SENTIMENT ANALYSIS OF TEXTUAL DATA USING MATRICES AND STACKS FOR PRODUCT REVIEWS

Akhil Krishna,
CSE department,
CMR Institute of technology, Bangalore, Karnataka 560037
akhil.krsn@gmail.com

Suresh Kumar Reddy
ISE department,
CMR Institute of technology, Bangalore, Karnataka 560037
sureshreddy@engineer.com

ABSTRACT The main contribution of the paper is to categorize emotion context from a single or a group of textual inputs and determine their polarity. The emotional context from each sentence is categorized as a polar output which signifies the positive or negative emotional context. We start with a basic sentence evaluation function that parses the input and rejects unnecessary words while assigning values to the ones that are required, based on a simple clause analysis algorithm. These individual values generated are represented in a two dimensional matrix where the location within the matrix specifies a certain emotional polarity and this categorization can be used to evaluate individual sentences at each level or generate the review of the product as a whole.

1 Introduction

A sentence evaluation function is a function that takes a variable-length input string, rejects expressionless words that are of no computational use and assigns value to the emotionally useful keywords thereby effectively calculating the **polarity of the sentences**.[1]

It then searches for emotional context intensifiers such as “very” “not” “kinda” to either **strengthen or weaken** the value of the emotional context considered in previous parts of the function.

Stacks provide a potential avenue to look for negations and intensifiers, as we place them in a stack by parsing the sentence and then popping out to compute the effective emotional value for each sentence.

Each word relocates itself within a two dimensional matrix during computation which initially had occupied the location based on **preassigned values.** Summary of each sentence contribute to the final emotional value.

Each location of the matrix indicates an emotional value held, using probabilistic estimation, the emotional state held by the entire sentence is then calculated using a general row-column formula.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

The matrix is populated with the words’ **count** value, each location representing the number of words which is of same polarity residing in the same location. This generates a matrix which has an inclination to be populated around locations where **i>j** in **A[i][j]** which represent positive emotional context or areas such wherein **i<j** in **A[i][j]** which represent negative regions. Similarly when a matrix is populated towards the centre, i~j in **A[i][j]**, it implies a neutral context.

2. Background

The early work by Turney[1] and Pang[2], where they used different methods for detecting the polarity of product reviews and movie reviews respectively inspired this paper. This analysis is done at a document level. One can classify a document's polarity a numerical scale, the basic task of classifying a document, as either positive or negative to predict star ratings on a scale between -3 to 3.

The accuracy of a sentiment analysis system is considered to be good when it is about 70% accurate**[3],** Since humans are bound to disagree with the system at least 20% of the times even if it is 100% accurate leaving space for errors in evaluation.**[4]**

3. Evaluation of sentences with stacks and matrices.

A sentence is simply a collection of words which makes some sense. So we treat words as objects which are to be evaluated for obtaining the final emotional polarity. In practical, each word has its origin polarity. Such as positive, neutral or negative among which few could act as intensifiers or negations.

Prerequisite: Each word has to be pre assigned its origin polarity. Initialize **A[i][j]=0**

1. The input string is parsed into words **P**i (where i=1,2,3,....n)
2. For every **P**i, based on its polarity, the matrix is populated. If more than one word has same polarity, the **count** of the particular matrix location is incremented.
3. The user sentence is now read from the beginning checking for intensifiers and negations. The following operations are performed as soon as they are encountered :

   a> If an intensifier or negation has been encountered, Push it into a stack. Repeat this process until a word(W) which is neither intensifier, nor negation has been encountered.

   b> Start popping out from the stack.

   - If the popped out word is a negation, swap the values of i and j in the matrix for the word W’s position. i.e., W relocates from **A[i][j]** to **A[j][i]**.
   - If the popped out word is an intensifier, values of i and j of the word W is compared. The one with the greater value is incremented. W is relocated into this new location **A[i][j]**.
   - If the popped out word is an intensifier and the next member to be popped out is a negation, the nullifier is popped out. values of i and j of the word W is compared. The one with the greater value is decremented. W is relocated into this new location **A[i][j]**. This condition as a whole acts as a nullifier.

   c> This process is repeated until the end of sentence(i=n)
4. The matrix **A[i][j]** is now processed to obtain the final value of the sentence.

As discussed earlier, each location in the matrix represents its own polarity. So the final result is the sum of the product of the count and polarity values at their respective location.

S = (A[1][1]*polarity[1][1]) + (A[1][2]*polarity[1][2]) + …. + (A[n][n]*polarity[n][n])

If we generalise, **S = Σ{i,j =1 to n} (A[i][j]*polarity[i][j])**

3.1 Test cases and validation

Let us consider few test cases and validate the proposed design.

<u>Example1:</u>

User input: “Siri is a very good assistant.”

1. P1: Siri, P2: good, P3: assistant (words such as ‘a’, ‘is’ are not necessary for polarity categorization. Hence they’re discarded)
2. Assuming the polarities Siri=A[1][1], good=A[2][1], assistant= A[2][2]. So the initial matrix will look like :

Matrix **A[3][3]**

1	0	0
1	1	0
0	0	0

3. Scan the sentence

Pi	Initial matrix location	Stack contents	Final matrix location
0	A[1][1]	-	A[1][1]
1	A[2][1]	very	A[3][1]
2	A[2][2]	-	A[2][2]

Final Matrix **A[3][3]**

1	0	0
0	1	0
1	0	0

4. S = (1*polarity[1][1]) + (1*polarity[2][2]) + (1*polarity[3][1]). Which is of **positive** polarity.

<u>Example2:</u>

User input: “Siri is not a good assistant.”

1. P1: Siri, P2: good, P3: assistant
2. Assume the polarities Siri=A[1][1], good=A[2][1], assistant= A[2][2]. Initial matrix:

Matrix **A[3][3]**

1	0	0
1	1	0
0	0	0

3. Scan the sentence

Pi	Initial matrix location	Stack contents	Final matrix location
0	A[1][1]	-	A[1][1]
1	A[2][1]	not	A[1][2]
2	A[2][2]	-	A[2][2]

Final Matrix **A[3][3]**:

1	1	0
0	1	0
0	0	0

4. S = (1*polarity[1][1]) + (1*polarity[1][2]) + (1*polarity[2][2]). Which is of **negative** polarity.

<u>Example3:</u>

User input: “Siri is not a very good assistant.”

1. P1: Siri, P2: good, P3: assistant
2. Assume the polarities Siri=A[1][1], good=A[2][1], assistant= A[2][2]. Initial matrix

Matrix **A[3][3]**

1	0	0
1	1	0
0	0	0

3. Scanning Pi from i=0 to n

Pi	Initial matrix location	Stack contents	Final matrix location
0	A[1][1]	-	A[1][1]
1	A[2][1]	not, very	A[1][1]
2	A[2][2]	-	A[2][2]

Final Matrix **A[3][3]**:

2	0	0
0	1	0
0	0	0

4. S = (2*polarity[1][1]) + (1*polarity[2][2]). Which is of **neutral** polarity.

3.2 Shortcomings of current method

Using the currently specified techniques, there are a lot of inherent flaws that are encountered in mining for opinions or contextually relevant data using “sentiment” as a pivotal characteristic for a given sentence. For example, while looking through reviews of an app such as Instagram on the Google Play Store, only about a quarter of all reviews actually contain sentimental value , either positive or negative, which means 75% of the reviews, reveal no sentiment and are ignored by the analysis. However, these reviews do in fact contain some valuable data and decisions are being based on what 25% of the posts are saying.

The pie chart fig.4 shows a representation of the reviews of the Instagram app which clearly shows how 75% of the reviews, although having some valuable feedback are not using to the developer as they are not distinguishable through the use of this sentiment analysis.

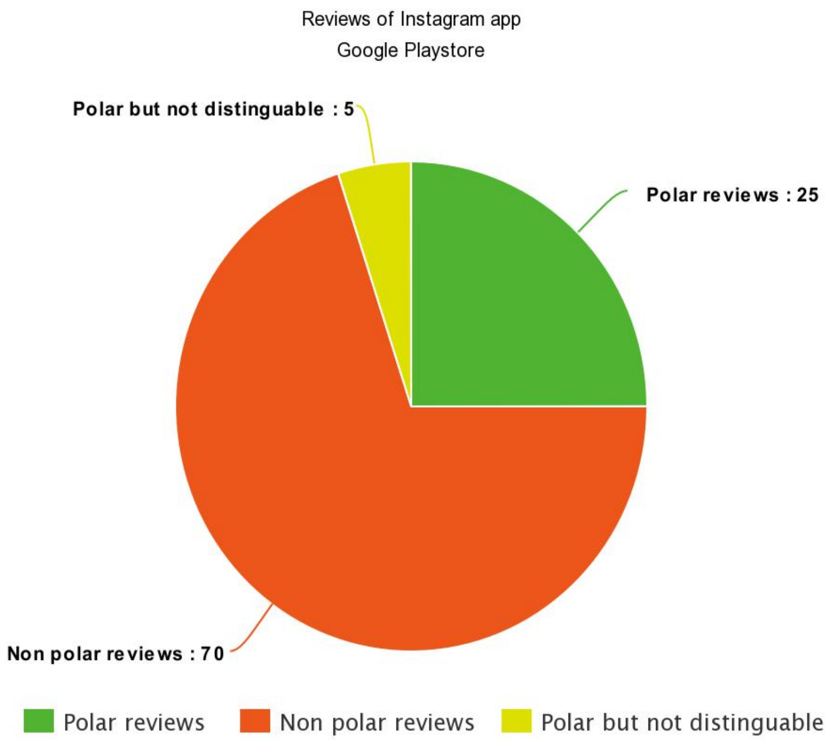

Fig. 4 Review distribution of the Instagram app for Android

Another issue faced using the current technique is the requirement of the analysis system to have a large predefined set of words that are emotionally significant and whose values have been evaluated and store in accessible format for evaluation by the algorithm. If the sentiment analysis encounters a word which does not have any predefined value, it must be treated as a neutral word, however if the majority of a sentence is filled with such words, the evaluation of the sentence will be false.

4 Conclusion

Sentiment analysis allows us to analyze people’s sentiments, attitudes, or emotions towards certain entities, products such as movies, appliances, digital purchases. This paper tackles the problem of sentiment analysis through polarity categorization with the use of matrices and stacks. Online app reviews from Google Playstore for the Instagram app are selected as data used for this study. A sentiment polarity categorization technique has been proposed through the use of matrices and stacks along with explicit descriptions of each step. Examples for both sentence-level evaluation and review-level evaluation have been discussed in sections 3, 3.1 and 3.2

References

[1] Turney, Peter (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". Proceedings of the Association for Computational Linguistics. pp. 417–424.

[2] . Pang, Bo; Lee, Lillian; Vaithyanathan, Shivakumar (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques". Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 79–86.

[3] . Ogneva, M. "How Companies Can Use Sentiment Analysis to Improve Their Business". Mashable. Retrieved 2012-12-13.

[4] . Roebuck, K. (2012-10-24). Sentiment Analysis: High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors. ISBN 9781743049457.