# Classify posts as likely to have few or many comments

(Project for the Natural Language Process Course, Fall 2016)

Shoaib Haque Khan

Date: 12-05-2016

## Abstract:

In the social media and other open forums, predicting which post that will have more comments than others can be interesting to an individual from both pleasure and financial point of view. Our goal was to create an intelligent system that can predict which post will have more comments than the others. For this purpose we first used language models at the word level, with different configurations, only. Then we used Naive Bayes and Neural Networks along with the existing language models to classify posts which will have more comments than others. We have been able to produce better F1 measures with our models than our baselines.

## 1. Introduction:

We wanted to predict which post in a forum will have more impact and as a result will draw more comments than others. For example, for two posts with same core content, but different structure – language, posting time and sub-forum wise – may have different response from the community. This can have good market value for product advertisement or even for personal purposes. Our original hypothesis was that the post that will make higher impact, and as a result will have more comments, depend on the language and grammar, as well as the location where it is posted and the person who posted it. During this study we actually incorporated another feature – the time of the post – for consideration, and have decided that the author of the post is not significant for a large forum like reddit. This procedure is described in the following sections.

For our purpose we used stand-alone language models, and then language models along with two different classifiers. We observed that using the classifiers, more precisely the neural network, increases the performance of our model. We obtained weighted F1 measures of 0.72, 0.73 and 0.73 for three different experimental setups, compared to F1 measures of 0.27 and 0.71 for two different baselines over the whole dataset. We also calculated the accuracies and obtained up to 0.78 with our classifiers, whereas the baselines have accuracies of 0.50 and 0.81, this is due to the fact that our dataset is unbalanced and the second base line we are considering uses that weakness of the dataset to classify all the instances to one single class to obtain a good accuracy, though it gives lower F1 measure than our best classifier.

In section 2, we describe how the dataset was collected and the preprocessing done on the dataset.
In section 3, we describe the experimental design and show the results.
In section 4, we discuss implications of our results, and which part of the proposal we covered more than we intended to and which part we didn't – and why.
In section 5, we present a conclusion and probable future works.

## 2. Dataset:

From [1] we got the reddit data set. As this is a large enough dataset to analyze, we didn't look for any Facebook or Twitter dataset. We only obtained the part of the data that contains the posts along with some attributes for each post separated by commas. A screen shot what the  of what the dataset looks like is attached at the appendix section. From this large dataset, to extract the dataset we need, we followed the following steps:

1. As we are interested about predicting the ability of the post to draw comments before it is

posted, we considered the attributes that can only be fixed before posting a certain post. So we only considered the following attributes: *created_utc* (creation time of the post), *selftext* (text of the post), *subreddit* and *title* (title of the post). We also extracted the number of comments (*num_comments*) to compare it with the threshold to decide whether a post will be considered as having more comments or not. The selection procedure for the threshold has been described the the experimental design section. We also didn't consider the name of the author as an attribute for this experiment as we empirically decided that for a large forum like reddit, where potentially thousands of posts are being created every few minutes, the author name will not be a valid feature to consider for our purpose.

2. As we want to use Natural Language Processing procedure for our purpose, we then considered the posts that have some text, i.e., the *self_text* field is not blank. Further, we chose to have only the top 11 most popular subreddits depending on the number of posts having unblank self_text. So we only chose the following subreddits: *AskReddit*, *leagueoflegends*, *Fireteams*, *trees*, *Dota2Trade*, *friendsafari*, *reddit.com*, *tf2trade*, *circlejerk*, *GlobalOffensiveTrade* and *gaming*. For creation of language models, further classifiers and testing we considered the combination of the fields *self_text* and *title* together, rather than individually. Additionally, for our analysis we only considered posts that have been written using roman alphabet.
3. So we ended up with a dataset of having 2430865 individual posts. We randomized the sequence of the posts using the unix command “shuf”. We then divided the dataset into two equal parts and three equal parts. We did most of our experiments on the dataset having two equal parts. We once tested with the dataset of three parts to observe the affect of using less data for training the language models.

## 3. Experimental Design and Results:

**Threshold:** First, we selected the threshold for comments. For this purpose we calculated the mean and the median of the number of comments in out dataset. We got that the mean was 9.66308805824 and the median was 3.0. Choosing the threshold to be 3.0, the dataset will be divided almost two equal parts of being in the positive and negative class, i.e. having comments more than the threshold and not – respectively. Whereas, choosing the threshold to be 9.0, a real number, would divide the dataset into two parts of having approximately 20% in the positive class and approximately 80% in the negative class. We empirically decided that 3 is too small to be a good threshold to indicate a post having “more” comments. So we decided to set the threshold at 9.0, Post having comments more that or equal to 9 are set to be in the positive class and the rest are set to be in the negative class.

**Baseline:** Secondly, we set the baseline. We considered two baselines. First one is random – a classifier without collecting any information sets 50% of the dataset to be in the positive class, and 50% in the negative class, which gets an accuracy of 50.00 and an F1 measure of 27.90. Second one is a special case where the classifier classifies all the instances as being negative and thus gets a good accuracy of 80.65, but an F1 score of 0 for the positive class as shown in the *Table 1*.

| | | **TP** | **TN** | **FP** | **FN** | **Precision** | **Recall** | **F1** | **Weighted F1** | **Accuracy** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Baseline 1** | **Positive** | 235199.5 | 980232.5 | 235199.5 | 980232.5 | 0.19 | 0.50 | **0.27** | **0.27** | **0.50** |
| | Negative | 980232.5 | 235199.5 | 980232.5 | 235199.5 | 0.50 | 0.19 | **0.27** | | **0.50** |
| **Baseline 2** | **Positive** | 0 | 1960465 | 0 | 470399 | 0 | 0 | **0** | **0.71** | **0.81** |
| | Negative | 1960465 | 0 | 470399 | 0 | 0.80 | 1.00 | **0.89** | | **0.81** |

*Table 1: The baseline classifiers. (TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative)*

**Experimental design:** Then, for experiments, we first used half of the dataset to train the language model for both positive and negative classes. Then using the language models, we predict the probabilities of each instance being in the positive or in the negative class, and measured the precision, recall, F1-measure and accuracy.

We then used the *sklearn* [2] python library to create two classifiers, one using Naive Bayes algorithm and other using Neural Networks using the following parameters: alpha = 1e-05, batch_size = 'auto', hidden_layer_sizes = 100, learning_rate_init = 0.001, max_iter = 200, shuffle = True, validation_fraction = 0.1. The features used as inputs to the classifiers are: the probabilities from the language models, the post creation time, the length of the title, the length of the self text and the name of the subreddit.

| | | | **Precision** | **Recall** | **F1** | **Weighted F1** | **Accuracy** |
|---|---|---|---|---|---|---|---|
| **All tokens** | Language model only | Positive | 0.29 | 0.87 | 0.44 | 0.61 | 0.57 |
| | | Negative | 0.94 | 0.50 | 0.65 | | |
| | Naive Bayes | Positive | 0.19 | 0.97 | 0.32 | 0.11 | 0.21 |
| | | Negative | 0.89 | 0.03 | 0.06 | | |
| | Neural Network | Positive | 0.28 | 0.16 | 0.20 | **0.73** | **0.76** |
| | | Negative | 0.82 | 0.90 | 0.86 | | |
| **Verbs and nouns (trigrams)** | Language model only | Positive | 0.30 | 0.89 | 0.45 | 0.62 | 0.58 |
| | | Negative | 0.95 | 0.51 | 0.66 | | |
| | Naive Bayes | Positive | 0.49 | 0.03 | 0.06 | **0.73** | **0.71** |
| | | Negative | 0.81 | 0.99 | 0.89 | | |
| | Neural Network | Positive | 0.33 | 0.51 | 0.40 | **0.73** | **0.65** |
| | | Negative | 0.86 | 0.75 | 0.80 | | |
| **Verbs and nouns (unigrams)** | Language model only | Positive | 0.25 | 0.94 | 0.40 | 0.47 | 0.45 |
| | | Negative | 0.96 | 0.33 | 0.49 | | |
| | Naive Bayes | Positive | 0.30 | 0.63 | 0.41 | 0.69 | 0.65 |
| | | Negative | 0.88 | 0.66 | 0.75 | | |
| | Neural Network | Positive | 0.23 | 0.11 | 0.15 | **0.72** | **0.75** |
| | | Negative | 0.81 | 0.91 | 0.86 | | |
| | | | | | | | |
| **Test with smaller training dataset for the language models (All tokens)** | Language model only | Positive | 0.18 | 0.89 | 0.29 | 0.30 | 0.30 |
| | | Negative | 0.90 | 0.18 | 0.30 | | |
| | Naive Bayes | Positive | 0.16 | 0.99 | 0.28 | 0.01 | 0.16 |
| | | Negative | 0.75 | 0.01 | 0.01 | | |
| | Neural Network | Positive | 0.23 | 0.13 | 0.16 | **0.76** | **0.78** |
| | | Negative | 0.84 | 0.92 | 0.88 | | |

*Table 2: Results for the different experimental setups.*

First, we used all the tokens in the training dataset to train the language models, trigrams for all the tokens, and test on the rest of the data. We got an accuracy of 57% and F1 measure also close to 0.44

for the positive class and 0.65 for the negative class [table 2]. We then again divided the second half of the dataset into two parts, using the first half this to train and the second half to test, to classify using the Naive Bayes and the Neural Network classifiers mentioned above. We got the accuracy of 21% and 76% respectively,  F1 measure for the positive class 32 and 20 and for the negative class 6 and 86 respectively.

Then, we did the same procedure for trigrams of verbs and nouns only, and then for the unigrams of verbs and nouns only. We used the nltk [3] python library for tokenization purposes, and for extracting the nouns and verbs from the sentences in the dataset. Further, we tested the effect of decreasing the dataset for training the language model, by doing so, increasing the data for training the Naive Bayes and Neural Network models. Here, one third of the dataset was used for creating language models, one third for training the other two classifiers and the rest for the final testing. We present the results for all the experiments in the table 2.

## 4. Discussion:

We have considered only the language models at the word level, not at the character level, as we have empirically decided that language models in the character level are more suitable for the purpose of language detection, whereas for our purpose language models at the token level makes more sense. We also didn't consider the chunking concepts for our project, as we now consider that implementing the chunking concept to obtain our objective can be considered a project on its own – rather than being a small part of this project.

The results [table 2] show that using the language model only, we can obtain F1 measure of up to 0.62 and accuracy of 0.58, using the trigrams of verbs and nouns. However, better results can be obtained by combining this with other classifiers like Naive Bayes and neural networks. Using Naive Bayes, we get F1 measure of 0.73 and accuracy of 0.71 when the language model consists of  trigrams of verbs and nouns. However, it gives us very poor performances for the other two language models. Where as, neural networks give us very consistent performance by giving us F1 measures of 0.73, 0.73 and 0.72, and accuracies of 0.76, 0.65 and 0.75 for the three setups respectively.

Further, when we reduced the amount of data for training the the language model and used the excess amount of data to train the neural networks and Naive Bayes, the performance of the language model decreased to F1 of 0.30 and accuracy of 0.30 – which is intuitive, as we have less data for the language model. However, the performance of the Naive Bayes also decimated, which indicates that this classifier largely depends on the language model and more importantly can only be helpful in specific experimental setups. The neural networks now give us even better F1 measure of 0.76 and accuracy of 0.78.

## 5. Conclusion:

From our experiments we conclude that using neural networks along with language models, and keeping larger portion of the data for training neural networks will give us much better performance. We think that as the length of the posts are sometimes too short for a language model to analyze, we couldn't get better performance with the language models only, than we have got here. However, given that we can move ahead with this study, we may want to extract more features like keywords and sentiments of the authors and combine that with the popularity of that word or topic during that month or week to train our classifier, which may increase the performance of classifier.

**References:**

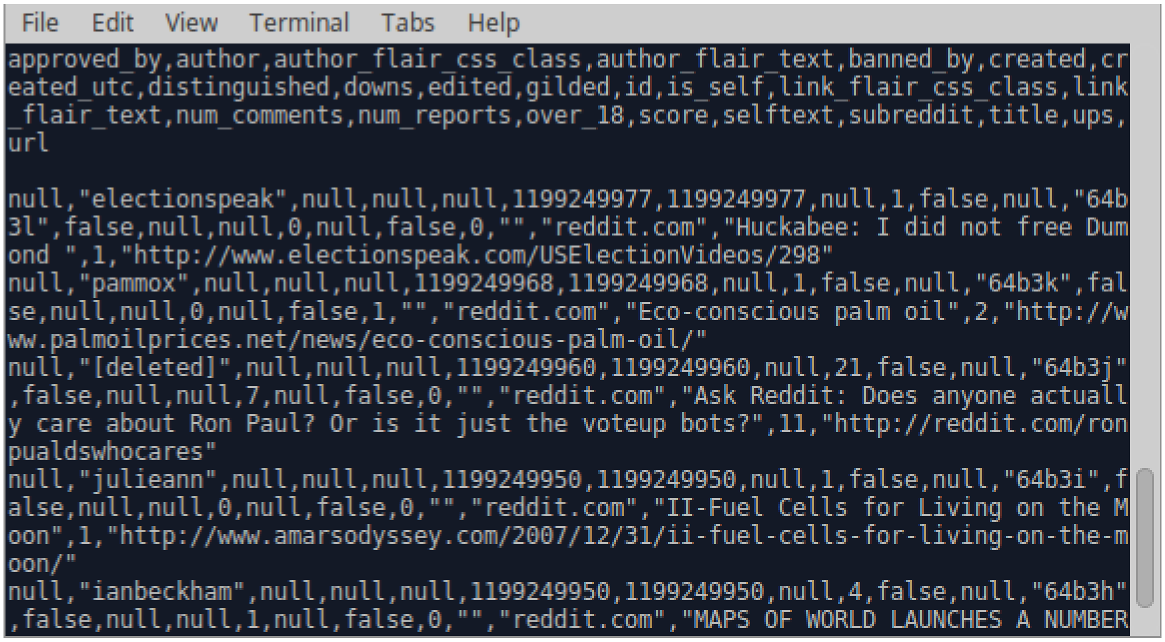
[1] Reddit dataset. Retrieved from https://github.com/reddit/reddit

[2] Sklearn python library. Retrieved from http://scikit-learn.org/stable/

[3] NLTK python library. Retrieved from http://www.nltk.org/

## Appendix:

### a. Few lines from the dataset:



```
approved_by,author,author_flair_css_class,author_flair_text,banned_by,created,cr
eated_utc,distinguished,downs,edited,gilded,id,is_self,link_flair_css_class,link
_flair_text,num_comments,num_reports,over_18,score,selftext,subreddit,title,ups,
url

null,"electionspeak",null,null,null,1199249977,1199249977,null,1,false,null,"64b
3l",false,null,null,0,null,false,0,"","reddit.com","Huckabee: I did not free Dum
ond ",1,"http://www.electionspeak.com/USElectionVideos/298"
null,"pammox",null,null,null,1199249968,1199249968,null,1,false,null,"64b3k",fal
se,null,null,0,null,false,1,"","reddit.com","Eco-conscious palm oil",2,"http://w
ww.palmoilprices.net/news/eco-conscious-palm-oil/"
null,"[deleted]",null,null,null,1199249960,1199249960,null,21,false,null,"64b3j"
,false,null,null,7,null,false,0,"","reddit.com","Ask Reddit: Does anyone actuall
y care about Ron Paul? Or is it just the voteup bots?",11,"http://reddit.com/ron
pualdswhocares"
null,"julieann",null,null,null,1199249950,1199249950,null,1,false,null,"64b3i",f
alse,null,null,0,null,false,0,"","reddit.com","II-Fuel Cells for Living on the M
oon",1,"http://www.amarsodyssey.com/2007/12/31/ii-fuel-cells-for-living-on-the-m
oon/"
null,"ianbeckham",null,null,null,1199249950,1199249950,null,4,false,null,"64b3h"
,false,null,null,1,null,false,0,"","reddit.com","MAPS OF WORLD LAUNCHES A NUMBER
```

*Figure 1: First few lines of the reddit dataset*