

Twitter review analysis and sarcasm detection

Mo'men Mohamed, Ahmed Magdy, Shymaa Hussien, Ahmed Samir, Hala Masoud,
Hana Morsy, Soha Mohamed

Faculty of Computers & Artificial Intelligence, Helwan University
Email: (dr.soha@fci.helwan.edu.eg)

Abstract Due to the increase in the number of users on the web, and the increase of the number of reviews that the user's giveaway, it becomes essential to understand and analyze this data. This paper provides a review analysis model for getting feedback from users about specific products found in tweets. This model predicts the polarity of tweet reviews. The main idea of this system is to give report with percent of positive and negative opinions about a specific product. Machine learning (ML) and Natural Language Processing (NLP) approaches are used to get a preliminary determination of the polarity of a tweet by analyzing public ones published on Twitter. In addition, this proposed model uses two techniques: topic modeling and word weight as a feature engineering and three ML algorithms: support vector machine, convolutional neural network (CNN) and naïve bays. The accuracy results of the three algorithms are compared to decide which one is better when using the same data-sets As a conclusion our model aims to provide a whole feedback picture about any product on the social network, but we will use here twitter because it is one of the most popular SN.

Keywords: Natural Language Processing, Machine Learning, Behavior analysis, Social media analysis , Twitter , Facebook, YouTube

1 - Introduction

Recently, Twitter became one of the most important and famous social media applications all over the world. In Feb 2019, it is believed that Twitter has 126 million daily active users, a figure that is 60 million smaller than Snapchat's and is dwarfed by Facebook's 1.2 billion [2].

One of the most popular tweets on twitter are tweets that contain feedback on products (reviews). The feedback can be good or bad from several perspectives. and these reviews are very important for companies as they represent users' feedback on this product. And it's very hard for companies to collect all these tweets which can be more than a billion tweets and make analysis on every tweet and detect its polarity. Another challenge is the increasing of emojis and what it means, and of course, some tweets became more sarcastic. This paper proposes an automatic review analysis and sarcasm detection system that should be seen as a helpful, complementary and accurate solution to detect the feedback or polarity from tweets and detect is it sarcastic or not. This work presents a discussion of classification methods to detect whether a tweet is positive or negative or natural and detect whether sarcastic or not, the tweet is classified into natural, positive or negative and sarcastic, or not sarcastic.

2 - Related work

A lot of recent research has been done on Tweets classification where tweets are classified into several classes. where Tweets are classified in Intent analysis as steps up the game by analyzing the user's intention behind a tweet and identifying whether it relates an opinion, news, marketing, complaint, suggestion, appreciation or query and in this project, we go to analyze the user's Opinion behind a tweet and identifying whether it relates positive,

negative or natural .

In [2] it focuses on the classification of text "sentiment analysis" as being positive, negative or being natural Different classification techniques, datasets, and features are used to evaluate, these techniques have to distinguish positive from negative and natural tweets... etc. They also examine which features appeared to be most important in facilitating this task.

In [3] used Twitter API to collect Twitter data. Their training data falls in three different categories (camera, movie, mobile). The data is labeled as positive, negative and non-opinions. Tweets containing opinions were filtered. Naive Bayes model was implemented and the Naive Bayes simplifying independence assumption was employed.

Finally, the orientation of the tweet is predicted. i.e. positive, natural or negative.

3 - Design and implementation

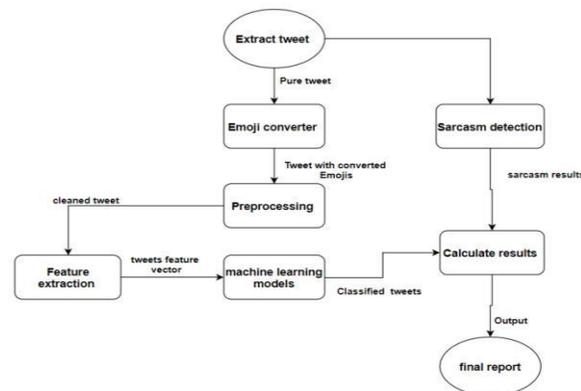


Figure 1: Framework

The proposed system is a standalone system which means that runs as a separate computer process, not an add-on of an existing process. It is going to get a collection of tweets from CSV file and input raw data in sarcasm detection next cleaning these tweets which is an important step to prepare them to the system, then features extraction is a necessary step before using machine learning algorithms , Furthermore, these cleaned collected tweets with their features (which is called features vector) will be entered to the methodology step "Machine learning algorithms" to determine sentiment of tweet finals results of sentiment and sarcasm merged to get final results .

3.1 Data collection

Twitter provides API to developers to get public tweets, We used **Twitter API (tweepy[16])** to collect data (tweets).

We used the **Search** function to get data by topic with mode we have 3 modes:

- **Recent:** which means get recent tweets about this specific topic.
- **Mixed:** get recent and old tweets about this specific topic
- **Popular:** get popular tweets which most retweets and likes about this specific topic.

Ayline website 2 APIs:

- **Text analysis API**
- **News API**

nd, we used ayline [17] text analysis API to know the senti- ment of tweets. We collect 30,875 tweets (10,255 mobiles tweets, 10,331 restaurants tweets,

10,053 drinks tweets)

Problems:

- To get different (Not duplicate) tweets we must get it in a different time.
- To use ayline API must pay or use it for 30 days in each day can enter only 1000 sentences.
- These problems lead to that we have only 30,875 tweets

3.2 Emoji converter

Emoji are symbols of faces and human expressions, which are used to express user's feelings so his tweet meaning more clear so most of the users put them in their tweets but these symbols computer can't deal with it so we have a challenge with it as emojis now a great feature to detect sentiment of tweet so we used emoji [18] package to convert emojis to words like :



Figure 2: Convert emoji example

3.3 Data set (Data Content)

We have 30,875 row (tweets number) and 7 columns:

1. ID (tweet id)
2. Tweet (full text of tweet)
3. Time (time of tweet)
4. Location (location that tweet from)
5. General topic (Mobile, restaurant, drink)
6. Specific Topic (product name)
7. Sentiment (positive, negative, natural)

3.4 Data preprocessing

We are doing the following preprocessing cycle.

- **Tokenization:** Split the text into sentences and sentences into words. Lowercase the words and remove punctuation.
- Words that have fewer than 3 characters are removed.
- All stop words are removed.
- Words are lemmatized words in the third person are changed to first person and verbs in past and future tenses are changed into the present.
- Words are stemmed words are reduced to their root form.
- Links and mentions are removed.

3.5 Feature extraction

3.5.1 Topic modeling

Definition: is the process that getting the topic of the text (tweet). The topics in our corpus are three topics Mobile, Drinks and restaurant. So if the topic of a tweet is "Mobile" that is meaning the tweet discuss the advantages or disadvantages of the mobile like battery, sound or camera.

Usage: we are using topic modeling in our task to help us in polarity classification. When we say "the mobile battery life is too long" you will indicate that this is a good review but if we say "the meal in restaurant x is taking too long time to be ready" it is indicating that it is a bad review for the restaurant. To the adjective or property too long is good for the first

context but bad for the second. So we need to predict the topic for the giving review (tweet) to get a high chance of predicting the right class of tweet (positive, negative or natural).

3.5.2 Word weight

Any tweet consists of many numbers of words each word has its own importance in this technique it called **weight**, the words which are rare are the ones that actually help in distinguishing between the data and carry more weight. To calculate it we use TF-IDF technique

- **Increases** with the number of occurrences within a document.
- **Increases** with the rarity of the term in the collection.

3.6 Training:

Our Data divided to **training data** and **test data** that contains data of three general topics they **mobiles, restaurants and drinks** and in every general topic we have a specific topic (Pepsi, Coca-Cola, spectra, KFC, iPhone X,.....). And we want to divide train data into three files **restaurants data, mobiles data and drinks data**. And also divided test data to three files **restaurants data, mobiles data and drinks data**. to use it in training and test.

3.7 Methods

3.7.1 Word weight

Using word weight technique only machine learning approaches learn from all data and its sentiment. The topic of data not to be used in this method.

3.7.2 Topic modeling and Word weight

A combination of results Topic modeling and word weight techniques to predict the sentiment of the data We build Word weight model for each topic in data so each model focuses on the features of its topic, on the other hand, we build Topic modeling model on data.

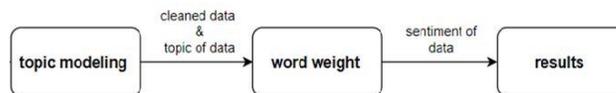


Figure 3: Topic modelling and word weight flow

After Topic modeling phase, we send predicted topics to Word weight models and each group of data has the same topic is predicted on separate model furthermore data is collected again

3.7.3 Merged columns

Our idea is to build a model was learned from 2 features (topic and sentiment). in the beginning, we tried to add the topic feature to the structure of Word weight (CRS matrix). So, we decided to combine two features into one column in the dataset. Sentiment & topic columns are combined so the number of classes becomes 9 classes instead of 3 it becomes positive, negative and neutral for each topic then machine learning approaches on this new row. this aims to make able to predict the sentiment of data with respect of topic and its specific features

3.8 Sarcasm

Sarcasm is saying has meaning but in a comedic way, to make fun of it or praise it nowadays most Twitter users use sarcastic way to represent their opinion and give a review on a product even if it is positive or negative review and to determine the sentiment of the sarcastic tweet

3.8.1 Sarcasm survey

Our goal is to produce the whole picture of the review and sarcasm is very important to see it, so we decide to combine results of sentiment analysis and sarcasm detection. Although there are no results or datasets our API combine these results, so we decide to make a survey and ask people on social media platforms about their opinions about how to combine these results.

● if tweet have negative meaning about a product but in a sarcastic wayso the meaning of tweet is?

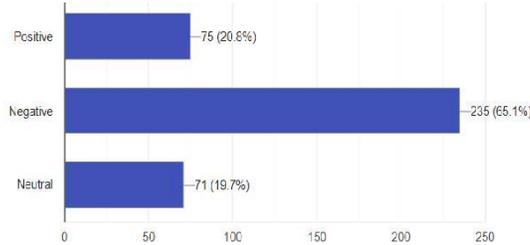


Figure 4: Survey results for negative

● if tweet have positive meaning about a product but in a sarcastic wayso the meaning of tweet is?

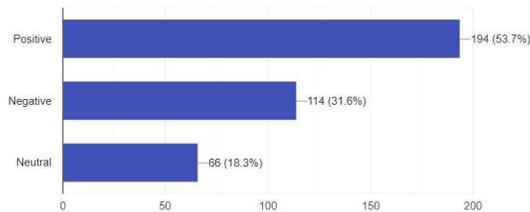


Figure 5: Survey results for positive

● if tweet have neutral meaning about a product but in a sarcastic wayso the meaning of tweet is

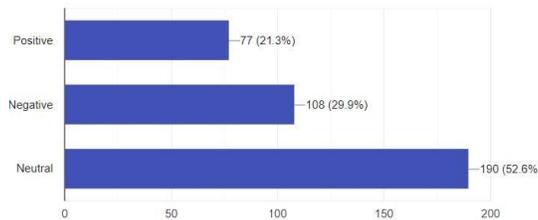


Figure 6: Survey results for neutral

3.8.2 Pre-trained model

In previous work they made sarcasm detection model also can deal with emoji they preprocessed the raw data using NLP. We would use other projects to reach our goal and not spend a lot of time and effort to reach what they have been reached already so we used a pre-trained model in sarcasm detection [7]

4- Results

1. Word weight

The first method we used is Word weight on all data without separating it into categories

We used word weight algorithm (TF-IDF) on all training tweets and build two models (Naïve Bayes and support vector machine (SVM)) and tested it from separated testing data and got the accuracy for each model:

SVM->68.8%

Naïve Bayes -> 63.11%

Which make SVM model is the best learning method in this approach.

But are these satisfying results?!

2. Word weight and Topic modeling

The second method we used dividing data into categories, and we used topic modeling to do this task ... What about topic modeling results?

□ Topic modeling

We made topic modeling according to three categories to separate on... these categories are:

Restaurants - Mobiles - Drinks,

And as we have the true categories, we started to calculate topic modeling accuracy according to three learning methods (support vector machine (SVM) - Naïve Bayes - Logistic Regression) on separated training data for topic modeling and we got these results:

SVM->97%

Naïve Bayes->95%

Logistic Regression->97%

So, we decided to use a logistic regression model to predict the topic in the next section.

□ Word weight for each topic

The first step done was separating data into categories manually without topic modelling then we used Word Weight to every category to get the best model to use.

We used 3 learning methods (Support vector machine (svm) - LogisticRegression - LogisticRegression with cross validation).

We separated data into 3 categories and tried the 3 learning methods on them to get scores. We depend on 4 criteria: Accuracy, recall, precision and F1 score.

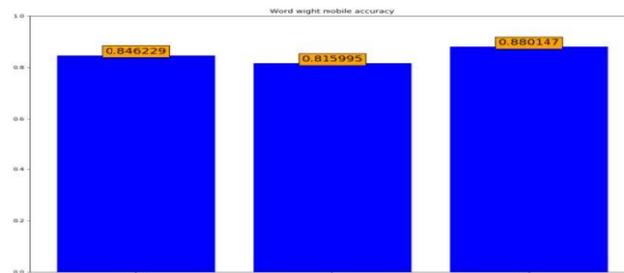


Figure 7: word weight mobile accuracy

This graph represents the accuracies of different models has been learned on mobile topic training dataset to achieve highest accuracy form the graph we can see that logistic regression cross-validation model get highest score 88%. From these results, we find that logistic regression cross-validation is the best model to use in prediction tweets of mobile topic as produces the highest score at 4 criteria

Second, we wanted to predict every tweet belongs to which topic (Not manually) so we used Topic modeling for this task, and we used logistic regression model. On testing Topic modeling model on our test data achieved accuracy 97.03%.

This graph shows the result of testing topic modelling on our test set as shown it predicted that

1. percentage of mobile tweets in dataset is 32.9% and the real is 33.5%
2. percentage of restaurants tweets in dataset is 36% and the real is 33.7%

3. percentage of drinks tweets in dataset is 32.8% and the real is 31.1%

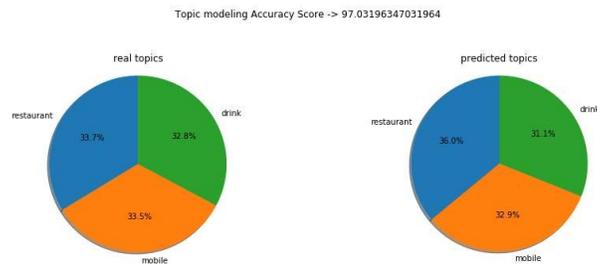


Figure 8: Topic modelling accuracy

After that we divided testing data according predicted topic and each group of data have the same predicted topic their sentiment is predicted by word weight model of this topic then we calculated the accuracy, precision, recall and F1 score for each topic:

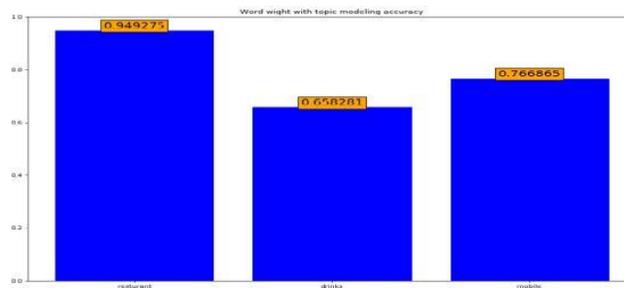


Figure 9: Word weight with topic modelling accuracies

This graph shows the result of each word weight model on its testing data with predicted topic as word weight model for: 1- restaurant data get 94.9% accuracy 2- drinks 65.8% accuracy 3- mobiles 76 % accuracy

From these results we calculated that the total accuracy for this method is 79%

3. Merging topic and sentiment

The third method we have as explained in section above is merged column as we have here 9 classes not 3, we used directly Word weight with three learning methods (Support vector machine (SVM) - Logistic Regression - Logistic Regression with cross-validation).

We calculated accuracy, recall and F1 score:

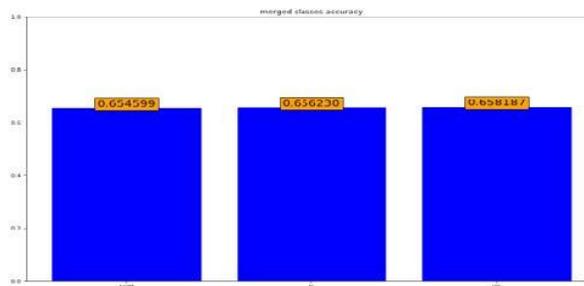


Figure 10: merging topics accuracy

This graph shows the accuracy of merged topics with word weight on its testing data with predicted topic: SVM gets 65% accuracy, logistic regression 65.6% and logistic regression with cross validation 65.8%

Best accuracy of logistic regression with cross validation in this method is

66%Not the Best

5 - References

- [1] Sentiment Analysis of Twitter Data: A Survey of Techniques:
<https://arxiv.org/ftp/arxiv/papers/1601/1601.06971.pdf>
- [2] Po-Wei Liang, Bi-Ru Dai, "Opinion Mining on Social Media Data", IEEE 14th International Conference on Mobile Data Management, Milan, Italy, June 3 - 6, 2013, pp91-96, ISBN:978-1-494673-6068-5, <http://doi.ieeecomputersociety.org/10.1109/MDM.2013>.
- [3] Pablo Gamallo, Marcos Garcia, "Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets", 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, Aug 23-24, 2014, pp 171-175.
- [4] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," in Proceedings of the 40th annual meeting on association for computational linguistics, pp. 417-424, Association for Computational Linguistics, 2002.
- [5] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138-1152, 2011.