Exploring Emoji-based Synthetic Annotations for Filipino-English Sentiment Analysis

Sean Timothy S. Co, Nicholas Rupert E. Custodio, Alexis Louis L. Dela Cruz, Martin Christopher B. Sanchez, Jason Jan C. Jabanes, and Edward P. Tighe*

Department of Software Technology and Center for Language Technologies
De La Salle University, Manila, Philippines
{sean_timothy_co, nicholas_custodio, alexis_louis_delacruz, martin_christopher_sanchez, jason_jan_c_jabanes, edward.tighe}@dlsu.edu.ph

Abstract. Studies that conduct sentiment analysis on Filipino or a mix of Filipino and English text data suffer from a lack of readily available language resources. Many studies result in losing linguistic information when translating to a high-resource language or undergo the painstaking and expensive process of manual annotation. As an alternative, we propose an emoji-based sentiment annotation scheme as a language-independent means to automatically assess sentiment labels. The scheme utilizes an existing emoji sentiment lexicon and labels a text document as either positive or negative based on the mean sentiment score of the emojis present. The scheme can then be used on documents containing emojis to produce an initial set of data points to learn sentiment from. To evaluate the scheme's effectiveness, we used our proposed tagging scheme on collected tweets from the Philippines that have emojis. We then conducted experiments in building a sentiment classification model centered around the usage of a convolutional neural network. We also trained our own Word2Vec model using tweets from the same domain. Our experiments showed that while a model trained with emojis achieved a higher kappa score, it suffered from overfitting. Hence, our best-performing model for generalizing on text data, which was trained with emojis removed, had a kappa score of 0.5331 and an F1 score of 0.7665. While there is still much room for improvement, our initial findings suggest that the emoji-based sentiment annotation scheme is a potential option to address the limited resources available for the task of modeling sentiment for Filipino and Filipino-English text data.

Keywords: Emoji-based Sentiment Analysis, Synthetic Sentiment Labels, Philippine Natural Language Processing

1 Introduction

Sentiment Analysis (SA), in the context of analyzing text data, involves the investigation of the "positive or negative orientation that a writer expresses toward some object" [9]. It is widely used in fields such as market research, social media monitoring, and public opinion analysis, where understanding users' attitudes and emotional reactions is of key interest [13,4]. Studies employing SA

J. Caro et al. (eds.), *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2025)*, Atlantis Highlights in Computer Sciences 24,
https://doi.org/10.2991/978-94-6239-638-8_26

typically vary in terms of how sentiment is represented, but in general, sentiment can be represented as either a categorical (e.g., positive, negative, neutral) or dimensional label (e.g., continuous value from -1.0 to +1.0). The choice of representation can influence how models are trained and evaluated, and may depend on the nature of the text data, the level of interpretation, and the specific goals of the application.

Sentiment analysis often relies on large volumes of user-generated content from websites and social media platforms [12]. In particular, social media data is produced rapidly and tends to capture emerging trends and public opinions in real time. Because of this, this type of data has become valuable for understanding consumer attitudes, market behavior, and community sentiment. Businesses and organizations can use SA to gain insights into how users perceive their products or services on platforms like Facebook and Twitter [27,21,8].

As social media continues to grow as a powerful means for communication, the use of emojis with text has become increasingly popular [16]. Emojis are employed in textual communications with non-verbal components to convey moods, sentiments, and emotions [24,23]. Other studies have explored including emojis as a feature in sentiment analysis, particularly for low-resource languages. Research in languages such as Japanese [10,25], French [24], Chinese [15,14], Arabic [6] has demonstrated that incorporating emojis into SA models can improve the interpretation of sentiment in social media text. These studies show that emojis can enhance sentiment classification accuracy by incorporating the contextual value of emojis – highlighting their impact on the overall sentiment of these texts.

In the Philippines, there are numerous studies that apply sentiment analysis to Philippine text. [20] focused on estimating customer sentiment of Filipino Internet Service Providers using blog articles. [26] examined public opinion on the implementation of the K to 12 educational program through sentiment analysis of tweets. [19] explored a bilingual lexicon approach for SA in evaluating teaching performance using faculty evaluation submissions. [17] analyzed sentiment in tweets during the 2022 Philippine Presidential Election using a semi-supervised learning technique. These studies depend on manual annotation, where human annotators tag and label text. According to [18], while manual annotation may be more reliable for ground truth labels, it remains a bottleneck for many NLP research due to its time-consuming nature. In addition, translation is sometimes seen as an alternative approach in annotating Philippine data [20]. However, this usually results in a loss of information since it is difficult to maintain the exact sentiment of the original text during translation [3].

Given the issues surrounding the annotation of Philippine text data for SA, we explore the potential of an emoji-based sentiment annotation scheme for synthetic labeling. Rather than relying on translation or manual labeling, we propose a simple language-independent approach that utilizes the sentiment scores of emojis found in the Emoji Sentiment Ranking (ESR) [11]. Our scheme assigns binary sentiment labels (i.e., positive or negative) to Twitter posts by computing the mean sentiment score of emojis present in each post. We collected tweets from

the Philippines that contain emojis, applied our annotation scheme, and trained a Convolutional Neural Network (CNN) sentiment classification model using our automatically labeled training data. We also trained a custom Word2Vec model on the same Twitter corpus to generate features aligned with our data's domain. Results of our work show promise as our best model, trained on emoji-labeled data with emojis removed prior to training, achieved a kappa statistic of 0.5331. While there is still much to explore and improve upon, our initial findings suggest that an emoji-based annotation offers a viable alternative for generating sentiment labels for English and Filipino text data.

2 Methods

The highlight of our methods involves the synthetic generation of sentiment labels using an emoji-based sentiment lexicon. To explore the effectiveness of the scheme, we first collected tweets from Twitter and applied the scheme on all tweets with emojis. We then trained predictive models using a custom-built Word2Vec model and a CNN. Evaluation of the models included the use of a test set coming from the collected data. A high-level visualization of our methods can be found in Figure 1. The rest of the section discusses how we went about implementing our activities in detail.

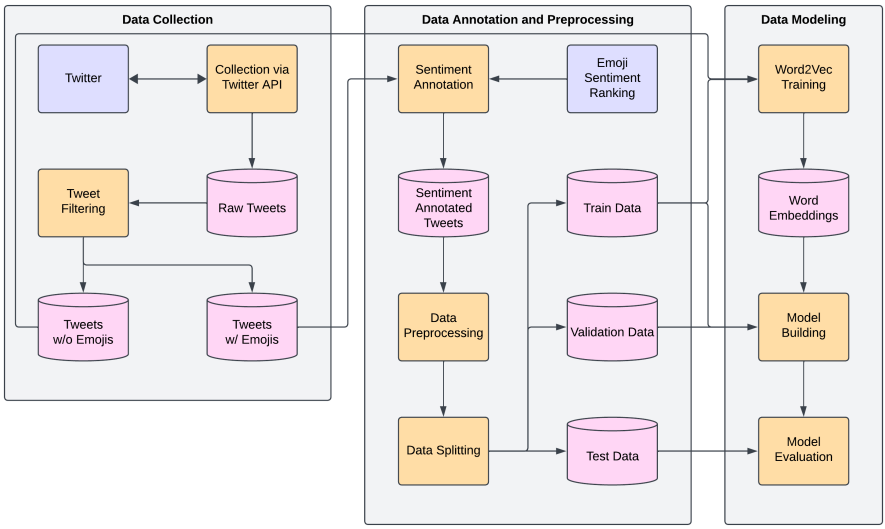


Fig. 1: An overview of the activities conducted for our study.

2.1 Emoji-based Sentiment Annotation Scheme

We designed an annotation scheme that utilizes an existing emoji sentiment lexicon and provides a positive or negative sentiment label for a document solely based on the emojis present. For the lexicon, we used the Emoji Sentiment Ranking [11], which contains a total of 969 emojis and their respective sentiment scores. However, we only used 751 of these emojis as these occurred five or more times in their data. According to [11], emojis with occurrence counts less than five were considered "not very reliable". Out of 751 emojis, 619 have positive sentiment score, 99 have negative sentiment score, and 33 have neutral (or zero) sentiment score.

As an algorithm, the annotation scheme takes an input document that contains at least one emoji present in the ESR lexicon. Emojis that are not part of the lexicon are ignored and any documents without a recognized emoji are not annotated. We compute the sentiment score of a document by getting the mean of sentiment scores of all recognized emojis. A tweet is then labeled positive if the mean sentiment score is greater than zero and negative if the mean is less than zero. As a limitation, tweets with mean sentiment scores equal to zero were not labeled as it isn't clear which label they should be given. For the sake of the initial experiment, we limited the possible sentiment labels to positive and negative.

2.2 Data Collection

As our annotation scheme leveraged emojis, we sought to collect data from Twitter (currently referred to as X), a social media platform with a large volume of informal text data. We utilized Twitter API v2 with academic access[1] to collect tweets made in the Philippines between the period of March to December 2020. We set our search queries to focus on the bounding box of the Philippines[2] so that the collection would mainly result in languages found in the Philippines – primarily English and Filipino.

Our data collection resulted in a total of 10,088,651 tweets after dropping all duplicates. Of the total tweets, around 2.79 million tweets or 27.65% of the initial data contained at least one emoji found in the ESR lexicon. We set this data aside for synthetic sentiment annotation. The remaining 7.30 million or 72.35% of the tweets (i.e., those that did not include recognized emojis or did not include any emojis) were reserved for training a Word2Vec model, as later discussed in Section 2.6.

To examine the language composition of the data, we extracted the language tag provided by Twitter when tweets were collected. We summarize the Twitter language tag distributions across (a) the entire dataset, (b) the subset of tweets with recognized emojis, and the (c) subset of tweets without recognized

[1] The type of access used in this study is no longer available due to changes that X made to how they managed 3rd party applications.

[2] Defined using the coordinates 117.17427453, 5.58100332277, 126.537423944, and 18.5052273625.

emojis in Table 1. As anticipated, more than > 82% of the tweets are labeled as either Tagalog[3] or English. We also note a number tweets labeled as undefined, Indonesian, and Spanish in the top five language tags. We attribute this to noise – either a tweet was too short or informal (leading to an undefined tag) or likely misclassified as Indonesian or Spanish (two languages with lexical similarities with Tagalog). As a result, we decided to retain tweets across all language tags to capture a more authentic representation of how people in the Philippines express themselves online. This approach prevents the dataset from being restricted to strictly English or Tagalog text.

Table 1: The percentage of Twitter language tags found across (a) All Tweets (n=10,088,651), (b) Tweets with Emojis (n=2,789,512), and (c) Tweets without Emojis (n=7,299,139). The top 5 tags are shown with all other tags aggregated into the "Others" tag.

Twitter Language Tag	**All Tweets**	**w/ Emojis**	**w/o Emojis**
Tagalog (TL)	55.8%	47.1%	59.1%
English (EN)	28.4%	34.9%	25.9%
Undefined (UND)	8.2%	11.3%	7.0%
Indonesian (IN)	3.0%	2.6%	3.1%
Spanish (ES)	0.8%	0.7%	0.8%
Others	3.8%	3.4%	4.0%

2.3 Sentiment Annotation

Applying our annotation scheme to the subset of tweets with emojis resulted in roughly 82.19% of the tweets being labeled as positive, while around 17.78% were negative. Additionally, a small number of tweets (0.03%) were removed due to their sentiment scores being exactly zero. The high imbalance of positive to negative instances poses a problem to the learning process; hence, data balancing is applied and later discussed in Section 2.5.

2.4 Text Preprocessing

We built a custom tokenizer that performed the following tasks:

- User handles, URLs, time expressions, phone numbers, and generic numbers are each identified and converted to a corresponding placeholder token
  - @username → <USERHANDLE >

[3] The Tagalog language is the basis for Filipino, the national language of the Philippines. Twitter's language tags, at least of the time of collection, did not include a Filipino tag.

  - http://example.com/page → <URL >
  - 1234 → <NUMBER >
- Reduce repeating characters greater than or equal to three
  - "Hiiiii" became "Hiii"
- Lowercase said tokens except those that are entirely in uppercase.
  - "Hello" → "hello"
  - "WORLD" → "WORLD"

After tokenization, the tweets are filtered to retain only those that contain at least one alphanumeric character. In other words, tweets are removed if they are only composed of emojis, punctuations, and special symbols (e.g. Unicode characters). Finally, duplicate tweets and tweets that are empty after preprocessing are removed from the dataset.

2.5 Data Splitting and Balancing

With 2,514,165 tweets labeled using our synthetic sentiment annotation scheme, we split the dataset using random sampling into 64% for training (n=1,609,065), 16% for validation (n=402,267), and 20% for testing (n=502,833). These proportions were obtained by first dividing the data 80%–20% for training and testing, then allocating 20% of the training portion for validation.

Following the random sampling used to split the dataset, we observed a significant class imbalance across the training, validation, and test sets – with approximately 82% of the samples labeled as positive and only 18% as negative. This imbalance poses a risk of overfitting due to the overrepresentation of positive samples. To address this, we applied random undersampling to each subset using the size of the minority class (i.e., negative) as the reference. We opted for undersampling over oversampling techniques as oversampling runs the risk of overemphasizing biases in the minority class. As we have a large volume of data, we judged that the trade-off of losing some linguistic variance in the majority class was acceptable. This process substantially removed roughly 77% of the positive tweets; however, we were still left with a relatively large dataset especially when compared to manually annotated sentiment datasets. The resulting balanced datasets contain equal numbers of positive and negative samples for each split. The final sample counts for the train, validation, and test sets are summarized in Table 2.

2.6 Word2Vec Model Training

While off-the-shelf pretrained Word2Vec models are widely available, we opted to train our own model from scratch as we have a sizable amount of the data that was discarded due to not having emojis. This would ensure that both English and Filipino words are represented in the embedding space and that the Word2Vec model's domain aligns with our sentiment classification dataset.

We trained a word embedding model using the skip-gram implementation in Gensim [22]. For training data, our model learned from a total of 8,922,888

Table 2: A summary of the sample counts for each sentiment labels across each of the train, validation, and test datasets.

Dataset	Positive	Negative	Total
Train	291,406	291,406	582,812
Validation	72,826	72,826	145,652
Test	90,632	90,632	181,264
Total	454,864	454,864	909,728

tweets, which is composed of (a) 7,292,733 rejected tweets (i.e. without emojis), (b) 21,090 tweets that were given a sentiment score of zero, and (c) 1,609,065 tweets from the entire training dataset before undersampling. The data contained a total of 99,016,632 tokens with a vocabulary size of 1,583,053. Each tweet had an average of 11.1 tokens with a standard deviation of 10.2. As for configurations, we utilized the following:

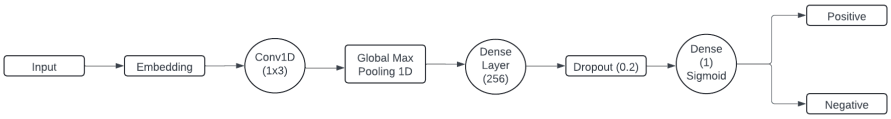
- Vector size: 200
- Minimum word count: 5
- Negative sampling: 5
- NS exponent: 0.75
- Epochs: 5
- Window size: 1,2,3

One noteworthy item from the list of parameters is the window size. We decided to experiment with embedding models with window sizes 1, 2, and 3 as tweets are relatively short in terms of average words/tokens per document. Discussion on the conduct of an ablation study and final configuration can be found in Sections 2.9 and 3.1.

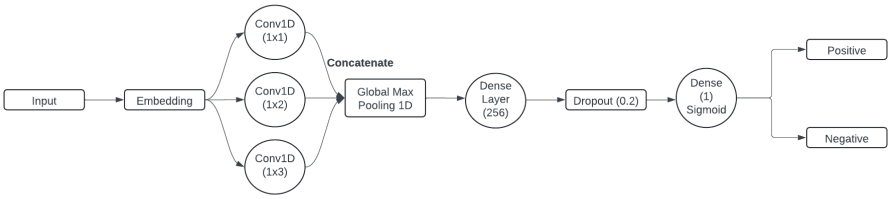
2.7 Sentiment Model Training

Our sentiment classifier is a neural network model designed to classify documents as either positive or negative. We utilize Keras [1] to implement our neural network. The network takes an input size of 236 – the max token length found in the training data. Documents less than 236 tokens are padded up to this length, while documents with more than 236 are truncated. We have an embedding layer with values initialized from our custom Word2Vec model, where each word vector has a dimension of 200. To optimize on training time, we freeze the embedding layer during training – preventing updates to the Word2Vec weights. Data is then fed into a convolutional layer to capture patterns between words in the input documents. We experimented with two configurations: (1) a single convolutional layer with a kernel size of 3 (single-channel) and (2) three separate convolution layers of kernel sizes 1, 2, and 3 (multi-channel). All convolutional layers have 100 filters and utilize a ReLU activation function. The outputs from the convolutional

layer are then passed through a max pooling layer and then through a single dense layer of 256 units utilizing a ReLU activation function. To mitigate possible overfitting, we implement a dropout layer with a dropout rate of 0.2. However, we experiment with and without the dropout layers to better observe any effects. Finally, data is fed through a single-unit dense layer with a sigmoid activation function to classify each document as either positive or negative. The network uses the Adam optimizer and binary cross-entropy for the loss function. More details on the conduct of an ablation study and final configuration can be found in Sections 2.9 and 3.1. We also visualize our neural network architectures in Figure 2.



(a) Single-Channel CNN Architecture.



(b) Multi-Channel CNN Architecture.

Fig. 2: Two neural networks that were experimented upon in our study.

2.8 Hyperparameter Tuning

We used a grid search approach to choose the optimal hyperparameter configurations for the neural network. The tuning process was focused on exploring different batch size and learning rate values. For each combination, the network was trained on 100 epochs with an early stopping mechanism in place to monitor and stop training once the binary cross-entropy loss stopped decreasing for at least 10 epochs. The search space for tuning is defined as follows:

- Batch size: 32, 64, 128, 256
- Learning rate: 0.00001, 0.0001, 0.001

2.9 Conduct of the Ablation Study

An ablation study is performed to determine the optimal model configuration, specifically exploring factors such as (a) the inclusion/exclusion of dropout lay-

ers, (b) the choice between single-channel or multi-channel CNN architecture, and (c) word embedding models trained on window sizes 1, 2, and 3. The data used for the ablation study involves the training data and the validation data, and emojis were stripped from both sets. Configurations from the best performing model (based on validation binary cross-entropy loss) are used for the final model configuration.

2.10 Test Evaluation

We utilize the neural network configurations from the ablation study's best performing model. The network is trained using the training data and evaluated using the testing data. We report metrics such as accuracy, kappa statistic, AUC, and F1-score, although due to the unbalanced nature of the test data, kappa statistic serves as the main metric for evaluation.

Additionally, we look to observe the effects of the inclusion or exclusion of emojis from the training and test sets. We recognize that keeping emojis in both the train and test data would bias the model as emojis were the primary basis for the sentiment labels. Our ideal model is one that can infer the sentiment of a text based on the words surrounding an emoji without the help of the emoji itself. This evaluation would allow us to observe how well the emoji-based annotation scheme works from the most biased setup to an ideal setup.

3 Results and Discussion

In this section, we present the results of the ablation study and the performance of our sentiment prediction models.

3.1 Results of the Ablation Study

For the ablation study, a total of 12 models were built using the different configurations discussed in Section 2.9. A summary of the models and their respective performances can be found in Table 3. The performance of each model is measured by its validation accuracy (ACC) and binary cross-entropy loss.

Results of the ablation study show that the model with the highest validation accuracy of 0.7673 was configured with dropout, a multi-channel convolutional layer, and a Word2Vec window size of 2. This configuration also achieved the lowest validation loss of 0.4843 – making it the best overall performer among the configurations tested. An analysis of the individual factors reveals a few trends. On average, models with a dropout layer and a multi-channel CNN architecture demonstrated a slight performance advantage over their counterparts. Similarly, models using a Word2Vec window size of 2 performed slightly better on average than those with window sizes of 1 or 3. As the performance differences between all configurations are minimal, we conclude that no single factor had a major impact on the outcome. Therefore, we will proceed with the best-performing model's configuration for final testing.

Table 3: Results of the ablation study. Each configuration varies in terms of dropout inclusion, CNN architecture type (single or multi-channel), and Word2Vec (W2V) window size, while also listing the best hyperparameters used (i.e., batch size and learning rate). Reported metrics are validation accuracy (ACC) and binary cross-entropy loss (Loss). The best-performing configuration for both metrics is shown in bold.

Dropout	**CNN Type**	**W2V Window Size**	**Hyperparameters**		**Validation**	
			Batch	**LR**	**ACC**	**Loss**
With	Single Channel	1	256	1.00E-03	0.7644	0.4872
Without	Single Channel	1	128	0.001	0.7631	0.4894
With	Multi-Channel	1	256	0.001	0.7664	0.4866
Without	Multi-Channel	1	128	0.001	0.7639	0.4878
With	Single Channel	2	32	0.0001	0.7660	0.4861
Without	Single Channel	2	256	0.001	0.7646	0.4883
With	Multi-Channel	2	256	0.001	**0.7673**	**0.4843**
Without	Multi-Channel	2	64	0.001	0.7644	0.4867
With	Single Channel	3	256	0.001	0.7648	0.4856
Without	Single Channel	3	32	0.0001	0.7639	0.4880
With	Multi-Channel	3	256	0.001	0.7658	0.4853
Without	Multi-Channel	3	128	0.001	0.7663	0.4847

3.2 Final Testing

For the final testing phase of our study, we used the configurations identified in Section 3.1 to train models across 4 scenarios where emojis are stripped or remain in the train and test sets. For simplicity, we refer to each of the 4 scenarios as follows:

- **M1**: Train with Emoji; Test with Emoji
- **M2**: Train with Emoji; Test without Emoji
- **M3**: Train without Emoji; Test with Emoji
- **M4**: Train without Emoji; Test without Emoji

We also reiterate that kappa statistic is the main basis for model performance given the unbalanced test dataset. A summary of the models and their respective performances can be found in Table 4. We also report train accuracy, test accuracy, test area under the curve (AUC), and F1 measure. We also report the best batch size and learning rate hyperparameters.

The results show that the model with the highest performance across all evaluation metrics was M1, which had a test accuracy of 0.9972 and a kappa statistic of 0.9943. Given that the data was annotated using an emoji sentiment lexicon, this high performance is unsurprising, as M1 was able to map emojis to sentiment scores almost perfectly. However, this high performance proves to be an over-reliance on emojis. When observing M2, where emojis are removed from the test data, the kappa statistic plummets from 0.9943 to 0.2964 – a

Table 4: Results of our final tests where we observe the effect of emojis being present in the train and test sets. We report the training accuracy (ACC), kappa statistic, area under the curve (AUC), and F1 score, as well as the identified best hyperparameters.

ID	w/ Emojis?		Best Hyperparameters		Train	Test			
	Train	Test	Batch Size	Learning Rate	ACC	ACC	Kappa	AUC	F1
M1	Yes	Yes	256	1.00E-03	0.9980	0.9972	0.9943	0.9999	0.9972
M2	Yes	No	128	1.00E-05	0.9152	0.6482	0.2964	0.7618	0.6286
M3	No	Yes	256	1.00E-04	0.7741	0.7587	0.5174	0.8372	0.7586
M4	No	No	256	1.00E-03	0.7789	0.7666	0.5331	0.8464	0.7665

0.7008 reduction in agreement. This significant drop indicates that M1 is unable to associate the surrounding words with the sentiment score, making models trained with emojis an inappropriate choice despite their high accuracy, AUC, and F1 metrics.

In a more promising observation, models that were trained without emojis (i.e., M3 and M4) showed significantly better performance than M2, which was trained with emojis but tested without them. M3 and M4 achieved kappa statistics of 0.5174 and 0.5331, respectively, representing a significant 0.22-point improvement compared to M2's kappa of 0.2964. This demonstrates that training with emojis leads to the model memorizing how emojis are mapped to the lexicon's sentiment scores and not associating sentiment to surrounding words. The M1 and M2 models were clearly overly dependent on emoji tokens. Additionally, when comparing M3 and M4, we see that the model performs best when the training and testing conditions are consistent. M4 (trained and tested without emojis) outperforms M3 (trained without emojis, tested with emojis) across all test metrics, suggesting the model is most effective when handling the same type of data it was trained on. Overall, the results of M3 and M4 show that patterns of sentiment can be found in the surrounding words of an emoji despite weak supervision.

4 Conclusion and Future Direction

In this study, we aimed to provide an alternative automatic annotation process for sentiment analysis in Philippine-based tweets through the use of an emoji-based sentiment annotation scheme. Our data collection efforts resulted in a dataset of around 2.5M labeled tweets out of the initial 10M collected, with 2.05M being positive tweets and 450,000 being negative tweets. Using 8.9M of the 10M tweets collected, we were able to create word embeddings that capture the language patterns and vocabulary of Philippine-based tweets. Our final testing demonstrated that while the model trained and tested with emojis (M1) achieved a near-perfect kappa statistic of 0.9943, the higher performance was the result of overfitting on emoji patterns. The model that proved to be the most

robust and reliable for generalizing on text data was M4, which was trained and tested without emojis and achieved a kappa statistic of 0.5331. Our findings suggest that approaching sentiment annotation through an emoji-based scheme has potential to create large useful datasets. Even without further fine-tuning of the processing pipeline, we clearly see that patterns were learned through a large volume input source. However, the approach requires further refinement to avoid training models that are overly dependent on emoji cues.

Looking ahead, there are several avenues for future work. Our current approach is limited to binary sentiment labels (i.e., positive/negative). A more sophisticated approach should be explored for creating sentiment labels as our current methodology used a naive approach of averaging scores from the Emoji Sentiment Ranking and splitting at a score of 0.0. Additional research is needed to determine a better way of splitting positive and negative sentiments and to even look into the possibility of determining how the neutral label can be incorporated into the process. Expanding the annotation scheme to account for neutral sentiment would allow us to compare against existing sentiment datasets focused on Philippine-based text data, such as that in [7] and [2]. In addition, our reliance on the ESR, which is a relatively older emoji lexicon, presents another area for improvement. Newer emoji sentiment lexicons, such as the Multidimensional Lexicon of Emojis (MLE) [5], could be explored as alternative sources of sentiment information. Using a lexicon like MLE would allow for the exploration of fine-grained affect labels beyond high-level positive or negative sentiments. Finally, while we show the potential of our synthetic annotation scheme, it is important to ensure that human judgment remains in the loop. Future steps should include additional testing to evaluate the alignment of human judgment with both the synthetic annotation scheme and the sentiment predictions of the trained model. This validation step would be essential for improving the real-world applicability of our work.

5 Resource Availability and Reproducibility

To promote transparency and reproducibility, the resources developed in this study will be made available through the following public repository: https://github.com/dlsucelt/PH-Senti-Toolkit. The repository will include (a) tweet IDs corresponding to the dataset used in this work (in compliance with Twitter's data-sharing policy), (b) the pre-trained Word2Vec embeddings and best-performing sentiment classification models, and (c) core preprocessing and training scripts. These resources are intended exclusively for academic and research purposes. In other words, the models are experimental and not designed for commercial or end-user deployment. By releasing these resources, we aim to foster open collaboration and support continued research on SA in the Philippine context.

6 Declaration of Generative AI Usage

We would like to acknowledge that ChatGPT and Gemini were used to assist in the preparation of this paper. We mainly utilized these generative AI tools in reviewing and understanding related literature, outlining sections of the document, and refining the style and organization of the writing. While generative AI was used to support the creation of this document, we ensured that all output was thoroughly reviewed and evaluated. The final content, structure, and arguments presented in this paper reflect our intellectual work and critical judgment.

References

1. Chollet, F., et al.: Keras. https://keras.io (2015)
2. Cosme, C., De Leon, M.: Sentiment analysis of code-switched Filipino-English product and service reviews using transformers-based large language models. In: Iglesias, A., Shin, J., Patel, B., Joshi, A. (eds.) Proceedings of World Conference on Information Systems for Business Management. ISBM 2023. Lecture Notes in Networks and Systems, vol. 834. Springer, Singapore (2024), https://doi.org/10.1007/978-981-99-8349-0_11
3. Dashtipour, K., Poria, S., Hussain, A., Cambria, E., Hawalah, A.Y., Gelbukh, A., Zhou, Q.: Multilingual sentiment analysis: State of the art and independent comparison of techniques. Cognitive Computation 8, 757–771 (2016)
4. Feldman, R.: Techniques and applications for sentiment analysis. Communications of the ACM 56(4), 82–89 (2013)
5. Godard, R., Holtzman, S.: The multidimensional lexicon of emojis: a new tool to assess the emotional content of emojis. Frontiers in Psychology 13, 921388 (2022)
6. Hakami, S.A.A., Hendley, R.J., Smith, P.: Arabic emoji sentiment lexicon (Arab-ESL): A comparison between Arabic and European emoji sentiment lexicons. In: Proceedings of the Sixth Arabic Natural Language Processing Workshop. pp. 60–71 (2021)
7. Imperial, J.M., Orosco, J., Mazo, S.M., Maceda, L.: Sentiment analysis of typhoon related tweets using standard and bidirectional recurrent neural networks. arXiv preprint arXiv:1908.01765 (2019)
8. Joyce, B., Deng, J.: Sentiment analysis of tweets for the 2016 US presidential election. In: 2017 IEEE MIT Undergraduate Research Technology Conference (URTC). pp. 1–4 (2017)
9. Jurafsky, D., Martin, J.H.: Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. 3rd edn. (2025), https://web.stanford.edu/~jurafsky/slp3, online manuscript released January 12, 2025
10. Kimura, M., Katsurai, M.: Automatic construction of an emoji sentiment lexicon. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. pp. 1033–1036 (2017)
11. Kralj-Novak, P., Smailović, J., Sluban, B., Mozetič, I.: Sentiment of emojis. PLOS ONE 10(12), 1–22 (December 2015), https://doi.org/10.1371/journal.pone.0144296
12. Li, S., Liu, F., Zhang, Y., Zhu, B., Zhu, H., Yu, Z.: Text mining of user-generated content (UGC) for business applications in e-commerce: A systematic review. Mathematics 10, 3554 (2022), https://www.mdpi.com/2227-7390/10/19/3554

13. Liu, B.: Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies, vol. 5. Morgan & Claypool Publishers (2012)
14. Liu, C., Fang, F., Lin, X., Cai, T., Tan, X., Liu, J., Lu, X.: Improving sentiment analysis accuracy with emoji embedding. Journal of Safety Science and Resilience 2(4), 246–252 (2021)
15. Lou, Y., Zhang, Y., Li, F., Qian, T., Ji, D.: Emoji-based sentiment analysis using attention networks. ACM Transactions on Asian and Low-resource Language Information Processing (TALLIP) 19(5), 1–13 (2020)
16. Mabokela, R., Roborife, M., Celik, T.: Investigating sentiment-bearing words-and emoji-based distant supervision approaches for sentiment analysis. In: Proceedings of the Fourth Workshop on Resources for African Indigenous Languages (rail 2023). pp. 115–125 (2023)
17. Macrohon, J.J.E., Villavicencio, C.N., Inbaraj, X.A., Jeng, J.H.: A semi-supervised approach to sentiment analysis of tweets during the 2022 Philippine presidential election. Information 13(10), 484 (2022)
18. Neves, M., Ševa, J.: An extensive review of tools for manual annotation of documents. Briefings in bioinformatics 22(1), 146–163 (2021)
19. Pacol, C., Palaoag, T.: Bilingual lexicon approach to English-Filipino sentiment analysis of teaching performance. In: IOP Conference Series: Materials Science and Engineering. vol. 1077, p. 012044. IOP Publishing (2021)
20. Patacsil, F.F., Malicdem, A.R., Fernandez, P.L.: Estimating Filipino ISPs customer satisfaction using sentiment analysis. Computer Science and Information Technology 3(1), 8–13 (2015)
21. Pravalika, A., Oza, V., Meghana, N.P., Kamath, S.S.: Domain-specific sentiment analysis approaches for code-mixed social network data. In: 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). pp. 1–6 (2017)
22. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
23. Santamaría-Bonfil, G., López, O.G.T.: Emoji as a proxy of emotional. Becoming Human with Humanoid: From Physical Interaction to Social Intelligence p. 45 (2020)
24. Saroufim, C., Almatarky, A., Hady, M.A.: Language independent sentiment analysis with sentiment-specific word embeddings. In: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. pp. 14–23 (2018)
25. Tomihira, T., Otsuka, A., Yamashita, A., Satoh, T.: What does your tweet emotion mean? Neural emoji prediction for sentiment analysis. In: Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services. pp. 289–296 (2018)
26. Torio, J.O., Bigueras, R.T., Maligat, D.E.: Sentiment analysis on Kto12 program implementation in the philippines. In: 2018 Fourth International Conference on Advances in Computing, Communication Automation (ICACCA). pp. 1–5 (2018)
27. Tumasjan, A., Sprenger, T., Sandner, P., Welpe, I.: Predicting elections with Twitter: What 140 characters reveal about political sentiment. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 4, pp. 178–185 (2010)