839 projecct 1 report

Zhendong Zhou Yunwen Jiang Yixin Chen

In this project, we plan to extract names from documents. All the documents are articles and novels from the internet. We tagged the names in each document with labels like :<name>…</name>. For examples:

My name is <name>Michael</name>
The old <name>Josh</name> is sleepy.

Among 300 documents, We have marked 1346 names.

In set I, there are 200 documents, and there are 904 mentions within them.

In set J, there are 100 documents, and there are 442 mentions within them.

All the negative variables are chosen to be started with capitalized letter, which will make it much harder. But we think that it's worth trying.

First, we decided to apply 18 features:

1. If there is honorific in front of the sample, like: Miss, Mr, professor,etc.
2. If all the characters are capitalized.
3. If there is comma before and after the word.
4. If there is "a", "an", "the" before or after the word.
5. If there is special adj. before the word like "poor" and "cute".
6. If there is special adj after the word.
7. If there is special word before the word like "sister" and "brother".
8. If there is some special word before the word like "said" and "named".
9. If it's the end of the sentence.
10. If it's ended up or started with some important words like "who".
11. If there is prep. before or after the word.
12. If the word before it is ended up with "ed".
13. If the word after it is ended up with "ed"
14. The length of the word.
15. If the word starts with capital letter.
16. If there is "'s" after the word.
17. If it's the start of the sentence.
18. If it's ended up with "an", like "American". "African" etc.

Then we transferred the document into a data frame containing the features and fitted the machine learning algorithms on that.

The algorithms I applied are Logistic, Linear Regression, Decision Tree, Random

Forest, SVM and MLP.

Finally we got some conclusions.

But after taking a look at the conclusions, we found that our data is not cleaned. There are some words that are not labeled right, so we just re-label the data and do the process again. The output is shown below:

Classifier M is MLP. The output is shown below:

MLP

	precision	recall	f1-score	support
negative	0.88	0.91	0.90	2014
positive	0.79	0.73	0.76	887

Unfotuenately, it's not good enough. So we decide to debug it.

After taking a look at the samples that are not classified right, we added two new rules:

1. If the names are Father, Mother, Grandfather, Grandmother, etc. For we tried a lot and found that we can't deal with this kind of words, They are almost the same structure with the real names. Though it's easy for us human to recognize it, but there is no way to tell the machine.
2. If there are words like "a", "an" after the word, which we missed before but actually very important.

Considering that we are only interested in the precision and recall of the positive samples, so we dedcide to change the parameter of the algorithm when making the decision.

After debugging, the best classifier X is still MLP. The output is shown below:

MLP

	precision	recall	f1-score	support
negative	0.88	0.92	0.90	2014
positive	0.80	0.71	0.75	887

Which seems to be much better, but still not what we want. Then we go back to see the words that are not predicted right, we got the output like below:

```
Coney Island          1
Merchant              1
Alarm                 1
Department            1
African               1
Station               1
                     ..
Zoological Society    1
Whatever              1
Complaint             1
Sewing Society        1
Warwickshire          1
Old World             1
Australia             1
London                1
Those                 1
Someone               1
Le Fanu               1
Russian               1
For God               1
Something             1
Infantry              1
Thin                  1
England               1
Fort Phil Kearney     1
Darkness              1
Netherfield           1
Enfield               1
Housekeeping          1
The Golden Notebook   1
Spring                1
What                  1
Winesburg             1
Darwin                1
Virginian             1
Equally               1
Sixth Avenue          1
```

There is nothing we can conclude from! All the wrong cases seem to be unique, and we can hardly do anything to improve it. After attempting for couple of times, we find that we can't make it better, so we decide to try it on the Set J, the output is shown below:

One thing need to say is that, due to the randomness of the RandomForest, the output will change every time we run the code, so it's normal not succeeding the threshold in some situations.

MLP

	precision	recall	f1-score	support
negative	0.85	0.93	0.89	1033
positive	0.81	0.63	0.71	453

Unfortuately, we still can't make any change.

The reason, we think, should because that our data are downloaded from internet, and from multiple sources, which may make it not structured enough, so the features we derived may not suitable to some samples. The machine learning itself can't deal with this problem, so we should have a better, cleaner data source next time.

Actually, we have considered to add some black list to make it better, but finally we think that it's not what we want, so we just choose to keep this not-so-good output.

Sorry about that, but we have tried our best.

The most important thing is that: we all learnt a lot through this project:

The data itself, should always from the same sample. For example: we have documents from different kind of novels, so some animals may act like human like in fairy tale, etc. These situations are ones we can't deal with. In the future, we need to pay more attention on the data itself before conducting the project.