International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

Volume 8, Issue 5 - May 2020 - Pages 29-36

Sentiment Analysis

[1]Shashidhar T Halakatti [2]Sangamesh S K, [3]Pavitra M Gadhar
Assistant Professor
RTE Society's Rural Engineering College Hulkoti, Karnataka-582101, India

Abstract –Sentiment Analysis refers to the task of Natural language processing to determine whether a piece of text contains some subjective information and what subjective information it expresses, i.e. whether the attitude behind this text is positive, negative, or neutral. Understanding the opinions behind user's generated content automatically is of great help for commercial and political use, among others. The task can be conducted on different levels, classification of polarity of words, sentence or entire document

KEYWORDS: Emotions,Positive,Negative, SVM.

I. INTRODUCTION

Machine learning is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within Machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, Machine learning is also referred to as predictive analytics.

II. SENTIMENT ANALYSIS

The world has revolutionized and phased into a new era, an era which upholds the true essence of technology and digitalization. The transformed and highly intelligent data mining approaches now allow organizations to collect, categorize, and analyze users' reviews and comments from micro-blogging sites regarding their services and products. This type of analysis makes those organizations capable to assess, what the consumers want, what they disapprove of, and what measures can be taken to sustain and improve the performance of products and services. This study focuses on critical analysis of the sentiment analysis by using SVM (support vector machine). Sentiment analysis refers to the task of natural language processing to determine whether a piece of text contains some subjective information and what subjective information it expresses i.e whether the attitude behind this text is positive, negative or neutral. Natural language processing is a part of Machine learning where it is a field of Computer Science that uses statistical techniques to give computer systems the ability to learn with data, without being explicitly programmed.

Emoticons

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 8, Issue 5 - May 2020 - Pages 29-36

Emoticons are considered to be handy and reliable indicators of sentiment, and hence could be used either to automatically generate a training corpus or to act as evidence feature to enhance sentiment classification.

Emoticons are introduced as expressive, non-verbal components into the written language, mirroring the role played by facial expressions in speech. Their role is mainly pragmatic: Emoticons give a positive or negative sense to written sentences by a visual expression. According to this consideration, there is a relationship between the sentiment orientation of emoticons and messages. Emoticons have been distinguished in two main categories, i.e. positive and negative. Instances of positive emoticons are :-), :), =), :D, while instances of negative ones are :-(, :(, =(, ;(. These emoticons surely are an important source of information for polarity classification. In fact, on social media, positive and negative messages have a high percentage of emoticons.

Emoticons in Sentiment Analysis

Figure 1: Emoticons in sentiment analysis

Using emoticons to boost sentiment analysis

Information contained in Social media, which became one of the major types of communication, make them an attractive source of data for sentiment analysis. Not only texts but also emoticons, which represent linguistic elements typically used on social media to elicit a given message, can be used to boost sentiment analysis.

The precision of recognizing emotions can increase and improve with the analysis of emoticons. They provide a crucial piece of information and this is essential for companies in order to better understand their customer's feelings. The way people communicate is constantly changing, Thanks to technology; in order to understand what they are saying, it is important that companies adjust the way they are listening to suit these changes, and this means taking also emoticons into account.

III. PROBLEM STATEMENT/IDENTIFICATION

The problem in sentiment analysis is classifying the polarity of a given text or at the document, sentences, or feature/aspect level. Whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral.

To implement an algorithm for automatic classification of text into positive, negative, or neutral. Sentiment analysis to determine the attitude of the mass is positive, negative or neutral towards the subject of interest. Graphical representation of the algorithm to be implemented.

→Sentence Level Sentiment Analysis: Given a message containing a sentence, a word or a phrase, to determine its sentiment in that context.

→Emoticon-Hashtag Level Sentiment Analysis: Given a message containing hashtags or

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 8, Issue 5 - May 2020 - Pages 29-36

emoticons instance of a word or a phrase, to determine its sentiment in that context.

IV. RELATED WORK

An Introduction to Sentiment Analysis- In the last decade, sentimental analysis is also known as opinion mining, has attracted an increasing interest. It is a hard challenge for Language Technologies, and achieving good results in much more difficult than some people think. The automatically classifying a text written in a Natural language into a positive or negative feeling, opinion or subjectivity, is sometime so complicated that even different human annotators disagree on the classification to be assigned to a given text. An important part of our information-gathering behavior has always been to find out what other people think. With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people can, and do, actively use information technologies to seek out and understand the opinions of others. The sudden eruption of activity in the area of opinion mining and sentiment analysis, which deals with the computational treatment of opinion, sentiment, and subjectivity in text, has thus occurred at least in part as a direct response to the surge of interest in new systems that deal directly with opinions as a first-class object.

V. MOTIVATION FOR THE WORK

SUPPORT VECTOR MACHINE

Support Vector Machine" (SVM) is a Supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well.

Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

VI. METHODOLOGY

Our project involves the usage of SVM algorithm to analyze real time reviews. The objective of our case study is to find the polarity of the words (in data) retrieved. Each step in the framework involves several sub-tasks.

1. Data collection: Data in the form of raw is retrieved which provides a package for real time streaming API. The API requires us to filter data by using keywords. Filters supports to retrieve data which match a specific criterion defined by the developer. We used this to retrieve data that is related to specific keywords which are taken as input from users. Initially, the data is taken into consideration in application mode. We execute the program .Then, input array of keywords is provided .For example, on inputting multiple keywords like, 'Awesome', 'Terrible', the output we obtained from 15 seconds' window time was the live stream of data associated with these keywords. Only caveat of using filters is that famous keywords like "India" have more tweets compared to niche words like "Focusrite" which makes it difficult to get data for niche specific keywords.

2. Data Processing: Data processing involves Tokenization which is the process of splitting the raw data into individual words called tokens. Tokens can be split using whitespace or punctuation characters. It can be unigram or bigram depending on the classification model used. The bag-of words model is one of the most extensively used model for classification. It is based on the fact of assuming text to be classified as a bag or collection of individual words with no link or interdependence. The simplest way to incorporate this model in our project is by using unigrams as features. It is just a collection of individual words in the text to be classified, so, we split each raw

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 8, Issue 5 - May 2020 - Pages 29-36

data using whitespace. For example, the data "Met Aziz today!!" is split from each whitespace as follows. {Met Aziz!!"} The next step in data processing is normalization by conversion of data into lowercase. Data are normalized by converting it to lowercase which makes its comparison with a dictionary easier.

3. Data Filtering: A review acquired after data processing still has a portion of raw information in it which we may or may not find useful for our application. Thus, these raw data is further filtered by removing stop words, numbers and punctuations. Stop words: For example contain stop words which are extremely common words like "is", "am", "are" and holds no additional information. These words serve no purpose and this feature is implemented using a list stored in file. We then compare each word in a review with this list and delete the words matching the stop list. Removing non-alphabetical characters: Symbols such as "#@" and numbers hold no relevance in case of sentiment analysis and are removed using pattern matching. Regular expressions are used to match alphabetical characters only and rest are ignored. This helps to reduce the clutter from the data stream.

4. Feature Extraction: TF-IDF is a feature Vectorization method used in text mining to find the importance of a term to a document in the corpus. Feature extraction involves library. The recommended API is the Data Frame based API. This feature is useful for a case where we need to find trending topics or to create word clouds. However, this project is more focused towards finding sentiment in data streams so TF-IDF is implemented for considering the unique words in the data and analyzing its sentiment.

3.1. How does it work?

When we are looking for purchasing a product, then a will look for the reviews of the product. This at the customer level. Now let us consider the company level, company should understand whether the product need some change, to increase their selling in market level.

Let us consider a statement *"The Movie was great!"*

Step 1: Tokenization

Tokenization is dividing a statement into different set of words. Now the statement, the movie was great is segregated into different words.

 The Movie Was Great !

Step 2: Cleaning data

Cleaning the data, this means remove all special characters or the words which do not add any meaning to the analysis parts.

 The Movie Was Great

Step 3: Removing stop words

Removing the stop words, like said earlier we do not need any words which do not add any meaning to the analysis part.

 Movie Great

Step 4: Classification

So that we have left with just two words the task will be to classify whether the words are positive, negative or neutral word. For a positive word the sentiment score is +1. For a negative word the sentiment is -1. For neutral word the sentiment is 0.

We can model/train our data with a bag of words or even use lexicons (dictionary of pre classified set of words).

And we will also classify the data based on more accuracy score.

Step 5: Conclusion

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 8, Issue 5 - May 2020 - Pages 29-36

Now we have just two words which finally add some meaning to the analysis part. Hence Movie (0) + Great (+1) =1. Since the polarity is greater than 0, so the given statement is positive.

Flowchart for the Proposed work:

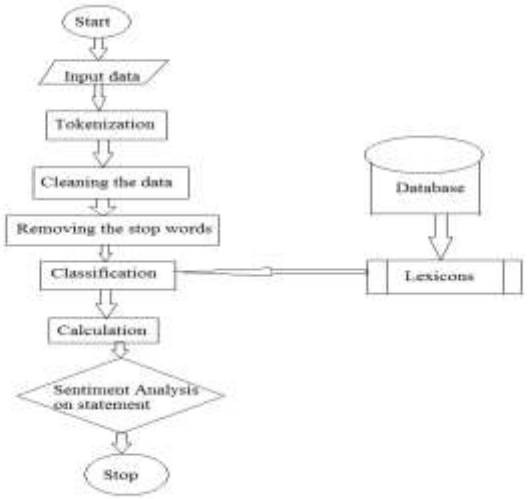


Fig 1: Flow chart of Sentiment Analysis

VII. IMPLEMENTATION

The first step in modeling the document into a vector space is to create a dictionary of terms present in documents. To do that, you can simple select all terms from the document and convert it to a dimension in the vector space, but we know that there are some kind of words (stop words) that are present in almost all documents, and what we're doing is extracting important features from documents, features to identify them among other similar documents, so using terms like "the, is, at, on", etc.. Isn't going to help us, so in the information extraction, we'll just ignore them.

Let's take the documents below to define our algorithm:

Train Document Set:

1. ***d1***: The sky is blue.
2. ***d2***: The sun is bright.

Test Document Set:

3. ***d3***: The sun in the sky is bright.
4. ***d4:*** We can see the shining sun, the bright sun.

Now, what we have to do is to create index vocabulary (dictionary) of the words of the train document set, using the documents ***d1*** and ***d2*** from the document set, we'll have the following index vocabulary denoted as ***E(t)*** where the ***t*** is the term: Note that the terms like "is" and "the" were ignored as cited before. Now that we have an index vocabulary, we can convert the test document set into a vector space where each term of the vector is indexed as our index vocabulary, so the first term of the vector represents the "blue" term of our vocabulary, the second represents "sun" and so on. Now, we're going to use the **term-frequency** to represent each term in our vector space; the term-frequency is nothing more than a measure of how many times the terms present in our vocabulary ***E(t)*** are present in the documents ***d3*** or ***d4*** , we define the term-frequency as a counting function:

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 8, Issue 5 - May 2020 - Pages 29-36

tf (t, d) $=\Sigma fr(x,t)_{x\in d}$
Where the fr (x,t) is a simple function defined as:
fr (x,t) = { 1 ,$if\ x=t$ 0,$otherwise$
So, what the ***tf(t,d)*** returns is how many times is the term ***t*** is present in the document ***d***. An example of this, could be ***tf("sun",d4)*** since we have only two occurrences of the term "sun" in the document ***d4***. Now you understood how the term-frequency works, we can go on into the creation of the document vector, which is represented by: Each dimension of the document vector is represented by the term of the vocabulary, for example, the ***tf(t1,d2)*** represents the frequency-term of the term 1 or ***t1*** (which is our "blue" term of the vocabulary) in the document ***d2.***
Let's now show a concrete example of how the documents ***d3*** and ***d4*** are represented as vectors: which evaluates to:
As you can see, since the documents ***d3*** and ***d4*** are:

$$\vec{v_{d3}} = (0, 1, 1, 1)$$
$$\vec{v_{d4}} = (0, 2, 1, 0)$$

1. ***d3***: The sun in the sky is bright.
2. ***d4***: We can see the shining sun, the bright sun.

The resulting vector shows that we have, in order, 0 occurrences of the term "blue", 1 occurrence of the term "sun", and so on. In the , we have 0 occurrences of the term "blue", 2 occurrences of the term "sun", etc.
But since we have a collection of documents, now represented by vectors, we can represent them as a matrix with |D| * F shape, where |D| is the cardinality of the document space, or how many documents we have and the F is the number of features, in our case represented by the vocabulary size. An example of the matrix representation of the vectors described above is:
As you may have noted, these matrices representing the term frequencies tend to be very sparse (with majority of terms zeroed), and that's why you'll see a common representation of these matrix as sparse matrices.
Environment Used: Python, Numpy, Scipy, Sklearn :
In this section we use Python to show each step of the tf-idf calculation using the Scikit.learn feature extraction module.
The first step is to create our training and testing document set and computing the term frequency matrix:

```
1. From sklearn.feature_extraction.text import CountVectorizer.")
2. train_set=("The sky is blue.","The sun is bright.")
3. test_set=("The sun in the sky is bright","We can see the shining sun, the bright sun.")
4. count_vectorizer = Countvectorizer()
5. count_vectorizer.fit_transform(train_set)
6. Print "Vecabulary:", count_vectorizer.vocabulary
7. #Vocabulary: {'blue': 0, 'sun': 1, 'brihgt': 2, 'sky':3}
8. Freq_term_matrix=count_vectorizer.transform(test_set)
9. Print freq_term_matrix.todense()
10. #[[0 1 1 1]
11. #[0 2 1 0]]
```

International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 8, Issue 5 - May 2020 - Pages 29-36

Now that we have the frequency term matrix (called freq_term_matrix), we can instantiate the Tfidf Transformer, which is going to be responsible to calculate the tf-idf weights for our term frequency matrix.

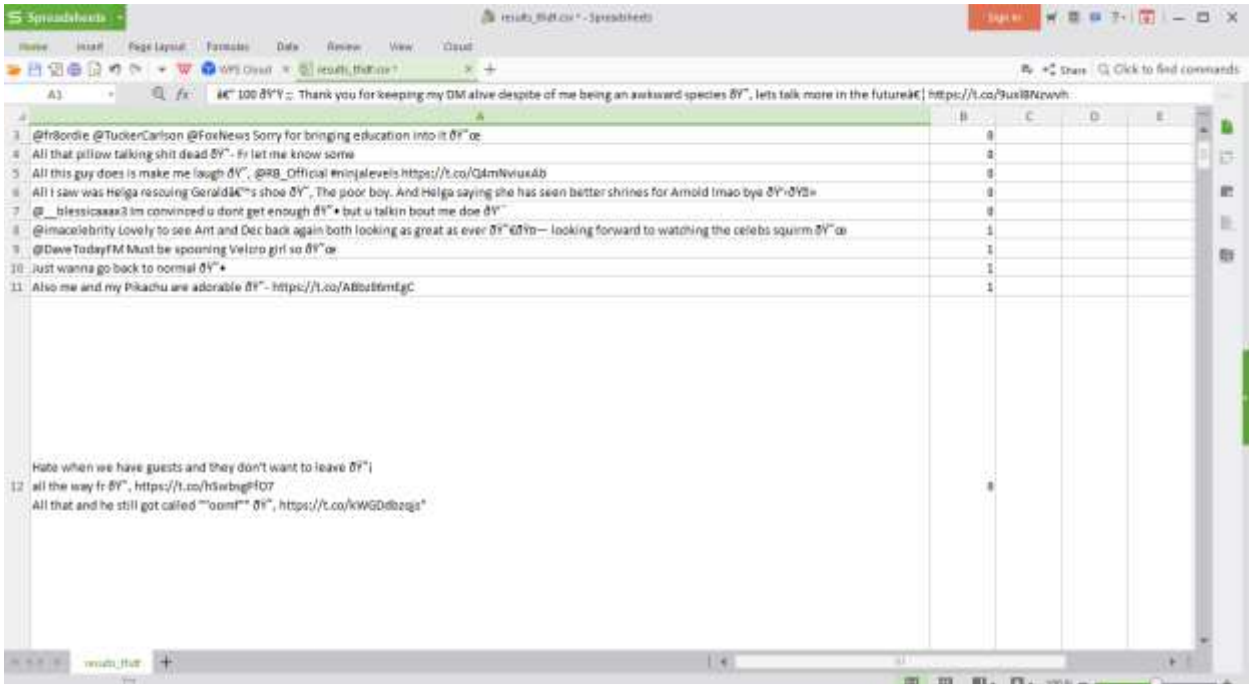
```
1. from sklearn.feature_extraction.text import TfidfTransformer
2. tfidf=TfidfTransformer(norm="12")
3. tfidf.fit(freq_term_matrix)
4. print "IDF:", tfidf.idf
5. #IDF: [0.69314718 -0.40546511 -0.4054645110]
```

Note that we have specified the norm as L2, this is optional (actually the default is L2-norm), but we have added the parameter to make it explicit to you that it's going to use the L2-norm. Also note that you can see the calculated idf weight by accessing the internal attribute called **idf_**. Now that **fit ()** method has calculated the idf for the matrix, let's transform the **freq_term_matrix** to the tf-idf weight matrix:

```
1. tf_idf_matrix=tfidf.transform(freq_term_matrix)
2. print tf_idf_matrix.todense()
3. # [[0. -0.70710678 -0,70710678 0.]
4. # [0. -0.89442719 -0.4472136 0.]]
```

And that is it, the **tf_idf_matrix** is actually our previous matrix. You can accomplish the same effect by using the **Vectorizer** class of the Scikit.learn which is a vectorizer that automatically combines the **CountVectorizer** and the **TfidfTransformer** to you. See this example to know how to use it for the text classification process.

VIII. SNAPSHOTS



International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Available online at: www.jrrset.com
ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395
Volume 8, Issue 5 - May 2020 - Pages 29-36

IX. CONCLUSION

In this paper we have been carried out to enhance the accuracy of prediction by using Sentiment Analysis techniques. Sentiment Analysis or Opinion Mining is computational treatment of opinions, sentiment and subjectivity text. This is branch of Natural language processing which aims to determine the attitude of writer with respect to some topic or the overall contextual polarity of a document. Opinion Mining is process of extraction knowledge from opinion of others about some particular topic or a problem. This work investigates whether sentiment analysis of public mood derived from large documents of feedback can be used to identify important events and predict movements of that particular topic.

X. REFERENCES

1) Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu National Research Council Canada Ottawa, Ontario, Canada K1A 0R6 {saif.mohammad, svetlana.kiritchenko,xiaodan.zhu}@nrc-cnrc.gc.ca
2) (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 2, 2018, Munir Ahmad1, Shabib Aftab2, Muhammad Salman Bashir3, Noureen Hameed4
Department of Computer Science, Virtual University of Pakistan Lahore, Pakistan.
3) International Journal of Computer Science Trends and Technology (I JCST) – Volume 4 Issue 3, May - Jun 2016, Bholane Savita D., Prof.Deipali Gore Department of Computer Science and Engineering Savitribai Phule, Pune University Pune – India.
4) 2014 IEEE 2014 International Conference on Computer, Communication, and Control Technology (I4CT 2014), September 2 - 4, 2014 - Langkawi, Kedah, Malaysia Nurulhuda Zainuddin and Ali Selamat Faculty of Computing Universiti Teknologi Malaysia 83100 Johor Bahru Johor, Malaysia.