Sentiment Evaluation: User, Business Assessment and Hashtag Analysis

Chetan Jha[1] and Ray Walshe[1]

[1] Dublin City University, Dublin, Ireland
chetan.jha3@mail.dcu.ie
ray.walshe@dcu.ie

Abstract. Social media has become an important platform for the public to express their opinions and comment on all manner of things. Analysis of social media data can yield interesting facts and views about another person, product, business or an issue. This paper focuses on content profiling based on sentiment analysis of Twitter data for a particular User, Business or a Hashtag by focusing on the emotions, reactions and opinions written on Twitter by different users in the form of tweets and using statistical, learning and natural language processing techniques.. To accomplish the task, this paper will also use various Machine Learning techniques along with a combination of Python Natural Language Toolkit (NLTK) VADER Library. The different machine learning classifiers are evaluated and results show that Artificial Neural Networks perform the best with 76.46% accuracy, Random Forest second most accurate classifier with 75.95% accuracy and Multinomial Naïve Bayes achieved 75.65% accuracy. The methods described here provide users with a robust and flexible way of profiling Twitter users using sentiment extracted from tweet data.

Keywords: Sentiment Analysis, User profiling, Classification algorithms, Artificial Neural Networks, Convolutional Neural Networks, Multinomial Naïve Bayes, RandomForest, K-Nearest Neighbour, Support Vector Machine, NLTK VADER.

1 Introduction

With the increase in use of social media and networked services, people share thoughts and emotions and in so doing generate a huge quantity of data. Twitter, [1] a major microblogging platforms with 1.3 billion accounts and 310 million active users, plays a major role in the research of social networks and social behaviour. Usesr on Twitter share their preferences in the form of tweets which are a free-format and fixed length (140 characters) texts which often contain rich information about likes and dislikes of consumer products, music, movies, or social issues. These opinions (often termed Sentiment) can be used by individuals, businesses or institutions to learn about sensitivities of people towards a certain product or issue. Although information-dense, it is often difficult to conclude anything from the raw data without effective ways to visualise or compare different aspects of this data. Sentiment analysis is a growing field of discussion by marketers and companies as it has been validated by describing how their customers are feeling. For example, Expedia Canada [2] corrected its marketing blunder of involving a violin sound which was reported as "Annoying" by people on Twitter. Sentiment analysis [3] [4] was able to provide useful insights about USA presidential election to understand how people were talking about the candidates on social media and also brought clarity to how the election was evolving.

2 Background and Related Work

Considering the internal implementation, sentiment analysis can be broadly classified into two categories – Lexicon based (unsupervised) [5] and Machine Learning based (supervised) [6]. Lexical methods use a dictionary of words annotated with their semantic polarity and strength which is then used to calculate polarity score of the document and is further classified into respective sentiment. While the machine learning methods require creating a model by training the classifier with annotated examples. This paper will look into sentiment prediction using both the ways (as above-mentioned) by using them in combination

The mapping of sentiments of US citizens about the weather was created by Dialogue earth project [7] devised to report the sentiments related to weather according to different states of USA on the map. Vrunda and Vipul conducted a comparative study [8] of Support Vector Machine, Naïve Bayes, Multi-Layer Perceptron, Decision Tree, Subjective Lexicon Method, Case based Reasoning and K-Nearest Neighbour focusing on comparing different classification algorithms for sentiment analysis. They directly compare the final results of classifiers but stop short of comparing the results after applying different preprocessing methods, feature extractors or lexicon based methods for the combination.

Amit, Sourabh, Pratik and Akshay try to compare the pros and cons of Naïve Bayes, Max Entropy, Boosted Trees and Random forest for studying classification algorithms used in Sentiment Analysis [9].

All these papers include significant research in the field of sentiment analysis but trying to combine lexical method with machine learning methods to improve prediction remains novel. This paper will also look into various feature extraction methods and will try to combine their output with lexicon based predictions and further compare different Machine learning algorithms.

3 Data Sources

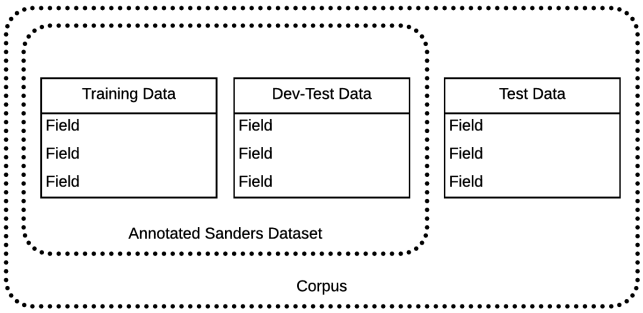


Fig. 1. - Data Sources

Data used for this paper is basically in the form of tweets. Due to the restricted length of 140 characters, people use acronyms, emoticons, abbreviations and slang words. Tweets also contain Hashtags which are used to mark topics or subjects starting with '#' symbol. People refer to another Twitter user using '@' symbol along with the username. Tweets can also contain URLs inserted by user linking to an image, document or website.

3.1 Training dataset

The training data used in this paper is the Sanders dataset [10]. Niek Sanders provided the Twitter sentiment corpus publically on October 24, 2011. This dataset consists of 5513 tweets which have been manually annotated by the annotator as positive, negative, neutral or irrelevant. These tweets are technology based tweets for four companies, i.e. Apple, Google, Microsoft and Twitter. Out of these tweets, there are 570 positive, 654 negative, 2503 neutral and 1786 irrelevant tweets.

The dataset contains tweet id and its corresponding sentiment. Approximately 4364 tweets were downloaded, of which there were 428 positive, 475 negative, 2003 neutral and 1458 irrelevant tweets.

3.2 Test dataset

This real-time Twitter data was acquired and stored in the database for the purpose of research this paper. This data is downloaded for few different kinds of Twitter users which are related to technology, political and sports. This data ranges from 20,000 to 60,000 tweets per user.

The system designed and presented in this paper will run for any Twitter user, and the data fetched from Twitter in real-time as per the limitations [11] applied by Twitter for search which is 180 tweets for a user-based authentication and 450 for an app-based authentication. The Twitter search API also restricts the search against a sampling of recent tweets published in past seven days.

4 Tools and Applications

4.1 Django Web Framework

Django [12] is a high-level Python Web Framework which is free and open-source. It follows a Model-View-Template (MVT) architecture and contains a set of components like user authentication, management panel, forms. These features

help alleviate some of the overhead while building a new site. Django also provides access to Python's subject related libraries like Tweepy, Scikit-learn, Pymongo.

This framework is required to implement the client interaction server for allowing the user to use the project being developed for this paper. For enhancing the user interface of the project, Metronic Theme [13] was used.

4.2 Scikit-learn

Scikit-learn [14] is a Python [15] based machine learning library which provides a range of supervised and unsupervised learning algorithms. Scikit-learn is built upon Scipy (Scientific python) and includes some of the most important Python libraries for data science like NumPy, SciPy, Matplotlib, IPython, SymPy and Pandas. Scikit-learn is focused on data modelling which include clustering, cross-validation, dimensionality reduction, feature extension and selection.

4.3 Keras

Keras [17] is a Python based open source, high-level neural network API. It is capable of running machine learning models like convolutional and recurrent neural networks using TensorFlow [18], CNTK [19] or Theano [20] on CPU as well as GPU. It makes implementation of libraries like TensorFlow, CNTK and Theano easy and user friendly.

4.4 TensorFlow

TensorFlow [18] is an open-sourced library developed by Google to meet their needs of building and training neural networks which is user-friendly, easy to implement [21]. TensorFlow is more popular in comparison to Theano and Torch [21]. It supports a large community and also has a very interactive visualisation dashboard called TensorBoard.

4.5 Tweepy

Tweepy [22] is an open-sourced library which enables Python to communicate with the Twitter platform and use its RESTful API. Tweepy has a Cursor object which helps to paginate the downloading and iterating of tweets. It can recursively download the number of tweets requested by the user while maintaining the limitations [11] applied by Twitter. Tweepy was used to fetch tweets for training database and is also responsible for fetching the tweets at runtime.

4.6 MongoDB

MongoDB [23] is an open-source database that uses document-oriented data model. It is a NoSQL [24] database which is highly scalable and performance efficient. Instead of having rows and columns which are a part of SQL databases, MongoDB is based on collections and documents. Each document comprises of key-value pairs which are the fundamental unit of data in MongoDB. Like other NoSQL databases, MongoDB also has a dynamic schema which allows documents to have different structures and fields. MongoDB performs faster than other relational databases like Oracle [25] and MySQL [26] and PyMongo [27] provides an easy and recommended way to work with MongoDB from Python.

4.7 NLTK VADER

VADER [28] is an abbreviation for Valence Aware Dictionary for sentiment Reasoning which was developed in 2014 [29]. It is an open-sourced, lexicon and rule-based sentiment analysis tool which is specifically attuned to sentiments expressed in social media and also works well on other domains. VADER can include sentiments from Emoticons [:-)], Sentiment related acronyms [LOL] and Slang [Meh]

When a piece of text is passed to VADER, it returns result in terms of polarity –

1. Negative polarity – Negativity score of the text.
2. Neutral polarity – Neutrality score of the text.
3. Positive polarity – Positivity score of the text.

4. Compound polarity – This is the overall polarity score of the text. It will be negative if the text is largely negative and positive if the text is overall positive. It will be zero if the text is neither negative nor positive and will mean the text is neutral.

5 System Design and Models

5.1 Pre-processing

The training data as well the testing data undergoes the following pre-processing steps to bring into focus the words which contain some sentiment. The pre-processing helps to reduce the feature space and also increase the accuracy of the Machine learning algorithm. Pre-processing involved the following steps:

1. Removal of : URLs , repetition of characters, numbers, stop words like "the", "to", "from"
2. Changing text to lower-case
3. Remove of user mentions, "#" symbol. For example – "#great" to "great" etc.
4. Stemming – Stemming is the process of reducing the word to its root. Stemming improves the system effectiveness and results in higher accuracy of predictions [33].
5. Dictionary checking implemented using PyEnchant [34], which is a spell-checking library for python

5.2 Sentiment Analysis

Feature Extraction and Selection. Feature extraction [35] is the process of transforming arbitrary data, such as text or image, into numerical features usable for machine learning. When dealing with a large set of data major problem arises from the number of variables involved. Analysis comprising of large number of variables requires significant amount of computational power and memory. It can also cause the machine learning algorithm to over-fit the training data and cause poor predictive performance with new data. So, feature extraction is a method of creating a combination of input variables to avoid such problems and to accurately predict the results.

Scikit learn provides two suitable feature extraction classes:

CountVectorizer. CountVectorizer [36] converts a collection of text documents into a matrix of token counts. Each text is separated into tokens and the number of times each token occurs is counted.

TfidfVectorizer. TfidfVectorizer [37] is similar to the CountVectorizer as it also creates document term matrix, but instead of filling it with word count it calculates term frequency-inverse document frequency value for each word.

$$TF - IDF = term\ frequency * \left(\frac{1}{document\ frequency} \right)$$

or

$$TF - IDF = term\ frequency\ *\ inverse\ document\ frequency$$

where,

term frequency = number of occurrences of the word in a document

inverse document frequency = inverse of the number of occurrences of the word in all the documents

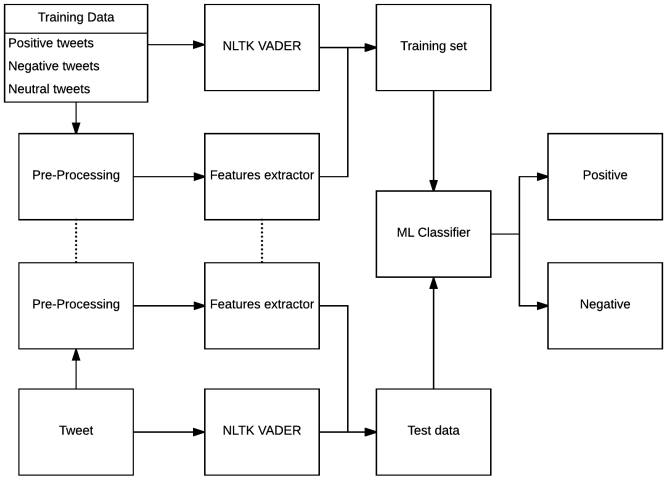


Fig. 2. - Machine Learning Model

Machine Learning Classifiers. Machine learning approach to classify the tweets as per the sentiment involves training set to develop a sentiment classifier that classifies sentiments of tweets.

The subsequent sections will explore some of the most commonly [8] [9] [38] used machine learning classifiers for text classification.

Multinomial Naïve Bayes (MNB). Naïve Bayes classifier is based on Bayes theorem and assumes that the classes for classification are independent. Despite this assumption usually being false, analysis has shown there are some theoretical reasons for the high efficiency of Naïve Bayes classifiers [39]. Though the probability estimates of Naïve Bayes are not good but the classification decisions are quite good [40]. Naïve Bayes classifier is also recommended in the presence of inadequate computational power and memory capacity.

The Multinomial Naïve Bayes [41] classifier is a specialised version of Naïve Bayes which is designed for the classification of data with discrete features. For example, word counts for text classification. MNB requires numerical feature counts as input.

Random Forest (RF). Random Forest [42] is an ensemble algorithm which means a combination of more than one same or different kind of algorithms for classifying objects. Random forest operates by constructing a large number of decision trees at training time and outputs the mode of classes classified by individual trees. By randomly selecting the trees, their correlation is reduced and prediction power is increased. The Random Forest supports parallelisation and concurrency of different trees. If there is a small size of data with a large number of trees, then Random Forest can over-fit the data. The Random Forest has been found to provide good accuracy in classification of datasets for opinion mining [9].

K-Nearest Neighbour (KNN). K-Nearest Neighbour [43] classifier is one of the simplest classification algorithms and is a non-parametric lazy learning algorithm which means that it does not make any assumptions on the underlying data distribution. Due to its lazy nature, its training phase is fast, and it does not use any generalisation of training data which means it requires training data during the testing phase. K-Nearest Neighbour assumes data points are in a metric space and requires it to be a scalar or multidimensional vector.

Support Vector Machine (SVM). Support Vector Machine [44] classifier can be used for both classification or regression problems. Support vector machine classifies data by finding the best hyperplane that separates all the data points of the particular classes. Support Vector Machine works for classification with exactly two classes but can be used for classification by reducing multi-class data to a binary problem by choosing random partitions of the set of classes, recursively. This increases the time taken but also enhances the accuracy of learning.

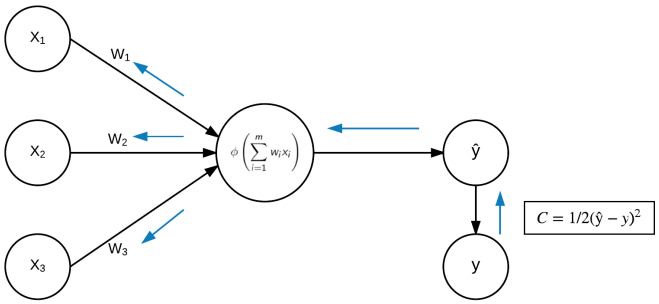


Fig. 3. - Learning in Neural Networks

Artificial Neural Network (ANN). Artificial Neural Networks [45] are computational algorithms intended to simulate the behaviour of biological systems composed of Neurons. They are constructed as a system of interconnected neurons which can compute values from inputs provided.

Neurons have multiple inputs, does some processing on received inputs and give the output. Neurons are organised in the form of layers and hidden layers. So, the input values to a neuron are received either from the input layer, if it is the first layer of neurons or the values are received from neurons of the previous layer. The connection between inputs and neurons is called Synapse, and by adjusting the weights, it can be decided how much input signal should be passed to the neuron.

$$Input\ received = Input(x) * weight(w)$$

Inside neurons, an activation function is applied to the summation of all weighted inputs. The output value of a neuron can be a continuous value (for example prices), Binary value or Categorical value.

The difference in output and the actual value is used to calculate the cost function C. Then the weights are adjusted to check the cost function once again. It keeps on repeating until the cost function is minimized.

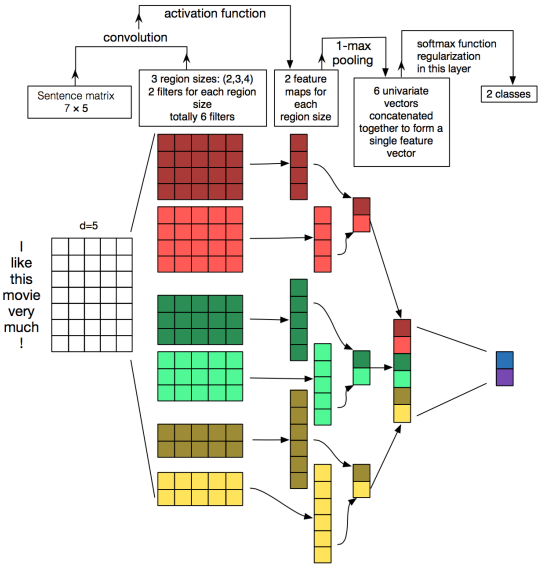


Fig. 4. - Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioner's Guide to) Convolutional Neural Networks for Sentence Classification. Available: https://arxiv.org/pdf/1510.03820.pdf

Convolutional Neural Network (CNN). Convolutional Neural Networks [46] [47] [48] [49] is responsible for breakthroughs in Image Classification and are the core part of most self-driving cars and image detection systems, for example, Facebook's automated photo tagging. CNN contains several layers of convolutions with non-linear activation functions like Rectifier function or Hyperbolic Tangent function applied to the results. In CNN, each input neuron is not connected to each output neuron in the next layer like traditional feedforward neural network. Instead, convolutions are used over the input layer to compute the output which creates local connections as all regions of input are connected to a neuron in the output. Each layer applies multiple filters and combines their results.

For Text Classification, input to the classifier is sentences or documents represented as a matrix. In this matrix, each row corresponds to one token (word). In image processing, the filters slide over local patches of an image, while in Natural Language Processing the filters slide over full rows of the matrix, which corresponds to a word. For example, see Figure 4, which explains the process of CNN.

User Interaction Portal. A user portal was developed using Django framework to allow users to interact with the project. A user can input either one or two Twitter usernames to profile and compare them. When usernames are provided, the system will respond to the request with the following results:

1. User Details: Name- Location- Profile picture- Profile background image- Description- Followers count - Tweets count
2. Tweets: Most recent 200 tweets by user - Most recent 200 tweets for user (@mention) - Most popular 200 tweets for the user (@mention)
3. Sentiment Profiling:
    a. Timeline – Number of positive, negative and neutral tweets for the Twitter user: - Past 24hr -Past week - Past month -Past year - All time
    b. Sentiment Reach – Number of people to whom the sentiments reached, i.e. the spread of positive, negative or neutral influence.
    c. Regional Stats – It was planned and proposed but was not feasible due to the limited location based data. For example, only 43 out of 33356 tweets contained location data. User location was tried as a fall back mechanism, but user location contained incorrect values.

Sentiment Analysis

DASHBOARD | HASHTAGS

User Analysis

@ google | @ facebook

COMPARE

Google

News and updates from Google

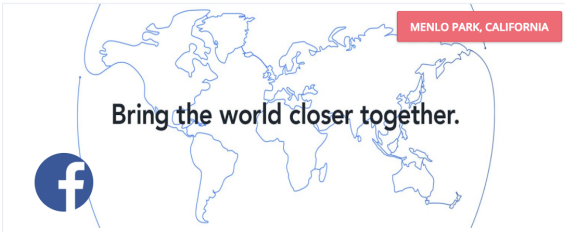
18734891 74845

TWEETS

By user | For user | Most Popular

- Today's #GoogleDoodle honors James Wong Howe, innovative Chinese American cinematographer and "master" of light →... https://t.co/XIkkkGSTEO — *3 hours & 5 min*
- @RCvictoryNet Glad it was caught in time! You might want to add these extra security layers to your Google account: https://t.co/4iM0oCSLgp. — *8 hours & 56 min*
- @Hellarryis Hi there. Just to confirm, are you still able to sign into your account? Let us know. — *8 hours & 58 min*
- @LouisatheLast Glad it was caught in time! You might want to add these extra security layers to your Google acct: https://t.co/4iM0oCSLgp. — *9 hours & 1 min*
- @K_Geezy74 Hi there. If you didn't request this code, you can ignore it. More info: https://t.co/4eap2FuXNG. Hope it helps. — *9 hours & 1 min*
- @RichardSkipper Mind if we jump in? Just to confirm, are you trying to — *9 hours*

SENTIMENT PROFILING



Facebook

Give people the power to build community and bring the world closer together.

14040327 7450

TWEETS

By user | For user | Most Popular

- @chadah Hi Chad. Please visit our Help Center to learn how to report this issue to us: https://t.co/Ljr5QBMqTS. Thank you. -MG — *4 minutes*
- @kara_hardy Hey Kara. Can you please try reinstalling the app to see if that makes a difference? Thanks in advance. -MG — *5 minutes*
- @mpjones26 Hi there, Matthew. Can you please provide more information as to what you are experiencing? Thanks! -AH — *9 minutes*
- @Amazing_Aye Hey Aileen. Can you please have your aunt visit the Help Center to learn more: https://t.co/x1NytXhOBC? Thanks. -MG — *4 hours & 24 min*
- @DavidLehman Hey David. For payment help please visit our Help Center: https://t.co/8BN72AJvyh. Thank you. -MG — *4 hours & 26 min*
- @jamilawardknott Hi Jamila. Can you please try reinstalling the app to see if that helps? Thanks in advance! -MG — *4 hours & 27 min*
- @W6leff Hey Jeff. Please visit our Help Center to learn how to unfollow — *2 days*

SENTIMENT PROFILING

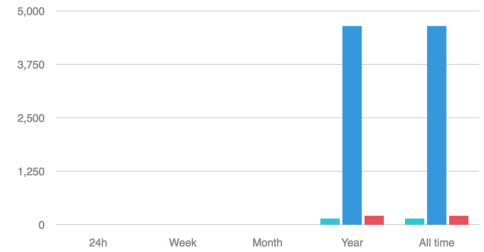
Sentiment Analysis

SENTIMENT PROFILING

Description of profile

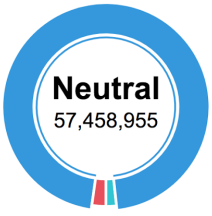
Tweet timeline

Past 24hr	0	0	0
Past week	0	0	0
Past 1 month	0	0	0
Past 1 year	144	4648	208
All time	144	4648	208



Sentiment Reach

Positive	399503 people
Negative	743438 people
Neutral	57458955 people



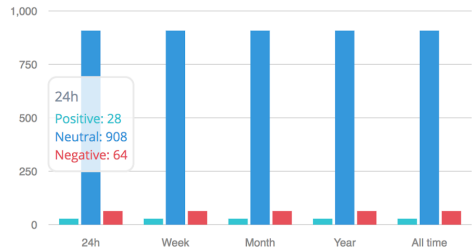
Regional Stats

SENTIMENT PROFILING

Description of profile

Tweet timeline

Past 24hr	28	908	64
Past week	28	908	64
Past 1 month	28	908	64
Past 1 year	28	908	64
All time	28	908	64



Sentiment Reach

Positive	44119 people
Negative	302759 people
Neutral	21922710 people

Regional Stats

		Machine Learning (10 fold cross-validated)				Neural Networks	
	Number of features	MNB	RF	KNN	SVM	ANN	CNN
Count Vectorizer							
Basic	5142	75.04	75.73	69.02	69.28	74.91	69.07
With stop words removal	5042	74.1	73.41	68.72	69.06	74.05	72.51
With Stemming	4207	**75.65**	**75.95**	**70.1**	**69.66**	**76.46**	**75.6**
Dictionary	**3479**	74.69	74.96	68.76	69.36	75.26	68.73
NLTK VADER	5107	75.08	75.34	69.96	69.58	75.43	70.62
Tfidf Vectorizer							
Basic	5064	68.76	73.84	72.42	74.78	73.71	68.56
With stop words removal	5048	**70.01**	74.7	72.16	73.92	74.74	69.42
With Stemming	4204	69.58	**75.6**	**74.27**	**75.6**	74.05	69.07
Dictionary	**3516**	69.67	74.83	70.61	74.18	**76.29**	68.73
NLTK VADER	5094	69.11	74.48	73.11	74.49	75.43	**70.45**

Table 1 - Accuracy results in percentage (Best results per classifier are highlighted)

6 Experiments and Results

For the purpose of this paper, as shown in Figure 2, annotated data by Sanders was downloaded, cleaned and pre-processed. This data was divided into 80:20 ratio for training and testing respectively. Different feature extractors and selectors were used to create sparse matrix of words. NLTK VADER results were obtained for the tweets and the positive and negative polarity of the tweet was inserted in two new columns in the previously created sparse matrix. This data was used to train the machine learning algorithms. Results were predicted for the test data and accuracy was calculated using 10-fold cross validation, wherein the original sample is divided into 10 samples, out of which 9 were used for training and one was used for validation. This process is repeated 10 times and therefore, each sample is used as a validation sample exactly once. The result is averaged to calculate the prediction accuracy. Table 1, shows the results of the accuracy of various classifiers where columns denote the classifiers used and the rows signify the type of methods used for enhancement before training the classifiers. The row termed as "basic" does not utilise the enhancement methods mentioned in other rows of the table.

Also, when NLTK VADER was used separately to predict the sentiment on the test data, based on its polarity, it gives an accuracy of 55.80% in comparison to human sentiment analysis.

When n-grams were tried, the bigrams and trigrams increased the feature space multi-fold for example, the data set contained 5091 unigrams, 18286 bigrams and 22,306 trigrams. When unigrams, bigrams and trigrams are used simultaneously they result in feature space of 45683 words. N-grams, in turn, increased the time required to train and predict a model for classifiers like Random Forests and K-Nearest Neighbour, along with significantly increased memory consumption of some classifiers like Convolutional Neural Network (up to around 22GB) which was beyond the scope of the system setup used. Due to increase in the demand of resources, and the fact that there was only a marginal increase in accuracy by 0.01% for Multinomial Naïve Bayes, N-grams were not used.

7 Conclusion

Twitter is one of the most popular social networks where users can tweet about different topics. The tweets have a maximum size of 140-characters which imposes a significant challenge in predicting sentiments of that text. In this paper, an evaluation method is designed where different machine learning classifiers are evaluated after changes like using different feature extractors, data preprocessing, combining NLTK VADER polarities.

Table 1, shows the results where Artificial Neural Network seems to perform the best as compared to other classifiers in the given conditions which resulted in 76.46% accuracy. On the other hand, stemming process using snowball stemmer on an average provided the best increase in the accuracy of classifiers, followed by inclusion of VADER polarity which handles emoticons, slangs and acronyms automatically.

Random Forest proved to be the second most accurate classifier with 75.95% accuracy. However, Multinomial Naïve Bayes along with being fastest and most light weight on resources achieved 75.65% accuracy, almost achieving RF accuracy.

The methods described here provide users with a robust and flexible way of profiling Twitter users, for example, comparing two hotels, two companies, political parties or products by providing their Twitter usernames.

8 Future work

For future work, further research is necessary to improve the profiling of Twitter users, increase user confidence and provide more meaningful insights about the reason regarding positive or negative sentiments. Some of these may include:

1. Bias removal - While profiling a Twitter user, for example, any product or company, research is required to find a way of fetching a list of employees of the company or its subsidiary companies who may be tweeting to induce a positive bias. Such list can be fetched from LinkedIn, but research needs to be done on finding a way to correlate the LinkedIn users with Twitter users.
2. Researching ways of predicting the location of users based on their friends or content of their tweets.
3. Researching to analyse and categorise the users based on their likes and favourites for predicting which kind of users are more likely to tweet positive or negative about a user or business.

References

1. Twitter, "Company | About," [Online]. Available: https://about.twitter.com/company. [Accessed 19 June 2017].
2. www.theglobeandmail.com, "No strings, please: Expedia Canada ad falls flat - The Globe and Mail," [Online]. https://www.theglobeandmail.com/report-on-business/no-strings-please-expedia-ad-falls-flat/article16476388/. [Accessed 23 July 2017].
3. MonkeyLearn Blog, "Donald Trump vs Hillary Clinton: sentiment analysis on Twitter mentions | MonkeyLearn Blog," [Online]. https://monkeylearn.com/blog/donald-trump-vs-hillary-clinton-sentiment-analysis-twitter-mentions/. [Accessed 22 July 2017].
4. SAS, [Online]. http://blogs.sas.com/content/sgf/files/2016/09/13550_Grover-Analytics-Conference-E_Poster-Final-Sid-Grover.pdf. [Accessed 23 July 2017].
5. Coursera, "5.2 Explanations of sentiment analysis with unsupervised learning - Yonsei University | Coursera," [Online]. Available: https://www.coursera.org/learn/text-mining-analytics/lecture/x2oe6/5-2-explanations-of-sentiment-analysis-with-unsupervised-learning. [Accessed 25 July 2017].
6. Coursera, "5.1 Explanations of sentiment analysis with supervised learning - Yonsei University | Coursera," 25 July 2017. [Online]. Available: https://www.coursera.org/learn/text-mining-analytics/lecture/hbTb7/5-1-explanations-of-sentiment-analysis-with-supervised-learning.
7. D. Earth, "Dialogue Earth Pulse Weather Sentiment," [Online]. http://www.dialogueearth.org/pulse/weather/. [Accessed 26 May 2017].
8. V. Joshi and V. Vekariya, "A Comparative Study on Classification Algorithms for Sentiment Analysis," International Journal for Scientific Research & Development, vol. 4, no. 10, pp. 428-430, October 2016.
9. A. Gupte, S. Joshi, P. Gadgul and A. Kadam, "Comparative Study of Classification Algorithms used in Sentiment Analysis," International Journal of Computer Science and Information Technologies, vol. 5, no. 5, pp. 6261-6264.
10. N. Sanders, "Sanders Analytics - Twitter Sentiment Corpus," [Online]. http://www.sananalytics.com/lab/twitter-sentiment/. [Accessed 23 May 2017].
11. Twitter, "Rate Limits: Chart — Twitter Developers," [Online]. https://dev.twitter.com/rest/public/rate-limits. [Accessed 27 May 2017].
12. Django Software Foundation, "The Web framework for perfectionists with deadlines | Django," Django Software Foundation, [Online]. https://www.djangoproject.com. [Accessed 23 May 2017].
13. keenthemes.com, "Metronic | #1 Selling Ultimate Bootstrap Admin Dashboard Theme," keenthemes.com, [Online]. Available: http://keenthemes.com/preview/metronic/. [Accessed 28 May 2017].
14. Python.org, "scikit-learn: machine learning in Python — scikit-learn 0.19.0 documentation," Python.org, [Online]. Available: http://scikit-learn.org/stable/index.html. [Accessed 02 June 2017].
15. O'Reilly Media, Inc., "Data scientists and data engineers like Python and Scala - O'Reilly Media," https://www.oreilly.com/ideas/data-scientists-and-data-engineers-like-python-and-scala. [Accessed 22 May 2017].
16. IPython development team, "The Jupyter Notebook — IPython," [Online].https://ipython.org/notebook.html. [Accessed 27 May 2017].
17. keras.io, "Keras Documentation," [Online]. Available: https://keras.io. [Accessed 14 July 2017].
18. Google Inc., "GitHub - tensorflow/tensorflow: Computation using data flow graphs for scalable machine learning," [Online]. Available: https://github.com/tensorflow/tensorflow. [Accessed 16 July 2017].
19. Microsoft, "The Microsoft Cognitive Toolkit," Microsoft, [Online]. Available: https://docs.microsoft.com/en-us/cognitive-toolkit/. [Accessed 30 July 2017].
20. Theano Development Team, "Welcome — Theano 0.9.0 documentation," [Online]. Available: http://www.deeplearning.net/software/theano/. [Accessed 30 July 2017].
21. Stanford University, "Lecture note 1: Introduction to TensorFlow," [Online]. Available: http://web.stanford.edu/class/cs20si/lectures/notes_01.pdf. [Accessed 02 Aug 2017].

22. Tweepy.org, Tweepy.org, [Online]. Available: http://www.tweepy.org. [Accessed 29 May 2017].
23. MongoDB, Inc., "MongoDB for GIANT Ideas | MongoDB," https://www.mongodb.com. [Accessed 03 July 2017].
24. MongoDB, Inc., "NoSQL Databases Explained | MongoDB," [Online]. Available: https://www.mongodb.com/nosql-explained. [Accessed 05 July 2017].
25. A. Boicea, F. Radulescu and L. I. Agapin, "MongoDB vs Oracle -- Database Comparison," IEEE, 20 November 2012.
26. C. Győrödi, R. Győrödi and G. Pecherle, "A comparative study: MongoDB vs. MySQL," IEEE, 16 July 2015.
27. MongoDB, Inc, "Python Driver (PyMongo) — Getting Started With MongoDB 3.0.4," [Online]. Available: https://docs.mongodb.com/getting-started/python/client/. [Accessed 05 July 2017].
28. nltk.org, "nltk.sentiment.vader — NLTK 3.2.4 documentation," [Online]. Available: http://www.nltk.org/_modules/nltk/sentiment/vader.html. [Accessed 18 June 2017].
29. C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," [Online]. http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf. [Accessed 22 July 2017].
30. Amazon Web Services, "Amazon Web Services (AWS) - Cloud Computing Services," Amazon.com, [Online]. Available: https://aws.amazon.com. [Accessed 22 June 2017].
31. Amazon Web Services, "AWS Free tier," Amazon.com, [Online]. Available: https://aws.amazon.com/free/. [Accessed 09 June 2017].
32. K. Ganesan, "Text Mining, Analytics & More: All About Stop Words for Text Mining and Information Retrieval," [Online]. Available: http://text-analytics101.rxnlp.com/2014/10/all-about-stop-words-for-text-mining.html. [Accessed 07 July 2017].
33. N. Kaur and M. Kakkar, "A balanced sentiment analysis approach with stemming porter for neutralized emotion weightage," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 10, pp. 466-467, October 2015.
34. scikit-learn.org, "4.2. Feature extraction — scikit-learn 0.19.0 documentation," [Online]. Available: http://scikit-learn.org/stable/modules/feature_extraction.html#feature-extraction. [Accessed 15 June 2017].
35. scikit-learn.org, "sklearn.feature_extraction.text.CountVectorizer — scikit-learn 0.19.0 documentation," http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html. [Accessed 17 June 2017].
36. scikit-learn.org, "sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 0.19.0 documentation," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. [Accessed 17 June 2017].
37. H. M. Ismail, S. Harous and B. Belkhouche, "A Comparative Analysis of Machine Learning Classifiers for Twitter Sentiment Analysis," [Online]. Available: http://www.rcs.cic.ipn.mx/2016_110/A Comparative Analysis of Machine Learning Classifiers for Twitte Sentiment Analysis.pdf. [Accessed 12 July 2017].
38. H. Zhang, "The Optimality of Naive Bayes," [Online]. Available: http://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf. [Accessed 29 June 2017].
39. The Stanford Natural Language Processing Group , "Properties of Naive Bayes," [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/properties-of-naive-bayes-1.html. [Accessed 27 June 2017].
40. scikit-learn.org, "sklearn.naive_bayes.MultinomialNB — scikit-learn 0.19.0 documentation," http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html. [Accessed 01 July 2017].
41. scikit-learn.org, "3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.19.0 documentation," http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed 03 July 2017].
42. scikit-learn.org, "sklearn.neighbors.KNeighborsClassifier — scikit-learn 0.19.0 documentation," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. [Accessed 05 July 2017].
43. scikit-learn.org, "1.4. Support Vector Machines — scikit-learn 0.19.0 documentation," [Online]. Available: http://scikit-learn.org/stable/modules/svm.html. [Accessed 07 July 2017].
44. Department of Psychology, University of Toronto, "What are Artificial Neural Networks," [Online]. Available: http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/neural2.html. [Accessed 16 July 2017].
45. Stanford Engineering, Computer Science, "CS231n Convolutional Neural Networks for Visual Recognition," [Online]. Available: http://cs231n.github.io/convolutional-networks/. [Accessed 27 July 2017].
46. Stanford, "UFLDL Tutorial," [Online]. Available: http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/. [Accessed 28 July 2017].
47. D. Britz, "Understanding Convolutional Neural Networks for NLP," http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/. [Accessed 27 July 2017].
48. cambridgespark.com, "Deep learning for complete beginners: convolutional neural networks with keras," https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html. [Accessed 29 July 2017].