Frontiers in Economics and Management Volume 3 Issue 1, 2022
ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

Classification of Messages in "Smart Government Affairs"

Zuyu Hou

College of Economics, Jinan University, Guangzhou 510632, China

Abstract

In recent years, the online political platform has gradually become an important channel for the government to listen to the voices of the people, solve their worries, gather their wisdom, and warm the people's hearts. The amount of text data on various social conditions, public opinions, policies and people's livelihood has continued to rise. The work of related departments and related hot spots has brought huge challenges. Based on natural language processing technology and text mining methods, this paper constructs a targeted "smart government system" for the classification of public messages in the context of this political trend, and becomes a bridge connecting the government and the people.For the mass message information data set given in a certain format, one-hot encoding is performed on the training set and the verification set under the 5-fold hierarchical cross-validation division, and the machine learning method based on the TF*IDF bag-of-words vector feature engineering and the Convolutional neural network CNN, recurrent neural network RNN and deep learning methods under long and short memory network LSTM are used for message classification. After selecting the best comparison, this paper finally uses the support vector machine classifier based on the part of speech level TF*IDF model under machine learning to classify. After this step, the average value of F1-Score under 5-fold hierarchical cross-validation reaches 0.9013, and the output The corresponding predicted label data set. In the end, the automatic division of the procedures for the masses' messages was realized, which greatly improved the efficiency of dividing the masses' messages in government work.

Keywords

TF*IDF, Machine Learning; Convolutional Neural Network; Recurrent Neural Network; Long Short-Term Memory.

1. Introduction

1.1. "Smart Government System" Background

In recent years, with the further popularization of mobile networks and related electronic products among the masses, as well as the further development of related big data processing technologies, some new government work processes have begun to emerge in response to the feedback of government issues and the collection of people's livelihood issues. It is more convenient for public feedback and the mainstream way of government departments to collect and manage relevant opinions. In the past, the vast majority of people used to go to relevant administrative units to mention their opinions, appeals, and submit relevant opinions face to face. The emergence of political platforms has made it more convenient for the masses to give feedback and make statements, and government agencies and departments can collect and manage relevant information more efficiently.

But the problem that comes with it is that the amount of text data related to various social conditions and public opinions continues to rise. At this time, if you still rely on the traditional way to manually divide messages and other related tasks, from another perspective, government departments work On the contrary, the efficiency of the Internet may be reduced, and it is impossible to maximize the advantages of the new method of network inquiry.

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

Therefore, in order to maximize the advantages of the new way of government work through the Internet, this article builds a "smart government system" based on NLP (natural language processing technology) and text mining technology to solve the problem of automatic classification of public messages.

1.2. Data Objects and Output Results

A system for business processing must have the data objects it needs to rely on. Specifically, in the "smart government system" constructed in this article, the data objects to be solved are the 8th Teddy Cup-Data Mining Challenge C. The attached unstructured data of Annex 1 and Annex 2. Compared with structured data, it is more difficult for computers to understand.

Completeness: According to statistics, the three-level classification label system attached to Annex I covers 15 first-level classifications, 103 second-level classifications, and 517 third-level classifications. We assume that the combination of these three can combine any reality the government-affair message issues are all classified into a certain intersection category. For example, message A can be divided into the intersection of three levels of urban and rural construction→safety production→accident handling; message B can be divided into three categories: land resources→marine meteorological earthquake→meteorology Among the intersection of class classification, etc.;

Practical feasibility: In the data set of the mass message question collected at a certain time, not all the intersections of the 15 first-level classifications, 103 second-level classifications, and 517 third-level classifications can be involved. Just like the message data set collected in Annex 2 shown in the figure below, after a simple screening, it can be found that there are only 7 first-level classification labels involved, and not all 15 first-level classifications are involved.

The purpose of emphasizing the completeness and practicality of the label system is to give certain warning signs to the mass message classification module in the constructed "smart government system". That is to say, the automatic division effect of the program this time may only be divided for the message problem of the 7 first-level label classifications involved this time. This requires us to regularly update and maintain the message classification module of the system.

The mass message classification issues in Annexes 1 and 2 require that the "smart government system" can automatically classify the mass message data set in Annex 2 according to the three-level label classification system in Annex 1, and evaluate the effect of the classification. For good or bad, the numerical indicator F1-Score is used. The higher the F1 value, the better the effect; the construction idea of the mass message classification module is the selection idea of selection and comparison of the best.

The machine learning model process of the crowd message classification module:

Preprocessing of text data→divide, stop, and segment words according to hierarchical cross-validation→feature processing→select the classification algorithm (classifier) corresponding to the model→get F1-Score and output the corresponding prediction label set.

In the machine learning model process of the crowd message classification module, in the main process, except for the word segmentation level of TF*IDF in the feature processing and the classifier can be changed, the other steps are basically in a fixed state, that is, the final The value of F1-Score depends on the TF*IDF word segmentation level and the different combinations of classifiers.

1.3. The Programming Language and Related Packages Used in the System Construction Process

The program code involved in this system is carried out under the operating environment of Python3.7 in windows10, and the program packages used are: gensim, sklearn and keras under tensorflow. In addition, due to the large amount of calculation in the program operation of

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

crowd message classification and hot issue mining, the corresponding part is run on the Colab platform.

2. Methodology

2.1. Related Assumptions

In the process of constructing a "smart government system", it is necessary to make a clear definition of the research object of the specific data set. The definition of the research object here refers to the definition of the term "message", whether it is the mass message classification module or the hot issue mining module, or even the final reply comment evaluation module, we can not avoid the reference to the "message" Distinguish between objects.

Under this circumstance, we define the research object of "message", which is: message details.

The reason for making this hypothesis is to consider the amount of information contained, because the subject of the message is only a general description of its details, and even the subject of many messages is missing. We are not neglecting the topic of the message. We can use the topic of the message to predict the matching relationship between the classification label and the message, follow-up experience preview comparison of the module output results, and extract the relevant message set keywords.

2.2. Introduction to Theoretical Tools

1) Word segmentation, word segmentation method:

The following introduces the basic concepts and methodology of text mining technology and natural language processing.

The so-called word segmentation is simply to divide a paragraph (a piece of text) into a combination of words according to a certain existing thesaurus according to a certain calculation principle. This combination form is generally expressed in a specific format. The more common one is that words are separated by spaces, and the general word segmentation process will first remove punctuation, spaces, line breaks and other text cleaning procedures.

The divided words are divided according to a dictionary that is certain to exist. The reference dictionary and the calculation principle of word segmentation based on different word segmentation programs will be different.

In the construction of the "smart government system", the word segmentation methods involved in this article are: jieba, thulac, and snownlp word segmentation, all of which can be implemented by calling related packages in Python.

2) Stop word list:

Stop words refer to certain words or words that are automatically filtered before or after processing natural language data (or text) in order to save storage space and improve search efficiency in information retrieval. These words are extremely common, and recording the number of these words in each document requires a lot of disk space, and due to their universality and function, these words rarely express information about the degree of document relevance alone. If you consider each word instead of a phrase in the retrieval process, these functional words are basically not helpful. The stop vocabulary is a collection of words that we want to filter out in the program. Generally, we will have it in the natural language processing process. Some established stop word lists in Chinese and English are available for us to choose from. Of course, going to stop means deleting the words that need to be filtered out in the program running memory. Generally, this step is performed after text cleaning and word segmentation.

3) TF*IDF weight:

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

TF*IDF is a commonly used weighting technique for information retrieval and data mining. Among them, TF refers to word frequency, and IDF refers to inverse text frequency index. In a given text, TF refers to the frequency of a given word in the text. This number is a normalization of the number of words to prevent it from being biased towards long documents.

For a word in a given text, its TF formula is:

$$TF_{ij} = \frac{t_{ij}}{\sum_{j} t_{ij}} \tag{1}$$

Which t_{ij} represents the number of occurrences of word j in the i-th text, i=1,2,... , N. n is the number of texts contained in the data set. As the number of occurrences of the j word in the i text increases, the TF value increases under the condition that the total number of words in the i text remains unchanged. IDF is a measure of the universal importance of words. The IDF of a specific word can be divided by the total number of text n by [the number of texts containing the word plus one], and then the obtained quotient is taken as the logarithm to the base 10 to obtain the following formula, where m Is the number of texts containing j.

$$IDF_j = \log_{10}\left(\frac{\mathrm{n}}{m+1}\right) \tag{2}$$

It can be seen from the defined formula that as the number of texts containing j words in the corpus formed by all texts increases, the IDF value decreases instead. At this time, we define the TF*IDF weight of a word as:

$$[TF * IDF]_{ij} = TF_{ij} \times IDF_j \tag{3}$$

That is to say, the main idea of TF*IDF weight is: the importance of a word increases in proportion to the number of times it appears in the text, but at the same time it will appear inversely proportional to the frequency of its appearance in the corpus formed by all texts. decline. If a word or phrase appears in an article with a high frequency of TF and rarely appears in other articles (the IDF is also high), it is considered that the word or phrase has good classification ability and is suitable for classification.

4) Text vector space model:

Most of the data for text problems are string type data. If the relevant text paragraph can be equivalently transformed into a space where the content can be represented by a value, it will be convenient for subsequent computer program operations and measurement calculations. Various types of indicators. The text vector space is such a vector space in which text and numeric vectors correspond one-to-one.

①Construction of the dictionary.

Before building a dictionary, generally speaking, we must first complete deduplication (remove the redundant text that may appear repeatedly, and only keep one as the subsequent modeling object), remove line breaks, spaces, punctuation, and garbled text cleaning the processing program is followed by word segmentation processing on the text.

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

The so-called dictionary is a collection of non-repeated words formed based on the text after text cleaning, word segmentation and stop processing. Each word in the collection has its own independent number, and its format is: {"a":0,"b":1,"c":2,.....};

Note: In the above expression, the letter represents a word, and the Arabic numerals represent the serial number corresponding to the word.

②Formation of text word vector (sparse vector).

Using the dictionary formed by the above steps, the text object words are vectorized. The so-called word vectorization is to project the text one by one into a numeric vector according to the dictionary. The dimension of the vector is the same as the number of words contained in the dictionary. Here, for the convenience of expression, let it be N. The element corresponding to each position of the numeric vector corresponds to the word with the same sequence number in the dictionary, and the value of the position is the number of times the corresponding word appears in the text.

That is, the text word vector formed after the text object word vectorization is:

$\vec{V} = (v_1, v_2, ..., v_N)$, v_j Represents the number of occurrences of the word j in the dictionary in the text object,j=1,2,...,N. In this way, all text objects can be transformed into their corresponding text word vector forms one by one.

③Construction of text vector space (sparse matrix).

In the problem of text classification, a data set to be processed often includes multiple texts, and we set it to n. According to the above steps, a dictionary based on these n texts can also be formed. According to the content of this dictionary, we can uniformly vectorize these n texts into an N-dimensional sparse vector:

$$\vec{V_i} = (v_{i1}, v_{i2}, v_{i3}, ..., v_{iN}), i = 1,2....., n.$$

It represents the sparse vector corresponding to the i-th text. Then, the data set formed by these n texts can be represented by the set of n corresponding sparse vectors. Calling it the text vector space V.

$$V = V_{N\times n} = [\overrightarrow{V_1}, \overrightarrow{V_2}, ..., \overrightarrow{V_n}] = \begin{pmatrix} v_{11} & v_{21} & & v_{n1} \\ v_{12} & v_{22} & & v_{n2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ v_{1N} & v_{2N} & & v_{nN} \end{pmatrix}, i = 1,2...n。 \quad (4)$$

$\vec{V_i}$ appears in the form of column vector expression, i=1,2,..., N. This is what we usually call a sparse matrix.

3. Results and Discussion

3.1. Crowd Message Classification Background

In order to facilitate face-to-face, point-to-point, targeted handling of the message questions of relevant people, in the actual government work process, generally, the staff will first follow a certain labeling system to give messages from the people collected on various online political platforms Classification is carried out to facilitate the subsequent distribution of different types of messages to the staff of different government departments, so that professional people can do professional things. However, relying on manual experience processing to allocate and divide messages will have problems such as large workload, low efficiency and high error rate.

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

Therefore, if a "smart government system" with automatic classification and division of message programs can be established, it can greatly improve the efficiency and accuracy of related work.

Specific to the text data set given in this question, the problem to be solved by the "smart government system" constructed in this article is to establish the first-level label classification model of the message content, and perform the first level of the mass message data set collected in Annex 2. The classification of the first-level labels is to "stick" each collected message of the masses with its unique first-level label.

3.2. Text Classification Result Evaluation Index F1-Score

In the text classification problem of natural language processing NLP, the classification model needs a specific quantitative index to evaluate the accuracy of its classification, and such an index usually refers to what we call F1-Score.

In the evaluation of the classification effect of the "smart government system", we further define F-Score as F1-Score, and its expression is:

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i + R_i} \tag{5}$$

$$P_i = \frac{True\ \ positive}{True\ \ positive + False\ positive} \tag{6}$$

$$R_i = \frac{True\ \ positive}{True\ \ positive + False\ negative} \tag{7}$$

Among them, P_i is the precision rate of the message of the type i, and R_i is the recall rate of the message of the i type.

The intuitive understanding of precision and recall is as follows:

Table 1. Label explanation table related to precision and recall

forecast result		1	0
1 indicates that the forecast falls into this category	1	True positive	False positive
0 means that the forecast does not fall into this category	0	False negative	True negative

The F value is the harmonic average of the precision rate and the recall rate. Its advantage is that the model fitting rate of each type of message is taken into account, and the concept of a comprehensive average is adopted, which makes the evaluation result more general and convincing. The larger the value, the better the fitting prediction effect of the model. After obtaining the evaluation indicators, we will further elaborate on the established first-level label classification model, that is, the specific solution steps for the classification of public messages.

3.3. Detailed Steps to Classify Messages from the Masses

In dealing with practical problems, if only one model is established, it is often not objective enough. Therefore, in response to the problem of mass message classification, we adopt the idea of comparing and choosing the best: first establish multiple models that may be feasible to solve this problem, and each model can use different classifier algorithms, and then calculate different The F1-Score of the model finally takes the model with the largest F1-Score value and

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

the classification algorithm used in it as the model and algorithm we chose to deal with the government message problem this time.

In text classification problems, commonly used classification methods include machine learning methods and deep learning methods. In machine learning, supervised learning is usually used to train a generative model on historical data to predict the category of text. Commonly used machine learning method steps include one-hot encoding of text using a bag of words model, and contextual word relationship representation using a collinear matrix Method, using TF*IDF feature weights and feature extraction methods, using naive Bayes classifiers, support vector machines, linear classifiers, random forests and other classifier methods; in deep learning, in order to deal with high-dimensional and highly sparse data Commonly used deep learning methods include word vector embedding method using Word2vec, deep learning method using TEXTCNN convolutional neural network, deep learning method using TEXTRNN recurrent neural network, deep learning method using LSTM long and short memory model, using FastText's fast text classification method and the NLP method using the Attension mechanism.

In this question, the text classification type is a multi-classification problem. This article uses a hierarchical cross-validation machine learning method for classification based on the bag of words model. The main python package used is sklearn; on the basis of the deep learning methods of TextCNN, TextRNN and LSTM, cross-validation is used for classification, and the main python used is the package is gensim and the keras package under tensorflow (TensorFlow's high-level API).

1) A first-level label classification model based on machine learning:

Step1: preprocessing of message data.

Read the data in Annex 2 into the "Smart Government Affairs System", filter the message details and the first-level label column, de-duplicate the details of each message, and use regular expressions to remove line breaks and spaces. Re-merge the details of each message and the corresponding label to make it a data set to be further processed.

Step2: Data division according to hierarchical cross-validation.

When using machine learning for message label classification, the data set needs to be divided into two modules: training set and verification set. The training set is used to train the model, and the validation set is used to evaluate the verification model. The model learns from the training set through a suitable method, and then calls the score method to evaluate on the verification set, and then score, you can know the current training level and F1-score of the model.

If there is only one division, the verification result is contingent: in a certain division, the data in the training set is easy to learn, and the data in the verification set is complicated, which will lead to deviations in the final result. Therefore, consider cross-validation and hierarchical cross-validation methods to divide the data set multiple times.

The principle of cross-validation:

Divide the combined message and label data into n pieces, where n-1 pieces are the training set, and the remaining piece is the validation set, and scored; each time the data filtering conditions meet the used validation set data no longer appears In the subsequent validation set, the data set after the validation set is removed is used as the training set. And so on, until all n data sets become validation sets. Where n is the number of partitions of the data set, when n=5, it is a 5-fold cross-validation. In this article, you can consider using the successfully divided data sets to perform n times of score verification and take the average value to get the final F1 value. Therefore, the n-fold cross-validation greatly improves the use efficiency of data compared with one-time partition verification, and reduces the probability of occurrence of accidental data.

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

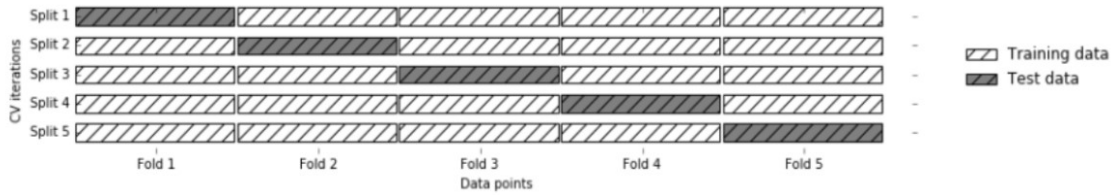

Figure 1. Visual illustration of 5-fold cross-validation

The principle of hierarchical cross-validation:

As can be seen from the above figure, this cross-validation method requires the data to be divided equally every time it is divided. According to the data in this question, there are 7 categories in the data set. Assuming that the extracted data is exactly 7 categories divided by categories, that is, fold1 is all 0 categories, fold2 is all 1 categories, and so on. Such a result will lead to model training without learning the characteristics of the data in the validation set, which makes the model score very low, even 0! Therefore, in order to avoid this situation, the "smart government system" in this article adopts a deepening method of n-fold cross validation: Stratified n-fold cross validation: it maintains the original in every compromise The proportional relationship of each category in the data. Assuming that the proportion of 7 categories in the message data is 1:2:2:3:2:2:1, using 5-fold stratified cross-validation, the divided 5-fold is divided, and each fold is divided The data categories maintain a ratio of 1:2:2:3:2:2:1, which makes the verification results more credible.

So far, the data division of the training set and the test set according to the hierarchical cross-validation has been introduced, and the fold number of the specific division should be determined by itself, such as n=5. The final fold number determination will be combined with the selection effect of the following classifiers, and then the fold number that maximizes the F1 value is selected as n (detailed training and validation set data can be seen in the attachment).

Step3: Word segmentation, remove stop words.

After dividing the training set and the test set, the two are segmented and stop words are removed.

Common word segmentation methods used to compare the effect of word segmentation in NLP are: jieba, thulac, and snownlp. In this article, the jieba, thulac, and snownlp word segmentation effects are similar, but the thulac Chinese word segmentation package and the snownlp word segmentation package have low operating speed and efficiency, and occupy a larger operating space. Therefore, for text classification, the jieba word segmentation package is mainly used, and the jieba word segmentation package is used. The precise mode, the cut_all parameter is set to False.

Step4: Feature engineering processing.

Initialize the training set vector space model.

Calculate the TF*IDF weight of each word in the details of each message in the training set;

At this time, the construction of the text vector space model and the weighting process of the corresponding message vector TF*IDF are all related to the level of word segmentation, which are:

(a) Single word level TF*IDF: The sparse matrix represents the TF*IDF score of each word in different message details. At this time, the word segmentation form is based on the selected corresponding word segmentation method attached to the vocabulary;

(b) N-gram level TF*IDF: N-gram is a combination of multiple words, and the sparse matrix represents the TF*IDF score of the N-gram. The so-called N-gram is an algorithm based on a statistical language model. Its basic idea is to perform a sliding window operation of size N on the content of the text according to bytes, forming a sequence of byte fragments of length N; N-

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

gram can better record the connection between words in a sentence, N The larger the sentence, the higher the completeness, but with it, the dimension of the word increases exponentially. So generally N=2 or N=3.

(c) Part of speech level TF*IDF: The sparse matrix represents the TF*IDF scores of multiple parts of speech in the corpus. The so-called part of speech refers to word attributes such as nouns, verbs, adjectives, adverbs, and pronouns. At this time, the word segmentation form is to divide the details of each message according to the word attributes and then carry out the corresponding weight vectorization.

As mentioned above, the three different levels of word segmentation used in the process of weighting the message TF*IDF vector will be an important variable control contrast in the construction of the mass message classification module.

The sparse matrix TF*IDF is weighted, and the value of each element in the sparse matrix is changed to the TF*IDF weight corresponding to the word.

$$V = V_{N\times \mathrm{n}} \Leftrightarrow M_{N\times \mathrm{n}} = [\overrightarrow{M_1}, \overrightarrow{M_2}, ..., \overrightarrow{M_{\mathrm{n}}}] = \begin{pmatrix} m_{11} & m_{21} & & m_{n1} \\ m_{12} & m_{22} & & m_{n2} \\ \cdots & \cdots & \cdots \ \cdots & \cdots \\ m_{1N} & m_{2N} & & m_{nN} \end{pmatrix} \quad (8)$$

Among them, $\overrightarrow{M_i}$ here appears in the form of a column vector, i=1,2,. . . , N. It is the text word vector formed after TF*IDF weighting. m_{ij} refers to the TF*IDF weight of the word with dictionary number j in the i-th message,j=1,2,3...,N;

Obtain feature words according to the weighted sparse matrix.

Obtain the TF*IDF value of the validation set and import it into the training set, and form the text vector space after TF*IDF weighting.

Step5: Select the classification algorithm (classifier) corresponding to the model, perform training classification, prediction, and calculate the corresponding F1-Score under the conditions of different classifiers naive Bayes, support vector machine, random forest, and linear classifier;

2) A first-level label classification model based on deep learning:

The first-level label classification model based on deep learning is mainly composed of two major modules, namely: the text preprocessing module and the model building module.

①Text preprocessing module:

Step1: Read the data set;

Step2: Convert text into digital features (choice of word segmentation: jieba, thulac, snownlp);

Step3: Convert each text to a list of numbers;

Step4: Set each text to the same length;

Step5: Convert each word encoding into a word vector;

The embedding matrix obtained after the entire text preprocessing is as follows: (the matrix dimension is (79125, 300)).

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

$$\begin{matrix} -0.9654 & -0.6791 & -0.6597 & \cdots & -0.1350 & 0.7178 \\ -0.0177 & 0.5692 & 0.7086 & \cdots & -0.0129 & 0.6746 \\ 0.9571 & 0.2398 & 0.9052 & \cdots & -0.6236 & -0.9969 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -0.4803 & -0.1936 & -0.5468 & \cdots & 0.1511 & 0.5450 \\ -0.2598 & 0.3846 & 0.5135 & \cdots & -0.5949 & 0.1284 \end{matrix}$$

②Model building module (selection of text classification algorithm):

This article mainly refers to the comparison of deep learning-based text classification algorithms TextCNN, TextRNN, LSTM, and a combination of CNN and RNN (connect the output of CNN to the output of RNN or merge the output of CNN and the output of RNN into one output):

(a) TextCNN model.

First, the text segmentation is done Embeding (word embedding) to obtain the word vector, the word vector is passed through a layer of convolution (Convolution), a layer of Max-pooling, and finally the output is externally connected to softmax for n classification.

Embedding: The first layer is the leftmost 7 by 5 sentence matrix in the figure. Each row is a word vector with dimension=5. This can be compared to the original pixels in the image;

Convolution: Then go through the one-dimensional convolution layer of kernel_sizes= (2,3,4), each kernel_size has two output channels;

Max-Polling: The third layer is a 1-max-pooling layer, so that sentences of different lengths can become fixed-length representations after passing through the pooling layer;

Softmax: Finally, a fully connected softmax layer is connected, and the probability of each category is output.

(b) TextRNN model.

Using RNN for text classification is also easier to understand, which is actually an N vs 1 model.

For English, it is word-based; for Chinese, you must first determine whether it is word-based or word-based. If it is based on words, the sentence must be segmented first. After that, each word/word corresponds to a moment in the RNN, and the hidden layer output is used as the input at the next moment. The hidden layer output h at the last moment is the abstract feature of the entire sentence, and then a softmax is used for classification. The following is a brief flow chart of its process:

RNN is different from the aforementioned CNN for text classification. After all, CNN still uses the convolution kernel to find n-gram features, and the size of the convolution kernel is a super parameter. On the other hand, RNN can handle time series, and it ensures the retention of "memory" through the output link before and after the time. However, the shortcoming of the RNN model is that the loop mechanism is too simple, and the simplest f=activate(ws+b) is used for the link before and after. In this way, there is a continuous multiplication operation in time when the gradient is propagated back, which leads to the problem of gradient disappearance and gradient explosion;

(c) LSTM model

Simply put, LSTM was born to solve the problem of long-term dependence, and LSTM is designed to avoid the problem of long-term dependence. Remember that long-term information is the default behavior of LSTM in practice, not a capability that can be acquired at a high price!

The RNN models mentioned above all have a chain form of repeating neural network modules. In a standard RNN, this repeated module has only a very simple structure, such as a tanh layer.

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

LSTM has the same structure, but the repeated modules have a different structure. Unlike a single neural network layer, there are four here, interacting in a very special way.

3.4. Comparison of Machine Learning and Deep Learning Results

1) Comparison of machine learning text classification results:

After randomly selecting part of the data for pre-running verification of the program, it can be found that when the word segmentation method adopts jieba word segmentation, hierarchical n-fold cross-validation, the fold number is 5 and the N-gram level TF*IDF N is (2,3), The overall operating efficiency of the program and the final F1-Score value are both high and stable.

2) Comparison of deep learning text classification results:

①TextCNN neural network operation results:

In this example, the TextCNN model structure is word embedding-convolution pooling*3-splicing-full connection-dropout-full connection layer (the activation function uses the softmax layer to produce 7 classification results). The rest of the parameter compile process: the optimal algorithm "adam" is the gradient descent method, the loss function is "categorical_crossentropy" (multi-classification problem cross-entropy), "acc" is the accuracy; set epochs=20, batch_size=64 during the fit process.

Table 2. TextCNN model structure

Model: "model_3"			
Layer (type)	Output Shape	Param #	Connected to
input_5 (InputLayer)	(None, 100)	0	
embedding_15 (Embedding)	(None, 100, 300)	23737500	input_5[0][0]
conv1d_13 (Conv1D)	(None, 100, 256)	230656	embedding_15[0][0]
conv1d_14 (Conv1D)	(None, 100, 256)	307456	embedding_15[0][0]
conv1d_15 (Conv1D)	(None, 100, 256)	384256	embedding_15[0][0]
max_pooling1d_10 (MaxPooling1D)	(None, 25, 256)	0	conv1d_13[0][0]
max_pooling1d_11 (MaxPooling1D)	(None, 25, 256)	0	conv1d_14[0][0]
max_pooling1d_12 (MaxPooling1D)	(None, 25, 256)	0	conv1d_15[0][0]
concatenate_4(Concatenate)	(None, 25, 768)	0	max_pooling1d_10[0][0] max_pooling1d_11[0][0] max_pooling1d_12[0][0]
flatten_5 (Flatten)	(None, 19200)	0	concatenate_4[0][0]
dropout_20 (Dropout)	(None, 19200)	0	flatten_5[0][0]
dense_14 (Dense)	(None, 7)	134407	dropout_20[0][0]
Total params: 24,794,275 Trainable params: 1,056,775 Non-trainable params: 23,737,500			

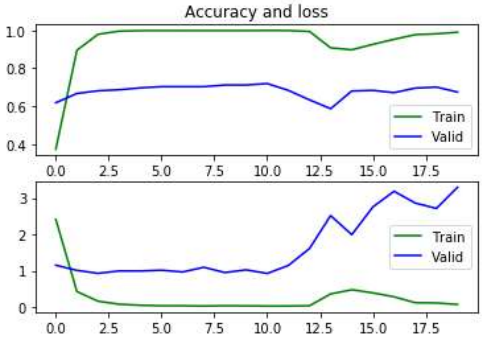

Figure 2. TextCNN model structure

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

In the figure above, the first row of graphs shows how the accuracy of the training set and test set changes with the increase of epochs; the second row of graphs shows the change process of the loss value generated by the loss function with epochs. In the process of the first ten times, the accuracy of the training set reached saturation roughly in the third time, so the loss value and accuracy did not change much after that, and the running efficiency was faster, but the fitting result was not too great. Good; in the process of the last ten times, the loss values of the training set and the test set are extended backward, and there is an overfitting situation. Therefore, the accuracy val_acc of the verification set selected for the tenth time is 0.7123, and the F1_score of the verification set is 0.6754;

②TextRNN neural network operation results:

In this example, the model structure is embedding layer + dropout layer (weight 0.5) + simpleRNN layer + dropout layer (weight 0.5) + fully connected layer (the activation function uses the softmax layer to produce 7 classification results). The rest of the parameter compile process: the optimal algorithm "adam" is the gradient descent method, "categorical_crossentropy" is the cross-entropy of the multi-classification problem, and "acc" is the accuracy; the fit process sets caepochs=20, batch_size=64.

Table 3. The TextRNN model structure

Model: "sequential_10"		
Layer (type)	Output Shape	Param #
embedding_13 (Embedding)	(None, None, 300)	23737500
dropout_18 (Dropout)	(None, None, 300)	0
simple_rnn_6 (SimpleRNN)	(None, 16)	5072
dropout_19 (Dropout)	(None, 16)	0
dropout_6 (Dropout)	(None, 100)	0
dense_13 (Dense)	(None, 7)	119
Total params: 23,742,691 Trainable params: 5,191 Non-trainable params: 23,737,500		

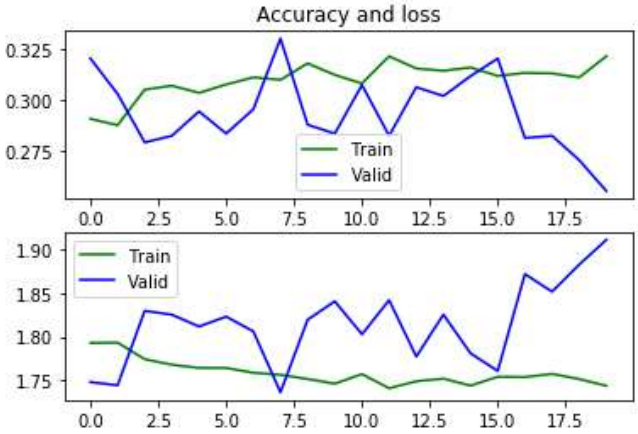

Figure 3. The TextRNN model structure

In the figure above, the first row of graphs shows how the accuracy of the training set and test set changes with the increase of epochs; the second row of graphs shows the change process of the loss value generated by the loss function with epochs. Because train and valid are messy

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

and the accuracy is always kept at a very low state, it indicates that the fitting result is poor, and the accuracy acc and F value are only about 0.3. Therefore, it is not recommended to use RNN for text classification in this example;

③LSTM neural network operation results:

In this example, the model structure is embedding layer + dropout layer (weight 0.5) + LSTM layer + fully connected layer (activation function is relu function) + dropout layer (weight 0.5) + fully connected layer (the activation function uses the softmax layer to produce 7 classification results). The rest of the parameters compile process: the optimal algorithm "adam" is the gradient descent method, "categorical_crossentropy" is the cross-entropy of the multi-classification problem, and "acc" is the accuracy; set epochs=20, batch_size=64 in the fit process.

Table 4. The LSTM model structure

Model: "sequential_3"		
Layer (type)	Output Shape	Param #
embedding_3 (Embedding)	(None, None, 300)	23737500
dropout_5 (Dropout)	(None, None, 300)	0
lstm_3 (LSTM)	(None, 300)	721200
dense_5 (Dense)	(None, 100)	30100
dropout_6 (Dropout)	(None, 100)	0
dense_6 (Dense)	(None, 7)	707
Total params: 24,489,507 Trainable params: 752,007 Non-trainable params: 23,737,500		

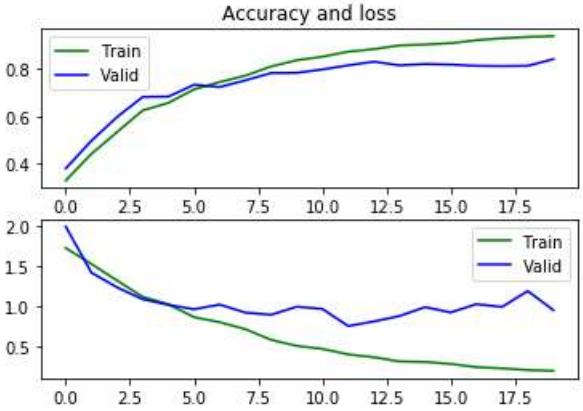

Figure 4. The LSTM model structure

In the figure above, the first row of graphs shows how the accuracy of the training set and test set changes with the increase of epochs; the second row of graphs shows the change process of the loss value generated by the loss function with epochs. Since train and valid are both rising and falling simultaneously, the model does not have over-fitting problems, and the fitting results are good. The verification set accuracy val_acc is 0.8404, and the verification set F1_score is 0.8409.

③CNN+RNN1 neural network operation results:

In this example, the output of CNN is used to directly splice the RNN.

The model structure is: embedding layer + convolutional layer pooling layer + GRU layer * 2 + fully connected layer (the activation function uses the softmax layer to produce 7 classification

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

results). Compile process of the remaining parameters: the optimal algorithm "adam" is the gradient descent method, "categorical_crossentropy" is the cross-entropy of the multi-classification problem, and "acc" is the accuracy; the fit process sets epochs=20, batch_size=64.

Table 5. The CNN+RNN1 model structure

Model: "sequential_2"		
Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, None, 300)	23737500
conv1d_1 (Conv1D)	(None, 100, 256)	230656
activation_1 (Activation)	(None, 100, 256)	0
max_pooling1d_1 (MaxPooling1	(None, 50, 256)	0
gru_1 (GRU)	(None, 50, 256)	393984
gru_2 (GRU)	(None, 256)	393984
dense_1 (Dense)	(None, 7)	1799
Total params: 24,757,923 Trainable params: 24,757,923 Non-trainable params: 0		

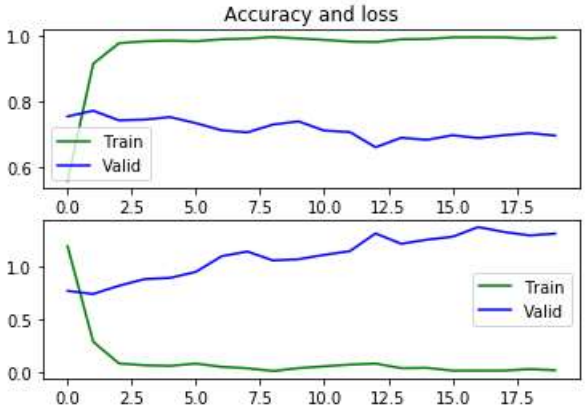

Figure 5. The CNN+RNN1 model structure

In the figure above, the first row of graphs shows how the accuracy of the training set and test set changes with the increase of epochs; the second row of graphs shows the change process of the loss value generated by the loss function with epochs. When the epochs is less than 3, the model achieves an accuracy of 0.7112. When the epochs is greater than 3, the accuracy of the training set does not change much, and the loss value of the loss function of the training set and the verification set has a backward trend, and the verification set accuracy The val_acc is 0.7112, and the F1_score of the verification set is 0.7077.

④CNN+RNN2 neural network operation results:

In this example, the output of CNN and the output of RNN are combined into one output.

The model structure is: embedding layer + convolutional layer pooling layer + fully connected layer + splicing layer + fully connected layer + two-way GRU + fully connected (the activation function uses the softmax layer to produce 7 classification results). Compile process of the remaining parameters: the optimal algorithm "adam" is the gradient descent method, "categorical_crossentropy" is the cross-entropy of the multi-classification problem, and "acc" is the accuracy; the fit process sets epochs=20, batch_size=64.

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

Table 6. The CNN+RNN2 model structure

Model: "model_1"			
Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 100)	0	
embedding_4 (Embedding)	(None, 100, 300)	23737500	input_2[0][0]
conv1d_3 (Conv1D)	(None, 100, 256)	230656	embedding_4[0][0]
max_pooling1d_3 (MaxPooling1D)	(None, 25, 256)	0	conv1d_3[0][0]
flatten_2 (Flatten)	(None, 6400)	0	max_pooling1d_3[0][0]
bidirectional_1 (Bidirectional)	(None, 512)	855552	embedding_4[0][0]
dense_3 (Dense)	(None, 256)	1638656	flatten_2[0][0]
dense_4 (Dense)	(None, 256)	131328	bidirectional_1[0][0]
concatenate_1 (Concatenate)	(None, 512)	0	dense_3[0][0] dense_4[0][0]
dense_5 (Dense)	(None, 7)	3591	concatenate_1[0][0]
Total params: 26,597,283 Trainable params: 26,597,283 Non-trainable params: 0			

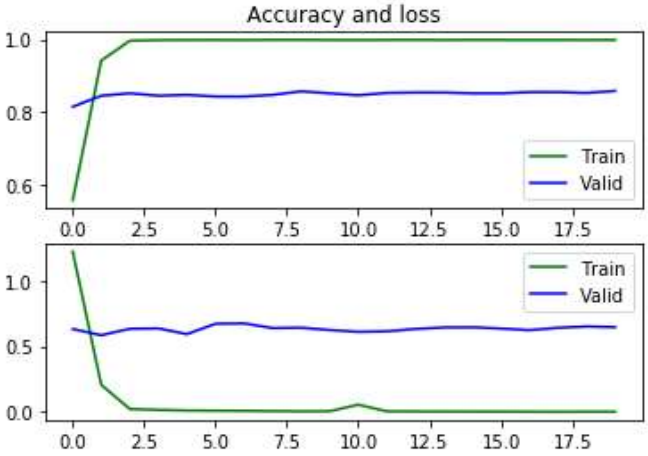

Figure 6. The CNN+RNN2 model structure

In the figure above, the first row of graphs shows how the accuracy of the training set and test set changes with the increase of epochs; the second row of graphs shows the change process of the loss value generated by the loss function with epochs. During the operation, when epochs=3, the result has been run to the best. At this time, the acc accuracy value is 0.8588, the effect is the best and the trend of the training set and the validation set are the same, and there is no problem of overfitting.

⑤Comparison of the final results of deep learning:

Analysis of deep learning results: In the deep learning of message classification in this article, since the results of accuracy and F1 values generally differ by only about 0.01, accuracy can be used to approximate the results of classification. From the table, RNN is not applicable in the text classification of this example, and the results of TEXTCNN and CNN+RNN (1) are general, but there may be over-fitting, and the results of using LSTM and CNN+RNN (2) are better. , Especially CNN+RNN(2), when the accuracy on the training set approaches 99%, the accuracy on the verification set can reach 0.8588 and there is no overfitting, so deep learning chooses CNN+RNN(2) The model is used as the result of message classification.

ISSN: 2692-7608 DOI: 10.6981/FEM.202201_3(1).0051

Table 7. Comparison of the final results of deep learning

Deep learning model	Model structure	Accuracy	F1
TextCNN	Word embedding-convolution pooling*3-splicing-full connection-dropout-full connection layer	0.7123	0.7023
TextRNN	Embedding layer-dropout layer-simpleRNN layer-dropout layer-fully connected layer	0.32	0.31
LSTM	Embedding layer-dropout layer-LSTM layer-fully connected layer-dropout layer-fully connected layer	0.8404	0.8409
CNN+RNN(1)	Embedding layer-convolutional layer pooling layer-GRU layer * 2-fully connected layer	0.7112	0.7077
CNN+RN(2)	Embedding layer-Convolutional layer Pooling layer-Fully connected layer-Splicing layer-Fully connected layer-Bidirectional GRU-Fully connected	0.8588	0.8562

4. Conclusion

Comparing machine learning and deep learning, machine learning uses part-of-speech level tfidf, selects support vector machine in the classifier, and the obtained 5-fold hierarchical cross-validation F1 value is 0.9013; while using deep learning CNN+RNN (2), the accuracy is 0.8588, therefore, according to the idea of comparison and selection, this paper chooses a machine learning method to establish a first-level label classification model. The final construction path of this module is summarized as follows:

Data preprocessing: Use regular expressions to remove line breaks and spaces for the details of each message in Annex 2, and re-merge the details of each message and the corresponding label to make it a further data set to be processed;

5-fold hierarchical cross-validation data division, and after dividing the training set and the validation set, perform jieba word segmentation and stop word list to stop both;

Feature processing: initialize the training set vector space model, calculate the TF*IDF weight of each word in the details of each message in the training set, and weight the sparse matrix TF*IDF (using the part-of-speech level TF*IDF), according to the weight The valued sparse matrix is used to obtain feature words, the verification set TF*IDF value is calculated and the training set is imported, and the text vector space is formed after TF*IDF weighting;

Select SVM (Support Vector Machine) as the classification algorithm (classifier).

Calculate the average F1-Score value of the hierarchical 5-fold cross-validation, and output the label prediction result predicted_label.xlsx of a certain test set from it. According to the above steps, the problem of classification of public messages in government work can be effectively handled, and the office efficiency of government departments can be greatly improved.

References

[1] Yoon Kim.Convolutional Neural Networks for Sentence Classification[J].2014.

[2] Ye Zhang,Byron Wallace.A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification.[J].2015.

[3] Nal Kalchbrenner,Edward Grefenstette,Phil Blunsom.A Convolutional Neural Network for Modelling Sentences.[J].2014.

[4] Chunting Zhou,Chonglin Sun,Zhiyuan Liu,Francis C.M.Lau.A C-LSTM Neural Network for Text Classification.[J].2015.