Report for Final Project : Analysis of Rumor

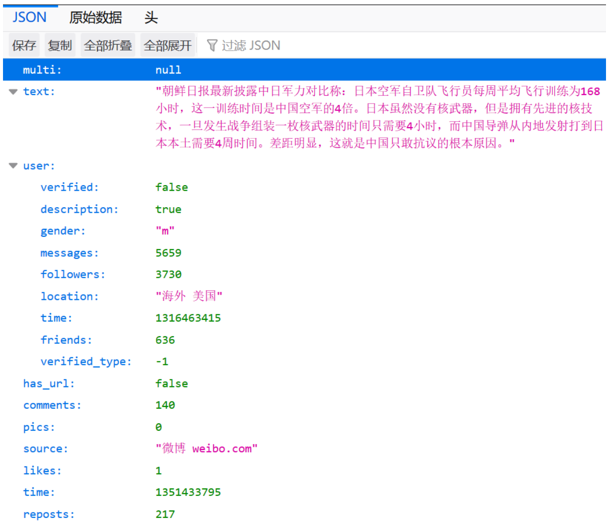
Chen Ying, Zhu Simo, Liu Guoding, Mao Haining

January, 2019

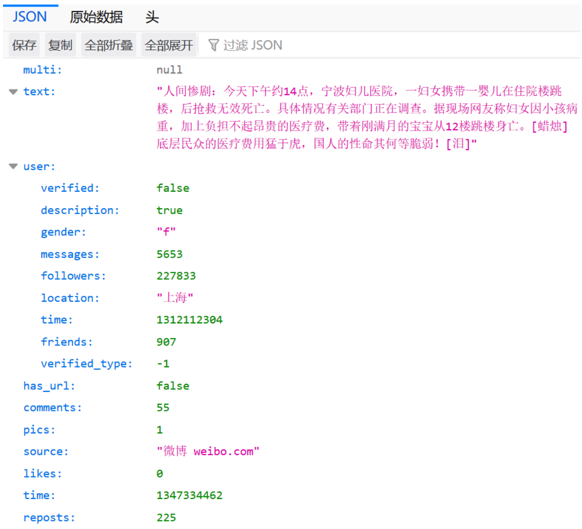
Contents

1 Preliminary Process (Data Reading)

The documents in the dataset are scores of JSON (JavaScript Object Notation) documents, the information contained in one json document has the construct below:



```
multi:              null
text:               "朝鲜日报最新披露中日军力对比称：日本空军自卫队飞行员每周平均飞行训练为168
                    小时，这一训练时间是中国空军的4倍。日本虽然没有核武器，但是拥有先进的核技
                    术，一旦发生战争组装一枚核武器的时间只需要4小时，而中国导弹从内地发射打到日
                    本本土需要4周时间。差距明显，这就是中国只敢抗议的根本原因。"
user:
    verified:       false
    description:    true
    gender:         "m"
    messages:       5659
    followers:      3730
    location:       "海外 美国"
    time:           1316463415
    friends:        636
    verified_type:  -1
has_url:            false
comments:           140
pics:               0
source:             "微博 weibo.com"
likes:              1
time:               1351433795
reposts:            217
```



```
multi:              null
text:               "人间惨剧：今天下午约14点，宁波妇儿医院，一妇女携带一婴儿在住院楼跳
                    楼，后抢救无效死亡。具体情况有关部门正在调查。据现场网友称妇女因小孩病
                    重，加上负担不起昂贵的医疗费，带着刚满月的宝宝从12楼跳楼身亡。[蜡烛]
                    底层民众的医疗费用猛于虎，国人的性命其何等脆弱！[泪]"
user:
    verified:       false
    description:    true
    gender:         "f"
    messages:       5653
    followers:      227833
    location:       "上海"
    time:           1312112304
    friends:        907
    verified_type:  -1
has_url:            false
comments:           55
pics:               1
source:             "微博 weibo.com"
likes:              0
time:               1347334462
reposts:            225
```

The useful information of it contains:

- Text
- Weibo Features
  - (a) Whether the weibo has URL
  - (b) Number of comments
  - (c) Pics
  - (d) Sources of this weibo
  - (e) Likes
  - (f) The time when this weibo is sent
- User Features
  - (a) Whether the user has description
  - (b) Whether the user is verified and the verified type
  - (c) The gender of user
  - (d) Number of followers
  - (e) Number of friends
  - (f) The location of the user
  - (g) The time when the user joined weibo
  - (h) Number of messages user had sent

To get this information, we preprocess these json documents. In this process, we use the os & json module to read all the json data and extract information. Since the dataset has been processed, there is no data missing (All the json documents have complete information). Then we encode each feature by the method ordinary encoding. First, we use list to collect the data and then transform it into array (except the text, the text will be processed in the feature extraction). Meanwhile we transform the data type into int (There are some examples below).

```python
has_url = np.array(has_url).astype(int)
verified = np.array(verified).astype(int)
description = np.array(description).astype(int)
gender = np.array(list(map(trans_gender, gender)))
followers = np.array(followers)
friends = np.array(friends)
category = np.array(category)
```

2 EDA & Feature Extract

At this project, we try to extract features from the information in each json document in the dataset. Before this project we read the paper written by Professor Liu Zhiyuan, we found there are some methods to detect rumors according to the features of the time series, but in that method we can't detect rumors as soon as they are sent. This idea determines the features we will choose. It should be attached great importance here that in this project we want to extract the early characteristics of rumors. That is to say the information of rumor after the time when rumor is sent, such as comments of this rumor isn't considered in this project.

First we use EDA to see the general features of rumors in this dataset and then we select the features which will be used in the model training process. (In the real process, we must construct feature first and then use EDA to see the characteristics. After this we determine whether we will use these features. The order this report describe is little different from the real order.)

2.1 EDA

In EDA process, we first explore the single feature of rumors as well as non-rumors and then compare them in one picture. For example we count the number of exclamation marks(the coordinate has been normalized) in one text and then describe it through frequency histogram.

In this process, we find the rumors will use more exclamation marks, question marks. Besides they tend to present negative attitude. Meanwhile, we find the number of friends of rumor-monger is usually large.

Then we explore the correlation between different features(variables). At the same time, we can see the features more clearly. There's no very strong feature, so the combination of different features may be a good method. We also find an interesting thing that the positive rate of the rumors is not always low, instead, it is very low or very high. That reminds us the rumors may be too positive or too negative, so decision tree may be a good choice. However, generally speaking, the rumors are always radical.

2.2 Feature Extract

After the exploration process, considering the construct of the dataset and the characteristics of the rumor texts. We select the features below to be the features we will extract in this project.

- Text Features

(a) Number of exclamatory mark
(b) Number of question mark
(c) Number of positive words
(d) Number of negative words
(e) Positive rate of the text

- Weibo Features Whether the weibo has URL
- User Features

(a) Whether the user has description
(b) Whether the user is verified
(c) The gender of user
(d) Number of followers
(e) Number of friends

Since the features except text features have been processed in the preliminary process, at this time we only need to extract the features from text.

This process is hard to handle. So we refer to the paper and conclude that rumors are always radical. So we thought about the features which can reflect whether the text is radical. According to the knowledge of language, we select:

- Number of exclamatory mark
- Number of question mark
- Number of positive words
- Number of negative words
- Positive rate of the text

We use this features for the reason that if one text is radical, it will tend to use more exclamatory marks and more question marks. Besides it will use more positive or negative words. In general the tonal of the text will be radical so we try to construct a feature which will reflect this point.

We know the character of one sentence will be represented by the “key word” of this sentence. The “key word” in this project is defined:

> The words appeared in this sentence but don’t appear in most of sentences we use in daily life.

So we want to get the “key word” and evaluate the text by the “key word” of the text. Then we use module jieba to split the text (before this we use regular expression to clean the text) and get the most importance “key words” of the text (jieba has a corpus and can return the importance words according to the TF). Number of importance words can be set, and this is a hyper-parameter. Once we get the “key words”, we use SnowNLP module to evaluate the emotion of each word. The value of neutral word is 0.5. The value will be larger if the word is positive and smaller if the word is negative. So we count the positive words and negative words of these “key words” according to the rule: if the value is larger than 0.6, the word is positive word. If the value is smaller than 0.4, the word is negative word.

At the same time, we calculate the positive rate of the whole sentence according to the “key words”. The algorithm is:

First get the evaluate value of each word and then subtract it by 0.5. Then get the average of this subtracted value. This value is the whole positive rate of the whole sentence.

Meanwhile, we count the number of exclamatory mark Number of question mark and then transform the data into array.

Then we get the whole features we want to extract. All the features have been transformed into array and all the data types are int or double.

3 Model Selection and Model Training

In this project, we want to select a model that can classify whether a document is a rumor or not.

Because we cannot easily tell the answer, we need a model that is highly interpretable. Here are some alternatives:

- KNeighborsClassifier
- Logistic regression
- Decision tree
- Random Forest
- Adaboost

The model of deep learning is so complex that we can hardly interpret it, so we don’t use neural networks.

Then we use cross-validation, split the dataset into training set(70%) and testing set(30%). Using the training-set to train the model and test it on the testing set, we can calculate the accuracy of the model. Initially we do not adjust the parameters of the model.

Random Forest is the best model for this problem.

In the process of adjusting the parameters of the LR model, we find that decreasing regular terms significantly increases model accuracy. This means we are dealing with an underfitting problem. So Random Forest perform better than LR. Then we use GridSearchCV to get better parameters. At last we find that when we do not limit the max-depth and max-leaf-nodes, set estimated quantity to 118 and criterion to gini, we can get the highest accuracy.

Limited by the quality of the original dataset, the accuracy of the model is 0.789.

Then we can calculate the importance of each feature to tell which feature we should care more.

We can find some interesting results from the importance of feature. As predicted previously, all features are weak. The most important feature is the number of followers. This is a counter-intuitive result that neither validation nor positive word rate matter. This may be because the dataset contains rumors of many low-level users who have few followers. Another interesting finding is that although both positive and negative words are weak features, the positive rate of a sentence is a relatively strong feature. This could offer us some insights when identifying rumors.