Facebook for Sentiment Analysis: Baseline Models to Predict Facebook Reactions of Sinhala Posts

Vihanga Jayawickrama*, Gihan Weeraprameshwara*, Nisansa de Silva*, Yudhanjaya Wijeratne†
* Department of Computer Science & Engineering, University of Moratuwa
†LIRNEasia
vihangadewmini.17@cse.mrt.ac.lk

Abstract — Research on natural language processing in most regional languages is hindered due to resource poverty. A possible solution for this is utilization of social media data in research. For example, the Facebook network allows its users to record their reactions to text via a typology of emotions. This network, taken at scale, is therefore a prime dataset of annotated sentiment data. This paper uses millions of such reactions, derived from a decade worth of Facebook post data centred around a Sri Lankan context, to model an eye of the beholder approach to sentiment detection for online Sinhala textual content. Three different sentiment analysis models are built, taking into account a limited subset of reactions, all reactions, and another that derives a positive/negative star rating value. The efficacy of these models in capturing the reactions of the observers are then computed and discussed. The analysis reveals that the Star Rating Model, for Sinhala content, is significantly more accurate (0.82) than the other approaches. The inclusion of the like reaction is discovered to hinder the capability of accurately predicting other reactions. Furthermore, this study provides evidence for the applicability of social media data to eradicate the resource poverty surrounding languages such as Sinhala.

Keywords — NLP, sentiment analysis, Sinhala, word vectorization

I. INTRODUCTION

Understanding human emotions is an interesting, yet complex process which researchers and scientists around the world have been attempting to standardize for a long period of time. In the computational sciences, sentiment analysis has become a major research topic, especially in relation to textual content [1, 2]. Several fields, scattered in diverse arenas from product marketing to political manipulations, benefit from the advancements in sentiment analysis. Studies such as those conducted by Rudkowsky et al. [3], Aguwa et al. [4], and Zobal [5] have described the potential of sentiment analysis and attempted to introduce useful tools for use in this field and discover new knowledge. Sentiment analysis of textual content can be approached in two ways:

1) Through the perspective of the creator
2) Through the perspective of the observer.

Many research projects attempt to follow the first approach, but only a few such as Hui et al. [6] have followed the second. Exploring the perspective of the observer would be quite important since the emotional reaction of the author and the reader to the same content is not necessarily identical. For certain fields, such as movie reviews [7] or product reviews [8], the perspective of the author is much more valuable than that of the reader; however, this relationship does not always hold true. Much effort is generally expended in the field of political polling, for example, where the public perception of a speech is studied to assess impact.

To the extent of our knowledge, no attempt has been made to do such analysis in Sinhala, the subject of this study. Sinhala, similar to many other regional languages, suffers from resource poverty [9]. Previous research and resources available for NLP in Sinhala are limited and isolated [10, 11]. This is therefore an experimental attempt in bridging this knowledge gap. The objective is to predict the sentimental reaction of Facebook users to textual content posted on Facebook. This study uses a raw corpus of Sinhala Facebook posts scraped through Crowdtangle3 by Wijeratne and de Silva [12], and analyses the user reactions therein as a sentiment annotation that reflects the emotional reaction of a reader to the said post [13]. Facebook reactions Like, Love, Wow, Haha, Sad, Angry, and Thankful are utilized as the sentiment annotation of a post within the scope of this project. Figure 1 illustrates the visual representations of Facebook reactions presented to the users and are included in the dataset.

Fig. 1. Facebook Reactions

This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Note: DOI: http://doi.org/10.4038/icter.v15i2.7247

© 2022 International Journal on Advances in ICT for Emerging Regions

3https://www.crowdtangle.com/
Overall, three models were created and tested. As for the first model, a reaction vector was created for each post with the normalized reaction counts belonging to Love, Wow, Haha, Sad, and Angry categories. Like and Thankful reactions, which are outliers at positive and negative ends of the spectrum respectively, were ignored. The results showed that the procedure could predict reaction vectors with F1 scores ranging between 0.13 and 0.52. The second model was highly similar to the first model, the only difference being the inclusion of Like and Thankful reactions for the prediction. The resultant F1 scores ranged between 0.00 and 0.96.

In the third model, the reactions were combined to create a positivity/negativity value for each post, following the procedure presented by De Silva et al. [8]. Here, Love and Wow were considered as positive, Sad and Angry were considered as negative, and Haha was ignored due to its conflicting use cases. The normalization was carried out as earlier for the four reactions included, and the difference between positive and negative values were re-scaled into the range 1 to 5, in order to map to the popular star rating system utilized by De Silva et al. [8]. The F1 score of this star rating value ranged between 0.29 and 0.30. In contrast, the binary categorization of reactions as Positive and Negative exhibited promising results, with F1 scores in the range 0.70-0.71 for Positive and 0.41 - 0.42 for Negative.

Thus, it can be concluded that such a binary categorization system captures the sentimental reaction to Facebook post more efficiently in comparison to the multi-category reaction value system, and presents a measure of reasonable accuracy in the imputation of such sentiment.

It should be re-iterated here that the values used here are completely independent from the intended or perceived sentiment of the original posts and are solely dependent on sentiment expressed by the audience reactions. Further, the model only attempts to predict the positivity or the negativity of Facebook reactions added to a post by users, and not of the actual emotion inflicted in the users by the post. While the duo might be correlated, the exact nature of the relation would have to be further explored before reaching a distinct conclusion. Figure 2 illustrates the scope of this research, where arrows indicate the influences among intended and perceived sentiments. This journal paper is an extension of our previously published conference paper [14].

II. BACKGROUND

Many of the studies on sentiment analysis are focused on purposes such as understanding the quality of reviews given for products presented in e-commerce sites [8, 15, 16] or understanding the political preferences of people [3, 17].

Among the research on review analysis, the work of De Silva et al. [8] is prominent. Rather than conducting a sentiment analysis following the more traditional procedures of identifying sentiments at the sentence level or at the document level, which assumes each sentence and each document to reflect a single emotion, this study had taken a path to determine sentiments on an aspect level. Different aspects were extracted from the review, and for each aspect, the sentiment value was calculated. Further, the study provides a set of guidelines to determine the semantic orientation of a subject using a sentiment lexicon while guiding how to handle negations, words that increase sentiment, words that shift the sentiment of the sentences, and groups of words that are used to express an emotion, all of which are important to convert sentiment in text into mathematical figures. The methodology presented by De Silva et al. [8] is crucial for this study since it provides the basis of one of the two workflows we discuss in this study to predict reactions for Sinhala text.

The work by Martin and Pu [16], a research done on creating a prediction model that could identify helpful reviews that are not yet voted by other users, emphasizes the value of sentiment analysis. Rather than solely relying on structural aspects of a review such as the length and the readability score, the emotional context was also utilized in rating the reviews, with the support of the GALC lexicon, which represents 20 different emotional categories. One of the most important findings of the project was that the emotion based model outperforms the structure based model by 9%. The work of Singh et al. [15] too has used several textual features such as ease of reading, subjectivity, polarity, and entropy to predict the helpfulness ratio. The model intends to assist the process of assigning a helpfulness value to a review as soon as the review is posted, thus giving the spotlight to useful reviews over irrelevant reviews. Both researches have highlighted the usefulness of understanding the reaction of the reader to different content. The studies on political preferences cover a massive area. Many governments and political parties use social media to understanding the audience. Therefore, the power vested in sentiment analysis cannot be ignored.

The research done by Caetano et al. [17] and Rudkowsky et al. [3] explain two different cases where sentiment analysis is utilized in politics. Caetano et al. attempts to analyse twitter data and define the homophily of the twitter audience while Rudkowsky et al. demonstrates the usability of word embedding over bag-of-words by developing a negative sentiment detection model for parliament speeches. Caetano et al. concludes that the homophily level increased with the multiplex connection of the audience, while Caetano et al. states that the negativity of the speeches of a parliament member correlates to the position he holds in the parliament. While these instances may not be immediately identifiable as direct results of sentiment analysis, they are great examples for the wide range covered by sentiment analysis.

Facebook data plays a major part in our research. Therefore, it is vital to explore the previous research done on Facebook data. The work by Pool and Nissim [18] and Freeman et al. [19] use datasets obtained from Facebook for emotion detection. The data scope covered through the work of Freeman et al. lacks diversity since the research is solely focused on Scholarly articles. However, Pool and Nissim has attempted to maintain a general dataset by using a variety of sources, ranging from New York Times to SpongeBob. The motivation behind this wide range of sources was to pick the best sources to train ML models for each reaction. Pool and Nissim too has looked into developing models with different features such as TF-IDF, embeddings, and n-grams. This comparison provides useful guidelines for picking up certain features in data. One of the most important aspects of the work by Pool and Nissim is that they have taken an extra step to test the models with external datasets; namely, AffectiveText [20], Fairy Tales [21], and ISEAR [22], to prove the validity of the developed model since those are widely used datasets in the field of sentiment analysis. This provides a common ground to compare different sentiment...
analysis models. The work of Graziani et al. [13] too follows the same procedure in comparing their model to those of others.

While all papers mentioned above provide quite useful information, almost all of them relate to English, which is a resource-rich language. On the contrary, our project will be based on the Sinhala language, which is a resource-poor language in the NLP domain [9]. Very few attempts have been made to detect sentiments in Sinhala content, and most of the attempts made were either abandoned or not released to the public [10]. This poses a major challenge to our work due to the scarcity of similar work in the domain.

Among the currently available research in this arena, Seneviratne et al. [23] is the state-of-the-art Sinhala text sentiment analysis attempt to the best of our knowledge. Through this paper, Seneviratne et al. has introduced a study of sentiment analysis models built using different deep learning techniques as well as an annotated sentiment dataset consisting of 15059 Sinhala news comments. The work was done to understand the reactions of the people reading. Furthermore, earlier attempts such as Medagoda et al. [24] provides insight into utilizing resources available for languages such as English for generating progress in sentiment analysis in Sinhala. The partially automated framework for developing a sentiment lexicon for Sinhala developed from this corpus by Wijeratne and de Silva [1] is a noteworthy study Sinhala text and a collection of stopwords. Both the raw content. Characters belonging to Unicode categories Cc, Cn, Co, and Cs were replaced with a single white space. Numerical content was removed due to their high unlikelihood to be repeated in the same sequence order. Finally, multiple continuous white spaces in the corpus were replaced with a single white space. Once cleaned, entries of which the Message column were merely null strings or empty strings were removed from the corpus. The final cleaned corpus consisted of 526,732 data rows.

B. Core Reaction Set Model

In selecting the core reaction set, Like and Thankful reactions were excluded due to their counts being outliers in comparison to other reactions; Like was an outlier on the higher end and Thankful being an outlier on the lower end. The total count of each reaction in the corpus along with their percentages are mentioned in table II. A probable reason for the abnormal behaviour of those reactions are the duration long with their reaction introduced to the platform, back in 2009 [28]. Love,
The dataset was then divided into train and test subsets for the purpose of calculating and evaluating the accuracy of vector predictions. The message column of the train set was tokenized into individual words, and set operation was used to obtain the collection of unique words for each entry. Then, a dictionary was created for each entry by assigning the normalized reaction vector of the entry to each word. The dictionaries thus created were merged vertically, taking the average value of vectors assigned to a word across the dataset as the aggregate reaction vector of that word. Equation 3 describes this process where \( V_W \) is the aggregate reaction vector for the word \( W \). \( R_i \) is the reaction vector of the \( i \)th entry \( (E_i) \), \( n \) is the number of entries, and \( \emptyset \) is the empty vector.

\[
V_W = \frac{\sum_{i=1}^{n} \begin{cases} R_i & \text{if } W \in E_i \\ \emptyset & \text{otherwise} \end{cases}}{\sum_{i=1}^{n} \begin{cases} 1 & \text{if } W \in E_i \\ 0 & \text{otherwise} \end{cases}}
\]  

(3)

The dictionary thus created was used to predict the reaction vectors of the test dataset. Entries in the test set were tokenized and then converted to unique word sets similar to the aforementioned processing of the training set. Then for each of the words in a set of a message which also exists in the dictionary created above, the corresponding reaction vector was obtained from the dictionary. For entries of which none of the words were found in the dictionary, the mean vector value of the train dataset was assigned. Equation 4 shows the calculation of the predicted vector \( \hat{V}_M \) for a message where, \( V_W \) is taken from the dictionary (which was populated as described in Equation 3), and \( N_M \) is the number of words in the message \( M \).

TABLE I
FIELDS OF THE SOURCE DATASET THAT WERE USED FOR THIS STUDY

| Field Name       | Description                               | Data Type |
|------------------|-------------------------------------------|-----------|
| Index            | Index of the dataset                      | int       |
| Like             | The number of Like reacts on the post     | int       |
| Love             | The number of Love reacts on the post     | int       |
| Wow              | The number of Wow reacts on the post      | int       |
| Haha             | The number of Haha reacts on the post     | int       |
| Sad              | The number of Sad reacts on the post      | int       |
| Angry            | The number of Angry reacts on the post    | int       |
| Thankful         | The number of Thankful reacts on the post | int       |
| Message          | Textual content of the Facebook post      | string    |

Wow, Haha, Sad, and Angry reactions were introduced in 2016 [29]; however, Like still retained its state as the default reaction which a simple click on the react button enforces. The Thankful reaction was a temporary option introduced as part of Mothers’ Day celebrations of Facebook in May 2016 [30]. The reaction was removed from the platform after a few days, and was reintroduced in May 2017 to be removed again after the Mother’s Day celebrations [31].

Thus, the core reaction set was defined considering only the Love, Wow, Haha, Sad, and Angry reactions. The percentages of the core reactions are also shown in Table II. Furthermore, Fig. 3 shows the core reaction percentages as a pie chart. Thus, initially, the normalization was done considering only the core reactions. Equation 1 obtains the sum of reactions \( T \) of an entry using the counts of: Love \( (n_L) \), Wow \( (n_W) \), Haha \( (n_H) \), Sad \( (n_S) \), and Angry \( (n_A) \). The Equation 2 shows the normalized value \( N_r \) for reaction \( r \) where \( n_r \) is the raw count of the reaction and \( T \) is the sum obtained in Equation 1.

\[
T = n_L + n_W + n_H + n_S + n_A
\]

(1)

\[
N_r = \frac{n_r}{T}
\]

(2)

TABLE II
TOTAL COUNTS OF REACTIONS IN THE CORPUS

| Reaction | Count     | Percentage | All | Core |
|----------|-----------|------------|-----|------|
| Like     | 528,060,209 | 95.43       | -   | -    |
| Love     | 12,526,942  | 2.26        | 49.56|      |
| Wow      | 1,906,174   | 0.34        | 7.54 |      |
| Haha     | 6,524,139   | 1.18        | 25.81|      |
| Sad      | 2,987,589   | 0.54        | 11.82|      |
| Angry    | 1,329,552   | 0.24        | 5.26 |      |
| Thankful | 13,637     | 0.002       | -   | -    |

Fig. 3: Core Reaction Percentages
\[
V_M = \frac{\Sigma N_M V_M}{N_M}
\] (4)

### C. Defining the Evaluation Statistics

To evaluate the performance of the prediction process, a number of statistics were calculated. Equation 5 shows the calculation of Accuracy \( A_r \) for reaction \( r \) where, \( N_r \) is the expected (actual) value for the entry as calculated in Equation 2 and \( M_r \) is the predicted value calculated in Equation 4 as \( M_r \in V_M \).

\[
A_r = \min(N_r, M_r)
\] (5)

The accuracy can be defined this way since we are solving a bin packing problem and the vector values are sum up to 1. Equations 21, 7, and 22 shows the calculation of Recall \( (R_r) \), Precision \( (P_r) \), and F1 score \( (F1_r) \) respectively where notation is same as Equation 5.

\[
R_r = \frac{A_r}{M_r}
\] (6)

\[
P_r = \frac{A_r}{N_r}
\] (7)

\[
F1_r = \frac{2 \times P_r \times R_r}{P_r + R_r}
\] (8)

The above measures were calculated for each entry of the dataset and the average value of each measure was assigned as the resultant performance measure of the dataset. Those values were then averaged across 5 runs of the code.

### D. All Reaction Set Model

The All Reaction Set Model was developed following the same procedure of the core reaction set model. In addition to the reactions included in the core reaction set, Like \((n_L)\) and Thankful \((n_T)\) were considered during this step. Equation 9 depicts how the sum of reactions is obtained while the normalized value \(N_r^*\) for each reaction could be obtained as mentioned in Equation 10. \(T^*\) refers to the sum of reactions obtained through Equation 9.

\[
T^* = n_L + n_L + n_W + n_H + n_S + n_A + n_T
\] (9)

\[
N_r^* = \frac{n_r}{T^*}
\] (10)

The sentiment vector for each entry was then generated following the same procedure as in III-B. The evaluation was done as mentioned in III-C.

### E. Star Rating Model

The next step of the study was inspired by the procedure proposed by De Silva et al. [8]. They propose using the star rating to generate sentiment vectors. Since the star rating take They propose using the star rating associated with amazon customer reviews to generate sentiment vectors. Since the star rating takes a value between 1 and 5 where 3 is considered neutral, and values more than 3 and less than 3 are considered positive and negative respectively by them. To adjust Facebook reactions to this scale, we classified the positivity of reactions as presented in table IV. The positivity of the Haha reaction is considered to be uncertain due to its conflicting use cases: the reaction is often used both genuinely and sarcastically on the platform [32]. Therefore, the experiment was carried out considering only the Love, Wow, Sad, and Angry reactions. The normalization process described in Section III-B for the Core Reaction Set Model was updated by modifying Equation 1 as shown in Equation 11 and modifying Equation 2 as shown in Equation 12, where \( \hat{T} \) is the modified sum of reactions of the entry. Figure 4 presents the distribution of selected reactions in the corpus.

\[
\hat{T} = n_L + n_W + n_S + n_A
\] (11)

\[
\hat{N} = \frac{n_r}{\hat{T}}
\] (12)

The positive sentiment value \((E_{(p,i)})\) for entry \( i \) was calculated by summing the Normalized Love \((\hat{N}_L)\) and Normalized Wow \((\hat{N}_W)\) values while the negative sentiment \((E_{(N,i)})\) was calculated by summing the Normalized Sad \((\hat{N}_S)\) and Normalized Angry \((\hat{N}_A)\) values, as shown in Equations 13 and 14. Using \(E_{(p,i)}\) and \(E_{(N,i)}\), the aggregated sentiment for entry \( i \) was calculated as shown in Equation 15.

\[
E_{(p,i)} = \hat{N}(\hat{L}_i) + \hat{N}(\hat{W}_i)
\] (13)

\[
E_{(N,i)} = \hat{N}(\hat{S}_i) + \hat{N}(\hat{A}_i)
\] (14)

\[
E_i = E_{(p,i)} - E_{(N,i)}
\] (15)

The Star Rating Value \((S_i)\) for entry \( i \) which is calculated over the entire dataset was computed as shown in Equation 16 where \( I \) is the set of entries in the dataset.

\[
S_i = 4 \times \left( \frac{\min(E_i)}{\max(E_i)} - \frac{\min(E_i)}{\max(E_i)} \right) + 1
\] (16)

The sentiment vector \((V_i)\) for entry \( i \) is defined in Equation 17 where \( E_{(p,i)}, E_{(N,i)}, \) and \( S_i \) were calculated as mentioned before.

\[
V_i = \left[ E_{(p,i)}, E_{(N,i)}, S_i \right]
\] (17)

Once the vectors were computed, the processing of test and train sets, building of the dictionary, and evaluating the
| Train (%) | Reaction | Core Reaction Set Model | All Reaction Set Model |
|-----------|----------|-------------------------|-----------------------|
|           |          | Accuracy | Recall | Precision | F1 Score | Accuracy | Recall | Precision | F1 Score |
| 95        | Like     | -        | -      | -         | -        | 0.9169   | 0.9651 | 0.9691 | 0.9626 |
|           | Love     | 0.3119   | 0.5863 | 0.7838    | 0.5164   | 0.0056   | 0.2510 | 0.6221 | 0.1769 |
|           | Wow      | 0.0298   | 0.3111 | 0.6373    | 0.2218   | 0.0005   | 0.1487 | 0.4550 | 0.0818 |
|           | Haha     | **0.1163** | 0.4241 | 0.6279    | **0.3060** | 0.0042   | 0.1646 | 0.6044 | 0.1068 |
|           | Sad      | 0.0497   | 0.2355 | 0.6206    | 0.1613   | 0.0015   | 0.1013 | 0.5829 | 0.0638 |
|           | Angry    | **0.0175** | 0.2059 | 0.5837    | **0.1318** | 0.0006   | 0.0880 | 0.5193 | 0.0495 |
|           | Thankful | -        | -      | -         | -        | 0.0000   | 0.0007 | 0.0440 | 0.0000 |
| 90        | Like     | -        | -      | -         | -        | 0.9170   | 0.9652 | 0.9691 | 0.9626 |
|           | Love     | 0.3119   | 0.5847 | 0.7833    | 0.5147   | 0.0056   | 0.2513 | 0.6225 | 0.1774 |
|           | Wow      | **0.0299** | 0.3110 | 0.6375    | 0.2216   | 0.0005   | 0.1486 | 0.4557 | 0.0818 |
|           | Haha     | 0.1160   | 0.4242 | 0.6261    | 0.3053   | 0.0042   | 0.1639 | 0.6043 | 0.1064 |
|           | Sad      | 0.0497   | 0.2360 | 0.6205    | 0.1616   | 0.0015   | 0.1009 | 0.5840 | 0.0636 |
|           | Angry    | 0.0174   | 0.2041 | 0.5834    | 0.1308   | 0.0006   | 0.0882 | 0.5162 | 0.0494 |
|           | Thankful | -        | -      | -         | -        | 0.0000   | 0.0007 | 0.0376 | 0.0000 |
| 80        | Like     | -        | -      | -         | -        | 0.9167   | 0.9649 | 0.9691 | 0.9625 |
|           | Love     | 0.3118   | 0.5854 | 0.7833    | 0.5153   | 0.0056   | 0.2515 | 0.6208 | 0.1770 |
|           | Wow      | 0.0298   | 0.3113 | 0.6370    | **0.2218** | 0.0005   | 0.1490 | 0.4527 | 0.0816 |
|           | Haha     | 0.1160   | 0.4238 | 0.6266    | 0.3052   | 0.0042   | 0.1647 | 0.6037 | 0.1067 |
|           | Sad      | **0.0499** | 0.2380 | 0.6176    | **0.1623** | 0.0015   | 0.1012 | 0.5825 | 0.0636 |
|           | Angry    | 0.0174   | 0.2045 | 0.5856    | 0.1314   | 0.0006   | 0.0889 | 0.5142 | 0.0497 |
|           | Thankful | -        | -      | -         | -        | 0.0000   | 0.0007 | 0.0297 | 0.0000 |
| 70        | Like     | -        | -      | -         | -        | 0.9167   | 0.9650 | 0.9690 | 0.9625 |
|           | Love     | 0.3117   | 0.5855 | 0.7829    | 0.5152   | 0.0056   | 0.2513 | 0.6216 | 0.1771 |
|           | Wow      | 0.0298   | 0.3110 | 0.6376    | 0.2217   | 0.0005   | 0.1484 | 0.4539 | 0.0814 |
|           | Haha     | 0.1158   | 0.4236 | 0.6263    | 0.3049   | 0.0042   | 0.1643 | 0.6045 | 0.1065 |
|           | Sad      | 0.0497   | 0.2368 | 0.6183    | 0.1616   | 0.0015   | 0.1014 | 0.5816 | 0.0637 |
|           | Angry    | 0.0174   | 0.2050 | 0.5847    | 0.1314   | 0.0006   | 0.0885 | 0.5155 | 0.0495 |
|           | Thankful | -        | -      | -         | -        | 0.0000   | 0.0007 | 0.0342 | 0.0000 |
| 50        | Like     | -        | -      | -         | -        | 0.9167   | 0.9650 | 0.9690 | 0.9625 |
|           | Love     | **0.3121** | 0.5863 | 0.7824    | 0.5156   | 0.0056   | 0.2513 | 0.6206 | 0.1768 |
|           | Wow      | 0.0298   | 0.3113 | 0.6361    | 0.2214   | 0.0005   | 0.1491 | 0.4519 | 0.0815 |
|           | Haha     | 0.1155   | 0.4236 | 0.6249    | 0.3043   | 0.0042   | 0.1643 | 0.6034 | 0.1063 |
|           | Sad      | 0.0496   | 0.2366 | 0.6195    | 0.1617   | 0.0015   | 0.1014 | 0.5815 | 0.0636 |
|           | Angry    | 0.0173   | 0.2041 | 0.5855    | 0.1310   | 0.0006   | 0.0886 | 0.5142 | 0.0494 |
|           | Thankful | -        | -      | -         | -        | 0.0000   | 0.0007 | 0.0330 | 0.0000 |
Equation 20 shows the calculation of precision of each star rating class, where $\hat{P}_x$ represents the precision value of class $x$, $n_{cc,x}$ represents the number of correctly classified posts in class $x$, and $T_i$ represents the trespass value of post $i$ in class $x$.

$$\hat{P}_x = \frac{n_{cc,x}}{n_{x}}$$

3) Recall: The recall value was calculated for each post in terms of its Class Gaussian Distance, which is defined as the Gaussian distance to the midpoint of the predicted Star Rating Class of the post from the midpoint of its true Star Rating Class, on a distribution which is centered on the midpoint of the true class. The recall value $R_x$ for a class $x$ consisting of an $n_x$ number of Facebook posts, each with a recall of $R_i$, was obtained as depicted by Equation 21.

$$R_x = \frac{\gamma n_x R_i}{n_x}$$

4) F1 Score: The F1 score of each class was then calculated based on the precision and recall values of the class, following the standard formula. Equation 22 portrays the calculation of F1 score $F1_x$ for class $x$.

$$F1_x = \frac{2 \times \hat{P}_x \times R_x}{\hat{P}_x + R_x}$$

5) Overall Performance: The overall performance measures for star rating were calculated by taking a weighted mean of performance measures of classes, with weights assigned based on the class size.

IV. RESULTS

Table III shows the results obtained for the preference measure defined in Section III-C for the Core Reaction Set Model introduced in Section III-B and All Reaction Set Model introduced in Section III-D. All reactions except Sad reach their highest F1 score at the 95% − 05% train-test division, while the Sad reaction reaches its peak F1 score at the 80% − 20% division. Interestingly, the performance of the model in predicting each reaction seems to roughly follow a specific pattern; reactions that were used more often in the dataset seem to have a higher F1 score than reactions that were used less often, with the exception of the F1 score of Wow being higher than that of Sad. Figure 5 portrays the F1 score for each reaction as the train-test division varies for the Core Reaction Set Model. In the case of All Reaction Set Model, as shown in Table III, while the F1 of Like was much higher than that of other reactions, its inclusion brought forth significant reductions in the F1 scores of the other reactions. The Thankful reaction had a F1 of almost zero.

The overall results obtained for Star Rating Model introduced in section III-E are shown in table V. In contrast to the results obtained for Positive and Negative components, aggregation of reactions into a single Star Rating value has caused a significant decrease in precision; possibly due to the discrete nature of the Star Rating value which is divided into bins at 0.5 intervals. Figure 6 portrays the change of F1 value with the train-test division.
Furthermore, the results obtained for each Star Rating Class are displayed in Table VI. It could be observed that the model exhibits better performance with regard to predicting more neutral star rating values. While accuracy and recall measures show comparable performance across all classes, this difference becomes much more prominent in precision. Consequently, a notable increase in performance is observed in more neutral classes in terms of F1 score. Further explorations revealed that the root cause of this issue is that the predictions of the model tend to lean towards more neutral classes, as portrayed in Table VII. It should be noted that the extremely positive and extremely negative classes are significantly larger in size, in comparison to comparatively neutral classes.

As portrayed by Figure 5, the performance of the models remains largely unaffected by the train-test division chosen. The reason could be the large size of the dataset; the number of unique words in the train dataset does not change significantly for different train-test divisions.

V. CONCLUSION

Upon comparing the Star Rating Model with the Core Reaction Set Model, it becomes evident that the F1 scores are significantly improved upon the accumulation of separate reaction values into two categories as Positive and Negative. A possible reason for this is the possibility of the intra-category measurement errors being eliminated due to merging. However, merging all reactions into a single Star Rating value accentuates errors. This could be accounted to the additional error margin introduced by discretization. Further, the model predictions for Star Rating Classes that are closer to the median proves to be better than those for the edge-classes. The negative effect of Like and Thankful reactions, which were eliminated in the Core Reaction Set Model due to their abnormal counts, could be proven as well. The inclusion of those reactions caused significant reductions in the F1 scores of the other reactions as can be seen from the results of the All Reaction Set Model.

This study represents modelling efforts that may be considered classical and limited in nature. Recent years have seen a significant growth in machine learning algorithms delivering exceptional results in many domains of text analysis, especially in finding non-linear relationships in the

![Fig. 5: Changes of the F1 score of the Core Reaction Model with Train-Test Division](image)
### TABLE VI

**Star Rating Model: Class Performance Measures**

| Star Rating Class | Train Set (%) | Accuracy | Precision | Recall | F1 Score |
|-------------------|---------------|----------|-----------|--------|----------|
| 1.0               | 95            | 0.5677   | 0.0001    | 0.5573 | 0.0021   |
|                   | 90            | 0.5700   | 0.0015    | 0.5598 | 0.0030   |
|                   | 80            | 0.5673   | 0.0007    | 0.5574 | 0.0015   |
|                   | 70            | 0.5686   | 0.0012    | 0.5586 | 0.0023   |
|                   | 50            | 0.5672   | 0.0013    | 0.5572 | 0.0026   |
| 1.5               | 95            | 0.5886   | 0.0151    | 0.5981 | 0.0293   |
|                   | 90            | 0.5888   | 0.0127    | 0.5985 | 0.0248   |
|                   | 80            | 0.5888   | 0.0142    | 0.5969 | 0.0277   |
|                   | 70            | 0.5873   | 0.0164    | 0.5961 | 0.0319   |
|                   | 50            | 0.5880   | 0.0176    | 0.5965 | 0.0341   |
| 2.0               | 95            | 0.6368   | 0.0841    | 0.6448 | 0.1457   |
|                   | 90            | 0.6392   | 0.1134    | 0.6470 | 0.1924   |
|                   | 80            | 0.6369   | 0.0953    | 0.6450 | 0.1653   |
|                   | 70            | 0.6373   | 0.1039    | 0.6449 | 0.1785   |
|                   | 50            | 0.6370   | 0.1074    | 0.6442 | 0.1839   |
| 2.5               | 95            | 0.7177   | 0.4403    | 0.7288 | 0.5481   |
|                   | 90            | 0.7162   | 0.4191    | 0.7280 | 0.5318   |
|                   | 80            | 0.7174   | 0.4324    | 0.7270 | 0.5421   |
|                   | 70            | 0.7150   | 0.4286    | 0.7251 | 0.5385   |
|                   | 50            | 0.7147   | 0.4248    | 0.7251 | 0.5356   |
| 3.0               | 95            | 0.7930   | 0.6707    | 0.8043 | 0.7314   |
|                   | 90            | 0.7892   | 0.6408    | 0.7981 | 0.7108   |
|                   | 80            | 0.8018   | 0.6696    | 0.8077 | 0.7320   |
|                   | 70            | 0.7932   | 0.6427    | 0.7982 | 0.7118   |
|                   | 50            | 0.7954   | 0.6543    | 0.8021 | 0.7206   |
| 3.5               | 95            | 0.8513   | 0.8456    | 0.8677 | 0.8565   |
|                   | 90            | 0.8455   | 0.8357    | 0.8630 | 0.8491   |
|                   | 80            | 0.8473   | 0.8390    | 0.8640 | 0.8513   |
|                   | 70            | 0.8485   | 0.8319    | 0.8615 | 0.8465   |
|                   | 50            | 0.8470   | 0.8283    | 0.8600 | 0.8439   |
| 4.0               | 95            | 0.8378   | 0.8135    | 0.8517 | 0.8321   |
|                   | 90            | 0.8334   | 0.7929    | 0.8443 | 0.8178   |
|                   | 80            | 0.8346   | 0.7888    | 0.8426 | 0.8148   |
|                   | 70            | 0.8309   | 0.7833    | 0.8405 | 0.8109   |
|                   | 50            | 0.8333   | 0.7913    | 0.8434 | 0.8165   |
| 4.5               | 95            | 0.7642   | 0.6144    | 0.7819 | 0.6879   |
|                   | 90            | 0.7630   | 0.6130    | 0.7822 | 0.6872   |
|                   | 80            | 0.7584   | 0.6011    | 0.7784 | 0.6783   |
|                   | 70            | 0.7604   | 0.6108    | 0.7805 | 0.6853   |
|                   | 50            | 0.7604   | 0.6110    | 0.7803 | 0.6853   |
| 5.0               | 95            | 0.7154   | 0.1554    | 0.7047 | 0.2545   |
|                   | 90            | 0.7144   | 0.1564    | 0.7037 | 0.2558   |
|                   | 80            | 0.7135   | 0.1564    | 0.7028 | 0.2558   |
|                   | 70            | 0.7160   | 0.1646    | 0.7054 | 0.2669   |
|                   | 50            | 0.7134   | 0.1653    | 0.7030 | 0.2677   |
data. Kowsari et al. [33] highlights a number of pre-processing steps (such as dimensionality reduction using topic modelling or principal component analysis) and algorithms that may be combined with the feature engineering work presented here (especially the selection of useful data classes and reduction to a star rating) for potentially more accurate models in the future. As noted therein, deep learning techniques hold particular promise. This is further explored in the work of Weeraprameshwara et al. [34], [35] that can be considered as a continuation of the research, which tests new models and develops a new embedding system using the Facebook data.

The study uses a word embedding developed by the work of Seneviratne et al. [23] for the Facebook dataset. However, developing an embedding structure based on the dataset may provide better sentiment annotation. Further enhancements can be done by introducing granularity to the embedding structure such as sentence embeddings.

An alternate approach to sophisticated modelling would be to examine pre-processing techniques therein that may not be possible in Sinhala as of the time of writing, due to limited or missing language resources and tooling, as noted by de Silva [10]; building these tools may further yield increases in accuracy even with a simplistic model.

References

[1] V. Gamage, M. Warushathiththana, N. de Silva and others, “Fast Approach to Build an Automatic Sentiment Annotator for Legal Domain using Transfer Learning,” in Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2018.

[2] P. Melville, W. Gryc and R. D. Lawrence, “Sentiment analysis of blogs by combining lexical knowledge with text classification,” in SIGKDD, 2009.

[3] E. Rudkowski, M. Haselmayer, M. Wastian and others, “More than bags of words: Sentiment analysis with word embeddings,” Communication Methods and Measures, vol. 12, p. 140–157, 2018.

[4] C. Aguwa, M. H. Oly and L. Monplaisir, “Modeling of fuzzy-based voice of customer for business decision analytics,” Knowledge-Based Systems, vol. 125, p. 136–145, 2017.

[5] V. Zobal, Sentiment analysis of social media and its relation to stock market, Univerzita Karlova, Fakulta sociálních věd, 2017.

[6] J. L. O. Hui, G. K. Hoon and W. M. N. W. Zainon, “Effects of word class and text position in sentiment-based news classification,” Procedia Computer Science, vol. 124, p. 77–85, 2017.

[7] R. Socher, A. Perelygin, J. Wu and others, “Recursive deep models for semantic compositionality over a sentiment treebank,” in EMNLP, 2013.

[8] S. De Silva, H. Indraraje, S. Premaratha and others, “Sensing the sentiments of the crowd: Looking into subjects,” in 2nd International Workshop on Multimodal Crowd Sensing, 2014.

[9] Y. Wijeratne, N. de Silva and Y. Shamugarajah, “Natural language processing for government: Problems and potential,” International Development Research Centre (Canada), 2019.

[10] N. de Silva, “Survey on publicly available sinhala natural language processing tools and research,” arXiv preprint arXiv:1906.02358, 2019.

[11] S. Ranathunga and N. de Silva, “Some languages are more equal than others: Probing deeper into the linguistic disparity in the nlp world,” arXiv preprint arXiv:2210.08523, 2022.

[12] Y. Wijeratne and N. de Silva, “Sinhala language corpora and stopwords from a decade of sri lankan facebook,” arXiv preprint arXiv:2007.07884, 2020.

[13] L. Graziani, S. Melacci and M. Gori, “Jointly learning to detect emotions and predict facebook reactions,” in ICANN, 2019.

[14] V. Jayawickrama, G. Weeraprameshvara, N. de Silva and Y. Wijeratne, “Seeking Sinhala Sentiment: Predicting Facebook Reactions of Sinhala Posts,” in 2021 21st International Conference on Advances in ICT for Emerging Regions (ICter), 2021.

[15] J. P. Singh, S. Irani, N. P. Rana and others, “Predicting the “helpfulness” of online consumer reviews,” Journal of Business Research, vol. 70, p. 346–355, 2017.
[16] L. Martin and P. Pu, “Prediction of helpful reviews using emotions extraction,” in AAAI, 2014.

[17] J. A. Caetano, H. S. Lima and others, “Using sentiment analysis to define twitter political users’ classes and their homophily during the 2016 American presidential election,” Journal of internet services and applications, vol. 9, p. 1–15, 2018.

[18] C. Pool and M. Nissim, “Distant supervision for emotion detection using Facebook reactions,” arXiv preprint arXiv:1611.02988, 2016.

[19] C. Freeman, M. K. Roy, M. Fattoruso and H. Alhooi, “Shared feelings: Understanding facebook reactions to scholarly articles,” in JCDL, 2019.

[20] C. Strapparava and R. Mihalcea, “Sentiment analysis for Sinhala Language using Deep Learning Techniques,” arXiv preprint arXiv:2011.07280, 2020.

[21] N. Medagoda, S. Shanmuganathan and J. Whalley, “Sentiment analysis using SentiWordNet 3.0,” in ICNC, 2015.

[22] P. D. T. Chathuranga, S. A. S. Lorensuhewa and M. A. L. Kalyani, “Sinhala sentiment analysis using corpus based sentiment lexicon,” in ICTer, 2019.

[23] M. Davis and K. Whistler, “Unicode character database,” Unicode Standard Annex, vol. 44, p. 95170–0519, 2008.