ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

Sentiment analysis of informal text using a rule based model

M.S. Abirami* M. Uma, Manasa Prakash
Department of Software Engineering, SRM University, Chennai, India
***Corresponding author: E-Mail: abirami.ms@ktr.srmuniv.edu.in**

ABSTRACT

Past research on phrase level sentiment analysis has mainly involved parts of speech tagging; Identifying the different parts of a sentence as nouns, adjectives, adverbs is appropriate and individually considering each such part to compute an aggregate score to decide on the sentiment polarity. Most of these results are binary or ternary, and although they are accurate to an extent, difficulties in working with raw social media content are observed. This paper proposes an alternative approach to sentiment analysis by computing the polarity, the intensity by looking at various phrase level features such as the aspect, and takes into consideration the presence of over one opinion, emoticons and deliberate spelling errors. The algorithm initially identifies the nature of opinion, base or comparative, and then the aspect, the above mentioned features to determine the polarity and strength. The evaluation of this algorithm is done by comparing the results to a baseline produced by manual scoring.

Keywords: Sentiment Analysis, NLTK, CMU POSTagger, AlchemyAPI

1. INTRODUCTION

The Web that we experience today is more than a network to just share and exchange information- it's a major intelligence tool. It's no longer just about putting data out and receiving it. The amount of data posted by the general public is humongous, Facebook for instance gets over 100 petabytes of data a day and Twitter gets 6000 tweets every second. With such excess information, it makes sense to extract meaningful data and make smart use of it. We have engines working on that and applying it to various spheres of web experience- the applications range from giving product or network recommendations they consider information like PC usage, general blog posts, websites viewed recently to nation-wide opinion analysis, election propaganda etc. Sentiment Analysis happens to be one such major application.

Interests in developing improved opinion mining algorithms for accuracy and developing a more efficient understanding of the dynamics of the human sentiment have only increased. This paper addresses three major issues in analysing social media content- computing the intensity of a polarity, identifying the more effective sentiment in case of mentioning of over one and context dependency on opinion words. This task is defined as a text classification, ranking problem with the goal of producing mathematical score for the sentiment observed. The idea presented here takes into consideration every aspect of the text and uses it to compute the final score - social media tendencies such as emoticons, punctuation, casing, vowel stretches, abbreviations, interjections and context in which the opinions are mentioned.

Twitter is used by people for various reasons (business, personal, emotional observation etc.) and hence analysis of tweets has proved to be highly interesting. A corpus has been used to collect tweets, analysed manually to study language usage. A set of rules have been made from the clues collected and tweets are scored for the sentiment expressed. The project hopefully opens up to rigorous training and commercial usage, apart from inspiring more research insights. This paper is broken into four main categories: Data collection, Investigation, Implementation and Evaluation. This paper focuses on studying the existing solutions on sentiment analysis and performing a detailed reasoning on the limitations of each and then implementing suggested solutions and comparing the performances of the models. The study extensively focused on analysis common patterns in informal web content.

The paper aims to identify if the body of the text can be used to identify the emotional tone of the user through polarity classification as follows:

1. Identify the ternary results: positive, negative, neutral scores (-5 to +5 for each aspect)
2. Identify the compound nature of the text. (If there are mixed emotions)
3. Identify a single scale result: (-5 to +5 for the overall aspect)

The algorithm focuses on short informal text expressed on social media. The decisions are based by remarks and identifications by manually scoring a set of tweets and subsequently use these results to score the next set of tweets. The idea is that terms that occur and are in a sentence scored with a high score, would get a bigger score. The algorithm is written as a python code runnable through the command line. The final prototype however is developed using a desktop application interface. The final implementation is a strict demo that takes a user input and gives the results as specified above. The tweets are stored to the system's data folder, from which the dictionaries are also taken. Future goals are to implement a strong classifier over the rules and enable API use in commercial or research based domains.

Background and related work: Sentiment analysis has been worked on for well over a decade. A variety of algorithms have been used and most of them follow two approaches:

Keyword Analysis: Computing the emotion associated with each word NLP- computing the emotion with words as well as nature of language and other features such as dependencies. For example, an algorithm based on pos (part of speech) parser.

The former has the advantage of being simple and predictable but gives limited, inaccurate results. The latter may be difficult to implement but has a better performance. The algorithms used may be classified into two approaches and further sub classified as follows:

1. Lexicon/ Dictionary Based
2. Machine Learning Based

Lexicon Based techniques work on an assumption that the collective polarity of a document or sentence is the sum of polarities of the individual words or phrases. For example Word Net was used by Andrea (Andrea Esuli, 2005). Machine learning approach was initially used by Melville, Rui, Ziqiong, Songho, Qiang and Smeureanu (Ion SMEUREANU, 2012). This method takes a set of training data with phrases and pre-computed sentiments and attempts to train the algorithm to apply it on a set of unknown test data.

The issues faced in the current scenario may be summarized as follows:

Accuracy : Almost all algorithms are accurate up to 70%. This is because of the following reasons:

Language Issues: Negation word can reverse the polarity of any sentence as shown by (Michael Wiegand, 2010). Blind negation and intensification can also play a role**.** Acronyms, jargons and emoticons also play a role in the computation of sentiment (Kevin Gimpel, 2011). For example, the movie eludes a clear lack of respect for women.

Context Issues: Certain words have different meaning in different contexts and sometimes, each sentence contains more than one opinion. Multiple opinions again, may be about the different aspects of one subject, or about same aspects of different subjects or different aspects of different subjects. For example, The script is average but the acting is good!

Domain Issues: Words have different connotations in different domains. For example, There was little music, which I appreciated. But the screen time was too little.

Subjectivity Issues: Opinions are subjective in nature. Sarcasm or irony may be expressed. Phrases that may be positive to entity may not be positive for another. For example, That was too nice of you. Hmm.

Performance : The performance faces two main challenges:

Data volume: ML methods not too scalable

System nature: Usually tested in a standard environment but implemented in different kinds.

Time : Many approaches are time consuming in terms of

Implementation: Resource extraction, training time for ML methods.

Testing: Finding or extracting the data set.

General challenges: Opinions being subjective in nature may be confusing even to humans. Sentences may contain both positive and negative polarities with different impacts as shown in Figure 1. They may be grammatically or structurally incorrect. The strength of opinions is generally not understood.

This study focuses on defining a better approach for analyzing informal texts keeping in consideration the social-media nature of the texts involved. The aim is to provide a result that indicates the strength of each polarity in a sentence and also it's richness in opinion.

1. Anaphora Resolution: The problem of identifying the pronoun or the noun
2. Sentiment towards an entity is not consistent.
3. Informal language
4. Implicit Statements
5. Ungrammatical occurrences
6. Detection of depth of sentiment
7. Identifying positivity/ negativity of subjects
8. High level of subjectivity which is difficult to deal with even manually

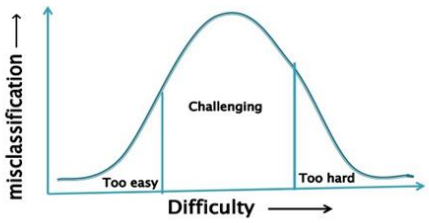


Figure.1.Learning Curve

This research deals with the mentioned issues using a combination of various existing tools and technologies, and presenting the results in a different format.

Data collection and investigation: Data is the core of any data mining problem. The importance of good data is normally understated. Most data found on social media is mostly noise and it's important that we clean and process it before use (Finn Arup Nielsen, 2011).

It's important to use good data in 4 stages:

1. Dictionary or the lexical resource used
2. The data used to train on
3. The data used to test on
4. The data used to implement on

This study has used data from Twitter as tweets are simple to access and provide real opinions of people. Being a micro blogging site it is rich in both expert and naive opinions. The tweets follow usage of typical web jargons and rely on extensive use of punctuation, casing and emoticons.

Algorithm: Streaming Twitter

Import tweety
Public _tweets=tweety api:public_timeline()
For teet in public_twwets:
Print tweet,text

The dictionary used for the analysis has been borrowed from a study conducted by Finn Arup Nielsen in 2009-2011- it is called the 'AFINN', along with words found using the method of web scraping.

Algorithm: Scraping Method

```
import urllib2 #to fetch the web page
from bs4 import BeautifulSoup #to fetch url elements

#specifying the url
online_slang = str(raw_input("Enter url:"))
#querying the website
page = urllib2.urlopen(online_slang)
#parse the page
soup = BeautifulSoup(page)
#test: print the structure
#print soup.prettify()

for div in soup.findAll('b'):
    title = div.a.string
    print (title)
```

Classification: The Classification approach as shown in Figure 2 is focused on two tasks:

a. Obtain a dictionary containing emoticons and abbreviations and score them.

b. Analyze phrases using the custom dictionary as well as the dictionary containing words and scores from AFINN.

To obtain the dictionary, tweets were collected for a week and then stripped for non-literal words. They were then automatically and manually examined for being emoticons or abbreviations and then scored empirically. To analyze the phrases, a rule based system that combines both keyword and natural language processing methodologies is suggested. The algorithm mentioned as a pseudo-code below is used. It focuses on looking at the phrases in two forms- one as a normal sentence and another as a sentence with the irregular social media tendencies. It also deals with negation and words that change the polarity. It checks for use of special case idioms and take into account the use of intensifiers, diminishers, irregular punctuation and casing. It also deals with words not present in the dictionary by looking at words around it and scoring accordingly. The algorithm not only scores the words individually, but changes the score according to the word combinations. Example: The movie is TOO good and funny!!

Algorithm: Classification

```
for text in collection:

    terms = text.split()
    words = removeNonLiterals(text).split()
    score = 0

    for each word in words:
        v=0
        if word.length > 1:
            isTerminDict: v = getDictScore(text)
        else:
            isTermPreceededByDiminisher(text): v = reduceScore(v, preceedingWords)
            isTermNegative: v = adjustScore (v)
            for each in permutations(prev_and_word AND prev_word_and_next) in idioms: v =
                adjustScore(v)
            isButPresent: v = adjustScore(v)

    for each term in terms:
        isTermPunc:
            isExclamation:
                isCount > 4: +=ep_amplifier
            isQuestion:
                isCount > 2: +=qm_amplifier

    if v > 0: v+=(ep_amplifier+qm_amplifier)
    else: v-=(ep_amplifier+qm_amplifier)

    pos, neg, neu, com = splitScore(v)

    return
```

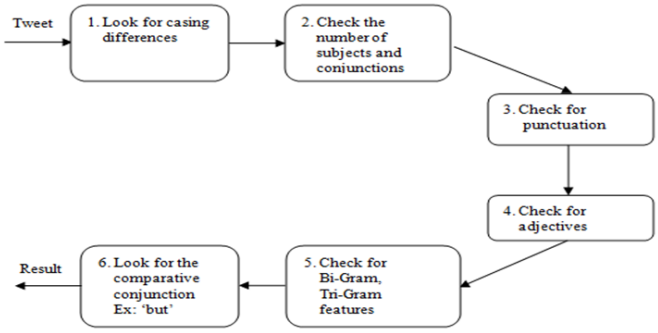
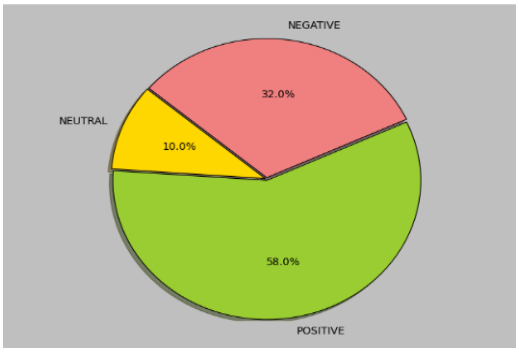


Figure.2.Classification Flow diagram

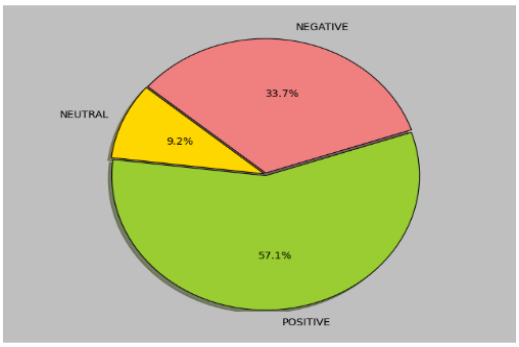
Evaluation: The performance of the algorithm was checked against the baseline set of manually scored set of tweets. The purpose of this algorithm was quickly computing the sentiments of data sets without being affected by the informal tendencies in social media language. The program was executed on a laptop with 4 GB RAM, running on an i7 CORE Intel processor. 100 tweets are used for the analysis. They were manually annotated for comparison; these scores were taken as the baseline. It was observed that the rule based approach took 2.69 seconds to perform. The accuracy for the models was comparable with respect to the manually scored tweets.

Table.1.Results

Method	Sample	Time	Accuracy
Rules Based	100	2.69	Comparable
Rule Based	Paragraph	0.96	Comparable
Manual Scoring	100	N/A	Comparable
Manual Scoring	Paragraph	N/A	Comparable



(a) Manual Scoring



(b) Rule Based Scoring

Figure.3.Scoring Breakdown

The issues with this breakdown (Figure 3) method are:

1. The algorithm rests on empirical values. A more statistically accurate methodology may be used.
2. A trainer could be implementing to teach the machine of each step involved in the algorithm.
3. A lot of corner cases are still incorrectly shown.
4. Testing for only a small sample was done. Manual scoring for over 1000 tweets and subsequent testing with the algorithm might be the next step.

2. CONCLUSION

This paper talks about the evaluation of the rule based system for sentiment analysis. Using a combination of lexical and keyword based system, a list of informal words from various sources such a web lists and Twitter data are collected. The system aims towards social media content. Hence, these features have been used with a set of rules observed and a ternary level sentiment score, label and presence of multiple opinions is detected. This algorithm has performed well in most cases.

This system giving deeper insights also took lesser time and performed better. However, a machine learning outlook on this would be more useful for future works. For this a huge training data set that is cross-domain and culturally and linguistically varied text would be appreciated. A more NLP approach that handles dependencies between the natures of the words may also be considered.

Meta-data such as the user's location may be used such as culture plays a role in the thought process and language varies with region. Their opinion on other issues or their previously expressed opinion on similar aspects may be taken into consideration. This problem evidently has the potential to expand into further areas and detail.

REFERENCES

Bishan Yang, Claire Cadre, Context-aware Learning for Sentence-level Sentiment Analysis with Posterior Regularization, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 325–335, Baltimore, Maryland, USA, June 2014, 23-25.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan,and Noah A, Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments, Smith School of Computer Science, Carnegie Mellon Univeristy, 2011.

Finn Arup Nielsen, A new: Evaluation of a word list for sentiment analysis in microblogs, Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, May 2011, 93.98 (CEUR Workshop Proceedings; No.718).

Melville, Wojciech Gryc, Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification, ACM 978-1-60558-495-9/09/06, KDD09, June 28-July 1, Paris, France, 2009.

Rui Xia, Chengqing Zong, Shoushan Li, Ensemble of feature sets and classification algorithms for sentiment classification, Elsevier Information Sciences, 181(6), 2011, 1138-1152.

Ziqiong Zhang, Qiang Ye, Zili Zhang, Yijun Li, Sentiment Classification of Internet restaurant reviews written in Cantonese, Elsevier Expert Systems with Applications, 38, 2011, 7674-7682.

Songbo Tan, Jin Zhang, An empirical study of sentiment analysis for chinese documents, Expert Systems with Applications, 34, 2008, 2622–2629.

Qiang Ye, Ziqiong Zhang, Rob Law, Sentiment classification of online reviews to travel destinations by supervised machine learning approaches, Expert Systems with Applications, 36, 2009, 6527–6535.

Ion Smeureanu, Cristian BUCUR, Applying Supervised Opinion Mining Techniques on Online User Reviews, Informatica Economica, 16(2), 2012.

Andrea Esuli and Fabrizio Sebastiani, Determining the semantic orientation of terms through gloss classification, CIKM, Bremen, DE, 2005.

Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, Andres Montoyo, Survey on the role of negation in sentiment analysis, Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, 2010, 60-68.

Liu B, Indurkhya N and Damerau FJ, Sentiment Analysis and Subjectivity, Handbook of Natural Language Processing, Second Edition, 2010.