Classification of Optional Practical Training (OPT) comments using a Naive Bayes classifier

Anand
a3anand@ucsd.edu

Sampath Krishna
svelaga@ucsd.edu

Jorge A. Garza Guardado
jgarzagu@ucsd.edu

Adithya Apuroop K
akaravad@ucsd.edu

ABSTRACT

This project aims to classify the optional practical training comments using a naive Bayes classifier. We demonstrate the effectiveness of the naive Bayes approach and further enhance its performance using a simplified form of an expectation maximisation algorithm. We explore how sentiments change over time, and also provide preliminary results that help in understanding how sentiments vary with ethnicity.

1. INTRODUCTION

OPT is a scheme in which students with F-1 visas are permitted by the United States Citizenship and Immigration Services (USCIS) to work for at most one year on a student visa towards getting practical training to complement their field of studies. On April 2, 2008, the department of homeland security(DHS) announced an extension to the OPT which was passed by USCIS as an interim rule. This rule allows students in Science, Technology, Engineering or Math (STEM) majors, an OPT extension for up to 17 months.

In August 2015, a US federal court gave its verdict on a lawsuit challenging the 17-month OPT STEM extension. The court has decided that the interim rule was deficient as it was not subjected to public notice, comments and opinions and it vacated the 2008 rule allowing the 17-month extension. However, a stay was put in place until February 12, 2016. DHS will have until then in order to take action regarding the fate of the STEM extension program. This rule was open to public comments for a one month duration, ending on Nov 18th. The comments are publicly available at [1].

2. THE DATA SET

2.1 Data collection

Data was collected from the Department of Homeland Security (DHS) web page forum [1] containing at the time, 42,925 comments. This data was obtained over a period of 30 days, ranging from 19th October to 18th November. The DHS web page provides a CSV file containing all user names and comments, but the comments are stored as a web page link. A script was written to download and then parse each web page containing the comment for each user and the resulting data was stored in a JSON file. The data we used contained the fields: *'userName'* , *'comment'*, *'docID'*,*'receivedDate'*, *'postedDate'*

2.2 Dataset Preprocessing

As a pre-processing step, we removed all the punctuations from the words. We also changed all words to lower case, although a more rigorous model could make use of the upper case information to identify stronger sentiments. Finally, all the common stop words were removed as they convey little meaning.

2.3 Dataset Labeling

Since the original dataset is unlabeled, we manually labeled the first 900 comments as *support* or *oppose*. Out of these, the first 600 were used for training, comments from 601-700 constituted the validation set and 700-900 were used for testing . We used validation set to pick the best possible model from a pool of possible models.

2.4 Data visualisation/Exploratory analysis

Figure (see Fig. 1 and Fig. 2) contain the word clouds of the most common words (after removing stop words) on the train data for positive and negative labels. Some of the most commonly found words in supporting comments were {*benefit, support, economy, STEM, international, students, good*} etc meaning that people supporting the OPT extension feel that the extension will benefit the economy and is good for international students. While in opposing comments we found words like {*American, job, worker, student, foreign, program*} etc meaning that people opposing are concerned about the jobs being taken away by the foreign students.

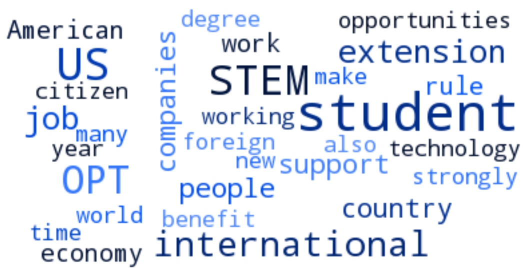

Figure 1: Positive comments word cloud

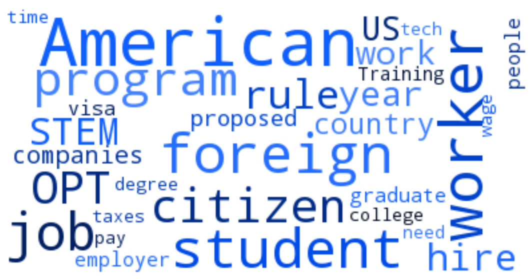

Figure 2: Negative comments word cloud

The distribution of frequencies of words in documents is believed to follow *Zipf's law*. Zipf's law suggests that the frequency of each word is close to inversely proportional to its rank in the frequency table to the power of a where $a \sim 1$. We observe a behavior consistent with the hypothesis from the plot in Fig. 3

$$P \propto 1/n^a$$

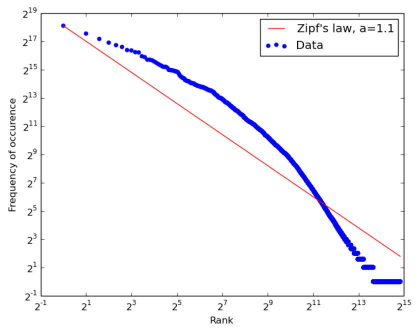

Figure 3: Zipfs law vs frequency of words in the dataset

2.5 Predictive task

Our main goal here is to classify whether a given comment is supporting or opposing. In addition, based on the classifier we obtained, we also examined how the proportions of supporting and opposing reviews varied with time. Finally, we tried to examine the trends on an ethnicity basis. The main idea behind this is to check if the voting pattern supports the hypothesis that most Americans oppose OPT extension, while people from other ethnicity support it.

3. PREVIOUS WORK

Since the data set is recent and relatively small, we didn't find any literature specific to this data set except for [2] which serves as motivation for our analysis in the first place. We read survey papers on natural language processing and algorithms for sentiment detection in similar data sets. We explored the algorithms in [7],[9],[10] for text classification. Apart from the naive Bayes and iterative maximisation approach, we came across other interesting algorithms for sentiment classification like decision trees, artificial neural networks and support vector machines. Decision trees are generally prone to over fitting on the training data and perform well in cases where there is a lot of labeled training data available. We decided to try naive Bayes and clustering as performant classification techniques. We would have liked to explore support vector machines and neural networks as well but time constraints persuaded us to focus our analysis on these two approaches.

4. ALGORITHMS AND MODELS FOR CLASSIFICATION

Broadly speaking, there are two classes of algorithms that could be tried to classify the comment labels - supervised and unsupervised. For unsupervised learning, we tried clustering based on the *tf-idf* features extracted from the text with the Eucledian distance metric.

Hierarchical clustering runs in time $O(n^3)d$, where n is the number of data points and d is the number of dimensions of the feature vector, making it very slow for large data sets. Therefore, we implemented K-means clustering which is much faster. However, no useful clusters were identified and the accuracies were no better than those of a random classifier. This is to be expected because there is no coherent structure across the different comments - they are of varying lengths and contain different kinds of vocabulary to express the same sentiment, thus rendering Eucledian distance as a very bad distance measure.

Naive Bayes performs particularly well for text classification despite the aggressive assumption it makes about independence. The reason for this is thought to be because, although naive bayes fails to produce good estimates of the probabilities, we do not require the absolute values of these, but only the relative ordering to estimate the MAP estimate. Reports by [2] suggested that Naive Bayes indeed performs well on this data set. There are at least two popular versions of naive Bayes - *multinomial* and *Bernoulli*.

Bernoulli naive Bayes makes the assumption that each document belonging to a class contains occurrence of some words that are described by the probability distribution of the words belonging to that class. The Probability of the document given the class can then be modeled by:

$$P(doc|class) = \prod_{unique\ w \in doc} P(w|class) \prod_{w \notin doc} (1 - P(w|class))$$

On the other hand, multinomial naive Bayes assumes that the document of a particular class is generated by the following generative process - First, the length is chosen according to some distribution(which we don't care, as the length does not depend on the class labels). Then, every word in the document is generated by a multinomial distribution over the words belonging to that class. In this case, the corresponding probability can be modeled by:

$$P(doc|class) = P(|length(doc)|) \prod_{w \in doc} P(w|class)$$

We implemented both multinomial and Bernoulli naive Bayes, but we explored only the multinomial version because it was faster, while giving similar results as the Bernoulli version

Multinomial models using just uni-grams, just bi-grams, and using both uni-grams and bi-grams were considered. Initial results showing the performance on the training set and validation set can be seen in Table1

From Table1, we observe that the bigram only model

Model	Train acc	Validation acc
Unigram	98%	86%
Bigram	99.5%	86%
Unigram+Bigram	99.33%	88%

Table 1: Table showing the validation errors on the 3 schemes being considered

possibly over fits on the validation set, therefore we only consider the unigram only model and the unigram+bigram model. While the naive Bayes in itself performs reasonably well, its performance can be boosted by augmenting it with a simple fix.

4.1 Semi supervised estimation

It has been suggested by the authors in [3], that in cases where the number of training examples is small, the performance of the naive bayes classifier can be improved by combining it with an expectation maximization algorithm. In short, the authors suggest to do this :

1. Predict the class probabilities $P(class|data)$ for all examples in the dataset

2. Retrain the model based on *class probabilities* estimated in the previous step

The first step above is an expectation step in disguise, and the second step corresponds to the maximisation. Although the second step requires us to retrain the model based on the probabilities in the previous step, we use a relaxed version of this step as follows:

1. Predict the class probabilities $P(class|data)$ for all examples in the dataset

2. Retrain the model based on *class labels* estimated in the previous step

This algorithm, which we'll refer to as *classification maximisation (CM)* algorithm is a convenient approximation to the more rigorous expectation maximization. What this means, is that we use the predicted labels as the actual labels and retrain the model based on these labels until convergence. These iterations significantly improve the accuracy of the naive Bayes model by incorporating the knowledge from the large pool of unlabeled examples. Refer to Figure 2 to see the performance comparison of the classification maximisation and naive Bayes algorithms. Note that the classification maximisation algorithm achieves significantly better TPR and TNR on the test set as compared to naive Bayes. Note that TNR is a particularly important term to evaluate the performance of the classifier, as the negative examples are relatively rare and one would want to classify them correctly. After all, an "all positive" classifier would achieve an accuracy of about 85% on this dataset.

5. RESULTS AND OBSERVATIONS

From the predictions of the classification maximisation algorithm, we find that approximately 85.17% of the users support OPT extension while 14.83% oppose it.

5.1 Excerpts of comments from the labeled set

There are certain cases where our naive Bayes model fails to predict the sentiment correctly. Consider a false negative classification in our test set :

> *"OPT is helping to find better workers for the jobs, not simply give the jobs to foreigners."*

The classifier recognises the words *jobs* and *foreigners* as predictive of negative sentiment but doesn't notice the negation in the original clause.

Also, a complex viewpoint expressed via contrast and juxtaposition stumps our classifier. Consider the comment :

> *"Admittedly, there are Americans who can not find a job. But there are also foreign students who can not find a job. The majority of US companies already give priorities to US workers. As a result, the unemployment percentage of international students is already higher than that of native Americans. It is unfair to say that more US workers can not find a job. We should compare the percentage instead of the absolute number."*

which actually supports the OPT proposal but is predicted to be a disapproval since the bag-of-words approach lumps *jobs, Americans* and *workers* with negative sentiment. The (+,-) log likelihood is (-307.6, -305.9) : which indicates an edge case for our classifier.

There are a few false positives as well. Let us evaluate a straightforward negative comment which manages to fool the classifier :

> *"I oppose the extension of OPT. Schools, especially public schools, welcome foreign students because they pay high tuition. And then, with extension of OPT, they earn back what they invest and maybe much more. Who is the winner? Obviously, foreign students who get more than they invested, schools which get a lot of money and companies which get a lot of comparable cheap workers. Who is the loser? Obviously not the government, the working Americans are loser, the middle class is loser. They take risk of losing their jobs but they don't get any benefit from having more and more foreign students."*

Here the classifier is incapable of parsing the topic sentence but counts phrases like *welcome foreign students* and *lot of money* towards a false positive prediction. The log likelihood for positive and negative prediction are -505 and -513 respectively. The word *oppose* is present in our list of words which appear only in negative comments. However the frequency is a weak 560 which doesn't sway the negative probability sufficiently.

However, for a majority of comments, the bag-of-words approach combined with iterative maximisation works surprisingly well. We will next look at a few comments drawn at random from the unlabeled dataset and see how our classifier performs.

5.2 Excerpts of comments from the unlabeled set

Here are some comments from the unlabeled set. We mention the log likelihood for positive and negative predictions alongside the comment. A higher log likelihood implies a greater probability for the classification.

Model	Train acc	Validation acc	Test acc	TNR	TPR
naive Bayes(Uni)	98%	86%	90.5%	40.7%	98.2%
naive Bayes(Uni+Bi)	99.3%	86%	88.5%	29.7%	97.7%
CM Algorithm(Uni)	96.33%	97%	95.5%	92.5%	96%
CM Algorithm(Uni+Bi)	96.5%	96%	96.5%	92.5%	97.1%

Table 2: Table showing the comparison of various algorithms

Consider comments like

> *"Please stand up for American citizens and say NO to this travesty."*

with a (+,-) log likelihood of (-67.3, -57.9) : clearly classifying it as a negative comment. True positive comments like

> *"International students bring money, skills and jobs in USA. This rule is not taking away any job from us. In fact because of this rule more and more jobs are being created for American citizen with or without degree in STEM field."*

show a wide margin between the pos/neg log likelihoods of -485.7 and -518.7 respectively.

> *"I oppose the Department of Homeland Security's proposed rule that would expand the Optional Practical Training program. This expansion would allow U.S. tech companies to hire . . . a de facto shadow H-1B program, in violation of Congressional intent."*

(+, -) log likelihood = (-961.6,-819.0). is also classified correctly.

With a test accuracy of 0.965, the classification maximisation approach combined with naive Bayes performs competitively compared to other advanced techniques like neural networks or support vector machines.

5.3 Other interesting observations

We also tried to analyse the data based on the ethnicity of the users. We were curious to know how people supported/opposed OPT based on the country of origin. For this, we have collected the publicly available common first names and surnames of Americans and Chinese ethnicity. We couldn't get the corresponding data for India to perform such an analysis. The results are summarized in Fig 4

It was initially quite surprising that a significant fraction of Americans support the OPT. There might be several reasons for this. Firstly, the database for American names contains many foreign names as well, and this might have created conflicts with true foreigners who supported the extension. Secondly, all American names are not in the database and that might have resulted in some conflicts too. Nevertheless, upon examination of some reviews, we found that many Americans were supportive of OPT due to the positive talent it brings to the country.

We ran the classifier on the entire corpus of around 42,000 comments and plotted the distribution of sentiment over time. The graph [5] suggests that the initial sentiment was overwhelmingly positive with negative comments beginning to trickle in a week after the voting started.

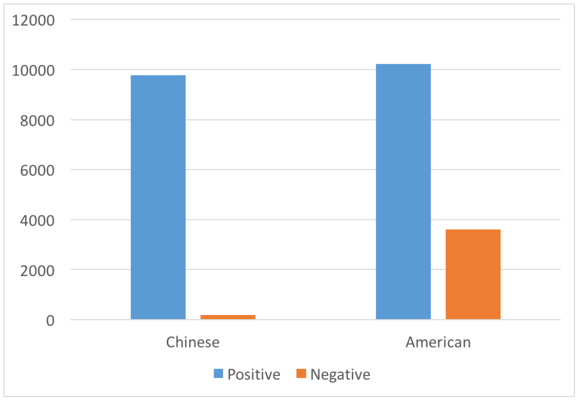

Figure 4: Sentiment breakdown by ethnicity

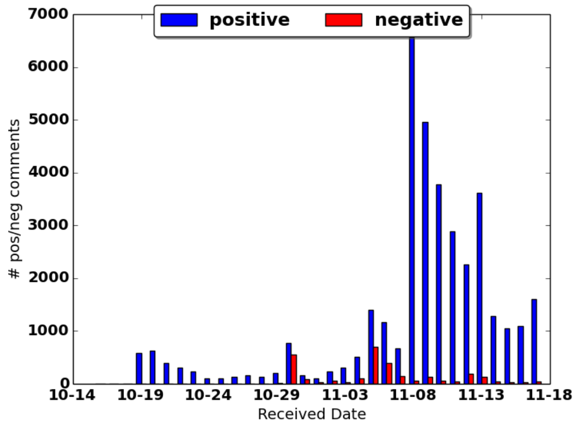

Figure 5: Sentiment breakdown over time

An interesting result was obtained from trying to get the most negative words that do not occur as often or are not even listed in the positive words list. For doing so, the entire dataset was segregated based on the predicted labels, and the frequency of each negative word was subtracted from its matching positive word frequency after normalization. If a negative word occurred a large number of times in the positive word list, it was removed . Resulting words from this list can bee seen in Fig. 6

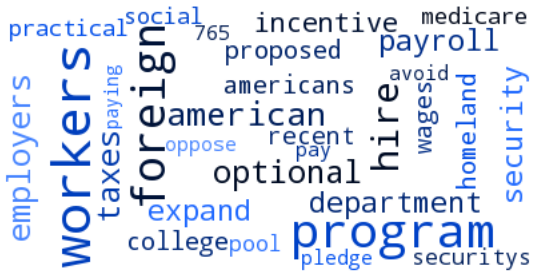

Figure 6: Negative words which do not appear as much in positive comments

From Fig. 6 it can be noticed for example, that words like {*workers, program, foreign, oppose, homeland, taxes, wages, medicare, taxes, paying*} now appear in the negative only words list. These words represent that most people that **oppose** are concerned mainly with **foreign** people stealing jobs from American **workers** in their own **homeland** with this new **program** and foreigners not paying **taxes**. Another interesting thing was to see the number "765" which refers to the form I-765 that needs to be filled when applying for an OPT, and words like *"medicare"*, meaning that people opposing are also somehow concerned with this. For example, consider this comment where words like {*wage, medicare, taxes, pay, social, security*} appear.

"American IT jobs should be done by natural born Americans, not foreigners, who will work for substandard wages and be exempt from the taxes that are paid to help support our economy and social security and Medicare."

Our code is currently hosted at [8].

6. ACKNOWLEDGEMENTS

This project was inspired by a similar analysis done in [2].

7. REFERENCES

[1] http://www.regulations.gov/#!docketDetail;D=ICEB-2015-0002

[2] https://medium.com/@heretic/on-opt-optional-practical-training-10ced7051066#.goc1w933d

[3] Nigam, Kamal; McCallum, Andrew; Thrun, Sebastian; Mitchell, Tom (2000)."Learning to classify text from labeled and unlabeled documents using EM"

[4] http://immigrationgirl.com/breaking-news-on-opt-stem-extension-court-says-uscis-rule-allowing-17-month-stem-extension-is-deficient/

[5] http://vikeshkhanna.webfactional.com/opt

[6] http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1359749&reason=concurrency

[7] http://dl.acm.org/citation.cfm?id=288651

[8] https://github.com/ananducsd/opt-data-mining

[9] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee*, Khairullah khan."A Review of Machine Learning Algorithms for Text-Documents Classification"

[10] http://www.time.mk/trajkovski/thesis/text-class.pdf