Comparative Analysis of Convolutional Neural Networks and Recurrent Neural Networks for Twitter Sentiment Analysis

Christian Jericho Katigbak
dept. name of organization (of Affiliation)
name of organization (of Affiliation)
Oamaru, New Zealand
cjkatigbak14@gmail.com

Laurence EJ Manjares
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
Calamba, Laguna
laurencemanjares109@gmail.com

Georges grad Neilsen San Juan
dept. name of organization (of Affiliation)
name of organization (of Affiliation)
Calamba, Laguna
georgessanjuan@gmail.com

Abstract*— Social media platforms like Twitter have become invaluable sources of real-time public opinion and sentiment. Analyzing sentiment from these platforms is challenging due to the noisy and dynamic nature of the data. In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have shown promising results in sentiment analysis tasks. This paper compares the two algorithms on a publicly available Twitter sentiment dataset from Hugging Face. The research findings indicate that CNNs exhibit superior overall performance, suggesting their efficacy in sentiment analysis tasks on Twitter. The study underscores the importance of preprocessing techniques and the impact of resampling methods on model robustness. These insights offer valuable guidance for researchers and practitioners aiming to leverage deep learning models for sentiment analysis in social media contexts, emphasizing the significance of meticulous preprocessing and resampling strategies to optimize model efficacy. Future research avenues include integrating additional features and advanced preprocessing techniques for further refinement of model accuracy across diverse social media platforms and languages.

Keywords—Sentiment Analysis, Twitter, CNN, RNN, Deep Learning

I. INTRODUCTION

In the digital age, social media platforms have become ubiquitous channels for expressing opinions, emotions, and sentiments on a wide range of topics. Among these platforms, Twitter stands out as a prominent source of real-time information and a rich reservoir of public sentiment. The succinct nature of tweets, limited to 280 characters per post, coupled with the platform's vast user base, make Twitter an invaluable resource for sentiment analysis—a task vital for understanding public opinion, monitoring brand perception, and gauging societal trends.

Sentiment analysis, also known as opinion mining, involves the automated extraction of sentiment or subjective information from textual data. With the advent of deep learning techniques, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), sentiment analysis has witnessed significant advancements in accuracy and efficiency. CNNs excel at capturing spatial patterns in data, making them well-suited for tasks like image recognition and text classification. On the other hand, RNNs are adept at modeling sequential dependencies, making them ideal for processing time-series data such as language.

This paper embarks on a comparative exploration of CNNs and RNNs for Twitter sentiment analysis. Leveraging a publicly available sentiment dataset sourced from Twitter, we delve into the performance, interpretability, and computational aspects of both architectures. By conducting a systematic evaluation, we aim to elucidate the strengths and weaknesses of CNNs and RNNs in the context of sentiment analysis tasks on Twitter data.

The introduction of this paper delineates the significance of sentiment analysis in the realm of social media analytics and elucidates the rationale behind the comparative analysis of CNNs and RNNs. Subsequent sections will delve into the methodologies employed, the experimental setup, results, and discussions, culminating in comprehensive insights into the efficacy of deep learning models for Twitter sentiment analysis. Through this study, we aspire to provide valuable guidance to researchers and practitioners seeking to leverage state-of-the-art techniques for sentiment analysis in the dynamic landscape of social media.

II. RELATED WORK

In [1] The paper presents a deep learning system tailored for sentiment analysis of tweets, showcasing a novel approach to initializing parameter weights within convolutional neural networks. The proposed model innovatively employs an unsupervised neural language model to train initial word embeddings, subsequently fine-tuning them through a deep learning framework on a distant supervised corpus. Notably, this technique eliminates the need for additional features injection, enhancing the model's accuracy. The study mentions minimal preprocessing of the tweets before training the word embeddings. This includes tokenizing the tweets and normalizing URLs and author IDs. The study also evaluates the system using the supervised training data from the Semeval-2015 Twitter Sentiment Analysis campaign, where it exhibits remarkable performance. Notably, the results position the model among the top contenders in both phrase-level and message-

level subtasks, as evidenced by its potential ranking in the first two positions. Moreover, the comparison with official rankings underscores the robustness of the proposed approach, outperforming competing systems across various test sets.

In [2], various configurations of deep learning methods based on Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), specifically Long Short-Term Memory (LSTM) networks, are explored for sentiment analysis in Twitter data. While CNNs excel in tasks like image processing, RNNs, particularly LSTMs, have demonstrated success in natural language processing tasks. The preprocessing steps mentioned in the study include converting all letters to lowercase, removing special characters and emoticons, as well as tagging URLs. These tasks were carried out to enhance the system's performance during training for sentiment analysis on Twitter data. The evaluation conducted in this work reveals that while these configurations yield slightly inferior but comparable results to state-of-the-art methods, they provide valuable insights into the strengths and limitations of CNNs and LSTMs in sentiment analysis. Interestingly, the combination of CNNs and LSTMs outperforms their individual usage, attributed to CNNs' effective dimensionality reduction and LSTMs' ability to preserve word dependencies. Moreover, employing multiple CNNs and LSTMs enhances system performance. Discrepancies in accuracy across different datasets underscore the importance of dataset quality in system performance. Thus, while experimenting with various configurations is valuable, dedicating efforts to crafting robust training sets appears to be more advantageous.

In [3], authors introduce a novel sentiment analysis model specifically designed for social media, notably Twitter, where conventional approaches struggle due to tweet characteristics and user behavior. The authors propose a Convolutional Neural Network (CNN) model that incorporates both textual content and user behavioral signals, surpassing baseline methods like Naive Bayes and Support Vector Machines in accuracy, recall, precision, and F1 scores. This CNN-based approach demonstrates resilience against issues like imbalanced datasets and reduces the reliance on large-scale training data. Preprocessing involves several techniques to prepare data for analysis or modeling. These include data cleaning, which addresses errors and missing values; data normalization, which scales numerical features to similar ranges; and feature selection, which identifies and uses the most relevant features to reduce dimensionality. Additionally, encoding categorical variables converts them into numerical formats, while handling imbalanced classes involves methods like oversampling, under sampling, or synthetic data generation. Feature engineering creates new features from existing data to enhance model performance, and data transformation applies methods like log or square root transformations to make the data more suitable for modeling algorithms. The paper's findings suggest promising avenues for future research, including exploring the model's features for non-binary sentiment tasks and investigating alternative neural network architectures like Recurrent Neural Networks (RNNs) for further improvements in sentiment analysis.

However, after careful observation and review of existing studies, the researchers found that only a few have focused on preprocessing techniques to address common issues in Twitter NLP, such as hashtags, repeated vowels, contractions, and emojis/emoticons. The researchers also found out that few researchers focused on the effect of resampling methods with the proper preprocessing techniques. Thus, we present the significance of text preprocessing in improving Twitter text analysis and provide experimental results showing how these preprocessing and resampling methods enhance the performance of sentiment classification.

III. Methodology

A. Data Collection

The researchers obtained the necessary data for their project from their professor during their course. They utilized the Twitter Financial News Sentiment data sourced from huggingface.io, which was provided by their professor.

Table I shows the class distribution of each dataset. It can be seen that the majority class of both datasets are neutral with negative and positive sentiments making up not more than half of the distribution. The datasets are skewed to the neutral sentiments, and researchers should keep this in mind while doing the experiment.

TABLE I. Dataset Class Distribution

Dataset	Class	Distribution
Training	Negative	1442
	Neutral	6178
	Positive	1923
Validation	Negative	347
	Neutral	1566
	Positive	475

B. Data Preprocessing

Initially, incorrectly spelled terms were rectified using a predetermined lexicon. Subsequently, all content was transformed to lowercase to ensure consistency. Uniform Resource Locators (URLs) and number signs were eliminated using regular expressions, while distinct symbols like tabs, carriage returns, and quotes were replaced to maintain standardized text formatting. Punctuation marks, excluding apostrophes, were deleted to emphasize the fundamental textual content. Repetitive vowels were condensed to a maximum of two instances to avoid exaggerated expressions. Emoticons were changed into emotional descriptors using an alternative established lexicon, enabling a more methodical depiction of feelings. Numerals and foreign characters were excluded to prevent inconsequential data from impacting the assessment.

Following this, a Term Frequency-Inverse Document Frequency (TF-IDF) weighting approach was implemented to convert the textual information into numerical attributes. This phase helped illustrate the significance of expressions within

the records. Word stemming was carried out using the Porter Stemmer to reduce terms to their base forms, aiding in diminishing complexity and enhancing model abstraction. Subsequently, segmentation was performed to divide the text into separate terms, simplifying the model's analysis.

To identify the effect of category imbalance, a method called undersampling using the RandomUnderSampler, oversampling using RandomOverSampler and no resampling method was used. This is to see the how data imbalances affect the training and validation that data from real life Twitter sentiments suffers. This stage involved fitting and applying the said methods to the sequences of text data, generating a balanced dataset for model training. Label encoding was used to transform group classifications into binary form, if required.

C. *Modelling*

The textual data underwent initial tokenization and padding to ensure uniformity for both the RNN and CNN models. Sequential neural network architectures were built using Keras, with the RNN model featuring an LSTM layer with dropout regularization followed by an Embedding layer. Dense layers with ReLU activation and dropout were added as supplements. Conversely, the CNN model incorporated layers such as Embedding, GlobalMaxPooling1D, Dense, Conv1D, and MaxPooling1D. Both models were optimized using the Adam optimizer and categorical cross-entropy loss. Following training on the training set and evaluation using the validation data, the resulting accuracy provided insights into the suitability of each model architecture for the given task.

IV. RESULTS AND DISCUSSION

A. *Initial Results*

A relatively equal amount of data per class is crucial both in the training and validation of the model. A technique called oversampling and undersampling can be utilized for the class imbalance. An initial result was obtained for a CNN and RNN model with the same epochs and batch sizes using oversampling and undersampling. The results were as follows:

	precision	recall	f1-score	support
negative	0.55	0.68	0.61	347
neutral	0.87	0.85	0.86	1566
positive	0.72	0.64	0.68	475
accuracy			0.79	2388
macro avg	0.71	0.73	0.72	2388
weighted avg	0.79	0.79	0.79	2388

Fig. 1. CNN without over- and undersampling validation results.

	precision	recall	f1-score	support
negative	0.43	0.79	0.56	347
neutral	0.91	0.69	0.79	1566
positive	0.61	0.72	0.66	475
accuracy			0.71	2388
macro avg	0.65	0.73	0.67	2388
weighted avg	0.78	0.71	0.73	2388

Fig. 2. CNN with undersampling validation results.

	precision	recall	f1-score	support
negative	0.54	0.66	0.60	347
neutral	0.87	0.83	0.85	1566
positive	0.67	0.67	0.67	475
accuracy			0.77	2388
macro avg	0.69	0.72	0.71	2388
weighted avg	0.78	0.77	0.78	2388

Fig. 3. CNN with oversampling validation results.

	precision	recall	f1-score	support
negative	0.62	0.57	0.59	347
neutral	0.84	0.87	0.86	1566
positive	0.70	0.67	0.68	475
accuracy			0.79	2388
macro avg	0.72	0.70	0.71	2388
weighted avg	0.78	0.79	0.78	2388

Fig. 4. RNN without over- and undersampling validation results.

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	347
neutral	0.00	0.00	0.00	1566
positive	0.20	1.00	0.33	475
accuracy			0.20	2388
macro avg	0.07	0.33	0.11	2388
weighted avg	0.04	0.20	0.07	2388

Fig. 5. RNN with undersampling validation results.

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	347
neutral	0.66	1.00	0.79	1566
positive	0.00	0.00	0.00	475
accuracy			0.66	2388
macro avg	0.22	0.33	0.26	2388
weighted avg	0.43	0.66	0.52	2388

Fig. 6. RNN with oversampling validation results.

Based on the figures above, both CNN and RNN models that did not employ oversampling and undersampling approaches performed noticeably better than those that did. The RNN models diverge significantly from one another, even if the CNN models' accuracy was reasonably similar. In the RNN with undersampling model, only the positive data were correctly classified; in the RNN with oversampling model, only the neutral data were correctly classified.

Using these results, the CNN and RNN models without oversampling and undersampling techniques approach were utilized in the next step.

B. *Hyperparameter Tuning*

The researchers observed that batch size and epoch significantly impact the performance of the CNN model. To identify the optimal batch size and epoch, the researchers evaluated batch sizes of 32, 64, and 128 with epochs of 32, 64, and 128. Found in the Table 2 is the performance analysis for different configuration of the hyperparameters for CNN.

TABLE II. CNN HYPERPARAMETER TUNING

Model	Batch Size	Epoch	Validation Set			
			Prec.	*Rec.*	*F1*	*Acc.*
CNN	32	15	0.81	0.80	0.80	0.7994
		25	0.80	0.81	0.80	**0.8061**
		35	0.79	0.79	0.79	0.7866
	64	15	0.79	0.79	0.79	0.7877
		25	0.80	0.80	0.80	0.7994
		35	0.79	0.79	0.79	0.7898
	128	15	0.79	0.79	0.79	0.7881
		25	0.79	0.79	0.79	0.7881
		35	0.78	0.78	0.77	0.7763

For a batch size of 32, the model achieved its highest accuracy of 0.8061 at 25 epochs, with balanced precision (0.80) and recall (0.81). At 15 epochs, the performance was slightly lower, with an accuracy of 0.7994, precision of 0.81, and recall of 0.80. Extending the training to 35 epochs resulted in a minor decline, with an accuracy of 0.7866 and both precision and recall at 0.79.

With a batch size of 64, the model demonstrated consistent performance, particularly at 25 epochs, achieving an accuracy of 0.7994 and an F1 score of 0.80. At 15 epochs, the results were similar to those at 35 epochs, with an accuracy around 0.7877 and 0.7898, respectively.

For a batch size of 128, the model's performance was stable but lower compared to smaller batch sizes. At both 15 and 25 epochs, the accuracy was 0.7881 with F1 scores around 0.79. However, at 35 epochs, the performance further decreased, showing the lowest accuracy of 0.7763 and an F1 score of 0.77.

The findings highlight that the batch size and the number of epochs was critical hyperparameters for optimizing the CNN model's performance. The best results were observed with a batch size of 32 and 25 epochs, indicating the importance of fine-tuning these parameters to achieve optimal accuracy and balanced precision and recall in text classification tasks. Found in the Table 3 is the performance analysis for different configuration of the hyperparameters for RNN.

TABLE III. RNN HYPERPARAMETER TUNING

Model	Batch Size	Epoch	Validation Set			
			Prec.	*Rec.*	*F1*	*Acc.*
RNN	32	15	0.79	0.79	0.79	0.7915
		25	0.79	0.79	0.79	0.7927
		35	0.79	0.79	0.79	**0.7940**
	64	15	0.79	0.79	0.79	0.7923
		25	0.78	0.77	0.77	0.7705
		35	0.77	0.77	0.77	0.7680
	128	15	0.78	0.77	0.78	0.7722
		25	0.77	0.77	0.77	0.7697
		35	0.76	0.76	0.77	0.7539

For the RNN model with a batch size of 32, the performance is stable across 15, 25, and 35 epochs, with precision, recall, and F1 scores all consistently at 0.79. Accuracy slightly increases from 0.7915 at 15 epochs to 0.7940 at 35 epochs, indicating that additional epochs provide a marginal improvement in accuracy while maintaining balanced performance across other metrics.

When the batch size is increased to 64, at 15 epochs, the model's precision remains at 0.79, but recall and F1 score drop to 0.78 and 0.77, respectively, with an accuracy of 0.7705. This suggests a slight decline in the model's ability to generalize with a larger batch size. At 25 epochs, precision and recall decrease to 0.78 and 0.77, with no change in accuracy, indicating diminishing returns from additional training. By 35 epochs, the precision, recall, and F1 score all stabilize at 0.77, but accuracy slightly decreases to 0.7680, suggesting potential overfitting or the model's learning capacity being reached.

For a batch size of 128, the model at 15 epochs shows similar performance to the batch size of 64, with precision at 0.78, recall at 0.77, and an F1 score of 0.78, with accuracy at 0.7722. However, increasing to 25 epochs results in decreases across all metrics, with precision and recall at 0.77 and accuracy at 0.7672. At 35 epochs, the model performs the worst among all batch sizes and epoch combinations, with precision, recall, and F1 score at 0.76 and 0.77, and accuracy dropping to 0.7539, indicating significant overfitting.

Overall, while increasing batch size and epochs can initially improve model performance, there is a point of diminishing returns. For the RNN model in this case, a batch size of 32 with around 25 epochs appears to strike the best balance between performance and generalization. Found in Table 4 and Table 5 are the performance analysis for the highest accuracy configuration for CNN and RNN.

TABLE IV. CNN PERFORMANCE

Model		Prec.	Rec.	F1	Supp.
CNN	negative	0.70	0.54	0.61	347
	neutral	0.85	0.90	0.88	1566
	positive	0.70	0.69	0.70	475
	accuracy			0.81	2388
	macro avg	0.75	0.71	0.73	2388
	weighted avg	0.80	0.81	0.80	2388

TABLE V. RNN PERFORMANCE

Model		Prec.	Rec.	F1	Supp.
RNN	negative	0.59	0.64	0.61	347
	neutral	0.86	0.86	0.86	1566
	positive	0.70	0.64	0.67	475
	accuracy			0.79	2388
	macro avg	0.71	0.71	0.71	2388
	weighted avg	0.79	0.79	0.79	2388

C. Insights

The following insights were gained during the research, highlighting key factors that influence model performance and

efficiency. These findings provide practical guidance for optimizing various aspects of the machine learning pipeline.

Data preprocessing played a significant role in the training and validation of the model. Remove too much, and the data may become worthless as the context inside the text is lost. Researchers must properly process the data so that the context of the language and the essence of the sentiment will remain.

The researchers explored various methods to address the issue of imbalanced datasets, specifically by applying undersampling, oversampling, and no resampling technique. Interestingly, they found that using no sampling consistently resulted in the highest performance for both CNN and RNN. On the other hand, undersampling, while helpful in balancing the dataset, led to a slight drop in performance compared to oversampling. Lastly, when over resampling was applied, the performance of the models significantly decreased.

Lowering the learning rate from the default 0.001 does not significantly affect performance, except when using overly exaggerated values like 0.0001, where performance significantly drops. It is crucial to find an optimal learning rate that allows the model to converge efficiently without overshooting or stagnating.

Padding, although not contributing much, still helps in creating a more robust system. Padding ensures that all sequences are of uniform length, which is particularly important for batch processing in neural networks, helping maintain consistency and stability in model training.

Adding layers increases performance but should be balanced. Using more layers than necessary can increase training time without any benefits in performance. It is essential to find the right architecture that provides sufficient depth to capture the complexity of the data while avoiding overfitting and excessive computational costs.

V. CONCLUSION

In this study, we compared two deep learning algorithms for classifying sentiments in twitter posts. The comparative analysis is between CNN and RNN for sentiment analysis on Twitter data delineated distinct strengths and weaknesses for each model. CNN exhibited superior in terms of overall performance. The implication here could suggest that CNNs may be better suited for analyzing sentiment in Twitter posts compared to RNNs. This is noteworthy as RNNs are specifically designed for sequential data like text. The observation that CNNs outperform RNNs for Twitter analysis suggests several implications. Tweets' short, noisy nature may favor CNNs, which excel at capturing local patterns in data. CNNs' ability to learn hierarchical representations may help in understanding nested structures within Twitter conversations. They autonomously extract features, such as sentiment-indicative phrases, without explicit engineering, and their computational efficiency facilitates handling the vast Twitter dataset. The study emphasized the pivotal role of preprocessing techniques, in augmenting sentiment classification model performance. This implies that a considerable portion of model performance improvement can be attributed to feature extraction and transformation methods applied during preprocessing.

Pertaining to this subject, the effect of resampling methods like oversampling and undersampling were identified. The researchers have found out that no resampling method makes the model robust and representative for real world cases where data imbalances is the normal occurrence. These findings offer valuable insights for researchers and practitioners intending to harness deep learning models for sentiment analysis in social media contexts, stressing the significance of meticulous preprocessing and resampling strategies to optimize model efficacy.

Future avenues of inquiry may involve integrating additional features and advanced preprocessing techniques to further refine model accuracy, as well as broadening the scope to encompass other social media platforms and languages for a more comprehensive understanding of sentiment analysis across diverse contexts. Ultimately, the study underscores the efficacy of CNNs and RNNs in sentiment analysis tasks, furnishing a robust framework for future research endeavors and practical applications within the dynamic realm of social media analytics.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Sir Manolito V. Octaviano Jr., our esteemed course adviser, for his invaluable guidance, support, and encouragement throughout the duration of this study. His expertise and insights have greatly contributed to the completion of this research project. We are sincerely thankful for his mentorship and dedication to our academic and professional development.

REFERENCES

[1] Severyn, A., & Moschitti, A. (2015). Twitter Sentiment Analysis with Deep Convolutional Neural Networks. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15. doi:10.1145/2766462.2767830.

[2] 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.

[3] M Alharbi, A. S., & Doncker, E. (2018). Twitter Sentiment Analysis with a Deep Neural Network: An Enhanced Approach using User Behavioral Information. Cognitive Systems Research. doi:10.1016/j.cogsys.2018.10.001.