Build your own Text classification with less than 25 lines of code using fasttext

Ravindra Elicherla
Aug 30, 2018 · 6 min read

Text Classification

Text classification is a basic machine learning technique used to smartly classify text into different categories. There are plenty of use cases for text classification. Spam filtering, sentiment analysis, classify product reviews, drive the customer browsing behaviour depending what she searches or browses and targeted marketing based on what the customer does online etc. In this example, we will use supervised classification of text. It works on the principle of “training” and “validate” principle. We input labeled data to the machine learning algorithm to work on. After the algorithm is trained, we use the training dataset to understand accuracy of the algorithm and training data. The effectiveness of the output depends on quality of the data and strength of the algorithm. In this example, there is no need to write any algorithm, we will use fasttext internal algorithm.

fasttext

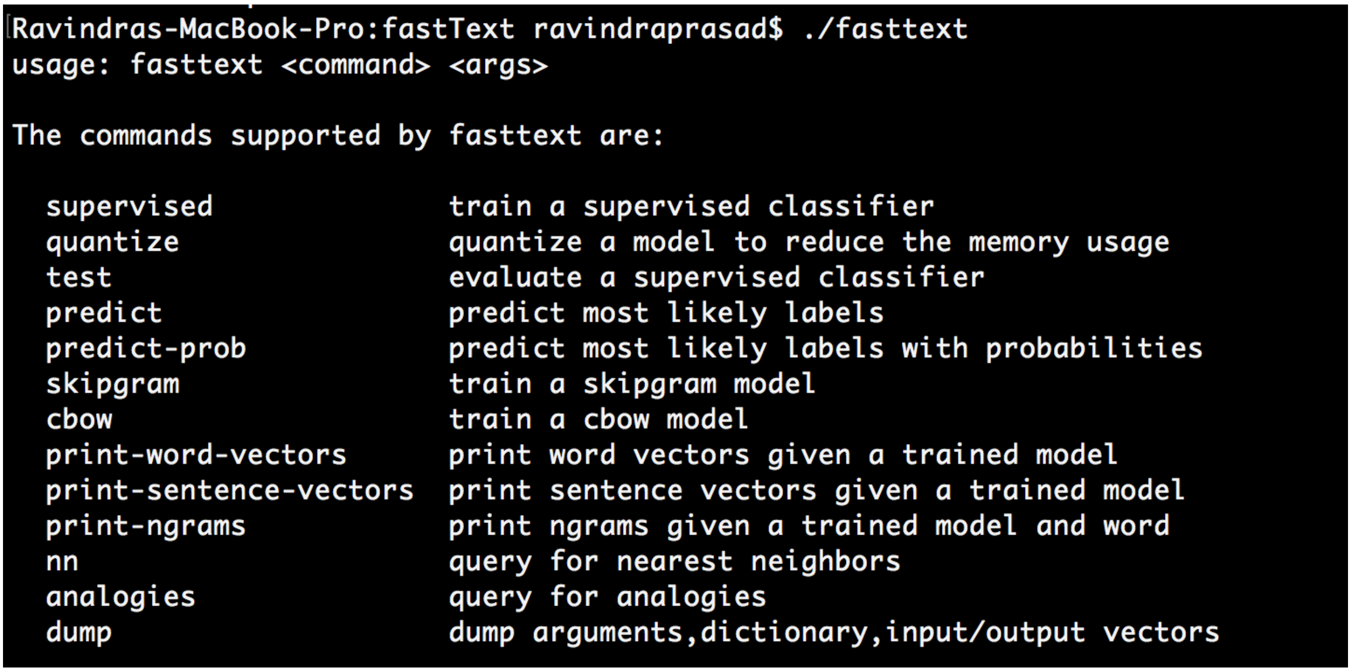
In this blog we will classify consumer complaints automatically into one or more relevant categories automatically using fasttext. FastText is an open-source, free, lightweight library that allows users to learn text representations and text classifiers. It works on standard, generic hardware. This is Open Sourced by Facebook.

Installing fasttext is very easy. I am using OSx.

```
$ git clone https://github.com/facebookresearch/fastText.git

$ cd fastText

$ make
```

You can check the success of fasttext installation by giving below command.

```
Ravindras-MacBook-Pro:fastText ravindraprasad$ ./fasttext
```



```
Ravindras-MacBook-Pro:fastText ravindraprasad$ ./fasttext
usage: fasttext <command> <args>

The commands supported by fasttext are:

  supervised              train a supervised classifier
  quantize                quantize a model to reduce the memory usage
  test                    evaluate a supervised classifier
  predict                 predict most likely labels
  predict-prob            predict most likely labels with probabilities
  skipgram                train a skipgram model
  cbow                    train a cbow model
  print-word-vectors      print word vectors given a trained model
  print-sentence-vectors  print sentence vectors given a trained model
  print-ngrams            print ngrams given a trained model and word
  nn                      query for nearest neighbors
  analogies               query for analogies
  dump                    dump arguments,dictionary,input/output vectors
```

Get and prepare data:

Download the Consumer complaints data csv file from here. This dataset has complaints received about financial products and services. These complaints are neatly categorised into various products. The CSV file has Date received, Product, Sub-

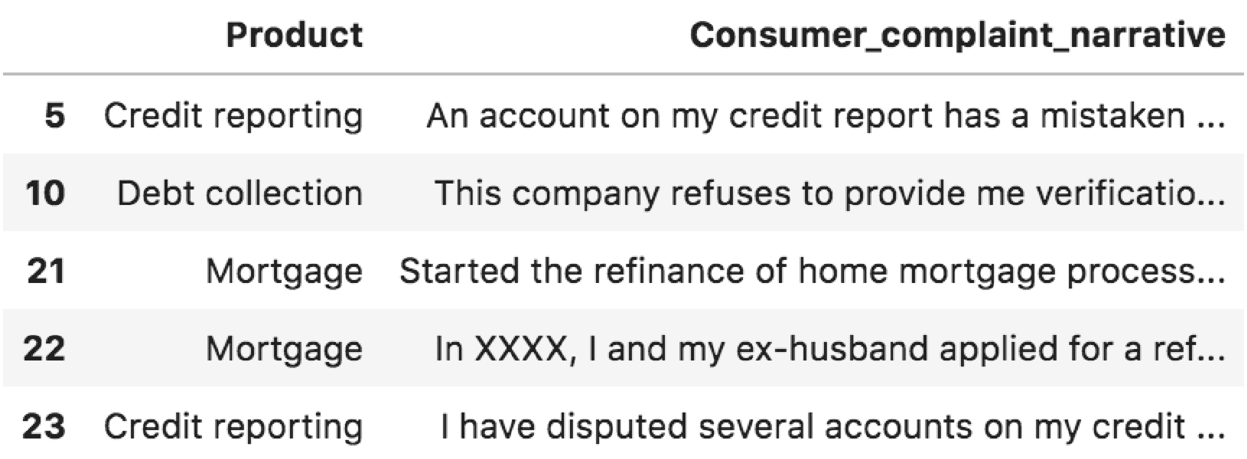
Product, Issue, Sun-Issue, Consumer Complaint Narrative etc. In this example we are interested in Product and Consumer Complaint Narrative. The problem statement we are trying to solve is "When a customer writes new complaint, how do we categorise into product automatically?" Is it not interesting? Let's start with the solution now.

Read and process the file using below python code.

```
import pandas as pd
consumercompliants = pd.read_csv('/Users/ravindraprasad/Ravindra/fasttext/fasttext/data/

from io import StringIO
col = ['Product', 'Consumer complaint narrative']
consumercompliants = consumercompliants[col]
consumercompliants = consumercompliants[pd.notnull(consumercompliants['Consumer complair
consumercompliants.columns = ['Product', 'Consumer_complaint_narrative']
consumercompliants.head()
```

preparefile.py hosted with ❤ by GitHub view raw

Your output will look something like this.



	Product	Consumer_complaint_narrative
5	Credit reporting	An account on my credit report has a mistaken ...
10	Debt collection	This company refuses to provide me verificatio...
21	Mortgage	Started the refinance of home mortgage process...
22	Mortgage	In XXXX, I and my ex-husband applied for a ref...
23	Credit reporting	I have disputed several accounts on my credit ...

fasttext expects the data to be some thing like this

> *__label__1 this is my text*
>
> *__label__2 this is also my text*

we now need to prepare product data. The output should look like

> *__label__credit_reporting An account on my credit report…*
>
> *__label__Debt_collection This company refuses to provide me…*

Let’s extend the previous code.

```
import pandas as pd
consumercompliants = pd.read_csv('/Users/ravindraprasad/Ravindra/fasttext/fasttext/data

from io import StringIO
col = ['Product', 'Consumer complaint narrative']
consumercompliants = consumercompliants[col]
consumercompliants = consumercompliants[pd.notnull(consumercompliants['Consumer complai
consumercompliants.columns = ['Product', 'Consumer_complaint_narrative']


#consumercompliants['Product'] = consumercompliants['Product'].replace(' ', '_', regex=
consumercompliants['Product']=['__label__'+s.replace(' or ', '$').replace(', or ','$').
consumercompliants['Product']

consumercompliants['Consumer_complaint_narrative']= consumercompliants['Consumer_compla

#consumercompliants['Consumer_complaint_narrative']=consumercompliants['Consumer_compla
consumercompliants.to_csv(r'/Users/ravindraprasad/Ravindra/fasttext/fasttext/data/consu
```

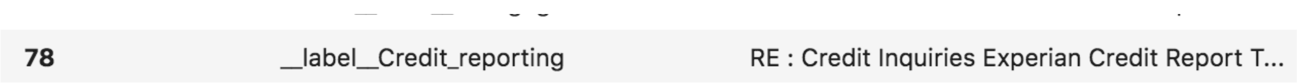
preparefil2.py hosted with ❤ by GitHub view raw

give this command to check the contents.

> *consumercompliants.head(200)*

bellow is output

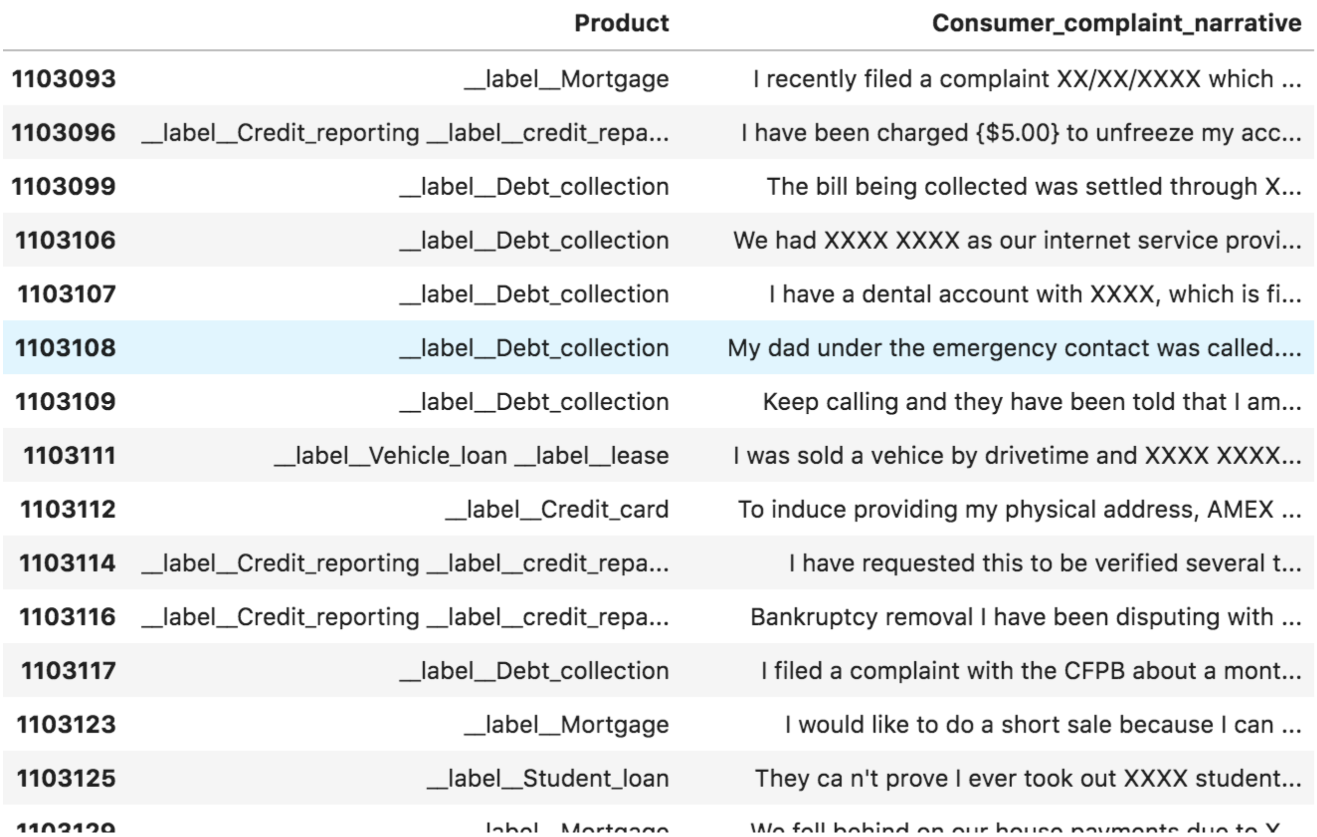
	Product	Consumer_complaint_narrative
5	__label__Credit_reporting	An account on my credit report has a mistaken ...
10	__label__Debt_collection	This company refuses to provide me verificatio...
21	__label__Mortgage	Started the refinance of home mortgage process...
22	__label__Mortgage	In XXXX, I and my ex-husband applied for a ref...
23	__label__Credit_reporting	I have disputed several accounts on my credit ...
24	__label__Mortgage	Mortgage was transferred to Nationstar as of X...
30	__label__Credit_card	Was a happy XXXX card member for years, in lat...
36	__label__Credit_card	Without provocation, I received notice that my...
41	__label__Debt_collection	I am writing to request your assistance in loo...
50	__label__Credit_reporting	I am disputing the inaccurate information the ...
53	__label__Credit_reporting	Checked my credit report after filing complain...
57	__label__Mortgage	Need to move into a XXXX facility. Can no long...
70	__label__Mortgage	I had an FHA loan at US Bank that was paid off...



78	__label__Credit_reporting	RE : Credit Inquiries Experian Credit Report T...

also check the tail

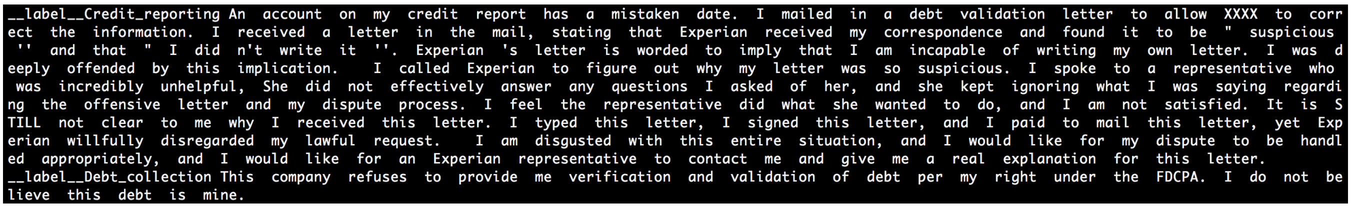
consumercompliants.tail(1000)



	Product	Consumer_complaint_narrative
1103093	__label__Mortgage	I recently filed a complaint XX/XX/XXXX which ...
1103096	__label__Credit_reporting __label__credit_repa...	I have been charged {$5.00} to unfreeze my acc...
1103099	__label__Debt_collection	The bill being collected was settled through X...
1103106	__label__Debt_collection	We had XXXX XXXX as our internet service provi...
1103107	__label__Debt_collection	I have a dental account with XXXX, which is fi...
1103108	__label__Debt_collection	My dad under the emergency contact was called....
1103109	__label__Debt_collection	Keep calling and they have been told that I am...
1103111	__label__Vehicle_loan __label__lease	I was sold a vehice by drivetime and XXXX XXXX...
1103112	__label__Credit_card	To induce providing my physical address, AMEX ...
1103114	__label__Credit_reporting __label__credit_repa...	I have requested this to be verified several t...
1103116	__label__Credit_reporting __label__credit_repa...	Bankruptcy removal I have been disputing with ...
1103117	__label__Debt_collection	I filed a complaint with the CFPB about a mont...
1103123	__label__Mortgage	I would like to do a short sale because I can ...
1103125	__label__Student_loan	They ca n't prove I ever took out XXXX student...
1103129	__label__Mortgage	We fell behind on our house payments due to X

In the terminal give this command to see if the file is loaded correctly

head consumer.complaints.txt



```
__label__Credit_reporting An account on my credit report has a mistaken date. I mailed in a debt validation letter to allow XXXX to corr
ect the information. I received a letter in the mail, stating that Experian received my correspondence and found it to be " suspicious
 '' and that " I did n't write it ''. Experian 's letter is worded to imply that I am incapable of writing my own letter. I was d
eeply offended by this implication.  I called Experian to figure out why my letter was so suspicious. I spoke to a representative who
 was incredibly unhelpful, She did not effectively answer any questions I asked of her, and she kept ignoring what I was saying regardi
ng the offensive letter and my dispute process. I feel the representative did what she wanted to do, and I am not satisfied. It is S
TILL not clear to me why I received this letter. I typed this letter, I signed this letter, and I paid to mail this letter, yet Exp
erian willfully disregarded my lawful request.  I am disgusted with this entire situation, and I would like for my dispute to be handl
ed appropriately, and I would like for an Experian representative to contact me and give me a real explanation for this letter.
__label__Debt_collection This company refuses to provide me verification and validation of debt per my right under the FDCPA. I do not be
lieve this debt is mine.
```

count number of records

wc consumer.complaints.txt

output is

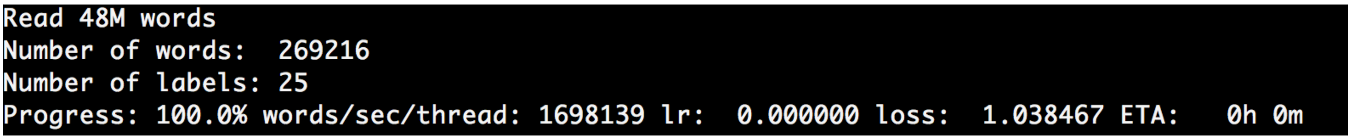
314263 62315198 415588908 consumer.complaints.txt file has 3,14,263 records.

We will make two datasets with about 80% data or 2,50,000 records for training and 20% data or 64263 records for validation (testing).

> *head -n 250000 consumer.complaints.txt > complaints.train.txt*
>
> *tail -n 64263 consumer.complaints.txt > complaints.valid.txt*

now train the model using fasttext

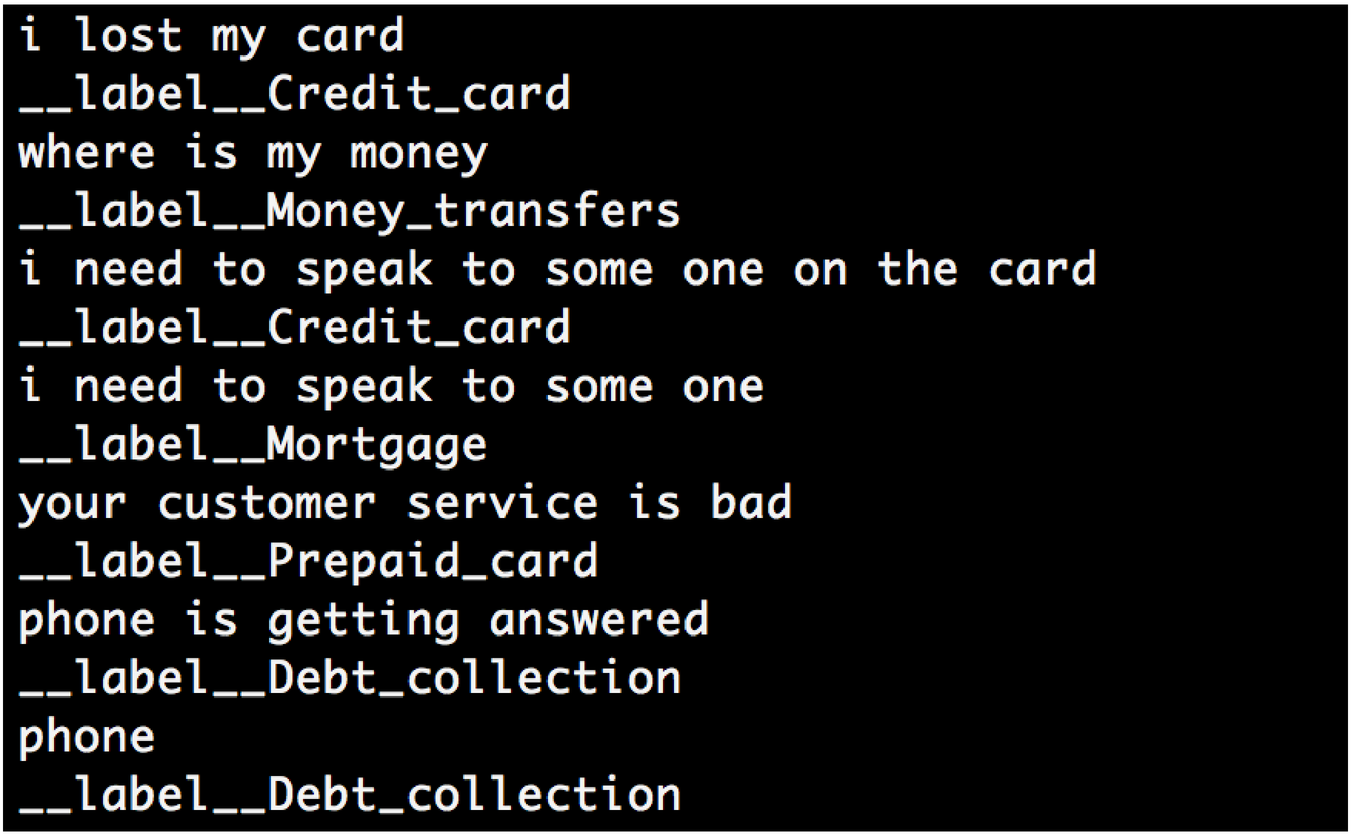
> *./fasttext supervised -input complaints.train.txt -output model_complaints*



```
Read 48M words
Number of words:  269216
Number of labels: 25
Progress: 100.0% words/sec/thread: 1698139 lr:  0.000000 loss:  1.038467 ETA:   0h 0m
```

we can now test the model using

> *./fasttext predict model_complaints.bin -*

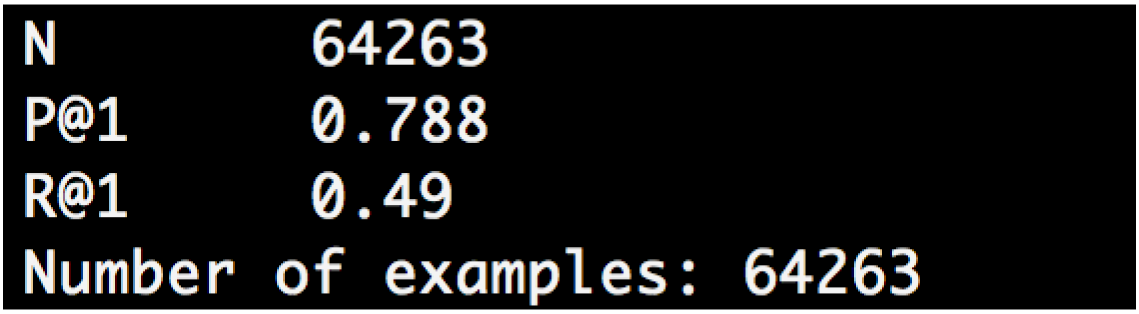


```
i lost my card
__label__Credit_card
where is my money
__label__Money_transfers
i need to speak to some one on the card
__label__Credit_card
i need to speak to some one
__label__Mortgage
your customer service is bad
__label__Prepaid_card
phone is getting answered
__label__Debt_collection
phone
__label__Debt_collection
```

Some complaints it is able to classify correctly. But some looks like did not go well. Especially phone and customer service is bad. I liked “Where is money?” The model correctly predicted it is related to money transfer. However, i would think “i need to speak to someone” should ideally be related to Service. But in this case it is showing under Mortgage. We will further enhance the model in later steps.

We can also check the effectiveness of the model using test data.

```
./fasttext test model_complaints.bin complaints.valid.txt
```



```
N       64263
P@1     0.788
R@1     0.49
Number of examples: 64263
```

Here N is number of test records, Precision at 1 (P@1) and Recall at 1 (R@1)

The Precision is the number of correct labels among the labels predicted by fastText. The Recall is the number of labels that successfully were predicted, among all the real labels.

We will try improving the model by getting rid of special characters and changing upper case letters to lower case.

```
cat consumer.complaints.txt | sed -e “s/\([.\!?,’/()]\)/ \1 /g” | tr “[:upper:]” “[:lower:]”
> consumer.processed.txt
```

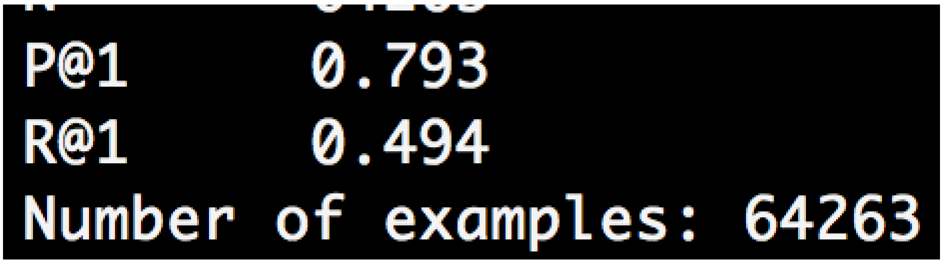
Lets again make data for training and testing.

```
head -n 250000 consumer.processed.txt > complaints.processed.train.txt

tail -n 64263 consumer.processed.txt > complaints.processed.valid.txt

./fasttext supervised -input complaints.processed.train.txt -output
model_complaints_processed
```

Number of words are reduced from 2,29,616 to 1,14,973 almost 50% drop.

```
./fasttext test model_complaints_processed.bin complaints.processed.valid.txt
```

```
N           64263
```

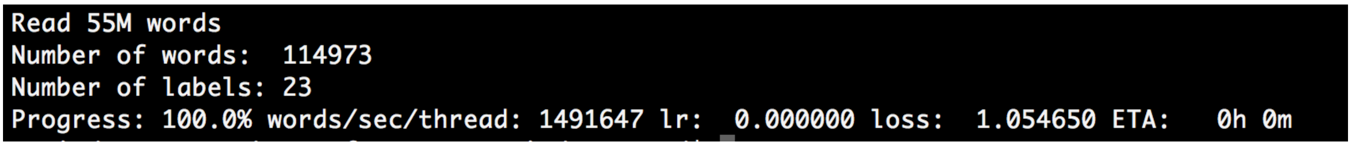


```
P@1     0.793
R@1     0.494
Number of examples: 64263
```

You can see that precision increased now from 0.788 to 0.793 about 0.8%. Not a big change.

The number of times each examples is seen (also known as the number of epochs), can be increased using the `–epoch` option:

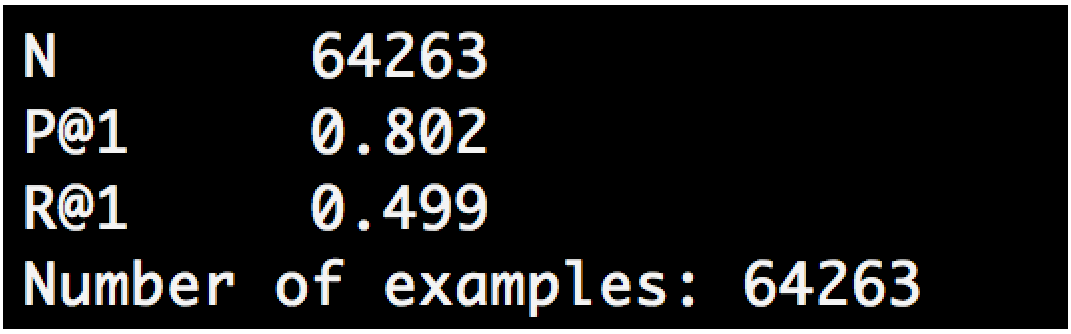
> *./fasttext supervised -input complaints.processed.train.txt -output model_complaints_processed -epoch 25*



```
Read 55M words
Number of words:  114973
Number of labels: 23
Progress: 100.0% words/sec/thread: 1491647 lr:  0.000000 loss:  1.054650 ETA:   0h 0m
```

check the model effectiveness now.

> *./fasttext test model_complaints_processed.bin complaints.processed.valid.txt*

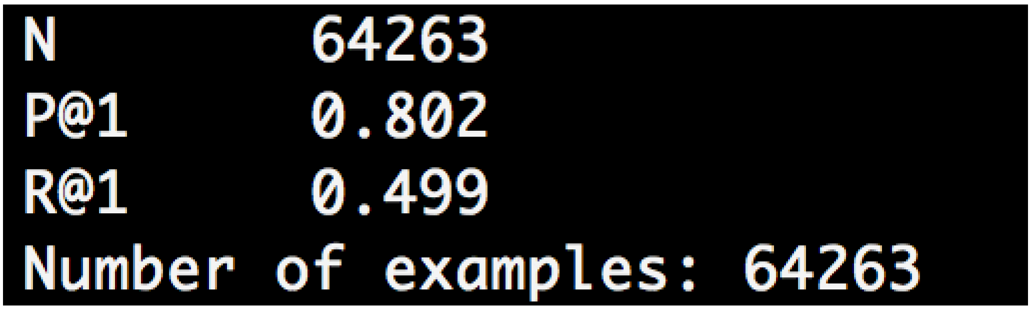


```
N       64263
P@1     0.802
R@1     0.499
Number of examples: 64263
```

Precision increased from 0.793 to 0.802 about 1%

Now let's change the learning rate. A learning rate of 0 would means that the model does not change at all, and thus, does not learn anything. Good values of the learning rate are in the range 0.1 to 1.0

> *./fasttext supervised -input complaints.processed.train.txt -output model_complaints_processed -lr 1.0*
>
> *./fasttext test model_complaints_processed.bin complaints.processed.valid.txt*

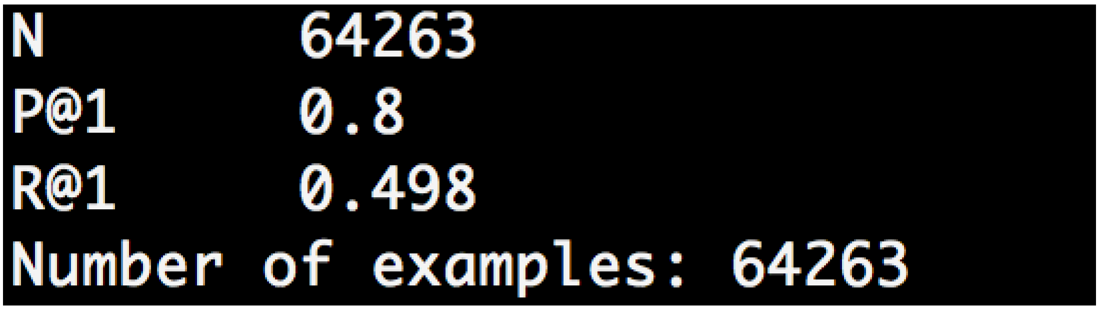


```
N         64263
P@1       0.802
R@1       0.499
Number of examples: 64263
```

There is no change.

Now lets try both epoch and learning rate.

> *./fasttext supervised -input complaints.processed.train.txt -output model_complaints_processed -lr 1.0 -epoch 25*
>
> *./fasttext test model_complaints_processed.bin complaints.processed.valid.txt*

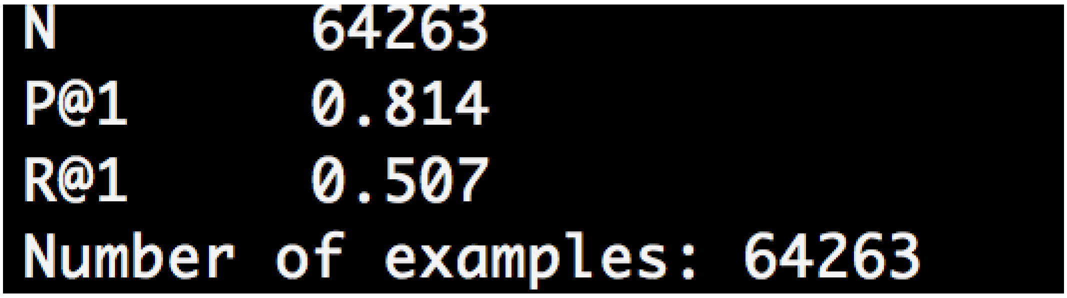


```
N         64263
P@1       0.8
R@1       0.498
Number of examples: 64263
```

Looks like not a good attempt. Precision went down. Now revert back to just epoch 25.

Finally lets try with word n-grams.

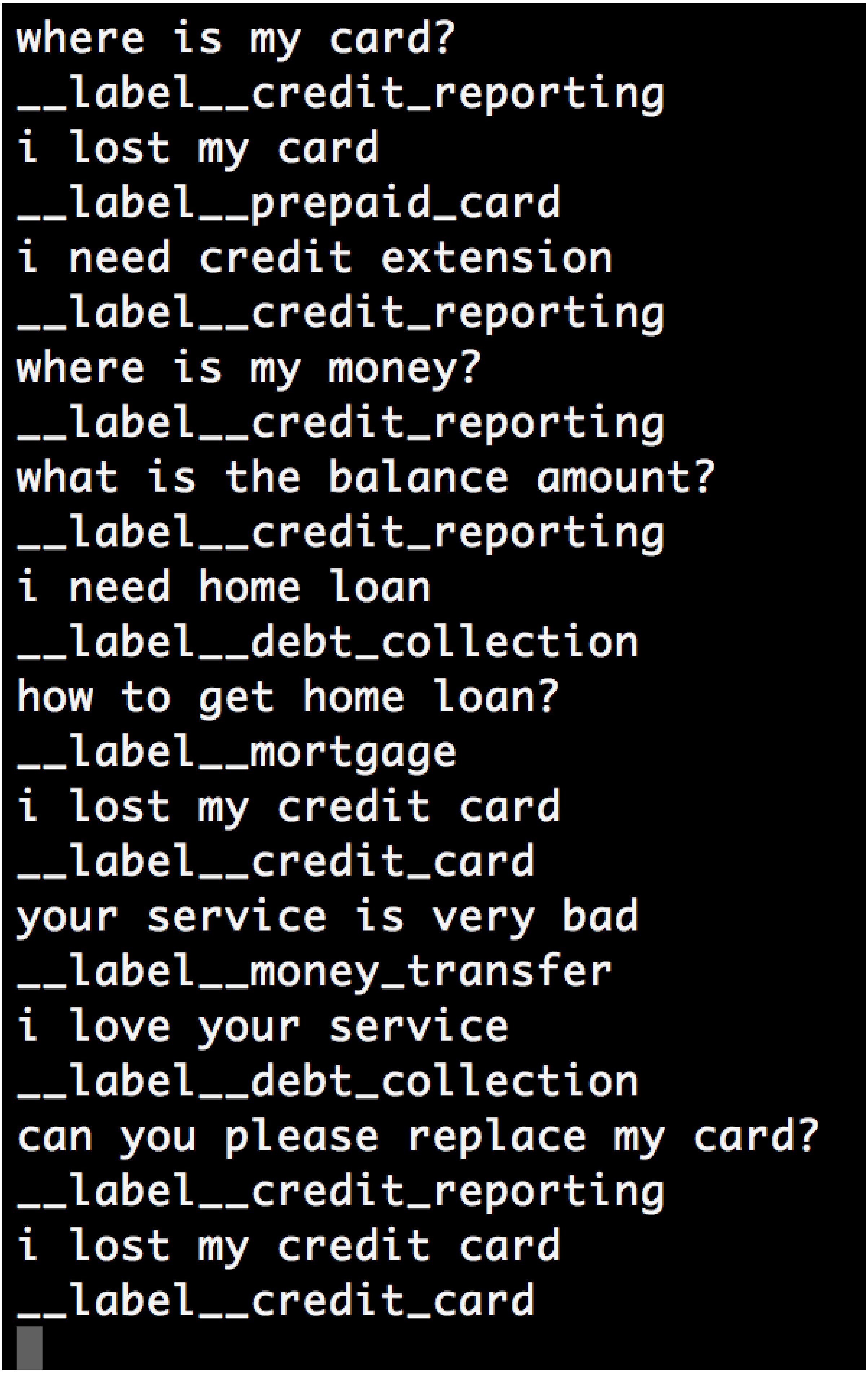
> *./fasttext supervised -input complaints.processed.train.txt -output model_complaints_processed -epoch 25 -wordNgrams 2*
>
> *./fasttext test model_complaints_processed.bin complaints.processed.valid.txt*



```
N         64263
P@1       0.814
R@1       0.507
Number of examples: 64263
```

Precision is now up from 0.802 to 0.814. This is 1.5% up. Lets try some manual examples on this model.

./fasttext predict model_complaints_processed.bin -



```
where is my card?
__label__credit_reporting
i lost my card
__label__prepaid_card
i need credit extension
__label__credit_reporting
where is my money?
__label__credit_reporting
what is the balance amount?
__label__credit_reporting
i need home loan
__label__debt_collection
how to get home loan?
__label__mortgage
i lost my credit card
__label__credit_card
your service is very bad
__label__money_transfer
i love your service
__label__debt_collection
can you please replace my card?
__label__credit_reporting
i lost my credit card
__label__credit_card
```

Not fully accurate. Well…. machine will learn with the time and more real time scenarios. When i get time, i will try removing the stop words and test the model.

Thats all folks. If you liked the article, please share and dont forget to clap.

Thanks to Sunil M for helping me with beautiful Python code.

Machine Learning Fasttext Text Classification