CSE 519 : Data Science Fundamentals

Classification of users belonging to eng vs non-eng speaking country based on twitter data

Ali Abbas Hussain
110951463
alhussain@cs.stonybrook.edu

Nidhi Mendiratta
110928535
nmendiratta@cs.stonybrook.edu

Shilpa Gupta
110948405
shilgupta@cs.stonybrook.edu

INTRODUCTION :

Twitter is a widely used microblogging platform that has a lot of content added to it on a daily basis. Analysing this content can give a great deal of trend analysis in a distributed domain. The whole idea of its presence in the Countries all across the world with its user base ranging from people belonging to different walks of life, gives it the benefit of having a diverse content. Exploiting this information base can help in solving various real world problems like sentiment analysis, gender prediction etc. We aim to solve one such problem of classifying the user into English vs Non english country given basic user profile information and his latest tweets.

PROBLEM STATEMENT:

The problem of predicting the location of any user given only specific traits can help in many situations like natural disasters responses or in other use cases like targeted advertising etc. wherein knowing user's location can help in accomplishing a task effectively. For example modifying the advertisements based on a user's location and targeting based on the local language can help in increasing the customer base.

We are aiming to build a classification framework which will classify given set of users into 2 classes. The task at hand is to identify if a user belongs to a english speaking or non- english speaking country given its basic user related features like followers count , friends count etc. and its most recent tweets.

DATA GATHERING :

Data gathering was one of the most crucial steps. For this we used the tweepy library of Python for which we have established authentication token to enable us to download user related information and also their latest tweets. To build an unbiased dataset, we took 5 countries from UK Visa and Immigration list of english speaking countries and 5 non-english speaking countries.

We are only considering tweets written in English alphabet i.e only those scenarios where any user is writing using english alphabets. This condition was necessary because if we consider all the tweets irrespective of language barrier, then it will be difficult to translate those into one common alphabet.

For English speaking category, we took countries like Australia, United States , United Kingdom , Ireland and Canada and 5 non english speaking countries were India, Pakistan, Saudi Arabia, Japan and Brazil. For each of these countries around 10,000 unique users were taken, giving a total of 10k*10 = 1 lac unique users in total.

For any user, we extracted the user related features (explained in detail in features section) and also the tweets written by the user, which we have identified as one of the most important features since there is a lot of information attached to the tweet text spread across the latest 150 tweets from every user.

DATA CLEANING :

Data collected above is in a raw format and there is a lot of scope of data cleaning because there is too much garbage information. For data cleaning the following criterias were considered :

- While downloading users using tweepy, we received lot of duplicate users since tweepy library doesn't return a unique list. Therefore before any processing, we first identified a list of equal number of unique users to ensure that dataset was complete.
- Secondly, we have done a further analysis on the number of tweets per user for our tweet text feature to carefully examine exact how many number of tweets are desired so that we have sufficient information and also to ensure that we are not overfitting the model. (Results section) For this we made sure we only considered those users which have at least 150 tweets.
- Once we had a clean dataset of unique users, we downloaded user related features and the tweets for each user. For cleaning the tweet data, we ensured that the downloaded tweets were not retweets so that only original source of information is taken into count and also that there is no duplicate data.
- From the combined tweet string, we are extracting features like number of urls, hashtags, emojis, tags etc. After that we are cleaning our tweet string to remove all invalid characters except digits and alphabets. Also at this step, we remove all the stopwords to ensure that only required information is passed to word2vec model.
- Also for user related features, we are normalizing the numeric figures and ensuring that outliers are removed effectively so as to ensure that model is built on right set of data range.
- We made sure that the end string combined of all the tweets to be fetched into word2vec model was uniform enough to be trained by our model and also that

there is no loss of information in the data cleaning step. Therefore we followed the approach of first calculating the tweet related features on all the tweets before actually dropping all these extra characters to get a clean dataset for the model.

FEATURES :

We classified our features into 2 categories :

1. User related features :

   Twitter user object comes with various values ranging from everything related to his personal profile like screen name , gender , age , time zone etc. to twitter profile related fields like followers count , friends count , Total number of tweets posted uptil now. Etc. From these set of available features we identified the following ones to be most relevant to our scenario :

- Followers count : Number of people following him/her on twitter
- Friends count : Number of people being followed by the user
- Statuses count : Total statuses posted uptil now
- Average number of tweets per day : Tweeting average i.e on an average, how many tweets are done by user in one day. This is identified by total number of tweets posted till now with the number of days from the time his profile was created till current date.

   Each of the above features signifies profile of any user and also his activity on twitter which can be efficiently exploited in predicting if he/she belongs to an English speaking country or not.

2. Tweet related features :

   A tweet posted by the user has a lot of information which signifies his writing style , the use of slang and other language related information. For each user we collected around 150 latest tweets. From this corpora of tweets per user, we identified the following features:

   - URL count: Number of urls stated by the user in all of 150 tweets.
   - Hashtag count: Total number of hashtags used by the user.
   - Emojis count: Total emojis count in the 150 tweets.
   - Tag count: Total number of tags done by the user for other profiles.
   - Retweet count: Number of retweets occurred for the user on all the

tweets considered.

Above features lay more emphasis on the writing style of any user and we are making this assumption that writing styles are influenced by the country he belongs to.

There are different attribute selection algorithms which help in selecting the significant features and also ranking them based on their influence on the classification model. The whole idea of not considering the irrelevant or redundant features can help in making an efficient model. Therefore, we chose algorithm of correlation based feature selection which helped us in stating that all the above mentioned features were significant in one way or the other and also gave the relative ranking among these features :

Rank	**Feature**
1	Average tweets per day
2	Number of tags
3	Number of hashtags
4	Friends count
5	URL count
6	Number of retweets for the user
7	Total number of tweets done uptil now
8	Number of emoticons
9	Followers count

Apart from the above features, we also considered the combined text string of 60 tweets of each user and this string was our most significant feature for which we created a 100 length vector using the below mentioned word2vec approach.

MODELS :

Word2vec:

Since there was a significant amount of text related information in the tweet text, therefore we adapted the approach of word2vec. We trained the model on a combined corpora of all the words in the tweets for all the users, which is how the lexicon for the model was built.

After creating the lexicon, we transformed our text into 100 length vectors generated from word2vec model which were the modified information carriers for the tweet text.

For training word2vec model , there are different parameters to be specified like :

- **Size :** This parameter is used to specify the vector size to be created for the word. We tried different values of this parameter ranging from 100-200. As we concluded in progress report 1 there was not much difference in the accuracy by increasing the vector size by more than 100. So we kept it 100 for our final analysis.

- **Training algorithm** : In word2vec there are two learning algorithms :
  continuous bag of words (CBOW) and skip gram model : We tried both these approaches for our dataset and as already shown in progress report 1 skip gram approach worked better for our scenario.

- **min_count** signifies minimum frequency with which a word should occur in order for its vector to be generated. For our algorithm , we have taken the min count to be 1. Since we have already removed stop words from the tweets , therefore we are not ignoring any other word because every word carries some information.

Let's say we want to generate a 100 length vector for each user in our dataset. For this, we converted the tweet to lower case first and then took the average of all 100 length vectors of the tweet words which are part of our word2vec corpora.
Let say Word2Vec corpora after the training looks like

hello → [h1, h2, h3 ….. h99, h100]
world → [w1, w2, w3, …. w99, w100]
Our tweet is “Hello good world”
Feature vector of this tweet will be :
[(h1+w1)/2, (h2+w2)/2, (h3+w3)/2, (h4+w4)/2, (h5+w5)/2, ……..,(h99+w99)/2, (h100+w100)/2]
Since “good” is not part of our Word2vec corpora, we are ignoring it.
These generated 100 length vector is part of our final user feature vector along with other user related features.

Support Vector Machine:

We chose SVM model because SVM generally works well in the scenarios of classification problems because of following reasons :

- **Optimum algorithm :** SVM being a classification method works on the principle of fitting a boundary to a region of points which are all alike (that is, belong to one class). The advantage of SVM is that once a boundary is established, most of the training data is redundant and SVM aims to run an optimization scheme to maximize the region between 2 classes.
- **Suitable for large feature vectors** : SVM’s ability to learn is independent of the dimensionality of feature vector and since in our scenario we have a large feature vector therefore svm runs efficiently without the risk of overfitting since in SVM once a fitting hyperplane is found, with incoming data there is not much change in the separating margin.

For the above mentioned reasons, SVM seems to be an ideal fit for our problem and the results obtained can be seen below in results section.

RESULTS :

As described in above sections our feature vector has length of 109 where 9 features are related to user’s profile and rest 100 features coming from the word2vec model by training it on every user’s X number of tweets and later getting a 100 length average vector representation of the tweet text of the user. We wanted to analysis how many tweets of every user we have to take into account in order to get maximum representative 100 length word2vec vector using which we can get maximum accuracy.

To achieve this during data gathering phase we downloaded 150 tweets of every user. If some user did not had 150 tweets we downloaded max tweets whatever was

available.

Now we took different number of tweets into account for training word2vec and generating feature vector. Since out of total unique users (95400) that we identified, around 43000 users had 150 tweets available so we created datasets of 43000 users by taking different number of tweets into account. For example we selected 43000 users which had at least 5 tweets available then we selected 5 tweets of these users randomly, and generated feature vector of each of these 43000 users by training word2vec only on 5 tweets keeping other features of user as it is. Similarly we did the same incrementally every time taking 5 more tweets into account. This way we created 30 data sets each one having 43000 user feature vectors. We trained and tested support vector machine (SVM) model on each of these data sets by 10 folds cross validation technique, (where we divided given data into 10 parts take each one of them as test set once, average out the final accuracy) and observed how the accuracy is changing for different number of tweets that we took into account while creating feature vectors.

- Every data set had same number of users (43000)
- Every data set had around equal representation of both classes (eng vs non-eng)
- All the data sets were train and tested on SVM model with 10 fold cross validation.
- Each time while creating data set we increased number to tweets taken into account for word2vec by 5 all other features were same.

Number of tweets considered while creating data sets	**Correctly Classified Instances Percentage (count)**	**Incorrectly Classified Instances Percentage (count)**
5	**84.0578 %** (36144)	15.9422 %(6855)
10	**88.2114 %**(37930)	11.7886 %(5069)
15	**90.2019 %**(38785)	9.7981 %(4213)
20	**91.4112 %**(39305)	8.5888 %(3693)
25	**92.1855 %**(39637)	7.8145 %(3360)
30	**93.0884 %(**40028)	6.9116 %(2972)
35	**92.1855 %**(39637)	7.8145 %(3360)
40	**93.4302 %**(40175)	6.5698 %(2825)

45	**93.9419 %**(40395)	6.0581 %(2605)
50	**94.0326 %**(40434)	5.9674 %(2566)
55	**93.9977 %**(40419)	6.0023 %(2581)
60	**93.9674 %**(40406)	6.0326 %(2594)
65	**94.3047 %**(40551)	5.6953 %(2449)
70	**94.286 %**(40543)	5.714 %(2457)
75	**94.4161 %**(40598)	5.5839 %(2401)
80	**94.3465 %**(40569)	5.6535 %(2431)
85	**94.4186 %**(40600)	5.5814 %(2400)
90	**94.3789 %**(40582)	5.6211 %(2417)
100	**94.3698 %**(40579)	5.6302 %(2421)
105	**94.4743 %**(40623)	5.5257 %(2376)
110	**94.2789 %**(40539)	5.7211 %(2460)
115	**94.4023 %**(40593)	5.5977 %(2407)
120	**94.4161 %**(40598)	5.5839 %(2401)
125	**94.3254 %**(40559)	5.6746 %(2440)
130	**94.2603 %**(40531)	5.7397 %(2468)
135	**94.0557 %**(40443)	5.9443 %(2556)
140	**94.1882 %**(40500)	5.8118 %(2499)
145	**93.9349 %**(40392)	6.0651 %(2608)
150	**93.8208 %**(40342)	6.1792 %(2657)

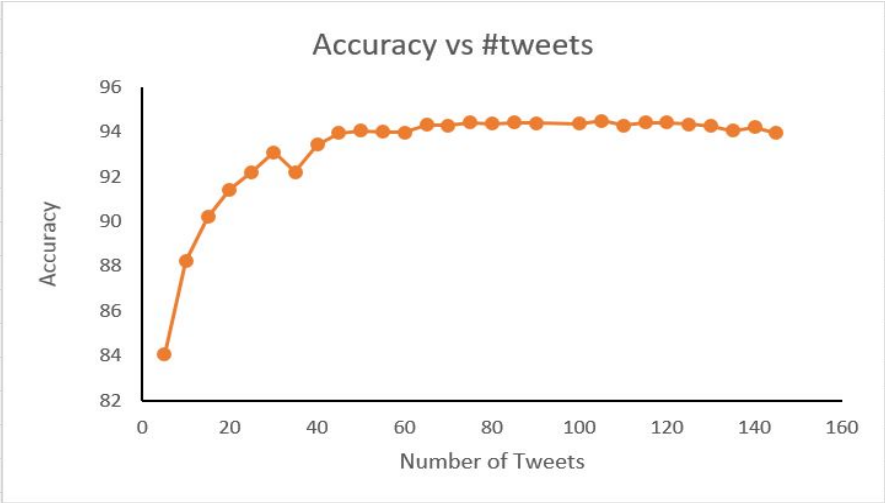


Figure 1 : shows the number of tweets vs accuracy plot as per above table

As we can see from above plot after around 60 tweets accuracy remains the same. For our final analysis we took 60 tweets into account and created a final data set of 86784 unique users with feature vector of size 109 taking random 60 tweets into account for word2vec vector. After generating data set we tried different approaches for testing and training our model on SVM.

- Training SVM model with 10 fold cross validation technique for dataset of 86784

Correctly Classified Instances	82109	**94.6152 %**
Incorrectly Classified Instances	4673	5.3848 %

- Training SVM model with 80% data as training set and rest 20% data as testing set.

Correctly Classified Instances	16409	**94.5437 %**
Incorrectly Classified Instances	947	5.4563 %

- To cross check that we are not overfitting the model this time we divided the full data set into 4 parts and created one training set and 3 test sets.
    1. Training data set (60k Instances)
    2. Testing data set 1 (10k Instances)
    3. Testing data set 2 (10k Instances)
    4. Testing data set 3 (7k Instances)

We first trained SVM on training data set (60k instances) and then tested on all 3 testing data sets results are shown below.

Testing data set 1	Correctly Classified Instances	9495	**94.95 %**
	Incorrectly Classified Instances	505	5.05 %
Testing data set 2	Correctly Classified Instances	9474	**94.74 %**
	Incorrectly Classified Instances	526	5.26 %
Testing data set 3	Correctly Classified Instances	6429	**94.7671 %**
	Incorrectly Classified Instances	355	5.2329 %

Since we have built our model with different approaches and tested with multiple data sets in addition to that we also tried to make sure that the projected results are not overfit, therefore we can conclude our results as follows.

Correctly Classified Instances 82109 **94.6152 %**

Incorrectly Classified Instances 4673 5.3848 %

=== Confusion Matrix (Total Number of Instances 86782)===

a b <-- classified as

44564 1533 | a = 0

3140 37545 | b = 1

Precision --- **94.7%**

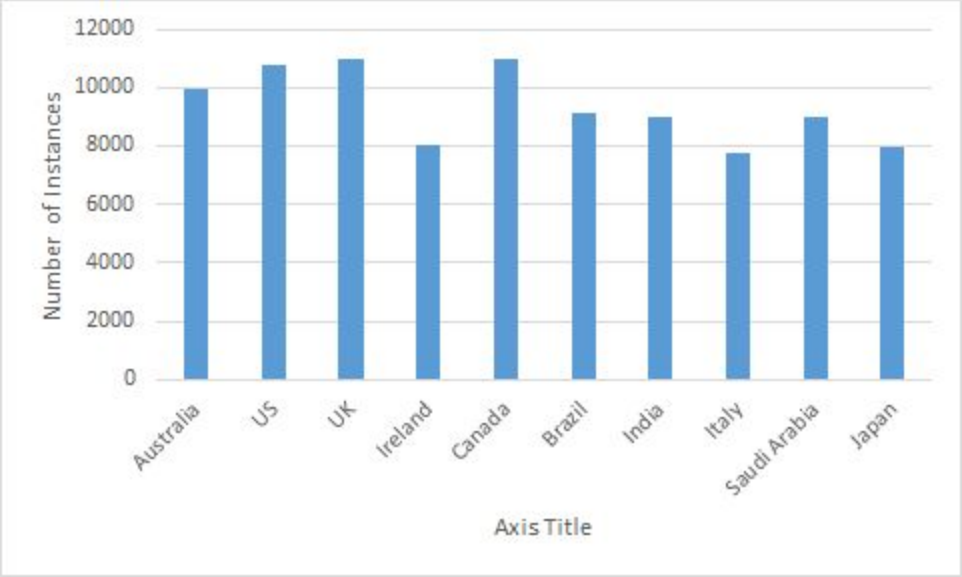
Recall --- **94.6%**

Predicting the country :

Extending our existing model, we also tried to classify each user according to their countries. Instead of a binary classifier as mentioned above we built a multi class classifier which will take the same feature vector as input and classify it into one of the country buckets.

Since we have collected the user information for a total of 10 countries, we aim to classify the users into one of the 10 classes.

Our data set for 10 countries look like this:



Our dataset had users of each country in the range of 8000 to 10,000 users to create an almost balanced database, which helped in making sure that results were not biased towards any particular class.

Total Number of Instances 92834

k cross validation for multi class:

Correctly Classified Instances 76780 82.7068 %

Incorrectly Classified Instances	16054	17.2932 %

80-20 split validation for multi class :

Correctly Classified Instances	15401	82.9482 %
Incorrectly Classified Instances	3166	17.0518 %

As we can see from the above results, our existing model works considerably well when extended to the scenario of multi class classifier. There is still scope of improvement for this, which can be tried by modifying a few of the features and also adding other location features like geo tagged location etc.

CONCLUSION:

From the initial phase of the project we tried and tested different approaches to solve the given problem. We went through different stages during this evolution process.

Stage 1 : Naive bayes algorithm with dictionary word mapping, taken into account single tweet, dataset size (33k)

Stage 2 : Naive bayes algorithm with word2vec feature vector , taken into account single tweet, data set size (33k)

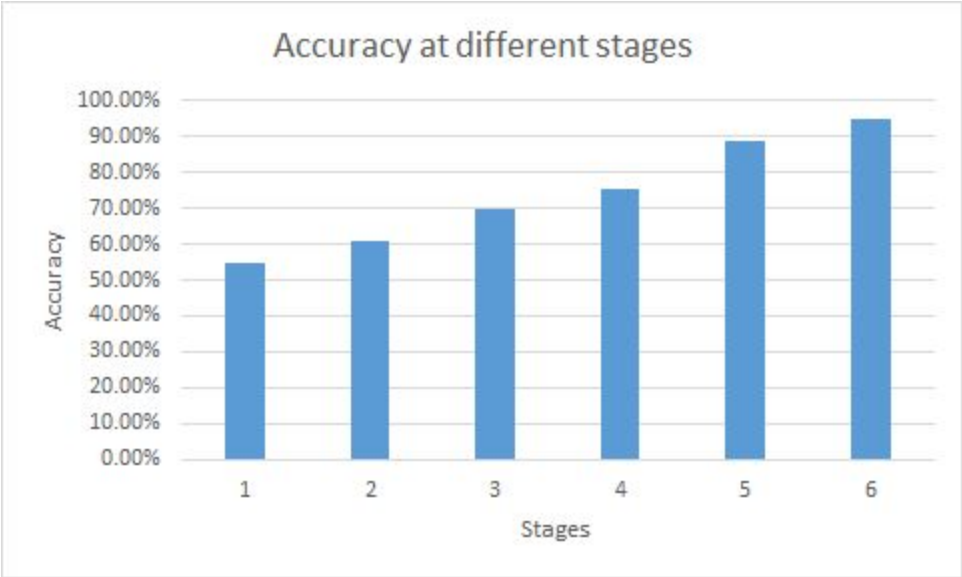
Stage 3 : Random forest algorithm with word2vec feature vector, taken into account single tweet, data set size (33k)

Stage 4 : naive bayes with multiple tweets per user, along with other user related features, data set size (33k)

Stage 5 : Support vector machine with multiple tweets per user, along with user related features, data set size (33k)

Stage 6 : Support vector machine with multiple (60) tweets per user, along with user related features, data set size (96k), increased our data size.

During these stages every time we tried to improve the accuracy further more and tried to build the model more robust. Below is the bar plot of the accuracy we obtained during these different stages.



Therefore as we can inferred the final model is a stable solution to the problem at hand and can also be extended to solve the bigger problem of correctly predicting the country of the user.

REFERENCES

- http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2886/3262
- https://www.jair.org/media/4200/live-4200-7781-jair.pdf
- http://www.simafore.com/blog/bid/112816/When-do-support-vector-machines-trump-other-classification-methods
- https://cs224d.stanford.edu/reports/JindalPranav.pdf
- https://www.reddit.com/r/MachineLearning/comments/44iy7u/differences_between_continuous_bag_of_words_cbow/
- https://radimrehurek.com/gensim/models/word2vec.html

FUTURE WORK :

- The existing approach can be extended to include other user related features for predicting the exact country of the user. Predicting the exact location(city) inside the country is another challenging problem which can be dealt with using this approach.
- Apart from twitter, there are many other similar social media platforms which have a diverse user base and are filled with information. Tapping those with the

same problem of identifying the location of the user can help in situations like disasters, emergencies etc.