Sentiment Classification using N-gram IDF and Automated Machine Learning
S. Harika¹, K. R. Tejaswini², Thrivani. C³, Saigeetha¹, Vinutha Prashanth³
Assistant Professor³
Department of CSE
VTU/RYMEC College Bellary, Karanataka, India

Abstract:
We introduce sentiment classification method with machine learning. For representing features, N-gram IDF is used. The main aim of using IDF is to extract software-engineering-related, positive, negative, neutral N-gram expressions. Automated machine learning tool is used for classifiers.

1. INTRODUCTION
We introduce sentiment classification method with machine learning. For representing features, N-gram IDF (Inverse Document Frequency) is used. The main aim of using IDF (Inverse Document Frequency) is to extract software-engineering-related, positive, negative, neutral N-gram expressions, where N-gram actually means a contiguous sequence of n-items from a given sequence of text or speech. Automated machine learning tool is used for classifiers. Nowadays it has became a challenge to extract the useful i.e. meaningful information from the documents and messages by identifying their effective states.

Sentiment analysis has been used to several practical purposes, such as identifying problematic API features, assessing the polarity of app reviews, clarifying the impact to the issue resolution time and so on. Because of the poor accuracy of existing sentiment analysis tools trained with general sentiment expressions, recent studies have tried to customize them with software engineering datasets. However it is reported that there is no tool that accurately classifies sentences to negative, neutral or positive, even they are specifically customized for that tasks.

2. PROPOSED METHOD
The fig 1 shown below gives the overview of our method with three components.

Text Processing:
The messages in the software documents may have special characters so we have to remove those characters that are neither English characters nor numbers. By using spacy library stop words can be removed. spacy tokenizes text and finds the part of speech and tag of each token and it also checks whether the token appear as in the stop word list.

Feature extraction using N-gram IDF:
N-gram IDF (Inverse Document Frequency) is used for representing the features. It is a theoretical extension of IDF used for handling words and phrases of any length and to extract software-engineering related positive, negative, neutral N-gram expressions. N-gram IDF can identify n-grams among overlapping ones. In this study we use N-gram weighting scheme tool. The result after using this tool is a dictionary of n-gram phrases. N-gram phrases appear only one time and they can be removed.

Automated machine learning:
For classifying the sentences to positive, negative and neutral we use a automated machine learning tool called auto-sklearn. To get the better performances automated machine learning runs multiple classifiers. N-gram actually means a contiguous sequence of n items from a given sequence of text or speech.
Table 1. (Obtained N-Gram Phrases)

<table>
<thead>
<tr>
<th>Stack overflow</th>
<th>Negative statements</th>
<th>Neutral statements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Positive statements</strong></td>
<td><strong>It is very bad question to answer,</strong> <strong>I don’t understand</strong> <strong>I don’t like this question</strong></td>
<td><strong>I am not clear about ur question to answer,</strong> <strong>I suggest u to do more work on it</strong></td>
</tr>
<tr>
<td><strong>Negative statements</strong></td>
<td><strong>It is very good question , This is useful to everyone,</strong> <strong>This site is more helpful to User</strong></td>
<td><strong>My project is still pending, I am getting lot of bugs,</strong> <strong>Still I need time to complete my project and recover bugs,</strong> <strong>Its is bad experiance</strong></td>
</tr>
<tr>
<td><strong>Neutral statements</strong></td>
<td><strong>My team members are very good and supportive,</strong> <strong>Successfully executing our project</strong></td>
<td><strong>My work is almost done we need to recheck once,</strong> <strong>Our project is done still we need time to add some specification</strong></td>
</tr>
<tr>
<td><strong>App reviews</strong></td>
<td><strong>This is good app</strong></td>
<td><strong>Very bad experience,</strong> <strong>Products quality is not good,</strong> <strong>App is too slow for normal users,</strong> <strong>I honestly don’t like your applications but your service is the best</strong></td>
</tr>
<tr>
<td><strong>Positive statements</strong></td>
<td><strong>This is useful to everyone,</strong> <strong>This site is more helpful to User</strong></td>
<td><strong>Most of the time it will work fast,</strong> <strong>During payment by scanning it is not responding’</strong></td>
</tr>
</tbody>
</table>

3. OBTAINED N-GRAM PHRASES

Why our method high accuracy performance in sentiment classification? The table shows that which are useful for classifying positive, neutral and negative statements, obtain in each datasets. For negative we see ‘It is very bad experience to work’ software engineering specific negative sentence and many negative expression we can also see reasonable n-gram phrases for positive cases, such as ‘This is good app’, ‘I like this app very much’ ,and so on, we can think that because of dataset specific positive, negative and neutral sentence, n-gram IDF worked well to differentiate the positive, negative and neutral sentence and our method have resulted in good performance. we consider three different classifications like stack overflow, jira issues and app reviews to differentiate the sentiment classification analysis using n-gram IDF algorithm.

4. EXPERIMENTAL DATA AND RESULTS

Experimental data:

We are using three data sets namely Stack Overflow datasets, app reviews datasets and jira issues datasets as input. Inspite of this we have also created a website like structure to examine the data when people posts something in their page after logging in using credentials and categorize them into positive negative and neutral and the classification also seen in the form of graphs namely bar graph and pie chart.

5. REPRESENTING THE CLASSIFICATION BY GRAPHS

The sentiment classification analysis can also be done using graphs. here we have include two graphs namely pi graph and bar graph. The graphs shows that percentage of positive, negative and neutral n-gram expressions. The datasets that are taken from stack overflow, jira issues and app reviews after classification their positivity, neutral and negativity can be represented in the form of pi graph and bar graph in terms of percentage.

6. CONCLUSION

In this paper, we proposed a sentiment classification method using n gram IDF and automated machine learning. We apply
this method on three datasets including question and answer from Stack Overflow, reviews of mobile app, and comment son Jira issue trackers. Our good classification performance is not based only on an advanced automated machine learning. N-gram IDF also worked well to capture. Datasets specific software engineering related positive, negative, neutral expressions. Because of the capability of extraction using sentiment expressions with n-gram IDF, our method can be applicable to various software engineering datasets.

7. REFERENCES


