Research and Application on Text Classification Model Based on Keywords

Kuncheng Li
Data Research Center, China Academy of Information and Communications Technology, Beijing, China
likuncheng@caict.ac.cn

Chunmei Fan
Network Education College, Beijing University of Posts and Telecommunications, Beijing, China
chunmei.fan@bupt.edu.cn

ABSTRACT

Text classification is the process of assigning predefined labels to text by its content. It is a common task of Natural Language Processing (NLP). The traditional way for text classification is machine learning and its effect is greatly depended on the amount and accuracy of the training data set which is difficult to obtain in most cases. The job of building the training data set is inefficient and expensive [1]. This has motivated a research word to break this barrier, with a method using keywords instead to complete text classification. In this work, we explore the usage of label-keywords pairs (each label has a set of keywords) for assigning text documents to one or more categories automatically even without the training data set, obtaining results comparable to those systems that classify the text manually.

CCS CONCEPTS

• **Computing methodologies**; • **Modeling and simulation**; • **Model development and analysis**; • **Modeling methodologies**;

KEYWORDS

Text classification, Multi-classification, Multi-label, Tf-Idf

ACM Reference Format:
Kuncheng Li and Chunmei Fan. 2021. Research and Application on Text Classification Model Based on Keywords. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487075.3487159

1 INTRODUCTION

1.1 Background

With the rapid development of the Internet and the abundance of network data, people have been used to obtaining all kinds of data through Internet. Text classification has become an indispensable means to analyze these data. For example, if you want to recommend movies to users according to the types of movies they are interested in (such as action movies, adventure movies, comedies, etc.), you can collect movie summaries from Internet, label movies with their types (i.e. classification) according to the summaries, and then recommend them to users according to their labels. Without any preprocessed data, the commonly used machine learning method of text classification will not work. This paper will focus on how to solve above problems efficiently in the absence of preprocessed data as a training data set.

1.2 Cases of Text Classification

Text classification includes binary classification and multiclass classification [2]. Binary classification is a task of classifying the elements of a set into two groups on the basis of a classification rule. Detecting an email is a spam or not is a typical binary classification issue. Multiclass classification is a task of classifying instances into one or more of three or more classes such as the movie classification mentioned before. Multiclass classification includes the single-label issue and the multi-label issue, and the difference between them is categorizing text into one or more classes. For the movie label example, it is a single-label issue if assigning one label such as adventure to each movie and it is a multi-label issue if assigning two or more labels to a movie for a better description of the film, such as action and crime.

1.3 Text Classification Based on Machine Learning

Machine learning is widely used in text classification [3]. There are many popular algorithms such as decision trees [4], Naïve Bayes [5], Support Vector Machines (SVM) [6], k-Nearest Neighbor (kNN) [7], Neural Network, etc. Machine learning needs a completed classified corpus data as training data set [8] and test data set, and the amount of the training data set should be large enough [9]. The effect of machine learning is ideal for binary classification, but it is not good enough for multiclass classification. Especially when there is a certain degree of similarity between the categories, or the classification belongs to multi-label issue, the accuracy is even worse. In Ka-Wing Ho's paper [10] the experiment result about movies' genres classification is that the average F-score of the single-label algorithm is 0.52 and the average F-score of the multi-label algorithm is 0.46. The same experiment in Prateek Joshi's paper [11] uses different multi-label classification algorithm and the F-score is only 0.44 after adjusting the parameter.

When machine learning cannot be used (such as training data set is insufficient) or does not work well (such as multi-label issue), manual classification is usually performed. What human will do in a manual classification is to read the text word by word to find the keywords and classify the text by them, which is feasible but

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487159

inefficient. Our method exploits the structural similarity of manual processing to create a model to classify huge amount of text documents automatically for a higher efficiency.

1.4 Tf-Idf

Tf-Idf, short for Term Frequency-Inverse Document Frequency, is a numerical statistic that is intended to reflect how important a word is to a document in corpus [12].

Tf gives the count of the terms present in a document. Tf is a function of term t and document d. Tf tells that the words appear more in the document are more important [13].

$$Tf\,(\mathrm{t},\mathrm{d}) = \left[\frac{number\ of\ occurances\ of\ the\ term\ \mathrm{t}\ in\ document\ \mathrm{d}}{number\ of\ terms\ present\ in\ document\ \mathrm{d})}\right]$$

Some words are obviously not important although they appear almost in every document many times. That means a term which appears frequently in a corpus does not add special information to the target document, such as “a”, “the”, “is”. It is why Idf is needed. Idf, short for Inverse document frequency, is a measurement of uniqueness of a term to a document with respect to a corpus. Idf is defined for each term t.

$$Idf\,(\mathrm{t}) = log\left[\frac{number\ of\ documents}{number\ of\ documents\ that\ have\ \mathrm{t}}\right]$$

Tf-Idf is the multiplication of Tf(t,d) and Idf(t, D).

$$TfIdf\,(t,\ d,\ D) = Tf\,(t,\ d) * Idf\,(t,D)$$

A term which can be seen almost everywhere in a target document and cannot be found in other documents in the corpus has high degree of uniqueness on the target document and hence a high Tf-Idf weight on that document. Terms with high Tf-Idf weight are called keywords which are the most meaningful for distinguishing documents.

1.5 Label-keyowrds Set

In order to deal with a large number of text documents accurately and effectively, the general method is to classify the documents according to their main meaning by selecting one or more correct class labels predefined respectively.

As an important feature of the text data, keywords can reflect the topic of the document [14]. The main meaning of the text can be condensed into a few keywords which are limited and repetitive in a specific domain. There is a many-to-many corresponding relationship between class labels and keywords.

A text document can be labeled based on label-keyword set which should be found first and the process is as follows.

- 1) Calculate the frequency of each keyword in a label.
- 2) Calculate the frequency of each label as label frequency by summing the frequency of each keyword in a label.
- 3) Sort the set of label frequency in a descending order.
- 4) The label corresponding to the top frequency is what the text document should be.

2 PROPOSED TEXT CLASSIFICATION FRAMEWORK

The basic process of text classification based on keywords is shown in Figure 1

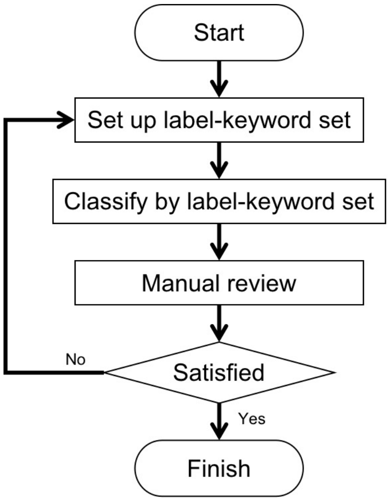


Figure 1: The Basic Process of Text Classification Based on Keywords.

2.1 Set Up Label-keyword Set

Generating workable label-keyword set is the premise of the whole work.

There are two steps to set a label-keyword set.

- 1) Set up label-keyword set manually.
- 2) We can classify a small part of the corpus manually, and then extract keywords from the classified texts by writing a TF-IDF algorithm or using some other NLP libraries. There will be some suitable keywords for classification in extracted keywords.

Rake-nltk [15] in python3 is a good choice for extracting keywords from a text. RAKE short for Rapid Automatic Keyword Extraction algorithm depends on nltk (Natural Language Toolkit). To install rake-nltk “pip3 install rake-nltk”. We’ll also need nltk, but it will automatically be installed when we install rake-nltk. It is very easy to use it.

Then we’ll add useful words to the label-keyword set. Rake-nltk works for English. If using other language we have to use other toolkit such as jieba [16] and pyhanlp [17] for Chinese.

2.2 Classify by Label-keyword Set

A label frequency can be obtained by summing up all frequency of each keywords included in the label. Normally, the label with the largest frequency is the label of the text.

There are some complicated cases.

For example, different keywords take different effect on the classification. That means some keywords are so important that the label related can be distinguished only if the keywords appear in the text. But other keywords do not work that much. So we can give each keyword a weight [18]. The more important the keyword is, the higher the weight is. The final keyword frequency is the actual keyword frequency times the weight [19]. All above is about the case of single label.

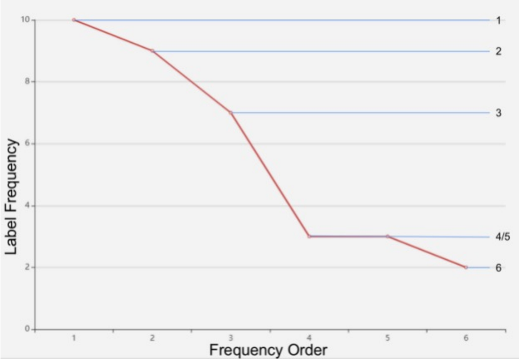


Figure 2: Ordered Frequency Number on the Axis.

If we need multi-label text classification for the corpus, the case will be more complicated. Assuming that we have ten labels (label1, label2, ..., label10) and the list of each label frequency is (0,2,3,3,0,7,0,10,9,0). After sorting, 0 excluded, the results are expressed as the following Ordered Label-frequency Pairs in descending numerical order:

[(label8,10), (label9,9), (label6,7), (label3,3), (label4,3), (label2,2)]

The corresponding simplified array is [9, 10,7,2, 3, 3].

Not all can be the labels of the text and there are two ways to choose labels by parsing the array.

One way is to find maximum difference between two adjacent elements in the array. For convenience, use two-dimensional coordinates to represent each number.

On Figure 2 Y-axis is label frequency and X-axis is order number from one. The numbers on the right of Figure 2 are also the order number and displays the distance of adjacent numbers. It is shown in Figure 2 that the maximum difference cut the labels into two parts. The labels before the maximum difference is for the text.

But sometimes there is no maximum difference or more than one maximum difference, then, another way is needed. Calculating the average of the label frequency and take the labels with the frequency which is higher than the average value as the text labels.

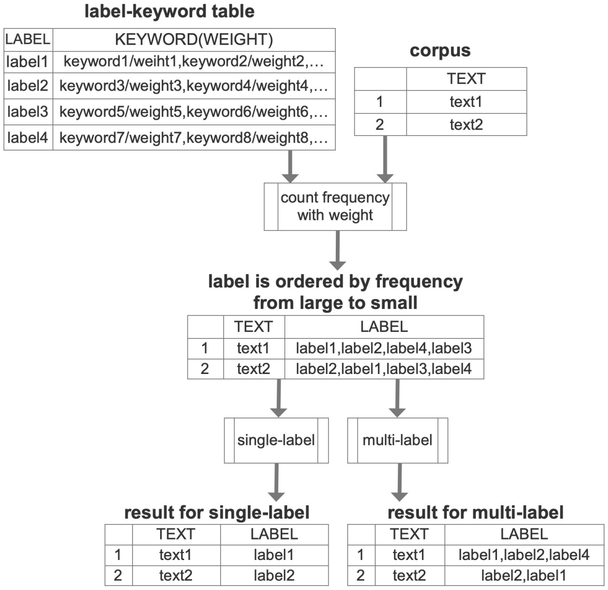
In actual cases, it is better to combine the two methods, which means to choose the labels whose frequency are higher than the average value and before the last maximum difference position.

The whole work flow of the classification by label-keyword set is Figure 3. From Figure 3 we can find that the classification by label-keyword set is divided into the following steps:

- 1) Prepare the label-keyword table and corpus.
- 2) Count label frequency for each text in corpus by the label-keyword table.
- 3) Order the labels by the frequency from large to small for each text.
- 4) If we want single-label, we choose the first label. If we want multi-label, we follow above description.

2.3 Manual Review

It's necessary to manually adjust the result of a labeling text. Initial classification result may not be satisfied, so we should improve the keywords of the label constantly by checking part of the classification result. The main job is to add or reduce the keywords and adjust the weight until we are satisfied with the result.



LABEL	KEYWORD(WEIGHT)
label1	keyword1/weiht1,keyword2/weight2,…
label2	keyword3/weight3,keyword4/weight4,…
label3	keyword5/weight5,keyword6/weight6,…
label4	keyword7/weight7,keyword8/weight8,…

	TEXT
1	text1
2	text2

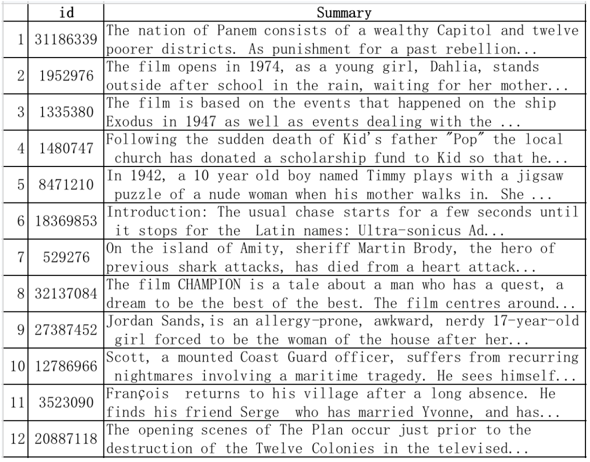
	TEXT	LABEL
1	text1	label1,label2,label4,label3
2	text2	label2,label1,label3,label4

	TEXT	LABEL
1	text1	label1
2	text2	label2

	TEXT	LABEL
1	text1	label1,label2,label4
2	text2	label2,label1

Figure 3: The Work Flow for Classifying by the Label-keyword Set.

Table 1: Movie Summary



	id	Summary
1	31186339	The nation of Panem consists of a wealthy Capitol and twelve poorer districts. As punishment for a past rebellion...
2	1952976	The film opens in 1974, as a young girl, Dahlia, stands outside after school in the rain, waiting for her mother...
3	1335380	The film is based on the events that happened on the ship Exodus in 1947 as well as events dealing with the ...
4	1480747	Following the sudden death of Kid's father "Pop" the local church has donated a scholarship fund to Kid so that he...
5	8471210	In 1942, a 10 year old boy named Timmy plays with a jigsaw puzzle of a nude woman when his mother walks in. She ...
6	18369853	Introduction: The usual chase starts for a few seconds until it stops for the Latin names: Ultra-sonicus Ad...
7	529276	On the island of Amity, sheriff Martin Brody, the hero of previous shark attacks, has died from a heart attack...
8	32137084	The film CHAMPION is a tale about a man who has a quest, a dream to be the best of the best. The film centres around...
9	27387452	Jordan Sands,is an allergy-prone, awkward, nerdy 17-year-old girl forced to be the woman of the house after her...
10	12786966	Scott, a mounted Coast Guard officer, suffers from recurring nightmares involving a maritime tragedy. He sees himself...
11	3523090	François returns to his village after a long absence. He finds his friend Serge who has married Yvonne, and has...
12	20887118	The opening scenes of The Plan occur just prior to the destruction of the Twelve Colonies in the televised...

3 EXPERIMENT

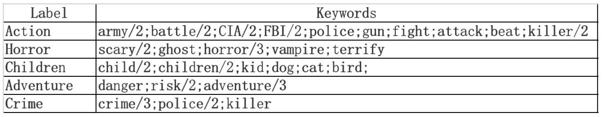
As a test, we try to classify movies by their summaries. First, we get the movie summary from the internet (http://www.cs.cmu.edu/~ark/personas/data/MovieSummaries.tar.gz), and save them in an excel file like Table 1

Second, we build the label-keyword table like Table 2

Finally, we run the python file and get the result like Table 3

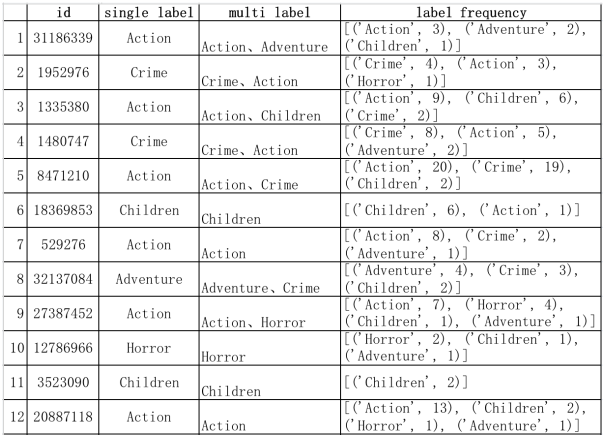
From the result we can find that the model works well. We can get the single label from the label with the maximum frequency,

Table 2: Label-keyword Table



Label	Keywords
Action	army/2;battle/2;CIA/2;FBI/2;police;gun;fight;attack;beat;killer/2
Horror	scary/2;ghost;horror/3;vampire;terrify
Children	child/2;children/2;kid;dog;cat;bird;
Adventure	danger;risk/2;adventure/3
Crime	crime/3;police/2;killer

Table 3: Classification Result for Movie Summary



	id	single label	multi label	label frequency
1	31186339	Action	Action、Adventure	[(’Action’, 3), (’Adventure’, 2), (’Children’, 1)]
2	1952976	Crime	Crime、Action	[(’Crime’, 4), (’Action’, 3), (’Horror’, 1)]
3	1335380	Action	Action、Children	[(’Action’, 9), (’Children’, 6), (’Crime’, 2)]
4	1480747	Crime	Crime、Action	[(’Crime’, 8), (’Action’, 5), (’Adventure’, 2)]
5	8471210	Action	Action、Crime	[(’Action’, 20), (’Crime’, 19), (’Children’, 2)]
6	18369853	Children	Children	[(’Children’, 6), (’Action’, 1)]
7	529276	Action	Action	[(’Action’, 8), (’Crime’, 2), (’Adventure’, 1)]
8	32137084	Adventure	Adventure、Crime	[(’Adventure’, 4), (’Crime’, 3), (’Children’, 2)]
9	27387452	Action	Action、Horror	[(’Action’, 7), (’Horror’, 4), (’Children’, 1), (’Adventure’, 1)]
10	12786966	Horror	Horror	[(’Horror’, 2), (’Children’, 1), (’Adventure’, 1)]
11	3523090	Children	Children	[(’Children’, 2)]
12	20887118	Action	Action	[(’Action’, 13), (’Children’, 2), (’Horror’, 1), (’Adventure’, 1)]

and get the multi-label from the labels which frequency is more than the average frequency and before the maximum difference. The result is what we need.

4 CONCLUSIONS

In this paper we classify the text corpus by label-keyword set. It is a easily controllable and modifiable way[18]. It solves the reliance on training data set of machine learning and it is even better than the machine learning especially in the case of multi-label for multi-class classification. The most important thing in this model is to establish the label-keyword set as full-scale and as accurate as possible. Manually reviewing and fixing the label-keyword set are also very important. In the practice we find that the quality of text features directly affects the text classification effect [20], so the process of improving the model includes improving the keyword table and improving the corpus as much as possible. After the continuous improvement we will get a good classification result.

REFERENCES

[1] McCallum A, Nigam K (1999). Text classification by bootstrapping with keywords, EM and shrinkage[C]//Unsupervised Learning in Natural Language Processing.

[2] Jeatrakul, P., Wong, K.W. (2009). Comparing the performance of differentneural networks for binary classification problems. In: 2009 EighthInternational Symposium on Natural Language Processing. pp. 111–115.

[3] Purohit A, Atre D, Jaswani P, *et al.* (2015). Text classification in data mining[J]. International Journal of Scientific and Research Publications, 5(6), 1-7.

[4] Sakakibara Y, Misue K, Koshiba T (1993). Text classification and keyword extraction by learning decision trees[C]//Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications. IEEE, 466.

[5] Purohit A, Atre D, Jaswani P, *et al.* (2015). Text classification in data mining[J]. International Journal of Scientific and Research Publications, 5(6), 1-7.

[6] Kamruzzaman, S. M. , & F Haider. (2010). A hybrid learning algorithm for text classification. Computer Science.

[7] Jason Kroll (2003). Decision Tree Learning for Arbitrary Text Classification," Sept 2003, www.cs.tufts.edu/~jkroll/dectree.

[8] Aly M (2005). Survey on multiclass classification methods[J]. Neural Netw, 19, 1-9.

[9] Park G , Kim S (2006). Web Document Classification Using Changing Training Data Set.[J]. Lecture Notes in Computer Science, 3984, 565-574.

[10] Ho K W (2011). Movies' genres classification by synopsis[J].

[11] Prateek Joshi (2011). Predicting Movie Genres using NLP – An Awesome Introduction to Multi-Label Classification. Analytics Vidhya, 2019.

[12] Rajaraman, A.; Ullman, J.D.. "Data Mining". Mining of Massive Datasets. pp. 1-17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.

[13] Isuru Boyagane (2020) "How important the words in your text data? TF-IDF answers..." in Towards Data Science, Oct 14,2020[online]https://medium.com/@isuruboyagane.16

[14] Ni P, Li Y, Chang V (2020). Research on Text Classification Based on Automatically Extracted Keywords [J]. International Journal of Enterprise Information Systems (IJEIS), 16(4), 1-16.

[15] RAKE_NLTK [online] https://github.com/csurfer/rake-nltk

[16] Jieba [online] https://github.com/fxsjy/jieba

[17] Pyhanlp [online] https://github.com/hankcs/pyhanlp

[18] Fonda W, Purwarianti A (2014). Experiments on keyword list generation by term distribution clustering for text classification[C]//2014 International Conference on Advanced Computer Science and Information System. IEEE, 297-301.

[19] Saad M K (2010). The impact of text preprocessing and term weighting on arabic text classification[J]. Gaza: Computer Engineering, the Islamic University.

[20] Yongxia J, Heping GOU, Wei SUN (2019). Text Classification Based on LDA and Semantic Analysis [J]. DEStech Transactions on Computer Science and Engineering, (iccis).