Sentiment Analysis on Book Reviews Using Convolutional Neural Network (CNN) Long Short-Term Memory (LSTM) Hybrid

Lenz Baron S. Balita
School of Information Technology, Mapua University, Makati, Philippines
slashkoopax2@gmail.com

Kyle Matthew A. Degrano
School of Information Technology, Mapua University, Makati, Philippines
kyledegrano@gmail.com

Andrei Daniel A. Pamoso
School of Information Technology, Mapua University, Makati, Philippines
adapamoso@gmail.com

Joel C. De Goma
School of Information Technology, Mapua University, Makati, Philippines
jcdegoma@mapua.edu.ph

ABSTRACT

Sentiment analysis is one of the most prominent methods on gathering and analyzing insightful textual data from various sources. The information produced from such a method can be imperative in understanding the general public's sentiment on a certain product or service. Over the years, countless sentiment analysis models have already been established using known algorithms such as Naive Bayes, Support Vector Machine, and many more. However, with the advent of novel technologies and neural networking, recent studies have employed Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) Recurrent Neural Network together to formulate more efficient and modernized models (Rehman, Malik, Raza, & Ali, 2019). As such, the study proposed a similar model to analyze the sentiments of book user reviews from GoodReads categorized according to three distinct genres – children's, young adults', and romance. The paper also aimed to determine the viability and effects of amalgamating features such as Word2Vec, POS, and SenticNet to the overall accuracy (Ayutthaya & Pasupa, 2018). Once the model was trained to the procured dataset, the results suggested that combining Word Embedding, POS, and SenticNet features drastically improves its performance in contrast to other tested variations. Amalgamating the three features to a CNN-LSTM hybrid model yielded an F1-score of 90%; whilst other variants with lacking features or a standalone CNN or LSTM model only resulted to F1-scores around 86% below. Graphing the performance of all the constructed models to an ROC curve also indicated the effectiveness of the proposed model – having an AUC value of 0.9588.

CCS CONCEPTS

• **Computing methodologies** → Artificial intelligence; Natural language processing; Information extraction; Artificial intelligence; Natural language processing; Lexical semantics.

ICEBI 2022, October 14–16, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9864-0/22/10…$15.00
https://doi.org/10.1145/3572647.3572666

KEYWORDS

Sentiment Analysis, Convolutional Neural Network, Long Short-Term Memory, Word2Vec, SenticNet, POS feature, GoodReads

ACM Reference Format:
Lenz Baron S. Balita, Kyle Matthew A. Degrano, Andrei Daniel A. Pamoso, and Joel C. De Goma. 2022. Sentiment Analysis on Book Reviews Using Convolutional Neural Network (CNN) Long Short-Term Memory (LSTM) Hybrid. In *2022 6th International Conference on E-Business and Internet (ICEBI 2022), October 14–16, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3572647.3572666

1 INTRODUCTION

Sentiment Analysis is an NLP (natural language processing) method that determines if the textual data is either positive or negative. This has multiple uses, an example would be for a business that requires you to process customer feedback by using the customer's textual data [8]. However, Sentiment Analysis can have inaccurate problems and the researchers will be attempting to make Sentiment Analysis more accurate [7] and compare the differences when techniques are applied [2]. A research paper on sentiment analysis on movie reviews employed Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) hybrid with great success to produce remarkable results [5]. Another study has also mentioned that combining CNN and LSTM networks allows them to perform better than when used as standalones [6]. A CNN-LSTM framework can overcome certain limitations of high-quality feature dependence [1] and can be able to handle enormous amounts of high-dimensional data with far greater efficiency and accuracy [4]. As such, the study aims to apply the hybrid framework alongside feature additions to conduct sentiment analysis on book reviews.

1.1 Statement of the problem

The study aims to construct a CNN-LSTM hybrid sentiment analysis model to examine social sentiments within book reviews. As such, the paper also attempts to answer the following questions: (1) How accurate is the proposed model sentiment analysis model in determining social sentiments within book reviews using accuracy, precision, recall, and F1 scores as parameters? (2) Will the improved hybrid model be able to yield better performance as opposed to traditional machine learning techniques? (3) Does integrating POS embedding, Sentic features, and Word embedding features to the proposed model improve its performance?

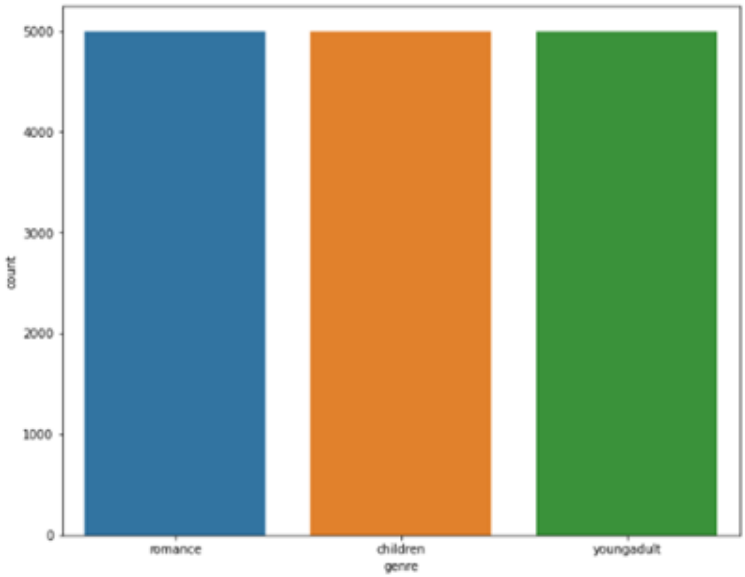

Figure 1: Amount of reviews in the dataset by genre

1.2 Objectives

The main objective of the study is to construct a sentiment analysis model for analyzing social sentiments within book reviews. The proposed model is formulated using a hybrid of CNN-LSTM frameworks due to their novelty and proven accuracy. As such, the paper aims to fulfill the following statements: (1.) To determine the model's precision and accuracy in terms of analyzing social sentiments within book reviews. (2.) To capture the sentiment polarity from book reviews with the use of Hybrid CNN-LSTM model with word embedding, POS embedding, and Sentic features. (3.) To compare the accuracy of our hybrid model to the Hybrid CNN-LSTM with word embedding, standalone CNN, and LSTM frameworks.

1.3 Significance of the study

The study aims to develop a sentiment analysis model using a combination of LSTM and CNN frameworks to analyze and gather sentimental insights from book reviews. The researchers proposed that the hybrid model be able to maximize efficiency due to the amalgamation of both algorithms. As such, the said model shall be beneficial to the following groups: Authors, Readers, and Researchers

2 METHODOLOGY

2.1 Dataset and structure

The proposed model utilized a dataset that was webscraped from GoodReads, which was conducted by researchers Julian McAuley and Menting Wan [3]. The researchers scraped various kinds of data like books and users' public shelves. Datasets on user reviews were then added and categorized according to different genres. The dataset employed in the study was organized by genres as reviews have varying densities, wordings, and word lengths based on the genre. As such, the researchers utilized 15,000 user reviews separated into three genres: children's, romance, and young adult's. Each genre consists of 5,000 reviews thus amounting to 15,000 in total as seen in figure 1. The reviews were procured manually from the main source to ensure that they are written in English, and no bad reviews were gathered and fed into the model.

Figure 2 illustrates the dataset with annotations that were utilized in the study to train the proposed hybrid model. As the reviews utilized in the study were categorized into three separate categories, their structures also differ by genre. Whilst exploring the corpus employed, the researchers discovered that children's reviews contain few words on average compared to romance reviews and young adults' reviews. The average number of words found in the romance reviews had a value of 185 words as such entries are often detailed and exploratory. Young adult reviews came in second to romance with an average word count of 170 words as it contains both short and long-written reviews. Figure 3 below illustrates the average word count of the three genres in contrast to the average word count of the entire dataset – which is 145 words.

2.2 Conceptual framework

The study's proposed conceptual framework begins with the procurement of data from the Amazon Review Polarity dataset to formulate the model's training and testing dataset. Data cleansing comes right after data gathering wherein the pre-processing stage commences to segment the textual data, tokenize, remove unnecessary stop words, and stemming and/or lemmatization. Once the raw data are cleaned, the researchers segregate them as either a part of training or testing datasets. Once the cleansing stage is complete to refine the two datasets, the textual data will undergo through the following layers for their sentiment polarities to be classified as embedding layer, convolution layer, global max-pooling layer, and dense layer. The embedding layer will be used for the features that

	review_text	genre	polarity	book_id
0	So the other day Elizabeth and I are in the bo...	romance	1	1893
1	It is very hard to believe this is all true bu...	romance	1	17939501
2	Enjoyable read I liked that Connie is not a ty...	romance	1	7840190
3	There are definitely too many book lately with...	romance	1	15463724
4	I really enjoyed this book Chick lit with brai...	romance	1	2718668

Figure 2: User reviews dataset with annotations

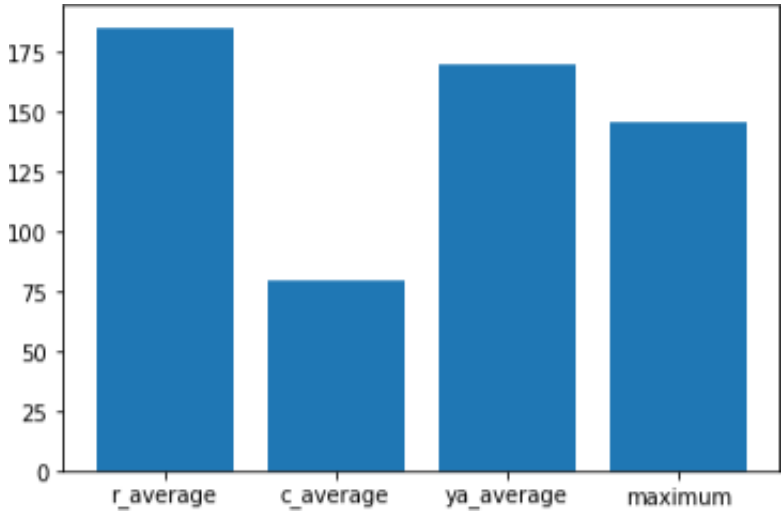

Figure 3: Average number of words found in each genre in contrast with the average number of words in all the reviews.

are to be extracted from the text with the use of POS Embedding, Word Embedding, and Sentic Embedding.

2.3 Features

- **Word Embedding Feature** - The researchers will use the model to take the input as word embeddings and feed them into the convolutional layers that allow it to extract local features. The researchers will use a word-to vector tool called "Word2Vec" that is to be combined with POS embedding and Sentic Features so that it can be applied to the CNN and LSTM hybrid model for Semantic Analysis.
- **POS Embedding Feature** - The POS embedding feature is done by POS2Vec model, which concatenates syntactic categories of words to the word2vec Model,. Instead of employing words represented in sentences in the corpus, types of POS are utilized as words in the corpus. From this, the POS embedding feature allows the model to understand the structure of sentences in a POS perspective.
- **Sentic Feature** - The feature is a five-dimension vector wherein the first four elements are pleasantness, attention, sensitivity, and aptitude values, respectively. The last element among the five-dimensions represents the polarity value wherein the data ranges between -1 and 1 [4] It identifies each word if it is either 'positive', 'negative', or 'neutral' to allow the model to identify the sentences easily.

3 RESULTS AND DISCUSSION

3.1 Exploratory data analysis

To test the viability of the proposed machine learning hybrid model with combined features (POS, Word2Vec, and SenticNet), the researchers utilized text datapoints from UCSD Book Graphs which contained webscraped GoodReads book reviews [9]. The raw corpus was reduced to 15,000 to accommodate to real-life financial expenses. To explore the data effectively, determining term frequencies play a key role in exploring the contents and complexities of the study's corpus. Sklearn's count vectorizer mode was employed to determine the most common words found in the corpus in both positive and negative text classes [10].

Figure 5 illustrates the study's term frequency data frame which showcases the ubiquity of stopwords and a couple of nouns, verbs,

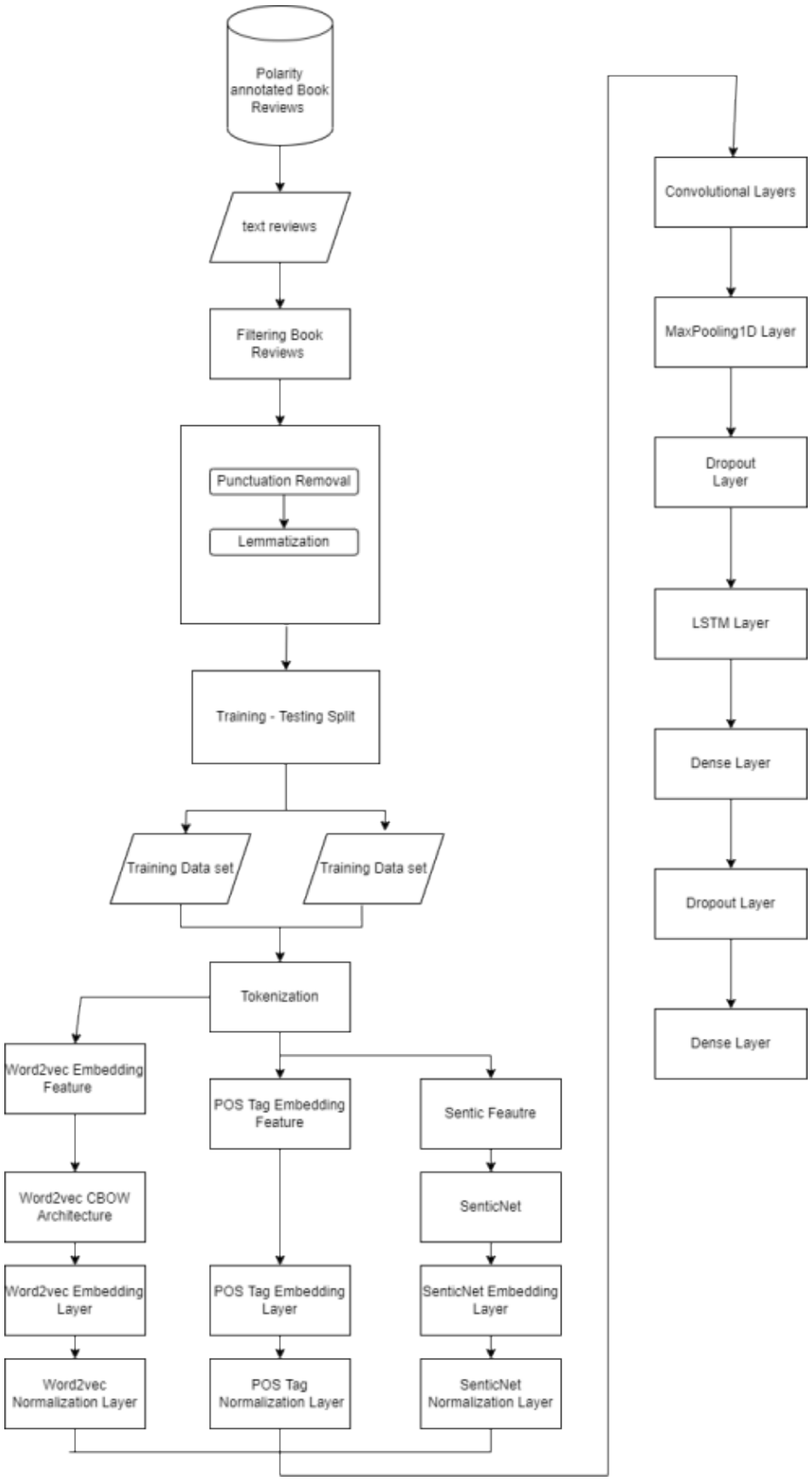

Figure 4: Proposed CNN-LSTM Sentiment Analysis Model Hybrid Conceptual Framework

	negative	positive	total
the	47542	52208	99750
and	26549	43091	69640
to	25222	31303	56525
of	19283	24169	43452
it	19803	16668	36471
is	12633	20369	33002
this	14378	15387	29765
in	12361	17228	29589
book	13189	14554	27743
that	13236	14237	27473

Figure 5: Top 10 words found in both negative and positive sentiment polarities of the text corpus

and adjectives within the datapoints. Afterwards, the researchers utilized bar graphs to be able to visualize the top 50 negative and positive words. As stop words are the most prominently used throughout the entire dataset, they were retained to maximize the efficiency of the LSTM portion of the hybrid model as remembering contexts is integral for its performance. In the top 50 negative tokens graph, stop words such as the, and, to, and it among others dominate the sentiment class as the most ubiquitous tokens. On the other hand, similar stop words such as the, and, to, and among others dominate the class of the positive token. From this, retaining such words was necessary for the structure of the reviews.

A log-log plot was employed to measure the zipf distribution across all tokens regardless of their polarity opr genre. This follows the Zipf law which asserts that given a list of a book's most common terms, the most common word will occur twice as often as the second most common, which will appear twice as often as the third most common, and so on. As seen in figure 6, similar tokens found in the top 50 words for positive and negative classes dominate the term frequency dataframe for the whole dataset. This suggests that such tokens play a significant role in the construction of most of the user reviews. After determining the most common words in the entire corpus, formulas for finding the positive rate, positive frequency pct, harmonic mean, and cumulative distribution function were employed to find out the words that bore great significance in each sentiment class. The formulas below showcase how the variables were calculated based on the term frequency dataframe.

$$\text{Positive Rate} = \frac{positive\ requency}{positive\ frequency\ +\ negative\ frequency}$$

Positive Rate Calculation

$$\text{Positive Frequency pct} = \frac{positive\ frequency}{\sum positive\ frequency}$$

Positive Rate Calculation

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

Harmonic Mean Formula

In the positive sentiment class, tokens such as sexy, heart, loved, hot, perfect, and others play a significant role in characterizing a positive polarity amongst the reviews with positive ratings. On the contrary, tokens such as boring, annoying, finish, unfortunately, and others generally characterize the negative polarity amongst the reviews with negative ratings. Modules and libraries such as Bokeh, Seaborn, and Matplotlob were then employed to visualize both positive cdf harmonic mean and negative cdf harmonic mean values. Figure 9 illustrates a curvilinear form wherein nodes from the upper left represents words that characterize the positive class, whilst the lower right displayed terms that characterize the negative

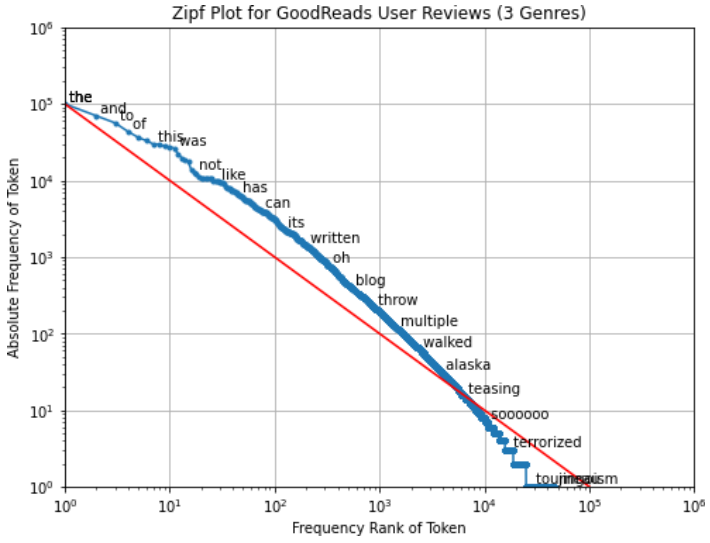

Figure 6: Zipf Plot for the user reviews

	negative	positive	total	positive_rate	positive_freq_pct	positive_rate_normcdf	positive_freq_pct_normcdf	pos_rate	pos_freq_pct	pos_hmean
sexy	99	1025	1124	0.911922	0.000915	0.790229	0.985684	0.911922	0.000915	0.001828
heart	178	1170	1348	0.867953	0.001045	0.757989	0.993887	0.867953	0.001045	0.002087
loved	535	2735	3270	0.836391	0.002442	0.733285	1.000000	0.836391	0.002442	0.004870
provided	73	597	670	0.891045	0.000533	0.775247	0.894836	0.891045	0.000533	0.001065
wait	148	780	928	0.840517	0.000696	0.736585	0.950815	0.840517	0.000696	0.001392
hot	177	810	987	0.820669	0.000723	0.720520	0.957138	0.820669	0.000723	0.001445
arc	119	635	754	0.842175	0.000567	0.737906	0.909185	0.842175	0.000567	0.001133
highly	81	533	614	0.868078	0.000476	0.758085	0.867086	0.868078	0.000476	0.000951
perfect	214	816	1030	0.792233	0.000729	0.696705	0.958321	0.792233	0.000729	0.001456
return	118	599	717	0.835425	0.000535	0.732509	0.895629	0.835425	0.000535	0.001069

Figure 7: Extracted Words that Characterize the Positive Sentiment Class

	negative	positive	total	positive_rate	positive_freq_pct	positive_rate_normcdf	positive_freq_pct_normcdf	pos_rate	pos_freq_pct	pos_hmean
boring	614	81	695	0.116547	0.000072	0.128185	0.549496	0.116547	0.000072	0.000145
didnt	3660	1345	5005	0.268731	0.001201	0.222614	0.998063	0.268731	0.001201	0.002391
annoying	473	77	550	0.140000	0.000069	0.140574	0.546032	0.140000	0.000069	0.000137
finish	531	160	691	0.231548	0.000143	0.196500	0.616815	0.231548	0.000143	0.000286
maybe	805	343	1148	0.298780	0.000306	0.245092	0.757183	0.298780	0.000306	0.000612
wasnt	1537	713	2250	0.316889	0.000637	0.259205	0.934005	0.316889	0.000637	0.001271
ok	436	116	552	0.210145	0.000104	0.182347	0.579619	0.210145	0.000104	0.000207
seemed	599	256	855	0.299415	0.000229	0.245580	0.693936	0.299415	0.000229	0.000457
unfortunately	436	129	565	0.228319	0.000115	0.194323	0.590699	0.228319	0.000115	0.000230
okay	555	229	784	0.292092	0.000204	0.239986	0.672924	0.292092	0.000204	0.000409

Figure 8: Extracted Words that Characterize the Negative Sentiment Class

Table 1: Overall Evaluation Report for the Models Constructed in the Study

MODEL	Accuracy	Precision	Recall	F1-Score	True Positive Rate	False Positive Rate
CNN + LSTM	77.20%	78%	77%	77%	74.53%	25.47%
W2V - CNN + LSTM	80.47%	81%	80%	80%	83.03%	16.97%
W2V + POS - CNN	81.07%	81%	81%	81%	79.23%	10.77%
W2V + POS - LSTM	84.27%	85%	84%	84%	87.46%	12.54%
W2V + POS - CNN + LSTM	86.07%	86%	86%	86%	87.21%	15.08%
W2V + POS + SENTICNET - CNN + LSTM	89.53%	89%	90%	90%	89.49%	10.04%

class. The saturated nodes in the middle of the scatterplot represents the stop words that are most prominent throughout the corpus. The nodes in the middle are numerous for these stop words are widely employed and appear ubiquitously in almost every user review in the 15,000 textual data points used by the researchers.

3.2 Model results overview

The outcomes indicate that accuracy improves when Word2Vec word embeddings and POS are applied to a CNN-LSTM model. The table also demonstrates that the CNN-LSTM model alone is

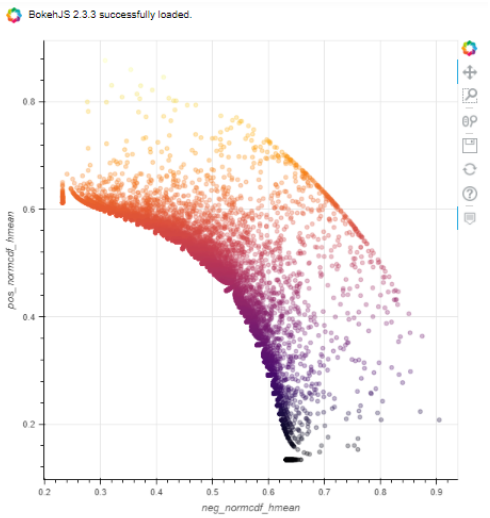

Figure 9: Bokeh Scatter Plot of Word Relationships with CDF Harmonic Mean as Metric for Both Positive and Negative Sentiment Classes

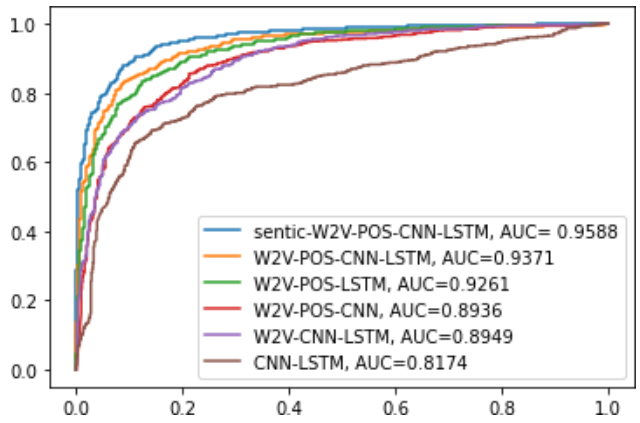

Figure 10: Overall ROC Curve Results for the Models Created in the Study

less accurate than the LSTM model with Word2Vec and POS and the LSTM model with Word2Vec and POS. This suggests that the POS improves an LSTM model's accuracy and potentially the CNN standalone as well. In addition to that, the F1-scores of the models also correspond with their accuracies – indicating that the CNN-LSTM model that combined word embedding and POS features outperform the CNN-LSTM model that solely employed word embeddings. To test the viability of utilizing Word2Vec in creating the word embeddings, the researchers incorporated the Word2Vec to the CNN-LSTM model to determine its findings first. Adding W2V, combining Skipgram and CBOW architectures [7], to the CNN-LSTM model resulted in the accuracy of 80.47%, precision of 81%, recall of 80%, and f1-score of 80%. Then to test the viability of combined W2V and POS features on a model, the researchers applied the features onto a CNN model and an LSTM model. When applied to a CNN model, it had an accuracy of 81.07%, and precision, recall, and f1-score of 81%. And when applied to an LSTM model, it had an accuracy of 84.27%, precision of 85%, and recall and f1-score of 84%. Then by adding W2V and POS to CNN-LSTM had an accuracy of 86.87%, precision of 87%, recall of 87%, and f1-score of 87%. Lastly, senticnet was added on the last model which resulted in an accuracy of 88.33%, precision of 88%, recall of 88%, and an f1-score of 88%, which is the highest accuracy out of all the models used, both created models of the researchers have further improved the accuracies of the proposed CNN-LSTM Hybrid model.

The ROC curves of the five sentiment analysis models that the researchers developed are shown in the picture above. By graphing the results of each model's True Positive Rate (TPR) and False Positive Rate calculations, it was graphed (FPR). In the ROC illustration, the AUC values of the models differ from each other where CNN-LSTM has the lowest percentage compared to others since it did not employ any features. However, when Word2vec and POS Tagging are applied the AUC curve increases by around 10% which is a significant increase. This model was further improved by 1% when all three (word2vec embedding, pos embedding, sentic embedding) were applied within the model which gathered around 95.88% on the AUC curve. It can be observed that applying the three features will yield better results for sentiment analysis.

4 CONCLUSION AND RECOMMENDATION

In terms of accuracy, the proposed hybrid CNN-LSTM models outperformed single CNN with Word2Vec + POS, single LSTM with Word2Vec + POS, and CNN-LSTM hybrid with Word2Vec on the filtered GoodReads Dataset for book reviews.As shown from the ROC curve, the overall ROC curves are seen, the models containing the POS and SenticNet Feature have the best performance based on the AUC compared to the other models, the CNN-LSTM(w/ Word2Vec + POS) having the second highest performance score of 0.9371 and CNN-LSTM (w/ Word2Vec + POS + SenticNet) having the best performance score of 0.9588. The Proposed Hybrid CNN-LSTM with Word2Vec + POS model achieved 86.07% accuracy, 86% precision, 86% recall, and 86% f1-score. The proposed Hybrid CNN-LSTM with Word2 + POS + SenticNet achieved 89.53% accuracy, 90% precision, 90% recall, and 90% f1-score, which are better compared to traditional machine learning and deep learning models. The proposed CNN-LSTM Hybrid model with W2V, POS, and SenticeNet features achieved its aim to prove that with the addition of features, the performance can be further enhanced.

To further enhance the study, the researchers will recommend for future researchers to increase the dataset size as the model will be enabled to learn more words and vocabulary as the dataset grows. The researchers also suggest that with the use of Bi-LSTM the study will improve further and even more if it is combined with CNN. This can be seen in a similar study by Mikram [3] where it employed a CNN-Bi-LSTM approach for sentiment analysis. The use of senticnet could also be improved on by future researchers as not all of its extracted features were used on the model which may or may not impact the performance of the model.

ACKNOWLEDGMENTS

We thank Professor Joel De Goma (Mapua University, Makati) for evaluating the thesis paper and assistance on form requirements and submissions. We thank Dean Ariel Kelly D. Balan for comments regarding thesis revisions. We thank Professor John Paul Q. Tomas (Mapua University, Makati) for answering our inquiries and for providing insightful remarks regarding thesis revisions. We thank Professor Mary Jane Samonte (Mapua University, Makati) for comments on thesis revisions and insights regarding deep neural networks and proper data set structuring. This work was supported by Mapua University – Makati, School of Information Technology.

REFERENCES

[1] Yadav, A., & Vishwakarma, D. K. (2019). Sentiment analysis using deep learning architectures: a review. Artificial Intelligence Review, 53(6), 4335–4385. doi:10.1007/s10462-019-09794-5

[2] Liao, S., Wang, J., Yu, R., Sato, K., & Cheng, Z. (2017). *CNN for situations understanding based on sentiment analysis of twitter data. Procedia Computer Science, 111, 376–381.* doi: 10.1016/j.procs.2017.06.037

[3] Goularas, D., & Kamis, S. (2019). Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data. 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML). doi:10.1109/deep-ml.2019.00011

[4] Rehman, A. U., Malik, A. K., Raza, B., & Ali, W. (2019). A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. Multimedia Tools and Applications. doi:10.1007/s11042-019-07788-7

[5] Rvia, M. (24 April 2021). Word Embeddings: CBOW vs Skip-Gram. https://www.baeldung.com/cs/word-embeddings-cbow-vs-skip-gram

[6] He, W., Li, J., Tang, Z., Wu, B., Luan, H., Chen, C., & Liang, H. (2020). A Novel Hybrid CNN-LSTM Scheme for Nitrogen Oxide Emission Prediction in FCC Unit. doi.org/10.1155/2020/8071810

[7] Rhanoui, M., Mikram, M., Yousfi, S., & Barzali, S. (2019). A CNN-BiLSTM Model for Document-Level Sentiment Analysis. Machine Learning and Knowledge Extraction, 1(3), 832-847. doi:10.3390/make1030048

[8] Wan M., McAuley J., "Item Recommendation on Monotonic Behavior Chains", in RecSys'18. https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home?authuser$=$0

[9] Wan, M., Misra, R., Nakashole, N, McAuley, J. (2019). Goodreads Dataset. Retrieved from: https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home?authuser$=$0.

[10] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011