[4].

In the research of text classification in China, due to the difference in language, we can't fully refer to foreign research results, so we need to have text classification suitable for Chinese. Professor Hou Hanqing made an important contribution to this. Later, in 1987, the experimental classification system of Chinese scientific and technological literature (computer) developed by Zhu Lanjuan and Wang Yongcheng was born. By 1995, China had developed a variety of practicality. High-level system, the field involves oncology professional literature, Chinese corpus automatic classification system and archives automatic classification system[4].

In 2000, more and more outstanding researchers made great improvements to the system, improved the classification accuracy, collected synonyms on the same concept word, and reduced the calculation of the overall classification. These outstanding researchers made great progress in text classification in China.

Now consider text classification based on natural language processing to improve the efficiency of text classification.

3. Overview of Natural Language Processing Technology

With the development of science and technology, the application of artificial intelligence is more and more extensive, and it can be applied in many directions, such as images, texts, etc. And natural language processing technology is the product of the combination of artificial intelligence and text technology. Language refers to everyday languages such as Chinese and English. It is a tool for communication in learning and life, not a programming language, so computers can't understand it. However, natural language processing refers to the use of computers to process natural languages, so as to realize the information interaction between man and machine, including intelligent word segmentation, part-of-speech word segmentation, named entity recognition, text classification, and reactionary advertisement auditing.

The basic tasks of natural processing technology can be divided into three categories, including part-of-speech analysis, syntactic analysis and text analysis. Part-of-speech analysis is the basic work of natural language processing technology, including word segmentation, part-of-speech tagging and named entity recognition. Syntactic analysis includes syntactic dependency analysis, semantic dependency analysis and text error correction. Text analysis includes keyword extraction, sentiment analysis and text classification. Among them, text classification is an important direction.

The general process of natural language processing technology is roughly divided into corpus acquisition, corpus preprocessing, feature engineering, feature selection, model training, evaluation index, and online application of models.

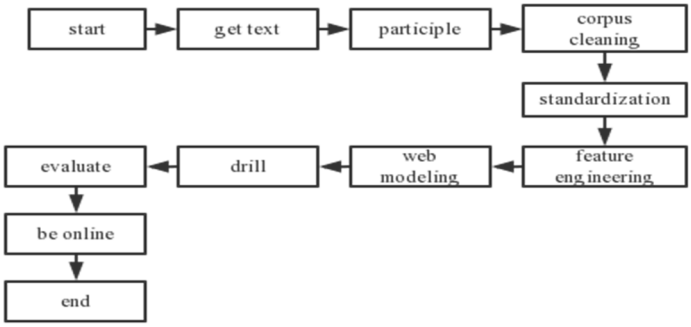

Figure 1 General process of natural language processing

4. Overview of text classification

4.1 Concepts

The so-called text classification means that the computer automatically classifies and marks the text sets according to a certain classification system. According to a classified training sample, he finds out the relationship between the text features and the new text sets, establishes a model, and classifies the new text sets.

With the advent of the information age, information is exploding. It is inefficient, time-consuming and labor-intensive to analyze and process information by hand, and the subjectivity of people is strong, which may have a certain influence on it. So, we consider using computers to deal with these problems, but computers do not understand the common texts between human beings, so we need natural language processing technology to build a bridge between human beings and computers.

Text classification is widely used, such as Sentiment Analyse, Topic Labeling, Question Answering, intention classification, Natural Language Inference, etc. Therefore, text classification is very important and has research significance.

4.2 Text classification infrastructure

Text classification includes two basic structures: feature representation and classification model. The purpose of feature representation is to transform text into a form that can be understood and processed by computers, so that computers can work instead of people and efficiently realize information processing. Common text feature representation methods of:

(1)BOW bag of words model, which represents the text as a feature vector. The basic idea is to ignore the word order and grammar of the text, and only regard the text as a collection of words, and classify them according to the words, that is, the text is regarded as a bag, and the words are regarded as a series of items in the bag, and classified according to the items. First of all, to get the features of words, it is necessary to count the frequency of times, and put the features together with the corresponding word frequencies, so that vectorization can be completed. Besides, standardization and some feature engineering are needed, so that classification can be carried out.

(2)N-gram is an algorithm based on statistical language model. N is a sequence of byte fragments with length N formed by sliding the window of text according to N bytes. gram is a byte fragment. Count the frequency of the byte fragments, and filter them to form a gram list, which is the feature vector space, and each Gram is a latitude. This model can be used to evaluate whether the model is reasonable or not.

(3)TF-IDF, word frequency-inverse file frequency. Simply put, if a word has a high frequency in this document and a low frequency in the document library, it has a good classification function and is suitable for classification. This model can be used in search engine, keyword extraction, text similarity and text summarization.

(4)Word2vec, which depends on the local context to obtain word vectors.

(5)Glove uses not only local context but also global statistical features.

The classification model includes shallow learning model and deep learning model. Shallow learning models include PGM model (probability diagram), KNN model (K nearest neighbor), SVM model (supporting cross product), DT model (decision tree) and RF model (random tree); Deep learning models include ReNN-Based model (recurrent neural network), MLP-based model (multilayer perceptron), RNN-based model (recurrent neural network), CNN-based model (convolutional neural network), Attention-based model (attention mechanism), Transformer-based model (pre-training model) and GNN-based model.

4.3 Data set of text classification

According to the application of text classification, text classification data sets are divided into:

(1) sentiment analysis data set: Yelp, including data of two sentiment classification tasks, namely, detecting fine-grained and predicting positive and negative emotions; IMDb, developed for the two-category emotional classification task of film reviews, including positive and negative reviews; Movie Review movie review is a collection of movie reviews; SST Stanford Emotion Tree Database includes two categories and five categories. MPQA Multi-perspective Q&A is an opinion corpus with two category labels. Amazon is a review collection of all the popular products on Amazon.com.①Yelp②IMDb③Movie Review④SST⑤MPQA⑥Amazon

(2) News classification NC. 20NG, 20 classes, with the same number of each class; AG: Academic news search engine, with 4 classifications; Sogou News, the news corpus of sogou, includes two corpora, Sogou CA and SogouCS.

(3) The theme is labeled TL, which aims to find a clear theme for each document, including a large-scale multilingual knowledge base generated by the most commonly used information frame of DBpedia Wikipedia, with many kinds; Ohsumed Biomedical Literature Database, 7400 articles

(4) Q&A task AQ, including extraction and generation.

(5) NLI, natural language reasoning, is used to predict whether the meaning of another text can be inferred from one text, including SNLI (Stanford Natural Language Reasoning), MNLI (Multi-genre Natural Language Reasoning), etc.

5. Text classification based on natural language processing

Through investigation, it is found that pure manual text classification can't completely meet the needs of information processing. Therefore, consider text classification based on natural language processing technology, use natural language processing technology to obtain more accurate features, select and extract features, comprehensively consider the complexity and extraction effect of the algorithm, and select the appropriate feature extraction algorithm to achieve text classification efficiently. In this paper, the convolutional neural network CNN structure in natural language processing technology is used to classify texts[5]. The paper is implemented based on tensoflow and python.

Convolutional neural network CNN is a deep learning model, which can also be understood as multilayer perceptron of artificial neural network. Its neurons in each layer are arranged in three dimensions, and three dimensions refer to width, height and depth, where depth refers to the number of layers of the network, while vectors are arranged in the depth direction. Convolution structure has many advantages, which can reduce the amount of memory occupied by deep network. Besides, it can also reduce the number of network parameters, and can be used to over-fit some models.

5.1 Data preprocessing

1) load data set

This paper selects Movies Review Data from Rotten Tomatoes, a data set of film reviews from Rotten Tomatoes, which includes more than 10,000 reviews, with positive reviews and negative reviews accounting for half each other.

2) Sentence cleaning

Clear sentences, clear all kinds of symbols, etc.

```
def clean_str(string):
    """..."""
    string = re.sub(r"[^A-Za-z0-9(),!?\'\`]", " ", string)
    string = re.sub(r"\'s", " \'s", string)
    string = re.sub(r"\'ve", " \'ve", string)
    string = re.sub(r"n\'t", " n\'t", string)
    string = re.sub(r"\'re", " \'re", string)
    string = re.sub(r"\'d", " \'d", string)
    string = re.sub(r"\'ll", " \'ll", string)
    string = re.sub(r",", " , ", string)
    string = re.sub(r"!", " ! ", string)
    string = re.sub(r"\(", " \( ", string)
    string = re.sub(r"\)", " \) ", string)
    string = re.sub(r"\?", " \? ", string)
    string = re.sub(r"\s{2,}", " ", string)
    return string.strip().lower()
```

Figure 2 Sentence cleaning

5.2 Model implementation

The basic idea is as follows: the first layer maps words to a set of vector representations; The second layer is convolution layer, which uses multiple filters to traverse 3-5 words at a time, and the next layer is a series of long feature vectors. Then, the probability of each class is obtained by using softmax.

(1) Input placeholder. tf.placeholder, as a placeholder for tensorflow, is used to assign values to input variables in the training and testing model stages like feed_dict. Not all dropout are suitable for training here, but only 50% are often used.

```
self.input_x = tf.placeholder(tf.int32, [None, sequence_length], name='input_x')
self.input_y = tf.placeholder(tf.float32, [None, num_classes], name='input_y')
self.dropout_keep_prob = tf.placeholder(tf.float32, name='dropout_keep_prob')
```

Figure 3 Input placeholder layer

(2) Embedding layer, which is used to convert all words into a set of variable representations. In this process, a parameter matrix needs to be defined, and the vector representation is searched through tf.nn.eding_lookup, and it should correspond to input_x, and the returned result is a three-dimensional vector. As the convolution layer of the latter layer has special requirements, it is required to input a four-dimensional vector, so to proceed to the next step, it is necessary to expand the result of this layer to four dimensions.

```
with tf.name_scope('embedding'):
    self.W = tf.Variable(tf.random_uniform([vocab_size, embedding_size], -1.0, 1.0), name='weight')
    self.embedded_chars = tf.nn.embedding_lookup(self.W, self.input_x)
    # TensorFlow's convolutional conv2d operation expects a 4-dimensional tensor
    # with dimensions corresponding to batch, width, height and channel.
    self.embedded_chars_expanded = tf.expand_dims(self.embedded_chars, -1)
```

Fig. 4 Embedded layer

(3) Convolution and maximum pool layer. The most important thing in convolution layer is filter. In convolution neural network CNN, there are three kinds of Fliter, each of which includes two. The input matrix will be processed by iterating each filter, and the final results will be combined into a whole feature vector.

```
pooled_outputs = []
for i, filter_size in enumerate(filter_sizes):
    with tf.name_scope('conv-maxpool-%s' % filter_size):
        # conv layer
        filter_shape = [filter_size, embedding_size, 1, num_filters]
        W = tf.Variable(tf.truncated_normal(filter_shape, stddev=0.1), name='W')
        b = tf.Variable(tf.constant(0.1, shape=[num_filters]), name='b')
        conv = tf.nn.conv2d(self.embedded_chars_expanded, W, strides=[1,1,1,1],
                            padding='VALID', name='conv')
        # activation
        h = tf.nn.relu(tf.nn.bias_add(conv, b), name='relu')
        # max pooling
        pooled = tf.nn.max_pool(h, ksize=[1, sequence_length-filter_size + 1, 1, 1],
                                strides=[1,1,1,1], padding='VALID', name='pool')
        pooled_outputs.append(pooled)
```

Figure 5 Convolution and maximum pooling

(4) Drop-out layer, which is to avoid the over-fitting of the model. It is necessary to set a threshold in advance, so that some neurons fail, so the failure is different every time.

```
with tf.name_scope('dropout'):
    self.h_drop = tf.nn.dropout(self.h_pool_flat, self.dropout_keep_prob)
```

Figure 6 Drop-out layer operation

(5) Score and prediction. Through the above operations, a series of feature vectors are obtained, and a score of each classification is obtained by operation. softmax is used to convert the score into probability, and the one with the highest probability among all results is selected as the final prediction.

(6) loss and accuracy, the task is not only to classify, but also to improve the accuracy of classification as much as possible. Therefore, socre is used to calculate the loss of the model, which makes the loss minimum. Generally, the cross_entorpy function is used to calculate the loss. Besides the minimum loss, an accuracy rate should be redefined, so that the classification result has the highest accuracy and the best effect.

```
    with tf.name_scope('loss'):
        losses = tf.nn.softmax_cross_entropy_with_logits(logits=self.score, labels=self.input_y)
        self.loss = tf.reduce_mean(losses) + l2_reg_lambda * l2_loss

    # accuracy
    with tf.name_scope('accuracy'):
        correct_predictions = tf.equal(self.prediction, tf.argmax(self.input_y, 1))
        self.accuracy = tf.reduce_mean(tf.cast(correct_predictions, 'float'), name='accuracy')
```

Figure 7 Defining Loss and Accuracy

5.3 Model training

(1) Create diagrams and session. Using Graph and Session of tensorflow, simply speaking, Graph is a summary, including all operations and vectors that need to be used, and Session is the environment. All operations in Graph are completed in Session, so these two are especially important.

```
checkpoint_file = tf.train.latest_checkpoint(FLAGS.checkpoint_dir)
graph = tf.Graph()
with graph.as_default():
    session_conf = tf.ConfigProto(
      allow_soft_placement=FLAGS.allow_soft_placement,
      log_device_placement=FLAGS.log_device_placement)
    sess = tf.Session(config=session_conf)
    with sess.as_default():
```

Figure 8 Creating diagrams and session

(2) Initialization, which initializes the model CNN and ensures that it will be recorded every time it runs.

(3) Overview, tensorflow is powerful in that it has many functions, which can avoid programming definition. You can use tensorflow's summary to record the changes of parameters or variables when training the model. Besides, you can also use tf.summary.filewriter () to write it into the hard disk and visualize it to tensorboard.

```
grad_summaries = []
for g, v in grads_and_vars:
    if g is not None:
        grad_hist_summary = tf.summary.histogram('{}/grad/hist'.format(v.name),g)
        sparsity_summary = tf.summary.scalar('{}/grad/sparsity'.format(v.name), tf.nn.zero_fraction(g))
        grad_summaries.append(grad_hist_summary)
        grad_summaries.append(sparsity_summary)
```

Figure 9 Overview

(4) Checkpoint, which is used to save the parameters of training model in each stage, and a group of parameters with the best classification effect is obtained by using accuracy and Loss screening.

(5) Initialize variables. Before training each model, you need to initialize variables. You can also use the already packaged function in tensorflow, tf.global _ variables _ initializer (), which is simple and convenient.

(6) Define a single training step. First, the first training model with only one step, and then use batch to continuously update the model parameters. sess.run () runs, and finally get the loss and accuracy of each step.

```
def train_step(x_batch, y_batch):
    """..."""
    feed_dict = {
        cnn.input_x: x_batch,
        cnn.input_y: y_batch,
        cnn.dropout_keep_prob: FLAGS.dropout_keep_prob
    }
```

Figure 10 defines a single training step.

Cycle, one-step training. After the definition of the model is completed, the next step is to cycle the model and batch it, which saves time and labor and has high efficiency.

```
batches = data_process.batch_iter(list(zip(x_train, y_train)), FLAGS.batch_size, FLAGS.num_epochs)
# training loop
for batch in batches:
    x_batch, y_batch = zip(*batch)
    train_step(x_batch, y_batch)
    current_step = tf.train.global_step(sess, global_step)
    if current_step % FLAGS.evaluate_every == 0:
        print('\n Evaluation:')
        dev_step(x_dev, y_dev, writer=dev_summary_writer)
```

11 cycle.

6. Result analysis:

```
step 2891, loss 0.011868, acc 1
step 2892, loss 0.0115379, acc 1
step 2893, loss 0.0142323, acc 1
step 2894, loss 0.0251607, acc 0.984375
step 2895, loss 0.0169635, acc 1
step 2896, loss 0.0171963, acc 1
step 2897, loss 0.0263007, acc 0.984375
step 2898, loss 0.0862797, acc 0.96875
step 2899, loss 0.00211942, acc 1
step 2900, loss 0.0243832, acc 1
```

Fig. 12 result chart

```
step 2900, loss 1.02165, acc 0.743902
```

Fig. 13 result evaluation chart

7. Conclusion

Through the results, it can be seen that the loss of using natural language processing technology for text classification is small, and the accuracy is good. Convolutional neural network CNN can be used for text classification well, and it is concluded that text classification based on natural language processing is feasible and efficient.

8. Summary

In this paper, the convolutional neural network CNN is used to classify the text. Unlike CV input in computer vision, the input of natural language processing technology is text, which includes articles, sentences, phrases, etc. These texts are represented as vector matrix by word2vec or glove, with one line representing a phrase and the number of lines representing the length of the sentence. Of course, it is necessary to clean up the sentences and remove symbols in advance, and the number of columns is latitude. In computer vision CV, filters slide through the whole image with arbitrary length and width, while in natural language processing, filter will cover everything. Filter convolutes the matrix obtained by input text processing to obtain features, and then extracts the best features. Although natural language processing often uses RNN architecture and then attention, I personally think that convolutional neural network will be simpler and more efficient for text classification. Science and technology are always improving, and I believe there will be more efficient methods for text classification in the future, which is worthy of continuous efforts.

References:

[1] Yu Shitomb. Introduction to Computational Linguistics[M].Beijing: The Commercial Press

[2] Sebastian F. A tutorial on automated text categorization[J]. In: Proceedings of ArgentineanSymposiums Artificial Intelligence(ASAI-99,1st) Buenos Aires,1999，7~35

[3] Cheng Ying,Shi Jiao-Lin. Research on the automatic classification: present situation andprospects[J].prospects[J]. Journal of the China Society for Scientific and Technical Information,1999，1: 20~27

[4] Spark J K,Willett P, et al.Readings of information retrieval.San Mateo,US: Morgan Kaufmann,1997

[5] 《Convolutional neural network for Sentence Classification[1]》