Green Intelligent Systems and Applications

Research Article
Volume 4(2), 2024, 66–79
https://doi.org/10.53623/gisa.v4i2.491

A Sentiment Analysis of Hate Speech in Philippine Election-Related Posts Using BERT Combined with Convolutional Neural Networks and Model Variations Incorporating Hashtags and ALL-CAPS

Micah Collette O. Mendoza, Wayne Gabriel S. Nadurata, Mark Gabriel E. Ortiz, Joshua Mari L. Padlan, Charmaine S. Ponay*

Department of Computer Science, College of Information and Computing Sciences, University of Santo Tomas, Manila, Philippines

*Correspondence: csponay@ust.edu.ph

SUBMITTED: 5 September 2024; REVISED: 1 October 2024; ACCEPTED: 5 October 2024

ABSTRACT: As the number of people who use X continually increases, the same thing is true for hate speech. A pressing need exists for automatic detection of posts that promote hate speech. The datasets gathered and validated from the base study were used to categorize posts as either hate or non-hate and classify them as positive, negative, or neutral using Conventional Neural Networks. The partitioning of the labeled data into training and testing sets adhered to a ratio scheme: 70%-30%, 80%-20%, and 90%-10%. The model of this study, BERT-CNN, had an overall better performance than the base study, fastText CNN. Notably, among the three splits, the BERT-CNN model for binary classification without the features of Hashtags and ALL-CAPS with the 90:10 split achieved the best performance with an accuracy of 93.55%, precision of 93.59%, and F1-score of 93.55%. For multi-label classification, the BERT-CNN model demonstrated its optimal performance when incorporating hashtags, specifically with the 90:10 split, achieving an accuracy of 69.14%, precision of 68.44%, recall of 68.40%, and an F1-score of 67.41%. The innovative use of BERT word embeddings paired with CNN proved to excel in classifying Philippine election-related posts as hate or non-hate.

KEYWORDS: X; hate speech; machine learning; deep learning; natural language processing

1. Introduction

The term "political hate speech" was commonly referred to as an act of marginalizing or dehumanizing groups or individuals for their political ideology, beliefs, or affiliation [1]. This involved using coded phrasing with underlying discriminatory intent, persuasion to boycott, "cancel," and even the dissemination of false or misleading information. Anyone could engage in this act at any time, with more frequent occurrences during elections or other political campaigns. Such hate speeches were often shared on various platforms, such as social media, which were being exploited to spread discrimination and misinformation [2].

Relevantly, X, formerly known as Twitter, served as a significant platform for expressing both filtered and unfiltered thoughts and ideas, including facts and opinions. In line with this,

X became the choice platform for netizens to voice political opinions, criticisms, and perceptions [3, 4]. Posts, formerly known as tweets, that included hate speech and offensive content aimed at opposing political factions or parties were frequently visible and could lead to negative consequences, such as threats or harassment, potentially generating fear among those targeted [5–7]. Posts with hate speech surged to nearly all-time highs during the 2022 Philippine Election period, driven by the election's polarizing nature, which caused deep divisions. During this time, supporters of particular candidates often berated, insulted, and belittled their opponents, or worse.

Given the sheer volume of data, automatically detecting the extent of hate speech became a challenge [8]. Computational models were developed to automate this process, such as a study from the University of Santo Tomas [9, 10], which detected Filipino election-related posts using fastText CNN. However, despite achieving a high accuracy of approximately 83%, there remained room for improvement. As such, researchers aimed to address the following questions related to the problem: Could the combination of BERT and the CNN model improve the performance of the existing model proposed by the base study in detecting hate speech in Philippine election-related posts? To what extent could the model reach optimal performance? How could this be further improved? What insights could be derived from the analyzed text data using BERT embeddings?

2. Related Works

2.1.Sentiment analysis.

The field of study known as sentiment analysis, also referred to as opinion mining, examines people's opinions, sentiments, assessments, attitudes, and emotions about entities and their attributes as expressed in written text [11–14]. These entities can be goods, services, businesses, people, events, issues, or topics [15]. The scope of the problem represented by this field is vast [6]. In another study, [7] proposed an analysis of the BERT word embeddings process with the recommended hashtag feature using neural networks. The prediction results showed that hashtags, particularly hashtag clustering, were beneficial for predictions and identifying semantic similarity. It was also noted that while hashtags used in BERT word embeddings and other approaches, such as Emhash[16, 17], improved output performance, the variety in their usage could also hinder predictions [18]. This makes hashtags both a strength and a weakness in NLP models [19]. In the study conducted by [7], an analysis was carried out on special orthographic characteristics present on social media platforms such as Twitter or X [20, 21]. This included the specific use of capitalization in text. Results showed that capitalization on Twitter or X followed a pattern termed meaningful capitalization, where capitalization is used with intent and expressive value, rather than merely for convenience, such as abbreviations. The study found that meaningful capitalization significantly impacted sentiment analysis and functioned as a conversational cue to clarify underlying meaning in text-based communication [22].

2.2.Convolutional neural networks (CNN).

A Convolutional Neural Network (CNN) was utilized in sentiment analysis on English posts related to the "Turkey Crisis 2018" topic, as conducted in the study by [8, 18]. The sentiment analysis process began with data retrieval, followed by classification using TextBlob to categorize posts as positive, negative, or neutral. After training and evaluation, the CNN

classifier model achieved an accuracy rate of 89%. During the testing phase with test data, the model reached an accuracy rate of 88%. These results were compared to those of the Naïve Bayes classifier model, which had an accuracy rate of 78%. It was concluded that the CNN model, utilizing a deep learning algorithm, performed better in sentiment analysis than the NBC model.

2.3.Bidirectional encoder representations from transformers (BERT).

The widespread use of social media has led to an abundance of user-generated data, which can be analyzed to determine emotions and opinions [10, 18]. Profanities can also be used in different contexts—either aggressively or non-abusively, depending on the situation [10]. Several studies have demonstrated the effectiveness of BERT in various NLP tasks [23–25]. However, the lack of publicly available Filipino post datasets, particularly for fire-related reports on social media, has hindered the development of classification models for Filipino posts. Only a few insights have emerged, such as the discovery that the BERT model can effectively detect and censor Tagalog profanity in text media content [12]. To address this gap, [12] conducted a study that aimed to design and implement a system for classifying Filipino posts using different pre-trained BERT models. Using a dataset of 2,081 fire-related posts, the authors created a model to organize Filipino posts and compared the accuracy of different fine-tuned BERT models. The results indicated a significant difference in accuracy among the pre-trained BERT models. The BERT Base Uncased WWM model performed the best, achieving a test accuracy of 87.50% and a training loss of 0.06, while the BERT Base Cased WWM model was the least accurate, with a test accuracy of 76.34% and a training loss of 0.2. These results suggest that the BERT Base Uncased WWM model is reliable for classifying fire-related posts in Filipino, though the model's accuracy may vary depending on dataset size.

2.4.BERT-CNN.

Studies on NLP have also shown that combinations of BERT and CNN models are more effective than using either one alone [14, 23, 26–28]. A study by [14] emphasized the importance of sentiment analysis in improving product quality and influencing consumer purchasing decisions. However, the accuracy of existing sentiment analysis models needed improvement. Therefore, the authors proposed a BERT-CNN model to enhance sentiment analysis accuracy in commodity reviews [23]. The results were compared to the Logistic Regression (LogReg) model used by [16]. The first CNN model used random word vectors and showed significant precision improvements over the LogReg model, though it had lower recall. The second model employed word2vec embeddings, which improved recall by 7.3% compared to the random vector model, achieving an F-score of 78.29% [29]. The third and fourth models incorporated character n-grams alongside word embeddings. The third model used only character n-grams as feature embeddings, while the fourth model combined word2vec embeddings with character n-grams. The fourth model achieved the best precision, though the word2vec model without character n-grams had the best overall performance, with precision, recall, and F-score values of 85.66%, 72.14%, and 78.29%, respectively. The CNN models outperformed the LogReg model in terms of precision and F1 score, while the LogReg model had better recall. In another study by [16], the authors combined pre-trained BERT and CNN models for text analysis. The study underscored the importance of pre-trained language models for downstream tasks such as offensive language detection. The results showed that combining BERT with CNN outperformed using the BERT model alone.

3. Methodology

3.1.Data gathering and preprocessing.

The input data for the system architecture were extracted from a labeled dataset used for binary and multi-label classification in the base study [4, 2, 10]. This dataset contained election-related posts gathered from the X API during the 2022 Philippine elections. The initial dataset comprised 20,000 tweets from various Twitter accounts located in the Philippines, with posts referencing the presidential candidates for the 2022 Philippine National Election, dated from October 8, 2021, to May 7, 2022. The dataset was manually labeled according to a set of criteria formulated based on the content of the tweets, focusing on sentiment (positive, negative, or neutral) and hate speech classification (hate and non-hate). An instructor from the University's Department of Political Science validated both the annotation criteria and the labeled dataset.

This validated dataset served as a benchmark for comparing the four datasets generated using the BERT-CNN models: (A) BERT-CNN, (B) BERT-CNN with Hashtags, (C) BERT-CNN with ALL-CAPS, and (D) BERT-CNN with Hashtags and ALL-CAPS. After extracting the data, a preprocessing phase involved cleaning the data to ensure accuracy and completeness. This process included removing account mentions, URLs, special characters (e.g., emojis, diacritics, and numbers), while retaining both English and Filipino stop words, unlike the base study. Hashtags and ALL-CAPS were selectively included in different datasets to test their impact on sentiment classification. For hashtag processing, they were excluded from the base BERT-CNN and BERT-CNN with ALL-CAPS models but retained in models that utilized hashtags. In the case of ALL-CAPS processing, although most text was normalized to lowercase, ALL-CAPS formatting was preserved in models that incorporated it. Finally, the textual data was tokenized using a BERT tokenizer, preparing it for subsequent vectorization and model training.

3.2.Hypotheses.

The primary objective of this study was to implement a model capable of detecting and identifying hate speech in Philippine election-related posts using the BERT-CNN model. The study aimed to enhance the performance of the fastText CNN model used in the previous study [2]. The researchers formulated the following hypotheses:

H_0: There is no significant difference in performance between the BERT-CNN model and the fastText CNN model in classifying Philippine election-related posts as either hate or non-hate.

H_1: There is a significant improvement in performance when using the BERT-CNN model compared to the fast.

Text CNN model for classifying Philippine election-related posts as either hate or non-hate. The following assumptions were assumed true in this study: (1) the labeled data used in the sentiment analysis was correct and accurate, (2) The dataset contained sufficient data for the system to output correct predictions, (3) The size input for training and testing of data was sufficient to form an accurate analysis of the BERT-CNN model's performance.

3.3.Training and testing.

After tokenization, the preprocessed data were divided into training and testing sets. During the training phase, the BERT-CNN model was trained to accurately classify posts as either hate

or non-hate and to assess their sentiment (positive, neutral, or negative). Adjustments to the training set's parameters were made to optimize the model's ability to detect underlying patterns and key correlations within the data. In the testing phase, the model was evaluated using new, unseen data to assess its performance based on what it had learned during training. The data was partitioned into training and testing sets using different ratios (70%-30%, 80%-20%, and 90%-10%) to evaluate the model's generalization capabilities across varying proportions of training and testing data.

3.4.System Architecture

The system architecture designed for this study illustrated the process flow, transforming the data into binary- or multi-labeled posts related to the 2022 Philippine presidential elections. Each phase of the architecture was designed to address the study's objectives. The architecture was divided into five modules, namely, the Preprocessing, Splitting, BERT Word Embeddings, Binary and Multi-Label CNN, and Evaluation. Figure 1 represents the system architecture, where dashed lines indicate the inclusion of features like hashtags and ALL-CAPS, and slanted boxes signify expected data flow (input or output). Each stage of the architecture played a critical role in transforming the data, contributing to the construction and refinement of the model. Detailed discussions on the methods, sub-components, and processes employed within the architecture follow in subsequent sections.

The input data for the system architecture were derived from the labeled dataset of the base study by [9, 10]. This validated dataset served as a key comparison point for the four (4) different datasets produced by the implemented BERT-CNN models, as compared to the fastText CNN model: (1) BERT-CNN, (2) BERT-CNN with Hashtags, (3) BERT-CNN with ALL-CAPS, and (4) BERT-CNN with Hashtags and ALL-CAPS. The system architecture is made up of five (5) major modules, namely, Preprocessing, Splitting, BERT Word Embeddings, Binary and Multi-Label CNN, and Evaluation. The Preprocessing is composed of eight (6) submodules: Extraction of Posts, Data De-Identification and URL Removal, Special Character Processing, Tokenization, as well as the innovative changes that were added to the system, namely, the Normalization with ALL-CAPS Processing, and Hashtag Processing. The next module, Splitting, segmented the preprocessed data into training at 70/80/90% portion and testing at 30/20/10% portion, wherein the train dataset was used for the added BERT word embeddings in replacement for the fastText embedding algorithm. The BERT token embeddings were passed to two (2) types of CNN namely, Binary CNN and Multi Label CNN with the additional layer of Embedding Output for the BERT algorithm. The Evaluation module then tested the trained Binary CNN and Multi label CNN which resulted in two types of outputs: Classified posts as either Hate, Non-Hate or Positive, Neutral, Negative.

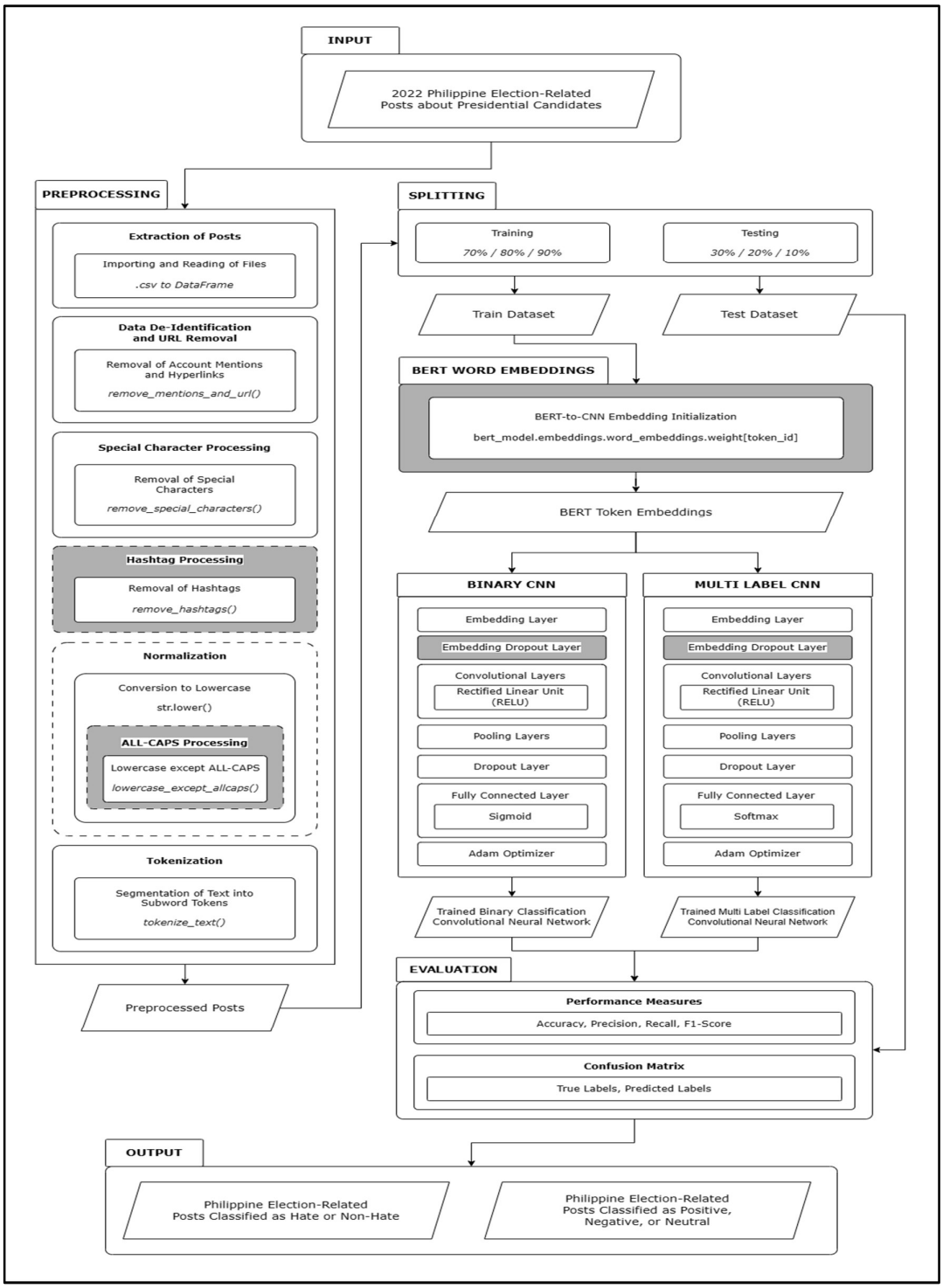

Figure 1. System architecture of the study.

4. Results and Discussion

4.1.Model comparisons.

The team conducted a study with four variations of BERT-CNN paired with a combination of features such as Hashtags and ALL-CAPS. This section of the study compared the models' performance measures namely, BERT-CNN and fastText CNN, BERT-CNN with Hashtags

and BERT-CNN without Hashtags, BERT-CNN with ALL-CAPS and BERT-CNN without ALL-CAPS, as well as BERT-CNN with Hashtags and ALL-CAPS and BERT-CNN without Hashtags and ALL-CAPS. Table 1 shows the comparison of performance measures for fast text CNN and BERT-CNN models for binary classification.

Table 1. Comparison of performance measures for fast text CNN and BERT-CNN models for binary classification.

Split	Accuracy		Precision		Recall		F1-Score	
	BERT-CNN	fastText CNN	BERT-CNN	fastText CNN	BERT-CNN	fastText CNN	BERT-CNN	fastText CNN
70:30	**90.30%**	85.61%	**90.30%**	85.97%	**90.30%**	85.70%	**90.30%**	85.59%
80:20	**90.62%**	85.94%	**90.64%**	86.02%	**90.63%**	85.92%	**90.62%**	85.92%
90:10	**93.55%**	86.52%	**93.59%**	86.85%	**93.58%**	86.47%	**93.55%**	86.49%

The comparison between the BERT-CNN and fastText CNN models for binary classification demonstrated significant differences in performance across various train-test splits, with BERT-CNN consistently outperforming fastText CNN in terms of accuracy, precision, recall, and F1-score. In the 70:30 split, the BERT-CNN model achieved an accuracy of 90.30%, while fastText CNN lagged behind with 85.61%. This performance trend remained consistent across other splits, with BERT-CNN maintaining higher precision and recall values. For example, in the 90:10 split, BERT-CNN showed a substantial improvement with an accuracy of 93.55% and an impressive recall of 93.59%, significantly surpassing fastText CNN, which displayed more modest performance gains as the training set size increased. Overall, BERT-CNN demonstrated a clear advantage, showcasing its superior predictive capabilities and suitability for binary classification tasks, regardless of the train-test split.

Table 2. Comparison of performance measures for BERT-CNN with hashtags and BERT-CNN without hashtags models for binary classification.

Split	Accuracy		Precision		Recall		F1-Score	
	BERT-CNN w/ HT	BERT-CNN w/o HT	BERT-CNN w/ HT	BERT-CNN w/o HT	BERT-CNN w/ HT	BERT-CNN w/o HT	BERT-CNN w/ HT	BERT-CNN w/o HT
70:30	**90.69%**	90.30%	**90.69%**	90.30%	**90.69%**	90.30%	**90.69%**	90.30%
80:20	90.14%	**90.62%**	90.16%	**90.64%**	90.15%	**90.63%**	90.14%	**90.62%**
90:10	92.38%	**93.55%**	91.76%	**93.59%**	92.86%	**93.58%**	92.31%	**93.55%**

The comparison between BERT-CNN models with and without hashtags (HT) for binary classification highlighted subtle variations in their performance metrics. As shown in Table 2, the 70:30 split, the BERT-CNN with HT achieved an accuracy of 90.69%, precision of 90.69%, recall of 90.33%, and an F1-score of 90.51%, surpassing the corresponding metrics of the model without HT by a small margin. However, BERT-CNN without HT consistently exhibited slightly higher accuracy, precision, recall, and F1-score values in the 80:20 and 90:10 splits. This trend showcased a consistent advantage for the model without HT. While both

models demonstrated strong binary classification capabilities, the absence of hashtags appeared to contribute to a modest but consistent improvement in performance metrics for BERT-CNN.

Table 3. Comparison of performance measures for BERT-CNN with ALL-CAPS and BERT-CNN without ALL-CAPS models for binary classification.

Split	Accuracy		Precision		Recall		F1-Score	
	BERT-CNN	fastText CNN	BERT-CNN	fastText CNN	BERT-CNN	fastText CNN	BERT-CNN	fastText CNN
70:30	**65.54%**	62.63%	**65.46%**	61.97%	**65.52%**	62.54%	**63.21%**	61.82%
80:20	**66.28%**	62.89%	**65.82%**	62.11%	**66.24%**	62.79%	**64.49%**	62.07%
90:10	**67.84%**	63.80%	**66.60%**	63.09%	**67.10%**	63.41%	**66.27%**	62.74%

Table 3 shows the comparison of performance measures for BERT-CNN with and without ALL-CAPS (AC) in binary classification revealed notable distinctions. The uncased BERT-CNN, without AC, consistently outperformed its cased counterpart with AC across all splits, demonstrating superior accuracy, precision, recall, and F1-score—except for the 70:30 split, where BERT-CNN with AC exhibited higher recall at 90.60%. Specifically, at the 90:10 split, the uncased model achieved an impressive accuracy of 93.55%, surpassing the cased model's 91.41%. This suggests that the uncased BERT-CNN model exhibited greater proficiency in handling the classification task. Notably, a study by [12] found that the uncased BERT model displayed enhanced resilience to inconsistencies in capitalization within noisy text data, whereas the BERT-base-cased model excelled in well-written text with clearly provided case information. The observed performance gap in this study's models may stem from the heightened sensitivity of the cased model to noise and inconsistencies, or it could be influenced by the prevalence of fewer ALL-CAPS instances in the dataset.

Table 4. Comparison of performance measures for BERT-CNN with hashtags and ALL-CAPS and BERT-CNN without hashtags and ALL-CAPS models for binary classification.

Split	Accuracy		Precision		Recall		F1-Score	
	BERT-CNN w/ HT	BERT-CNN w/o HT	BERT-CNN w/ HT	BERT-CNN w/o HT	BERT-CNN w/ HT	BERT-CNN w/o HT	BERT-CNN w/ HT	BERT-CNN w/o HT
70:30	**67.93%**	65.54%	**67.37%**	65.46%	**67.90%**	65.52%	**66.90%**	63.21%
80:20	**67.06%**	66.28%	**67.12%**	65.82%	**67.09%**	66.24%	**65.30%**	64.49%
90:10	**69.14%**	67.84%	**68.44%**	66.60%	**68.40%**	67.10%	**67.41%**	66.27%

The uncased BERT-CNN without HT and AC consistently outperformed its cased counterpart across all splits, exhibiting higher accuracy, precision, recall, and F1-score—except for the 80:20 split, where cased BERT-CNN with HT+AC exhibited higher accuracy, precision and F1-score, and for the 70:30 split with higher recall. As shown in Table 4, the performance of the cased model, while competitive, showed less variability across different splits, with minimal improvements in precision and recall. These findings emphasize the impact of preprocessing choices, such as the inclusion of hashtags and ALL-CAPS, on model

performance and underline the effectiveness of the uncased BERT-CNN configuration in this particular binary classification scenario.

Table 5. Comparison of performance measures for fastText CNN and BERT-CNN models for multi label classification.

Split	Accuracy		Precision		Recall		F1-Score	
	BERT-CNN w/ AC (Cased)	BERT-CNN w/o AC (Uncased)	BERT-CNN w/ AC (Cased)	BERT-CNN w/o AC (Uncased)	BERT-CNN w/ AC (Cased)	BERT-CNN w/o AC (Uncased)	BERT-CNN w/ AC (Cased)	BERT-CNN w/o AC (Uncased)
70:30	64.71%	**65.54%**	64.05%	**65.46%**	64.65%	**65.52%**	**63.42%**	63.21%
80:20	65.04%	**66.28%**	64.08%	**65.82%**	64.89%	**66.24%**	63.95%	**64.49%**
90:10	67.71%	**67.84%**	**67.56%**	66.60%	**67.23%**	67.10%	65.61%	**66.27%**

The results presented in Table 5 highlighted the superior performance of the BERT-CNN model in comparison to the fastText CNN model across various split ratios (70:30, 80:20, and 90:10). The evaluation metrics, including accuracy, precision, recall, and F1-score, consistently demonstrated higher values for the BERT-CNN model compared to the base study's model. Specifically, the BERT-CNN model exhibited a notable superiority in accuracy, surpassing the fastText CNN model by 2.91% absolute accuracy on the 70:30 train-test split, 3.39% on the 80:20 train-test split, and 4.01% on the 90:10 train-test split. These findings suggested that the BERT-CNN model outperforms the fastText CNN model across different data split ratios, emphasizing its effectiveness in the context of the study.

Table 6. Comparison of performance measures for BERT-CNN with hashtags and BERT-CNN without hashtags models for multi label classification.

Split	Accuracy		Precision		Recall		F1-Score	
	BERT-CNN w/ HT and AC (Cased)	BERT-CNN w/o HT and AC (Uncased)	BERT-CNN w/ HT and AC (Cased)	BERT-CNN w/o HT and AC (Uncased)	BERT-CNN w/ HT and AC (Cased)	BERT-CNN w/o HT and AC (Uncased)	BERT-CNN w/ HT and AC (Cased)	BERT-CNN w/o HT and AC (Uncased)
70:30	**66.45%**	65.54%	**65.98%**	65.46%	**66.36%**	65.52%	**65.49%**	63.21%
80:20	65.89%	**66.28%**	65.15%	**65.82%**	65.81%	**66.24%**	**64.96%**	64.49%
90:10	65.10%	**67.84%**	65.37%	**66.60%**	64.38%	**67.10%**	61.74%	**66.27%**

The significance of incorporating HT in multi-label classification, specifically distinguishing between positive, negative, and neutral sentiments, was evident in the performance of the BERT-CNN model. As indicated in Table 6, when hashtags were included, the model demonstrated a slight but consistent improvement across all evaluation metrics and for all split ratios compared to the model without HT. Furthermore, the BERT-CNN model, even with hashtags, outperformed the fastText CNN model across all metrics for every train-test split. The BERT-CNN model with HT achieved the highest absolute accuracy, surpassing the model without HT by 1.23% on the 70:30 split ratio, 0.78% on the 80:20 split, and 0.33% on the 90:10 split. These results highlighted the positive impact of incorporating hashtags in

the multi-label classification task, indicating that including this contextual information contributed to the model's ability to accurately classify sentiments. The consistent outperformance of the BERT-CNN model with HT, especially in terms of accuracy, reinforces the relevance and effectiveness of leveraging hashtag information for improved sentiment analysis in multi-label scenarios.

Table 7. Comparison of performance measures for BERT-CNN with ALL-CAPS and BERT-CNN without ALL-CAPS models for multi label classification.

Split	Accuracy		Precision		Recall		F1-Score	
	BERT-CNN w/ AC (Cased)	BERT-CNN w/o AC (Uncased)	BERT-CNN w/ AC (Cased)	BERT-CNN w/o AC (Uncased)	BERT-CNN w/ AC (Cased)	BERT-CNN w/o AC (Uncased)	BERT-CNN w/ AC (Cased)	BERT-CNN w/o AC (Uncased)
70:30	64.71%	**65.54%**	64.05%	**65.46%**	64.65%	**65.52%**	**63.42%**	63.21%
80:20	65.04%	**66.28%**	64.08%	**65.82%**	64.89%	**66.24%**	63.95%	**64.49%**
90:10	67.71%	**67.84%**	**67.56%**	66.60%	**67.23%**	67.10%	65.61%	**66.27%**

In Table 7, the evaluation of the BERT-CNN model as an uncased model without the AC indicated that its performance was marginally superior when compared to the cased BERT-CNN model with AC. This difference, however, was relatively small and inconsistent across various metrics. Despite the overall slightly better performance without AC, the BERT-CNN model with AC demonstrated specific improvements. Specifically, BERT-CNN with AC outperformed the model without AC by 0.21% in F1-Score on the 70:30 split ratio. Additionally, there were gains of 0.96% in precision and 0.13% in recall on the 90:10 split for the model with AC. These findings suggest that while the uncased model without AC showed a slight edge in overall performance, the inclusion of AC contributed to specific enhancements in precision, recall, and F1-Score, particularly in certain split ratios.

Table 8. Comparison of Performance Measures for BERT-CNN with Hashtags and ALL-CAPS and BERT-CNN without Hashtags and ALL-CAPS models for Multi Label Classification.

Split	Accuracy		Precision		Recall		F1-Score	
	BERT-CNN w/ HT and AC (Cased)	BERT-CNN w/o HT and AC (Uncased)	BERT-CNN w/ HT and AC (Cased)	BERT-CNN w/o HT and AC (Uncased)	BERT-CNN w/ HT and AC (Cased)	BERT-CNN w/o HT and AC (Uncased)	BERT-CNN w/ HT and AC (Cased)	BERT-CNN w/o HT and AC (Uncased)
70:30	**66.45%**	65.54%	**65.98%**	65.46%	**66.36%**	65.52%	**65.49%**	63.21%
80:20	65.89%	**66.28%**	65.15%	**65.82%**	65.81%	**66.24%**	**64.96%**	64.49%
90:10	65.10%	**67.84%**	65.37%	**66.60%**	64.38%	**67.10%**	61.74%	**66.27%**

In Table 8, the uncased BERT-CNN without HT and AC demonstrated superior overall performance compared to its cased counterpart with HT and AC. While the cased model showed superiority in the 70:30 split across all evaluation metrics, including a notable improvement in F1-score in subsequent splits, the overall trend revealed a decline and inconsistency in other metrics. Although the cased model had a brief advantage in specific split ratios, it was ultimately overshadowed by the uncased model's sustained and more consistent

performance. This highlights the nuanced impact of hashtag and ALL-CAPS inclusion on model performance, suggesting that the uncased BERT-CNN without HT and AC may offer more robust and reliable performance across a broader range of scenarios.

5. Conclusions

Having deployed the BERT-CNN model, this study achieved its primary goal of assessing the model's effectiveness in identifying political hate speech on X, demonstrating its proficiency in sentiment analysis of Filipino election-related posts. The comparison with the model proposed by [5] consistently favored BERT-CNN, validating its superior performance. The following conclusions were derived from the deployment of the presented system and the summary of findings: The BERT-CNN model demonstrated exceptional proficiency in classifying Philippine election-related posts into hate or non-hate categories, achieving high accuracy, precision, recall, and F1-scores, all exceeding 90%. The consistency of these outstanding results across a spectrum of train-test splits highlighted the model's robust performance. Regardless of variations introduced, such as the incorporation of hashtags, ALL-CAPS, or both, the BERT-CNN model maintained its superior classification capabilities. These findings underscore the model's resilience and effectiveness in handling diverse textual data related to Philippine elections, making it a reliable and versatile tool for sentiment analysis in this context. The comparison between the BERT-CNN and fastText CNN models revealed notable advancements in hate speech detection achieved by the BERT-CNN model. Demonstrating consistent superiority across various performance measures and splits, the BERT-CNN model exhibits a robust ability to classify effectively in different scenarios. The key strength of BERT-CNN lies in its capacity to capture intricate contextual relationships within language, rendering it particularly well-suited for complex sentiment analysis tasks. The model's persistent outperformance suggests its potential to offer more accurate and nuanced predictions compared to fastText CNN. Furthermore, the results of McNemar's Test with Holm-Bonferroni correction confirmed a significant improvement in the BERT-CNN model’s performance compared to fastText CNN in the sentiment analysis of election-related posts, specifically in hate or non-hate classification, across all train-test splits. The integration of BERT embeddings into the BERT-CNN model significantly enhanced its performance in the sentiment analysis of Philippine election-related posts. The study delved into the intricate dynamics of hate speech detection within the context of the Philippine presidential elections. One noteworthy aspect is the substantial influence of candidate names on the model's predictions, indicating a large word embedding value. Candidate names emerged as robust conversational cues, becoming integral to the classification of posts as hate or non-hate. This is exemplified by the detailed analysis of candidate names such as Marcos and Leni Robredo, showcasing their varying frequencies in hate and non-hate posts. The BERT-CNN model demonstrated its ability to learn from training data, identifying frequent words like "Bongbong" and "Marcos" as indicators for hate predictions, thereby improving its contextual understanding. However, the study also uncovered potential misclassifications, emphasizing the model's reliance on training data and the context-dependent nature of BERT embeddings. Despite the risk of misclassification, the BERT word embeddings approach was deemed advantageous in enhancing the model's overall performance, showcasing its ability to embed words based on contextual usage and providing valuable insights into the sentiment analysis of election-related posts.

Acknowledgment

The authors would like to acknowledge the authors of the previous research paper entitled " Hate Speech in Filipino Election-Related Tweets: A Sentiment Analysis using Convolutional Neural Networks" for sharing their dataset which was used in the training and testing phase of the model. The authors would like to thank all the teaching academic staff of the Computer Science Department, University of Santo Tomas for reviewing and scrutinizing the content of this paper.

Author contribution

Conceptualization: Mendoza, Nadurata, Ortiz, Padlan, Ponay; Methodology: Mendoza, Nadurata, Ortiz, Padlan, Ponay; Data Collection: Mendoza, Nadurata, Ortiz, Padlan, Arganosa et al; Data Analysis: Mendoza, Nadurata, Ortiz, Padlan; Writing: Mendoza, Nadurata, Ortiz, Padlan, Ponay; Supervision: Ponay

Competing Interest

The authors declare that no potential conflict of interest exists related to this article.

References

[1] Hate speech and incitement to hatred or violence. (accessed on 1 September 2024) Available online: https://www.ohchr.org/en/special-procedures/sr-religion-or-belief/hate-speech-and-incitement-hatred-or-violence#:~:text=As%20a%20matter%20of%20principle,peaceful%2C%20inclusive%20and%20just%20societies.

[2] Alfina, I.; Sigmawaty, D.; Nurhidayati, F.; Hidayanto, A.N. (2017). Utilizing hashtags for sentiment analysis of tweets in the political domain. Proceedings of the 9th International Conference on Machine Learning and Computing, 43–47. https://doi.org/10.1145/3055635.3056631.

[3] Hidayatullah, A.F.; Cahyaningtyas, S.; Hakim, A.M. (2021). Sentiment analysis on Twitter using neural network: Indonesian presidential election 2019 dataset. *IOP Conference Series: Materials Science and Engineering*, *1077*(1), 012001. https://doi.org/10.1088/1757-899x/1077/1/012001.

[4] Malik, P.; Aggrawal, A.; Vishwakarma, D.K. (2021, April). Toxic speech detection using traditional machine learning models and BERT and fastText embedding with deep neural networks. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 1254–1259. IEEE. https://doi.org/10.1109/ICCMC51019.2021.9418229.

[5] Alzahrani, E.; Jololian, L. (2021). How different text-preprocessing techniques using the BERT model affect the gender profiling of authors. *Advances in Machine Learning,* 1–8. https://doi.org/10.5121/csit.2021.111501.

[6] Solitana, N.T.; Cheng, C.K. (2021, December). Analyses of hate and non-hate expressions during election using NLP. 2021 International Conference on Asian Language Processing (IALP), 385–390.

[7] Velankar, A.; Patil, H.; Gore, A.; Salunke, S.; Joshi, R. (2022). L3cube-mahahate: A tweet-based Marathi hate speech detection dataset and BERT models. *arXiv preprint arXiv:2203.13778*.

[8] Alim, M.M.F. (2021, June). A sentiment analysis study for Twitter using the various model of convolutional neural network. *Journal of Physics: Conference Series*, *1918*(4), 042136. https://doi.org/10.1088/1742-6596/1918/4/042136.

[9] Arganosa S.;Marasigan, R.;Villanueva, J.; Wenceslao, K.; Ponay, C. (2022). Hate Speech in Filipino Election-Related Tweets: A Sentiment Analysis Using Convolutional Neural Networks. http://dx.doi.org/10.13140/RG.2.2.20961.52326

[10] Cabasag, N.; Chan, V.; Lim, S.; Gonzales, M.E.; Cheng, C. (2019). Hate Speech in Philippine Election-Related Tweets: Automatic Detection and Classification Using Natural Language Processing. *Philippine Computing Journal*, *XIV,* 1–14.

[11] Where is the love? Identifying hate speech in Philippine election-related tweets. (accessed on 1 September 2024) Available online: https://asite.aim.edu/data_science/where-is-the-love-identifying-hate-speech-in-philippine-election-related-tweets/.

[12] Mehta, R.P.; Sanghvi, M.A.; Shah, D.K.; Singh, A. (2019). Sentiment analysis of tweets using supervised learning algorithms. First International Conference on Sustainable Technologies for Computational Intelligence*,* 323–338. https://doi.org/10.1007/978-981-15-0029-9_26.

[13] Bello, A.; Ng, S.-C.; Leung, M.-F. (2023). A BERT framework to sentiment analysis of tweets. *Sensors*, *23*(1), 506. https://doi.org/10.3390/s23010506.

[14] Mastering BERT: A comprehensive guide from beginner to advanced in natural language processing. (accessed on 1 September 2024) Available online: https://medium.com/@shaikhrayyan123/a-comprehensive-guide-to-understanding-bert-from-beginners-to-advanced-2379699e2b51.

[15] De Goma, J., Hungria, C., Boquiren, A., & Garcia, R. (2022). Tagalog Sentiment Analysis Using Deep Learning Approach with Backward Slang Inclusion. 3rd African International Conference on Industrial Engineering and Operations Management, https://doi.org/10.46254/AF03.20220180.

[16] Kaviani, M.; Rahmani, H. (2020). EmHash: Hashtag recommendation using neural network based on BERT embedding. 6th International Conference on Web Research (ICWR). http://doi.org/10.1109/ICWR49608.2020.9122275.

[17] BERT transformers – how do they work? (accessed on 1 September 2024) Available online: https://www.exxactcorp.com/blog/Deep-Learning/how-do-bert-transformers-work.

[18] Statistical significance tests for comparing machine learning algorithms. (accessed on 1 September 2024) Available online: https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/.

[19] Chan, S.; Fyshe, A. (2018). Social and Emotional Correlates of Capitalization on Twitter. Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media. Association for Computational Linguistics: New Orleans, Louisiana, USA. pp. 10–15.

[20] Mingua, J.; Padilla, D.; Celino, E.J. (2021, November). Classification of fire-related posts on Twitter using Bidirectional Encoder Representations from Transformers (BERT). 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*,* 1–6. https://doi.org/10.1109/HNICEM.2021.9604867.

[21] Nanli, Z.; Ping, Z.; Weiguo, L.I.; Meng, C. (2012, November). Sentiment analysis: A literature review. 2012 International Symposium on Management of Technology (ISMOT)*,* 572–576.

[22] Imperial, J.M.; Orosco, J.; Mazo, S.M.; Maceda, L. (2019). Sentiment analysis of typhoon related tweets using standard and bidirectional recurrent neural networks. arXiv preprint arXiv:1908.01765.

[23] Chiorrini, A.; Diamantini, C.; Mircoli, A.; Potena, D. (2021). Emotion and sentiment analysis of tweets using BERT. Workshop Proceedings of the EDBT/ICDT 2021 Joint Conference, Nicosia, Cyprus.

[24] Galinato, V.; Amores, L.; Magsino, G.B.; Sumawang, D.R. (2023). Context-based profanity detection and censorship using Bidirectional Encoder Representations from Transformers. *SSRN,* 4341604. http://doi.org/10.2139/ssrn.4341604.

[25] Kaur, K.; Kaur, P. (2023). Improving BERT model for requirements classification by bidirectional LSTM-CNN deep model. *Computers and Electrical Engineering*, *108*, 108699. https://doi.org/10.1016/j.compeleceng.2023.108699.

[26] Dao, T.A.; Aizawa, A. (2023). Evaluating the effect of letter case on named entity recognition performance. In Natural Language Processing and Information Systems. NLDB 2023. Métais, E., Meziane, F., Sugumaran, V., Manning, W., Reiff-Marganiec, S., Eds.; *Lecture Notes in Computer Science*, 13913; Springer: Cham, Switzerland. https://doi.org/10.1007/978-3-031-35320-8_45.

[27] Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[28] Regalado, R.V.J.; Cheng, C.K. (2012, November). Feature-based subjectivity classification of Filipino text. 2012 International Conference on Asian Language Processing, 57–60.

[29] Sunarya, P.A.; Refianti, R.; Mutiara, A.B.; Octaviani, W. (2019). Comparison of accuracy between convolutional neural networks and Naïve Bayes classifiers in sentiment analysis on Twitter. *International Journal of Advanced Computer Science and Applications*, *10*(5). http://doi.org/10.14569/IJACSA.2019.0100511.