Replicating fastText

Arpan Rajput Akshit Tanwar Parth Kanubhai Patel

November 6, 2020

Abstract

Reproducible research put down to the notion of replicating the results of an academic research paper using its code and data to create a new work based on it. In current times Natural language processing is largely an experimental field, and how persuasive this technology is based on the reproducibility of these experiments. These experiments lay foundation for carrying out an empirical research. This when combined with theoretical analysis serves a much larger purpose of creating and improving technologies for the betterment of mankind. The project focuses on improved reporting and results of the experiments that are already conducted.

1 Replication of Original Work

Replication is the ability to re-run an experiment or a set of experiments that allow us to obtain the same results when provided the same computational environment. The resources are made available to us i.e. code and data so that we can replicate the results and add to the empirical investigation of the original work. The success of the investigation concludes its validity and encourage us to extend the work. Before proceeding further, we must keep in mind the changes that needs to be done to make sure the experiments do not fail because of any computational environment change. In this project we are replicating text classification which uses fastText as the code and the tutorial data as the original data. The project focuses on comparing the results obtained on the original data with the results that we get using the newly created data.

1.1 Issues with the original work.

The task that we have picked up is sentiment analysis. It is the process of determining whether a piece of writing is positive, negative, or neutral. It under the field of text classification which aims to assign documents to one or multiple categories. The original work is carried out on the data i.e. collection of stack exchange questions about cooking. The data is multi-labelled and fastText classifier recognise the topic of these questions automatically. Running our test data on this would give us faulty results as the tags are not in form of positive, negative, or neutral. The second issue is constraint on the format of the data that fastText accepts. In the original data each line of the text file contains a list of labels, followed by the corresponding document. All the labels start by the __label__ prefix, which is how fastText makes a distinction between a label and a word.

1.2 Resolving issues

- The first issue is addressed by replacing the stack exchange dataset with twitter US airline sentiment dataset. The data originally came from Crowdflower's Data for Everyone library. It is a sentiment analysis job about the problems of each major U.S. airline. Twitter data was scraped from February of 2015 and contributors were asked to first classify positive, negative, and neutral tweets. The data is in a csv file and available free on GitHub. This data is used to train the fastText which enable us to carry out our task of sentiment analysis. The data is available on `https://www.kaggle.com/crowdflower/twitter-airline-sentiment?select=Tweets.csv`
- The second issue is resolved by formatting the data. As mentioned, the above data is available in form of csv file which includes multiple parameters in different cells of the sheet. This data is formatted and is made readable by fastText. Below is the example of the required format for tweets and is transformed as such :

```
__label__POSITIVE congratulations you played very well yesterday.
__label__NEGATIVE disappointing result today.
...
```

- Lastly to perform a high performing model we need to train our model on a high performing data i.e. we must clean up the data and make sure that the data is something close to proper English. This is described in detail in the next section.

2 Construction of new data

This section of the report is further divided into three parts i.e. one per team member in which there is a description of data source, properties of the data and amount gathered. The data was gathered from twitter and reddit. The three datasets that we made are Tweets from the handles 'US Politics Polls' and 'Crunchyroll'. The third dataset was acquired from reddit , specifically from the subreddit 'Bitcoin'. The process of gathering all the three datasets is given in the following subsection. The cleaning of all the data required the same effort. What we did in order to clean the data was to write a piece of code common for all the three. Following is all the steps we took in order to clean .

Cleaning Tweets: The next essential task is cleaning tweets for sentiment, to obtain good results we must make sure data is free of anything which is difficult for machine to understand. I have inculcated some popular cleaning techniques. First one includes stemming. The goal is to reduce inflectional forms to a common base form. Next is removing stop words, these can be imported from python nltk library. Stop words add noise to the data. As we can see in the example above twitter API returns HTML characters, we need to escape them. Removal of hashtags/ accounts, web addresses, punctuation and emojis/smileys are essential. Converting everything to lowercase avoid case sensitive issues. Below is the example of a raw tweet and clean tweet for comparison. Below is the raw tweet with hashtags/accounts and web links, whereas the above is the clean tweet i.e. data ready to be fetched into the classifier.

```
' your airline is awesome but your lax loft needs to step up its game for dirty tables and floors '
```

```
df_check.iloc[73].text
```

```
'@VirginAmerica your airline is awesome but your lax loft needs to step up its game. $40 for dirty tables and floors? htt
p://t.co/hy0VrfhjHt'
```

Just like the cleaning of the data the result of all the three datasets was obtained based on the same metrics. Result is interpreted in form of precision and recall values. Precision is explained as the number of correct labels among the labels predicted by fastText. Recall is the number of labels that successfully were predicted, among all the real labels. Therefore, more the precision and recall value the better performing the model is. For a clear understanding of the concept, see the

2.1 Collecting tweets from the twitter handle 'US Politics Polls' using Twitter API.

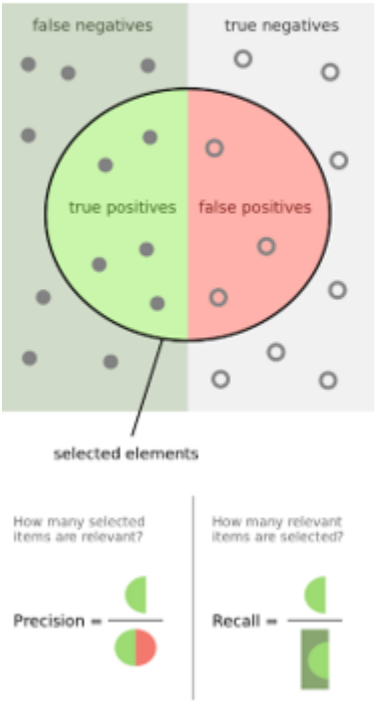
Making use of twitter's huge database makes it one of the most sought-after destinations of opinion data, in my case focussing the current US presidential elections. This has been made easy by tweepy. Tweepy is a python library for accessing the twitter API. It includes a set of classes that deal with open authentication, HTTPS requests and much more. Keeping in mind the primary goal of the task i.e. sentiment analysis I picked 200 recent tweets from the twitter handle 'US Political Polls' The tweets include opinions expressed by general public about the US elections and the two presidential candidates - Donald Trump and Joe Biden. Below is the snipped of the raw tweets extracted from the twitter handle.

Out[10]:

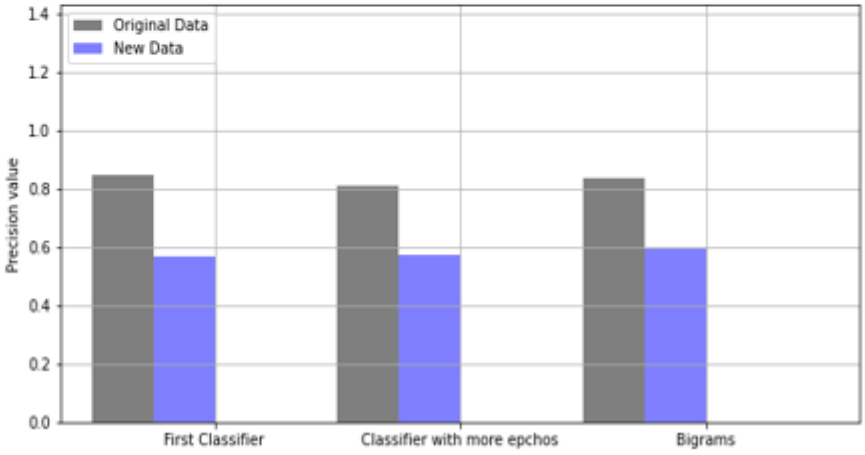
	text
0	😢 Good night everyone https://t.co/STrDTOECl8
1	MARICOPA DUMP NOW https://t.co/57nndQJWMl
2	Clap if you care https://t.co/qdwzFNeuhY
3	@jarriae @CooperCodes We know that but WSJ see...
4	@CooperCodes When did they call it?
...	...
195	We were telling y'all about this hours ago 🌚 h...
196	Yes, he is https://t.co/4fUgrDPU8Q
197	Pennsylvania, 85% in:\nTrump 51.6%\nBiden 47.1%
198	Georgia, 94% in:\nTrump 50.0%\nBiden 48.8%
199	RT @USPoliticsPoll: There's been a disproporti...

The raw tweets are conveying different opinions of different users. But we need tweets categorisation here for the specific use of sentiment analysis. This is achieved via text annotations. Sentiment annotations include polarity, feelings, emotions, and opinions. It is opinions we are interested in. Opinions can be recorded using surveys. Keeping in mind the goal of the task, to get the data annotated required me to conduct a survey. When conducting a survey, we must make sure the survey is statically significant and not biased because it does include human intervention. For this task I picked multiple choice question which included the category positive, negative, and neutral using Qualtrics. It is an online survey platform to build survey and collect data. To even out the biasness, the survey was forwarded to 5 different people and their responses were recorded. Next work is conducted manually and for each tweet maximum recoded option was chosen for the final text annotation. As in the example below :

```
Because they're trying not to get shot by Trump supporters https://t.co/NJS5zGykyB        negative
Because they're trying not to get shot by Trump supporters https://t.co/NJS5zGykyB        negative
Because they're trying not to get shot by Trump supporters https://t.co/NJS5zGykyB        negative
Because they're trying not to get shot by Trump supporters https://t.co/NJS5zGykyB        negative
Because they're trying not to get shot by Trump supporters https://t.co/NJS5zGykyB        neutral
```



Model	Original Data	New Data
First Classifier	0.845	0.565
Classifier with more epchos	0.810	0.570
Bigrams	0.835	0.595



Result : If we look at the table and graph, we see that in both the cases with a few steps, we are able

to significantly improve our precision score. In our original dataset precision score is increased moving up from a simple classifier to applying bigrams. Same is the case with New dataset. Although there is some difference in the precision values, but it is possibly due to sample size or human annotations. Hence, we can say that we have successfully replicated the results.

2.2 Construction of new data using Reddit

My new dataset is called Bitcoin Posts. My new dataset is derived from subreddit called "/r/Bitcoin". This dataset contains 400 posts on bitcoin, username, post data, comments, and amount of vote on the posts. Snippet of it stamped below in Figure. To extract data from Reddit, I have used software called Parse hub. It is a free web scrapping tool. We just must enter the website URL from which we have to scrap our data. Some of its features are that it is a machine learning relationship engine which understand the hierarchy of elements and extract automatically for you to download in in file format like JSON, CSV and google spreadsheet. Also, it provides API and web-hooks. By which we can integrate our extracted data anywhere. URL: `https://www.parsehub.com/`

Pre-processing and data annotations using Qualtrics survey: As you can see in the data there are many posts with URL, emojis, hashtags and unnecessary symbol, which are basically noise for our project requirements. To clean that up I have used Pre-processing of data and code of that provided in GitHub repo. After cleaning up data, so basically fastText library understand prefix _label_ , so to annotate the label to our data as positive, negative, and neutral. To do this, I have used website called Qualtrics Survey. Which basically a survey site. Where you can create your own survey with your own question, as the one we have in our new data, and we give user three options to choose from positive, negative, and neutral. Its snippet is shown below in Figure 2

	C1-T
	posts_name
1	Bitcoin Newcomers FAQ - Please read!
2	PSA: Ledger Has Been Hacked - Here's What You Need to Know
3	Monday Art - Anna Klepalova
4	Yep, we know this is a promoted ad. But we think this one is worth your time. In
5	Paypal 2018 vs 2020... bullish!
6	⚡Lightning Pool Is Open for Business: Lease Liquidity, Earn Returns, Stack Sats
7	Billionaire Jeffrey Gundlach Changes His Tune on Bitcoin
8	⚡Lightning Pool: A Technical Deep-Dive ⚡
9	Lightning Labs launches liquidity marketplace on the Lightning network
10	Monday Art - Oscar Tuayami
11	How Bitcoin Has Performed Throughout the Pandemic (With Graphs)
12	Crypto Cards
13	6 Years ago I animated Bitcoin Users In A Nutshell, and it's funny to see all th
14	BTC Paywall - Accept Lightning Payments for Wordpress Blog Content
15	Bought some crypto from Paypal and I'm a little disappointed
16	Lightning Labs Releases Channel Liquidity Marketplace
17	Stablewallets with Lightning and sockets
18	Make art. Make gifs. Make a story of birds with r/reallifedoodles arms. Whatever
19	Switzerland approves Gazprombank to offer Bitcoin trading and custody services
20	EU is considering implementing a digital euro

Results: Results for this data were not so great in terms of precision and recall value. Below is the snipped of the values recoded when the three didderent classifier were run on it.

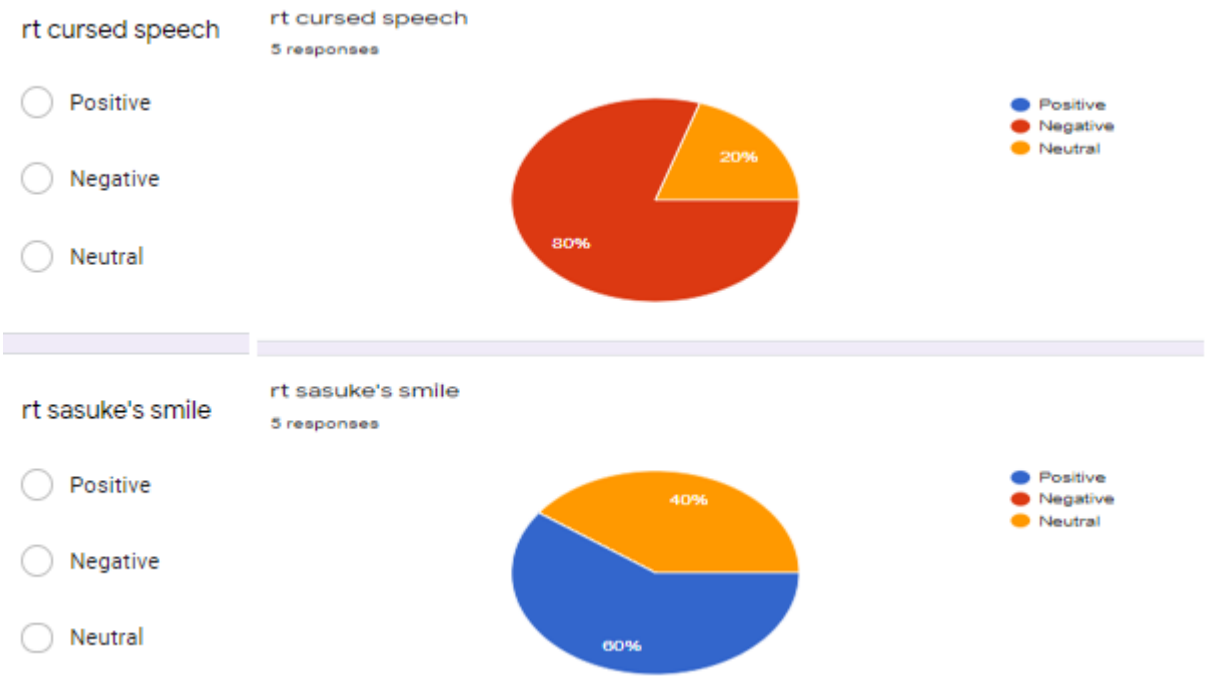
First Classifier	(399, 0.3283208020050125, 0.3283208020050125)
Second Classifier	(399, 0.3283208020050125, 0.3283208020050125)
Bigrams	(399, 0.3032581453634085, 0.3032581453634085)

2.3 Collecting tweets from the twitter handle 'Crunchyroll' using Twitter API.

For my dataset I selected twitter for my source of choice in order to pick data for the sake of sentiment analysis. At first my first thought was to pick a celebrity as a twitter handle. But out of curiosity I went to Crunchyroll's twitter handle. On their twitter page I came across various tweets that were not very common. These tweets consisted of various sentiment related words and various other things such as anime character names. I was interested in seeing if our model will work with all these hurdles. So, the original data set when extracted from twitter is show in the snippet below

	text
0	good night via
1	bakugo on the drums via my hero academia
2	watch how seconds of anime get made watch the ...
3	umaru i think it's time to divide the room
4	reigen's secret weapon via

But as seen in the above snippet , this data is completely unlabelled. This means that there is no indication if the data present is positive, negative or neutral. I used public annotation method in order to get labels for my dataset as not gold standard labels were available. This was achieved by creating a google form consisting all the tweets. This google form sent to 5 different people and the most suitable label was selected by observing the level of agreement between the responses. The Google form represent the level of agreement through a pie chart represent the percentage of responses. The following snippet represents the responses from the Google form.

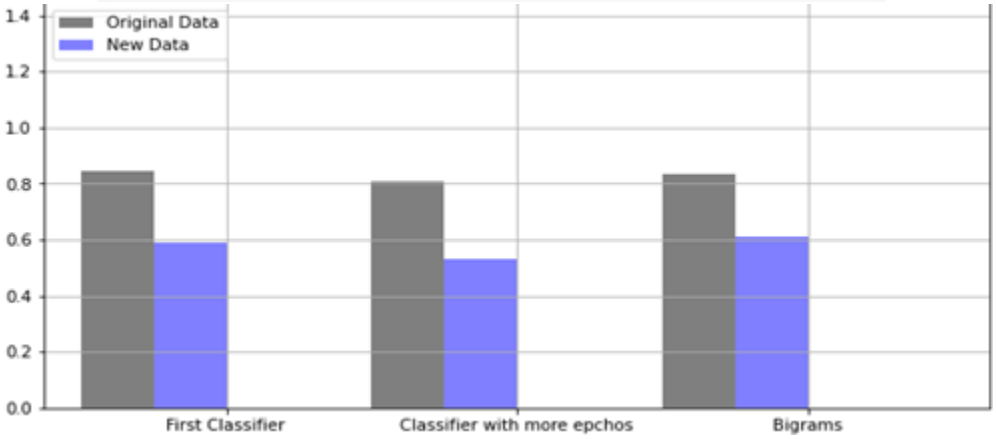


Now the labels were decided on the basis of these pie charts. If we take take the case of the first tweet, we can seen that 1 out of 5 people thought that the tweet was neutral and 4 people thought the tweet was negative. On the basis of this level of agreement the tweet was labelled negative. After this the process was to bring the dataset into a fastText compatible format. This process has been explained above. The final dataset is represented in the snippet below.

```
__label__positive good night via
__label__neutral bakugo on the drums via my hero academia
__label__neutral watch how seconds of anime get made watch the full video
__label__negative  umaru i think it's time to divide the room
__label__neutral reigen's secret weapon via
```

This dataset is completely clean and fully compatible for a fastText model. After this a fastText model was defined and trained on a part of the original data itself as well as the Crunchyroll data. The results obtained were based on precision score and recall value. These results are given below in a mathematical form and a graphical representation is provided as well.

	Model	Original Data	New Data
0	First Classifier	0.845	0.590
1	Classifier with more epchos	0.810	0.530
2	Bigrams	0.835	0.611



As inferred from these results the model well on the part of the original data and has a moderately well performance on the created dataset. The reason behind the lack of performance of the model on the new data might be because the words present in the tweets were not enough for the model in order to predict the labels properly or the labels that were fed in manually were not completely accurate. It was also observed that the best results were obtained when bigrams were combined with increased learning rate and epochs. With the help of the results that were obtained we were able to successfully able to replicate the model on our new datasets.

3 Other Reflections

While working on the dataset we observed that the performance of the model was decent on our datasets but was not up to par to the original dataset. In order to check if the labels were being predicted well enough, we trained the model on a small part of the original dataset, specifically on 200 tweets from the original dataset. Then we tested the model on our datasets and the observed poor results. So, we inferred that the problem resides in the amount of data that we gathered did not have enough information in order for the model to predict the labels properly. We decided to stick with the original results we obtained. Our GitHub link for code and data : `https://github.com/arajp011/Major-Project-COMP8240-fastText-Classification`