Sentiment analysis using Supervised Learning Methods

Tushar Baratam
tushar.baratam97@hotmail.com
GITAM Institute of Science, Visakhapatnam, Andhra Pradesh

ABSTRACT

As the organisations are getting bigger and bigger the need to make well informed decisions are ever more important. To do that however you need data. But is data enough. How is one to interpret the data to their advantage? How is one meant to sift through all of the data to find the specific ones that are meant to be relevant to the task at hand. For that you need a all-in-one solution. One that can handle all of that for us. This project targets the online product reviews. For an organisation that aims to sell products through online platforms like Amazon, Flipkart and other require a way to analyse all the product reviews to get information on what is performing well and what isn’t. For that it is important that the sentiments are analysed where it predicts where the review is positive or negative and what aspects of the review are negative or positive. It needs to give a numerical and visual representation which is helpful when it required to present the data to someone. All of these issues are addressed with the help of this application. It is built in python using pre-existing well-known libraries. Which also means that it is verified by many.

Keywords— Sentiment Analysis, Opinion mining, Logistic Regression, Machine Learning, Natural Language Processing

1. INTRODUCTION

There is a Natural Language Processing from Google which is considered to be one of the best in the market available. There are others that are also available. Given a piece of written text, the problem is to categorize the text into one specific sentiment polarity, positive or negative (or neutral). Based on the scope of the text, there are three levels of sentiment polarity categorization, namely the document level, the sentence level, and the entity and aspect level. In recent years the need for sentiment analysis has grown as we fishing for more sentiments and reviews from end users the requirement for analysing the text is higher than ever. The document level concerns whether a document, as a whole, expresses negative or positive sentiment, while the sentence level deals with each sentence’s sentiment categorization.

2. PROPOSED APPROACH TO BE EMPLOYED

Sentiment Analysis or Opinion Mining is a study that attempts to identify and analyse emotions and subjective information from text since early 2001, the advancement of internet technology and machine learning techniques in information retrieval make Sentiment Analysis becomes popular among researchers. Besides, the emergent of social networking and blogs as a communication medium also contributes to the development of research in this area Sentiment analysis or mining refers to the application of Natural Language Processing, Computational Linguistics, and Text Analytics to identify and extract subjective information in source materials. Sentiment mining extracts attitude of a writer in a document includes writer’s judgement and evaluation towards the discussed issue. Sentiment analysis allows us to identify the emotional state of the writer during writing, and the intended emotional effect that the author wishes to give to the reader. In recent years, sentiment analysis becomes a hotspot in numerous research fields, including natural language processing (NLP), data mining (DM) and information retrieval (IR) This is due to the increasing of subjective texts appearing on the internet. Machine Learning is commonly used to classify sentiment from text. This technique involves with statistical model such ad Support Vector Machine (SVM) , Bag of Words and Naïve Bayes (NB) [1].The most commonly used in sentiment mining were taken from blog, twitter and web review which focusing on sentences that expressed sentiment directly .The main aim of this problem is to develop a sentiment mining model that can process the text in the mobile reviews.

3. METHODOLOGY

• Acquiring sentiments for prediction: User will provide the program with a .csv file of all the sentiments that need to be analysed. This is done by QFileDialog.Options() which comes with the PyQt5 library.

• Selecting the appropriate training data: Selecting the appropriate training data for the sentiments are important as the machine learning will be train itself accordingly. For instance, if the user provides online product reviews and selects the training data for hospital reviews then the results might not be what you expect.
• **Splitting of Training data into Train and Test data:** For our purposes we have split the data in to 70% train and 30% testing data. This is done by train_test_split() which comes with the sklearn library.

• **Language processing:**
  - Removal of stopwords. Stopwords are the words in any language which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence.
  - Converting all text to lowercase which will help in the vectorising.
  - TfidfVectorizer is used and is what is the actual data that the model learns on. TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. It has many uses, most importantly in automated text analysis, and is very useful for scoring words in machine learning algorithms for Natural Language Processing (NLP).
  - A list of all the vectorised statements and another list is created with their sentiment whether it is negative or positive is created using vectoriser.transform().

• Now that the model is up and running the user data is vectorised and fed into the model to predict its sentiments. The sentiments are predicted and formed into a list.

• Keyword analysis is done on this data. It includes a Word cloud or most used words, a frequency graph of most common words and small widget to calculate what percentage of sentiments include that particular keyword that user is might be looking for. It does these three things for all verbatim, negative verbatims and positive verbatims.

• All of the keyword analysis is then exported into a .csv file and the metrics are exported as .png image as final output.

4. **TESTING**

One of the first things to test is the model’s metrics. To do this we do the following steps.

• The testing data is vectorised and converted into a list where it is not provided with the sentiments. The model will predict using what is has learned with the training data previously and will create a corresponding list for the list of testing vectorised data.

• The learning metrics are created using the results of the test predictions of the sentiments that were predicted by the model. It will calculate the following with respect to the actual sentiment with the predicted sentiment. They are True positive (where the actual sentiment and the prediction by the model is true), true negative (where the actual sentiment is positive however the prediction is negative), false positive (where the actual sentiment is negative however the model predicts it to be positive) and false negative (where the actual sentiment and the prediction are negative). The ideal model metrics would have none in the false positive and true negative in the results. Following metrics are calculated for this purpose.
  - Recall = (True positive + True Negative)/Total
  - Precision = (True positive)/(True positive +False Negative)
  - F1 score = 2*(Precision *Recall) / (Precision + Recall)

• The need to calculate F1 score is important for various reason. One is that it is not enough to judge a model with just its precision and its recall values. Another reason is that most models out there follow the same metrics more or less to be competitive.

• We can calculate these results once we have the confusion matrix. It gives is the values for True negative, True positive, False Positive and False negative.

Test title: accuracy
Test priority: high
Test objective: to ensure that the software gives correct sentiment for a given verbatim

**Description:** the code for the sentiment analysis is isolated and it given a string to analyse after the model is trained. The given verbatim must give the correct sentiment after analysis.
Now the metrics can be calculated and the results are shown below.

5. OUTCOMES
This shows us model’s competency and is a good indicator on its performance.

Test Case 2:
Test title: accuracy
Test priority: high
Test objective: to ensure that the software gives correct sentiment for a given verbatim
Description: the code for the sentiment analysis is isolated and it given a string to analyse after the model is trained. The given verbatim must give the correct sentiment after analysis.

6. RESULTS
The results are exported to a .csv document which shows the verbatim with the sentiment along with the model’s confidence for solving it. The lower score suggest that the model wasn’t so sure when making the prediction.
7. CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

This system was able to predict sentiments of user provided verbatim with a fairly high confidence with good accuracy. Furthermore, the model had a fairly high F1 score, a metric that allows it to compete with other NLP machine learning models available in the market. Furthermore, the ability to detect sarcasm with moderate precision is something that is currently lacking in the market. The flexibility provided with the system is immense and the portability of .csv files is also a huge bonus. It is not always practical to run a database server on every system wishing to run this application just for doing sentiment analysis on handful of verbatims. However, when this application is considered for a sentiment analysis of, let’s say, 10,000 verbatims or even a million verbatims it is easily able to handle that as there is no limit on the file size of the .csv file format and is only restricted by the ram capacity of the system which is at least a gigabyte. Allowing users to select their own training data is also a huge bonus as the model is only accurate as they data it is fed. When the training data has the similar context to the verbatim selected by the user it is certain that the learning model produces a highly accurate and precise prediction.

7.2 Future scope

As the application made to very flexible with its inputs the code can be easily modified to accept any type of file format and for data. It is also a possibility to scrap data from internet webpages and perform sentiment analysis on them directly. The code for sentiment analysis need not change. It is definitely possible to change the GUI for a more attractive version.

8. REFERENCES

[5] Aparoo Agarwal, Boyi Xie, Illia Vovsha, Owen Rambow, Rebecca, Passonneau, “Sentiment Analysis of Twitter Data”, Columbia University, New York, NY 10027 USA