Jacob Dineen

Text Mining

8/16/2018

Homework 5

With this assignment, we are tasked with using crowdsourcing methods for bulk labeling/classification of our collected documents from week 1. Amazon Mechanical Turk works by allowing for the completion of business tasks, mainly tasks involving some propensity of human intelligence, through their wide network of high quality, on demand workers at a cost-effective rate [1]. Appendix 1.1 contains results obtained when plugging individual documents into NLTK’s sentiment analysis tool, which maps a vectorized document to one of three classes responsible for describing polarity of an aggregation of text. What I had found through week 1’s analysis was that NTLK predicted the same sentiment as myself 40% of the time. As the confusion matrix and the classification report show, most of the mistakes (over half) were construing negative sentiment as neutral. It also had trouble with some of the positive comments, labelling them as having negative connotations or keywords that led to their classification as being negative in general. With a potential inherent bias pulling results toward neutral, and the kind of mistakes that NLTK appeared to be making, mainly lacking some contextual evidence of the current paradigm of AI, human annotation may be a more viable option for labelling and validation. As I’ve referenced in past assignments, crowd-source, bulk labelling would be beneficial for training sentiment classification model from scratch, as opposed to being reliant on pretrained weights – I say this because it’s entirely possible that pretrained models like NLTK are trained on corpuses that may not fully capture the variance expressed within our test set.

Design of the overall experiment was not incredibly well researched, as I had some trouble navigating and understanding how to set up a task at first. Appendix 1.2 details some of the results from my project, but, essentially, I set it up with default parameters:

Number of workers: 5

Numbers of items: 74

Time allotted: 1 Hour

Total number of completed submissions: 370 (74 \* 5)

Reward per Submission: $0.02

Total Cost: $11.20 \*AMT takes 33% in fees of the top, and the rest is split amongst the workers.

The entire task was completed in about 2 hours. As far as filtering out negligent workers, I simply required AMT expert workers, but did not limit some of the important economic and demographic that the assignment mentions. I’m partially concerned about this because of the potential that we introduce some other kind of bias – Perhaps AI is thought of differently in contrasting geographical settings, which is a concern because the sampled documents appeared to be coming from US citizens. A country that isn’t as technologically progressive might have an inherent neutral/negative stance on Artificial Intelligence, while more wealthy countries with early adoption may feel differently. One thing I found particularly interesting about some of the meta-data of my results was that it only took 5 minutes and 23 seconds, on average, for the whole task. This makes me a bit skeptical because I had some paragraph long documents scraped from Facebook, and I’m not entirely sure one could holistically analyze each of the 74 instances in that timeframe. These will be some things I look for when getting into the statistics below.

To prepare this data, I simply need to write a basic script that transforms/normalizes the Amazon Mechanical Turk Labels, along with my own ground truth label. This becomes a pretty trivial case while using pandas functions. I essentially convert everything that was labeled to a unified structure:

-2 & -1 mapped to 1

0 mapped to 2

1 & 2 mapped to 3

So, my 3 classes will be Negative, Neutral and Positive, ranked in ascending order. I had some issues calculating Kappa when I was using a negative index to define a negative review, so I’m hoping that this alleviates some of that concern. I also added a new feature that computes the element-wise sum of the answers and divides that by the length of the total labels, e.g. I made a new ‘average’ column.

For Spammer detection/identification, I created a new column computing the standard deviation of each row, in reference to the AMT labels. I then sorted by that column in descending order and was able to visualize from within the table what kind of mistakes were being made. As suspected, most of the difference in opinion came with identifying something positive as neutral, or negative as neutral. There weren’t stark differences in opinion on the polarity of each document. It’s also worth noting at this point that my ground truth annotations may be suspect here and there. I have been subjected to the internet paradigm where everyone expects AI to be some evil shift in the status quo, so I may have been a little hard handed in handing out negative labels. I also did some basic Kmeans clustering [Append. 1.4] to bin the AMT labels into different segments. I wanted to do this while holding out the ground truth to see which bins a document would fall into, and how that would differ from my label. Admittedly, this wasn’t the best solution as I was just computing the Euclidean distance between clusters based on the labels. An even more valuable exercise would be to vectorize the documents and see if we can identify any patterns in the respective clusters, perhaps seeing why they fall where they do – We could also concatenate both sets of features and try to cluster to detect outliers. I think that might be a bit beyond the scope of this assignment, so I’ll move into Kappa/distributions.

Distribution amongst annotators and classes:

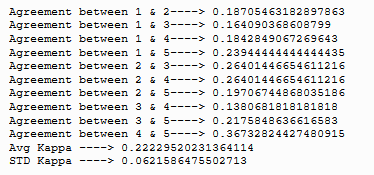
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Class** | **Ann. 1** | **Ann. 2** | **Ann. 3** | **Ann. 4** | **Ann. 5** |
| **1** | 36 | 41 | 38 | 36 | 35 |
| **2** | 14 | 16 | 12 | 27 | 32 |
| **3** | 24 | 17 | 24 | 11 | 7 |

So we can start seeing where some of the disagreement was occurring, at least holistically.

Kappa is an inter-coder agreement metric that helps us to understand the level of agreement between two annotators for classification problems.

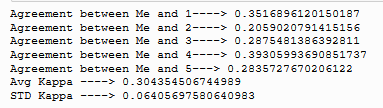
\kappa = (p_o - p_e) / (1 - p_e)[2]

Calculating the pairwise Kappa values for each combination of annotators was a bit arduous, but provided some insight:



Analyzing how this stacks up took a bit of research, but essentially, we are looking at a Kappa of 1 as being complete agreement, and values less than one as being interpreted as having no agreement. NCBI [3] notes 0.40 to 0.60 as moderate agreement. I fall into the bucket of 0.21-0.39, which means that I have minimal agreement between my annotators.

I compared my own results to those of each of the respective annotators brings along the following results:



In conclusion, I don’t think that, for this case, the results proved the worth of crowdsourcing annotation for sentiment analysis tasks. I will reiterate that most of my collected documents were long, and a tad difficult to decipher without context. An easier task would have likely produced better results. I also believe that, while a Neutral label is essential, if it were strictly a measure of polarity the results may have been better (Just positive or negative classes). I enjoyed this assignment, however, and I think the most important take away is the inter-coder agreement (Kappa), as that seems like a statistical measurement that can be used for many other projects where labels are crowdsourced.

References

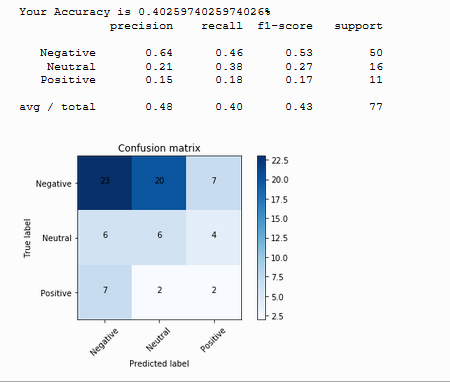
[1] <https://www.mturk.com/>

[2] <http://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html>

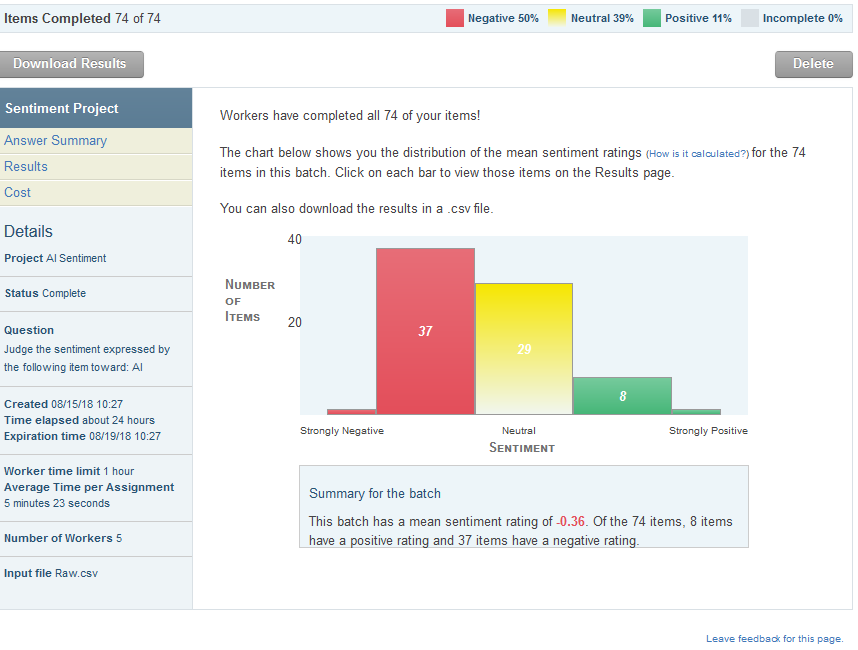
[3] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/>

Appendix

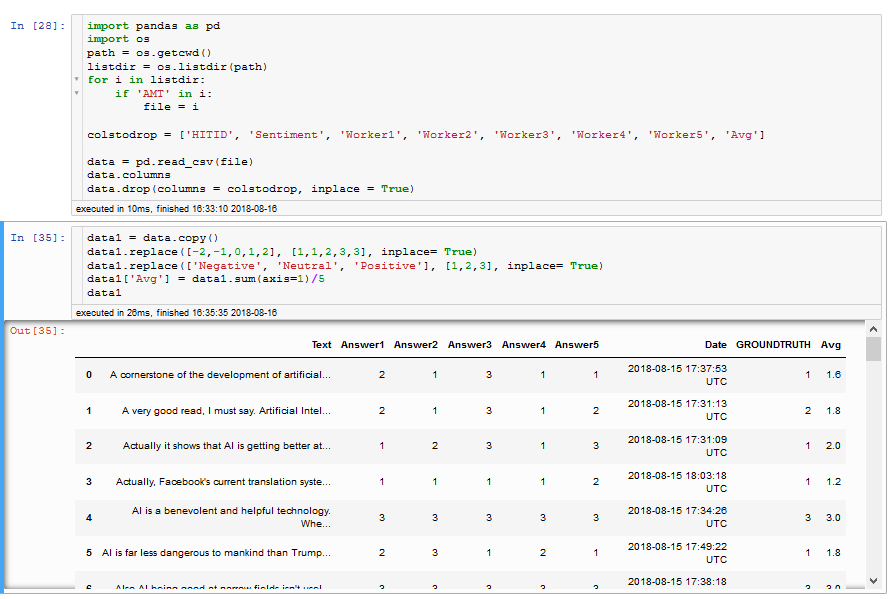
1.1 – Ground truth vs NLTK



1.2 – AMT Results



**1.3 – Reading in Data and Munging**



**1.4 – Some work with Kmeans & Sklearn**

