Published *online* on the journal's web page : http://innovatics.unsil.ac.id

Innovation in Research of Informatics (INNOVATICS)

| ISSN (Online) 2656-8993 |

Sarcasm Detection: A Comparative Analysis of RoBERTa-CNN vs RoBERTa-RNN Architectures

Sheraton Pawestri [1], Murinto[2], Muhammad Auzan[3,4]

[1]*Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Jl. Kapas No.9, Yogyakarta, 55166, Indonesia*
[2]*Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Jl. Kapas No.9, Yogyakarta, 55166, Indonesia*
[3]*Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan*
[4]*Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia*

[1]*sheraton@tif.uad.ac.id*, [2]*murintokusno@tif.uad.ac.id*, [3,4]*muhammadauzan@ugm.ac.id*

ARTICLE INFORMATION

Article History:
Received: July 10, 2024
Last Revision: September 28, 2024
Published *Online*: September 30, 2024

KEYWORDS

Sarcasm detection,
News headline,
RoBERTa,
Convolutional Neural Network,
Recurrent Neural Network

CORRESPONDENCE

Phone: +6285228224058
E-mail: sheraton@tif.uad.ac.id

ABSTRACT

Increasingly advanced technology and the creation of social media and the internet can become a forum for people to express things or opinions. However, comments or views from users sometimes contain sarcasm making it more difficult to understand. News headlines, sometimes contain sarcasm which makes readers confused about the content of the news. Therefore, in this research, a model was created for sarcasm detection. Many methods are used for sarcasm detection, but performance still needs to be improved. So this research aims to compare the performance of two text classification methods, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), in detecting sarcasm in English news headlines using RoBERTa text transformation. RoBERTa produces a fixed-size vector of numbers 1x768. The research results show that CNN has better performance than RNN. CNN achieved the highest average accuracy of 0.891, precision of 0.878, recall of 0.874, and f1-score of 0.876, with a loss of 0.260 and a processing time of 508.1 milliseconds per epoch. But overall highest can reach in fold 6, 0.897 for validation accuracy, 0.883 for F1-score and precision, and 0.882 for recall in CNN + RoBERTa model. On the contrary, RNN shows an accuracy of 0.711, precision of 0.692, recall of 0.620, f1-score 0.654, and loss of 0.564, with a longer processing time of 116500 milliseconds per epoch. The 10-fold cross-validation evaluation method ensures the model performs well and avoids overfitting. So it is recommended to use the combination of RoBERTa and CNN in other text classification applications that require high speed and accuracy. Further research is recommended to explore deeper CNN architectures or other architectural variations such as Transformer-based models for performance improvements.

1. INTRODUCTION

In the era of globalization and rapidly advancing technology, social media and the internet have provided a platform for people to express themselves, allowing their words to be seen by a global audience. On these platforms, individuals often express their opinions on various topics, ranging from personal issues to trending news. However, not all opinions are conveyed in straightforward language. Instead, users sometimes employ figures of speech, such as sarcasm, which can be challenging to detect. Sarcasm, especially in text form, can be difficult to interpret because it often relies on subtle contextual cues. Misunderstandings may arise when sarcasm in text, including news headlines, is interpreted literally, leading to confusion and potential misinterpretation [1].

Sarcasm is a style of language or a way of expressing thoughts through language, which usually shows the spirit and personality of the writer. One of the language styles commonly used are irony and sarcasm [2]. Sarcasm also relies on ambiguity where the literal meaning of a sentence differs from its intended meaning. For example, a headline that says, "Wow, It is the best weather ever!" during a storm would likely be sarcastic. According to Kreuz and Glucksberg in 1989, sarcasm is a verbal irony that expresses a negative and critical attitude towards people or events [3]. Sarcasm is figurative language that has a meaning that is harsh and hurtful. According to McDonald's in Dauphin's book page 486, sarcasm is a form of indirect expression deliberately used to produce a certain dramatic effect on the listener [4]. Sarcasm is defined as a type of language style that contains insults and insults so that it is unpleasant for the person who reads it to hear [5]. Sarcasm itself is quite easy to analyze if seen directly by observing someone's expressions and body movements, but sarcasm will be very difficult to recognize if it is in text form because, in text, it will be difficult for someone to guess other people's expressions and body movements [6]. So, sarcasm is a language style that uses irony or has an implied meaning and sometimes contains insults. Sarcasm is more complex than usual communication that varies across different social and cultural contexts. It can reflect the speaker’s attitude, personality, or critical stance toward a subject. For example, the interpretation of sarcasm may differ significantly between cultures, where certain expressions considered sarcastic in one culture might be interpreted literally in another. This makes it important to develop sarcasm detection models that are culturally aware and can account for these variances.

There have been many previous studies regarding sarcasm detection, especially in the last five years. Several methods have been used, but many methods can still be explored again in sarcasm detection, such as Neural Networks. Simple methods that are still used for research, especially in the field of text data, are Naive Bayes (NB), Support Vector Machine (SVM) [7], Random Forest (RF) [8], or even all three [9], and others. Previous research suggests using more modern methods such as Deep Learning (ANN, CNN, and RNN) [10]. CNN and RNN are still widely used, especially with various combinations of CNN and other methods. Some examples that use CNN and its development are research from Hazarika, D., et al. [11], Misra, R. and Arora, P. [12]. Misra also researched sarcasm detection using this dataset in 2023 [13]. Research from Wijaya, A.C., and Wibawa, I.G.A. [14] also used CNN. Research from Hazarika, D., et al. suggested using RNN in sarcasm detection. For data transformation, the method that will be used is RoBERTa. RoBERTa is also a popular method used in research [15] and [16] which has a higher level of accuracy than basic methods. Himawan 2022 research also uses RNN [17]. There is also research on sarcasm detection in Indonesian tweets [18]. Despite advances in sarcasm detection, several challenges remain. These include linguistic ambiguity, situational context, and emotional nuance, all of which play a significant role in how sarcasm is understood. For example, the same sentence could be interpreted as sarcastic or non-sarcastic depending on the context in which it is used. Therefore, accurately detecting sarcasm in text requires a model that can understand both the words and the context in which they are used.

Sarcasm detection has significant practical implications in various real-world contexts. For instance, recommendation systems can benefit from accurately detecting sarcasm to better gauge user sentiment. Understanding whether a review or comment is sarcastic can help systems avoid recommending products or services based on misleading or ironic statements. In content moderation, detecting sarcasm is essential for identifying potential harmful or inappropriate content that may not be flagged through traditional keyword-based approaches. For example, users might employ sarcasm to disguise hateful or offensive comments, and a robust sarcasm detection system could help content moderators more effectively identify and address such content. Similarly, in customer service and social media analysis, sarcasm detection enables more accurate sentiment analysis, allowing companies to respond appropriately to customer feedback and better manage their online reputation.

Therefore, it is important to create a model that is good at sarcasm detection. Traditional models like SVM and NB often struggle to capture the complexity and subtlety of sarcasm. Recent advancements in deep learning, particularly the use of neural networks such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have shown promise in improving performance [11-14]. In this research, those methods if combined with transformers like RoBERTa, can better capture the intricacies of sarcastic text. Based on research and suggestions from previous research, this research uses a deep learning approach for the classification method and RoBERTa for the transformation method. RoBERTa excels in understanding the context of words within sentences, making it ideal for sarcasm detection, where meaning often relies on the interplay between words rather than isolated terms. This capability allows RoBERTa to capture better the subtle and often ironic nature of sarcastic statements. Therefore, RoBERTa is very good to be applied in this research. The dataset used in this research is a dataset containing news headlines that have been labeled as sarcasm or not. News headlines usually use formal language, making them even more difficult to recognize. Therefore, this research makes sarcasm detection which aims to help in analyzing whether the news title sentence is sarcasm or not by comparing CNN and RNN methods with RoBERTa as a transformation method.

2. RELATED WORK

Research in 2018 used a modification of CNN and produced CASCADE for sarcasm detection [11]. This research uses various types of CNN to compare its method (CASCADE) with the main framework using CNN. The dataset used is the SARC public dataset. The results show that the highest accuracy in this study is the CASCADE method with an accuracy of 0.74 and an F1-score of 0.75. This research also suggests that future research can use RNN for better performance. The next research on sarcasm detection in 2020 uses the RoBERTa method combined with a deep neural network [14]. As a comparison, several methods are used, namely ELMO, USE, and BERT. The dataset used is the Twitter dataset by crawling. The results show that the highest performance was obtained from RoBERTa with a Precision accuracy of 0.77, Recall of 0.78, and F1-Score of 0.77. This research states that future research is expected to tune parameters for better results.

Other research in 2021 is about comparative studies of several methods. The methods used in this comparison are ELMo, MBSVM, XLnet, BERT, RoBERTa, CASCADE, and the last one is a combination of RNN and CNN with RoBERTa [15]. The datasets used are Twitter, Reddit, IAC, transcript, and dialogue. The results of this study show that the last method has the best performance with an accuracy of 0.78 and an F1-Score of 0.79.

The next research will be research on sarcasm detection using the hybrid neural network method in 2022, accompanied by the creation of a News Headlines dataset which will be used in this research. Dataset based on news titles on TheOnion's and HuffPost's websites [12]. The method used in this research is Hybrid LTSM combined with CNN. The results show good accuracy, namely 0.90, with suggestions to improve tuning parameters, better architecture, and more advanced knowledge so that the system can perform better. In the same year, the next research focused on detecting sarcasm and irony using CNN with seven layers [13]. The dataset used is a dataset from Twitter which has been labeled with sarcasm or irony. The results of this study show an average accuracy of 0.75.

Furthermore, in 2019, as a comparison for sarcasm detection using simple methods such as Support Vector Machine, Naive Bayes, and Random Forest for classification [9]. The dataset used in this research is the crawled Twitter dataset. This research also analyzes the impact of sarcasm on sentiment on Twitter in Indonesia. The results show that precision and recall were won by Naive Bayes with values of 0.63 and 0.64, accuracy was won by Random Forest at 0.61, and the highest F1-Score was obtained when using SVM at 0.58. In 2019, research about several methods such as feature extraction using unigrams and four sets of Boazizi features with Random Forest sarcasm detection and then feature extraction with TF-IDF with sarcasm detection using Naive Bayes [16]. The dataset used is the Twitter dataset resulting from manual crawling and labeling using experts. The results show the best average accuracy of 0.80, precision of 0.83, and recall of 0.91. The research states that things that need to be paid attention to are non-standard words and sentence context.

Another study in 2022 analyzed sarcasm detection using LSTM and BiLSTM which was then compared [17]. The dataset used in this research is the same as this research, namely the News Headlines Dataset. The best results obtained in this research were with the BiLSTM method with accuracy, precision, and recall of 0.83. Another research for sarcasm detection in 2020 using Support Vector Machine compared with Naive Bayes [7]. The dataset used is the Twitter dataset because the research contains the detection of sarcasm in politics at that time in Indonesia. The best results were obtained using SVM with an accuracy value of 0.86. The research suggests using other algorithms besides SVM and NB. The latest research in 2023 contained sarcasm detection using Random Forest as a classification method [8]. The dataset used in this research is the Facebook dataset with an accuracy of 0.53. The suggestions listed are for the use of loanwords and non-standard words which do not yet exist so there is still room for further development.

3. METHODOLOGY

The initial process for this research is data collection. In this research, the data uses a dataset containing news headlines that are labeled as sarcasm or not. News headlines usually use formal language. Therefore, it is interesting to study how a model can guess whether the headline is sarcastic or not sarcastic. Even though, sarcasm usually occurs in informal sentences. This dataset was taken from TheOnion and HuffPost which consists of 26,709 news headlines. This dataset was created by Misra, R. and Arora, P.[12] and already has its license, namely Attribution 4.0 International (CC BY 4.0), which means it can be used for research. This dataset can be accessed via GitHub https://github.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection. In the dataset, there are three features called is_sarcastic, headline, and article_link. The value of is_sarcastic is 1 if the headline contains sarcasm and 0 if the headline does not contain sarcasm. Feature headlines contain headlines from different sources and the article_link is the source link of the headlines.

After collecting the dataset, the dataset was cleaned by removing the duplicate and missing data. After that, the dataset was split into two parts, validation data and train data. Train data is 80% of the total data in the dataset. So data train contains 21281 data and the validation data is 5321 data or 20% of total data. Then the train data in the form of words will be represented in paragraph vectors using RoBERTa. The method is good for initial text representation and RoBERTa is an optimization of BERT [19]. After the initial representation of the text is carried out, the next stage is to form a model that can predict whether the sentence is sarcasm or not. The methods used for the models in this research are CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network). CNN is the traditional version of ANN (Artificial Neural Network). CNN and ANN both have neurons that always optimize themselves at each iteration. CNN is outstanding for looking for patterns in image data, but it is also very good at complex feature extraction processes [20]. However, in the same research, it was also stated that using RNN could produce better results. In addition, learning about the basics of RNNs can bring many benefits to machine learning [21]. So in this research, an evaluation will be carried out between CNN and RNN with the initial representation of words using RoBERTa.

After looking for patterns with CNN or RNN, the two models can predict which sentences are sarcasm and which are not. The models will be evaluated with their accuracy, precision, recall, and F1-Score. The evaluation method used 10-fold Cross-Validation. After the dataset was split into 80% train data and 20% validation data, then split again with different data until 10 splits (10 subsets), each time with a different fold serving as the test set and the remaining folds used for training. This method is used to avoid overfitting. After that, the performance of the two models and the weaknesses of the two models will be analyzed. This is useful for future research so that further research will know how to develop it, especially for these two models. The flow diagram for this research can be seen in Figure 1 as follows.

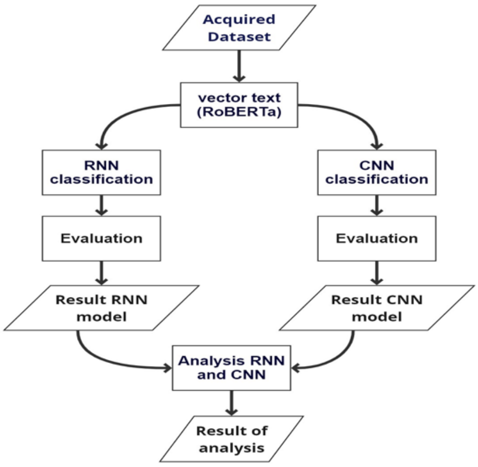

FIGURE 1. FLOWCHART SARCASM DETECTION

Research begins by taking a dataset. After the dataset is retrieved, data cleaning is performed. Data cleaning means eliminating empty data and eliminating duplicate data. After data cleaning, what is done is to carry out data transformation using RoBERTa. This research does not use traditional data preprocessing such as stemming and removing stopwords because this preprocessing is not significant for RoBERTa. This is because RoBERTa processes data based on the context of the sentence. So words that are unique and describe the context of a document have been summarized in RoBERTa. Apart from that, RoBERTa also recognizes words such as stop words and then ignores them. After that, RoBERTa creates a tokenization subword that functions like tokenization. This data preprocessing has no significant effect on RoBERTa or BERT [22], [23]. The output from Roberta is always a vector of size 768. After that, the vector is dimensioned so the classification process can be carried out using CNN and RNN. After classification using CNN and RNN, the classification results are evaluated using a confusion matrix of accuracy, recall, precision, and f1-score.

CNN and RNN are algorithms developed from Neural Network (NN) whose use is similar to NN. CNN and RNN both use certain layers whose uses vary. This research uses several layers for CNN, such as convolutional layers, pooling layers using maxpooling1D layers, flattened layers, and fully-connected layers or dense layers. Meanwhile, RNN uses several layers, such as bidirectional layers, fully-connected layers (dense layers), and dropout. The architecture of this research can be seen in Figure 2 as follows.

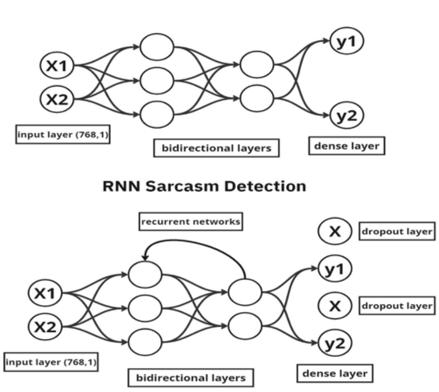

FIGURE 2. ARCHITECTURE CNN AND RNN SARCASM DETECTION

3.1 Convolutional Neural Network (CNN)

CNN is a deep learning algorithm that is commonly used for image data. CNN is also a Neural Network algorithm originally used to process image data. However, as time goes by, CNN is also capable of processing data other than image data. CNN models are designed automatically and adaptively with various layers such as convolutional layers, pooling layers, and fully-connected layers (dense layers). Convolutional layers are used for the initial filter of the input, pooling layers are used to reduce computation by setting dimensions into ones that are simpler but already describe the input, while fully-connected layers (dense layers) are used to determine predictions.

The concept of CNN was put forward by Kunihiko Fukushima in 1980 [24]. However, modern CNN was popularized by Yann LeCun et al. in 1998 with their research on the LeNet-5 architecture filled with seven layers, which contain convolutional layers, sub-sampling layers, and fully-connected layers [25]. The way CNN works is similar to Neural Networks in general. Input that has been entered will enter a series of hidden layers. Each layer has several neurons and each layer will be completely connected from one layer to another. After that, it is included in the output layer. The difference between NN and CNN is that CNN has neurons consisting of 3 dimensions (width, height, and depth). The general CNN architecture can be seen in Figure 3 as follows.

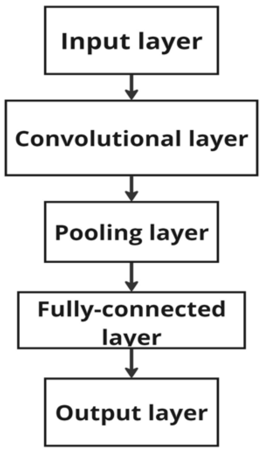

FIGURE 3. ARCHITECTURE CNN

3.2 Recurrent Neural Network (RNN)

RNN is a deep learning algorithm like CNN that is designed for sequence data. RNN has a direct connection to the cycle which makes it capable of processing hidden states and can obtain information from previous inputs in the sequence. These traits make RNN good for processing time series datasets, natural language processing, and voice data. RNN runs the process for each element in the sequence where the output depends on the initial computation. John Hopfield proposed RNN in 1982 and was later named Hopfield Network [26]. RNN was later developed into modern RNN by David Rumerlhart et al. in 1986 by introducing backpropagation for RNN training [27].

RNN became popular in 1997 with Long Short-Term Memory (LSTM) by Sepp Hochreiter and Jürgen Schmidhuber [28]. RNN has internal memory so it can remember all the information provided. RNN initially processes input in such a way that it produces output. This previous information is used as a benchmark for the next output, so it can be said that the RNN not only considers this input but also other information that has been obtained previously. The general RNN architecture can be seen in Figure 4 as follows.

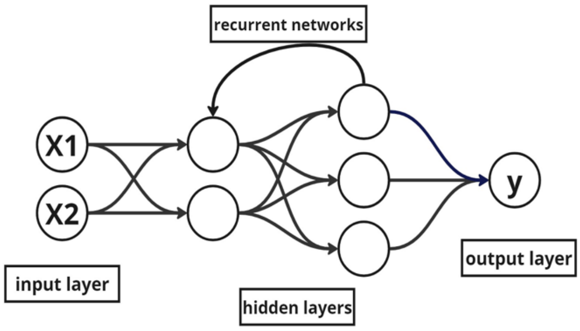

FIGURE 4. ARCHITECTURE RNN

3.3 Robustly Optimize BERT Approach (RoBERTa)

RoBERTa is a transformer method like TF-IDF, word2vec, and others. RoBERTa is another model of BERT developed by FacebookAI. RoBERTa improves BERT by optimizing data preprocessing, improving training data, and adjusting parameters to produce more accurate models in the field of Natural Language Processing (NLP). RoBERTa was first introduced by Yinhan Liu, et al. In 2019 [29]. RoBERTa aims to strengthen BERT optimization by optimizing data preprocessing and utilizing wider data. RoBERTa also uses dynamic masking which changes every epoch to improve the prediction of missing words. The general RoBERTa architecture can be seen in Figure 5.

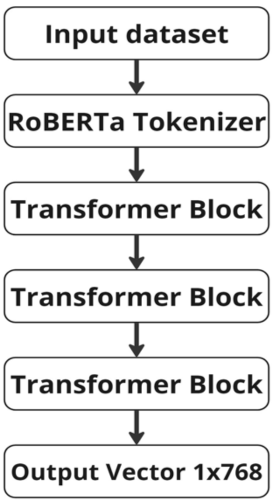

FIGURE 5. ARCHITECTURE ROBERTA

4. RESULT AND DISCUSSION

This research uses Python 3.10 on Google Colab Pro. The Python libraries used are Python and TensorFlow or Keras and other supporting libraries. The first step is to download a high-quality dataset regarding news titles labeled sarcastic or not. This data can be downloaded on GitHub https://github.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection. The dataset was taken from news sites TheOnion and HuffPost. After the dataset is downloaded. Dataset cleaning is carried out, such as deleting duplicate and empty data. After that, the word transformation process is carried out by RoBERTa as a text data vectorizer, which is then continued with classification using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) models.

Traditional data preprocessing is not performed because RoBERTa covers everything. The CNN and RNN performance is then compared. TensorFlow was used to train the CNN and RNN classification models, while PyTorch was used to utilize the RoBERTa model in the feature extraction process. The evaluation used in this research is 10-fold cross-validation with the evaluation parameters being accuracy, precision, recall, f1-score, and loss. This is used to avoid overfitting and ensure the model has good generalization. The data was divided into 10 subsets, and the model was trained 10 times, each time using one subset as test data and the other nine subsets as training data. Evaluation results with 10-fold cross-validation provide a more stable average performance and ensure that the model is not too dependent on one particular data set. The CNN and RNN implementation uses epoch parameters of 50, batch size of 512, and the loss function used is 'binary_crossentropy'. These parameters were chosen to optimize the model performance in text classification using vector representation from RoBERTa.

RoBERTa is a highly embedded transformer capable of converting any input text into a vector representation with dimensions of 1x768. RoBERTa also enables deep contextual coding of each text input making each text generate information-rich vectors. This research compares the performance of CNN and RNN when combined with RoBERTa. CNN is proven to be better than RNN. CNN shows a much faster training speed, with an average time of less than 1 second per epoch with better performance values.

Meanwhile, RNN takes hundreds of seconds per epoch with a lower performance value than CNN. These results show that CNN is more efficient in terms of training time and performance compared to RNN. The superiority of CNN compared to RNN in this study is due to CNN's ability to utilize highly embedded representations from RoBERTa more effectively than RNN. CNNs can better capture spatial features from the 1x768 vectors generated by RoBERTa, while RNNs, although they can handle sequence data, tend to be slower and less efficient in handling highly structured data such as the results of RoBERTa transformations. Apart from that, CNN is better able to avoid the vanishing gradient problem that often occurs in RNN.

The evaluation results show that the CNN model achieves superior performance with the highest average results per fold for validation accuracy of 0.891, precision of 0.878, recall of 0.874, f1-score of 0.876, loss of 0.260, and requires a processing time of 508.1 milliseconds per epoch. But overall highest can reach in fold 6, 0.897 for validation accuracy, 0.883 for F1-score and precision, and 0.882 for recall in CNN + RoBERTa model. The processing time required by CNN is much smaller because it is less than 1 second. This is different from RNN which requires an average processing time of 116.5 seconds. Results for accuracy, loss, and time for CNN can be seen in Table 1. Then, the results for precision, recall, and F1-Score for CNN can be seen in Table 2.

TABLE 1. CNN ACCURATION, LOSS, TIME RESULT

Fold	Accuracy	Loss	Time (ms)
1	0.890	0.263	554
2	0.895	0.254	479
3	0.887	0.271	490
4	0.880	0.285	498
5	0.888	0.265	491
6	0.897	0.249	463
7	0.894	0.254	548
8	0.890	0.257	542
9	0.895	0.252	553
10	0.897	0.255	463
Average	**0.891**	**0.260**	**508.1**

TABLE 2. CNN PRECISION, RECALL, F1-SCORE

Fold	Precision	Recall	F1-Score
1	0.878	0.870	0.874
2	0.883	0.878	0.880
3	0.873	0.871	0.872
4	0.869	0.855	0.862
5	0.875	0.868	0.871
6	0.883	0.882	0.883
7	0.877	0.882	0.880
8	0.875	0.875	0.875
9	0.884	0.875	0.879
10	0.883	0.881	0.882
Average	**0.878**	**0.874**	**0.876**

The highest average value per fold produced by RNN for accuracy was 0.711, precision was 0.692, recall was 0.620, f1-score was 0.654, and loss was 0.564. More detailed results for accuracy, loss, and time RNN are shown in Table 3. For precision, recall, and F1-score are shown in Table 4.

TABLE 3. RNN ACCURATION, LOSS, TIME RESULT

Fold	Accuracy	Loss	Time (ms)
1	0.723	0.551	117000
2	0.726	0.547	116000
3	0.713	0.560	113000
4	0.678	0.604	116000
5	0.714	0.559	116000
6	0.718	0.556	120000
7	0.719	0.558	116000
8	0.708	0.570	117000
9	0.703	0.573	116000
10	0.711	0.566	118000
Average	**0.711**	**0.564**	**116500**

TABLE 4. RNN PRECISION, RECALL, F1-SCORE

Fold	Precision	Recall	F1-Score
1	0.704	0.637	0.669
2	0.712	0.638	0.673
3	0.692	0.625	0.657
4	0.652	0.574	0.610
5	0.700	0.614	0.654
6	0.707	0.616	0.658
7	0.698	0.643	0.669
8	0.685	0.617	0.649
9	0.684	0.605	0.642
10	0.687	0.627	0.655
Average	**0.692**	**0.620**	**0.654**

The results of this study show that the combination of RoBERTa and CNN provides superior results compared to the combination of RoBERTa and RNN in text classification tasks. CNN is faster in the training process and more accurate and efficient in utilizing the rich text representation of RoBERTa. This makes CNN more feasible for large-scale or real-time applications, where quick model inference is crucial without sacrificing accuracy. CNN outperforms RNN in this study because it can more efficiently recognize local patterns in short texts like news headlines. The fixed-size embeddings produced by RoBERTa are highly structured, making them well-suited for CNN's convolutional layers. RNNs tend to struggle with such high-dimensional, context-rich embeddings, leading to longer training times and lower accuracy. The implementation of 10-fold cross-validation also ensures that the model built has stable performance and good generalization. The F1-score, which balances precision and recall, is particularly important in sarcasm detection, as both false positives and false negatives can lead to significant misunderstandings in sentiment analysis. With 10-fold cross-validation, the model can avoid the overfitting. The proof is seen in Figure 6 below.

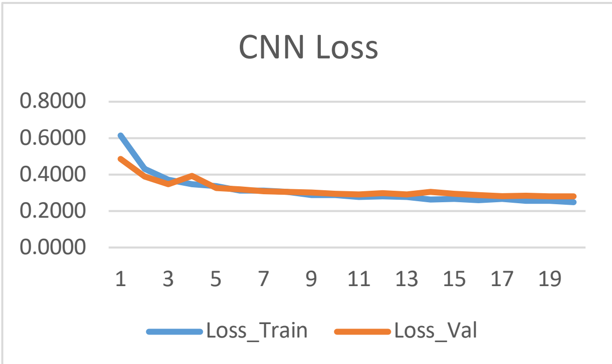

FIGURE 6. LOSS FOR MODEL CNN

Figure 6 with a low comparison of train loss and validation loss indicates that the model is not overfitting. This is also seen in the RNN model. In this research, early stops are used to avoid excessively long running times. So, the epoch of each model can be different. The loss of the RNN model is shown in Figure 7 below.

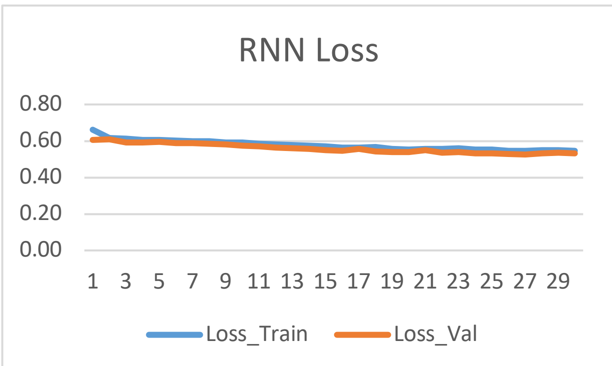

FIGURE 7. LOSS FOR MODEL RNN

5. CONCLUSIONS

This research succeeded in showing that the combination of RoBERTa and Convolutional Neural Network (CNN) provides superior results compared to the combination of RoBERTa and Recurrent Neural Network (RNN) in text classification tasks, especially in detecting sarcasm in English news headlines. The implementation using RoBERTa as a vectorizer produces vectors with fixed dimensions of 1x768, which is very effective for use in CNN models. CNN achieved superior performance with the highest average result per fold for accuracy of 0.891, precision of 0.878, recall of 0.874, f1-score of 0.876, loss of 0.260, and required a processing time of 508.1 milliseconds per epoch. In contrast to CNN, RNN has an accuracy performance of 0.711, precision of 0.692, recall of 0.620, f1-score of 0.654, and loss of 0.564. Meanwhile, the time required is quite long, namely 116,500 milliseconds per epoch. The 10-fold cross-validation evaluation method also ensures that the model has stable performance and can avoid overfitting.This research pro vides a clear demonstration of the superiority of CNN over RNN for sarcasm detection in news headlines, particularly when combined with RoBERTa embeddings. Given CNN's efficiency and accuracy, it holds significant potential for deployment in sentiment analysis systems, media monitoring tools, and AI-driven social media platforms. The paper shows that using CNN can be better than RNN in text classification but with the right transformation. Based on the results of this research, it is recommended to use a combination of RoBERTa and CNN in other text classification applications that require high speed and accuracy. For further research, it would be best to consider exploring deeper CNN and RNN architectures or alternative transformer models to refine further sarcasm detection capabilities that might improve performance. In addition, use vectorizer techniques that are more suitable for RNNs on text data.

6. ACKNOWLEDGMENT

The journal article entitled Comparison of RoBERTa CNN and RoBERTa RNN on Sarcasm Detection received research funds from the Universitas Ahmad Dahlan research and community service institute called Lembaga Penelitian dan Pengabdian Masyarakat (LPPM UAD) through the Batch 3 Internal Research Program. The contents of this article are the full responsibility of the author.

REFERENCES

[1] M. Nur Fitriyah and Y. Effendri, “BAHASA SARKASME WARGANET DALAM KOMENTAR AKUN INSTAGRAM PUAN MAHARANI DAN DPR RI,” *Bapala*, vol. 8, no. 4, pp. 112–119, 2021.

[2] A. Azis and L. Marlina, “Analysis of Sarcasm Found in Keith Alberstadt’s Comedy,” *English Language and Literature*, vol. 9, no. 2, p. 215, Aug. 2020, doi: 10.24036/ell.v9i2.7805.

[3] R. J. Kreuz and S. Glucksberg, “How to be sarcastic: The echoic reminder theory of verbal irony.,” *J Exp Psychol Gen*, vol. 118, no. 4, pp. 374–386, Dec. 1989, doi: 10.1037/0096-3445.118.4.374.

[4] P. V. Dauphin, “SARCASM IN RELATIONSHIPS.” Accessed: Oct. 07, 2023. [Online]. Available: https://ccat.sas.upenn.edu/plc/communication/valerie.htm

[5] F. Anshari and A. Hafiz, “Bahasa Sarkasme dalam Berita Olahraga - Studi Kasus bolatory.com,” *Prosiding Konferensi Nasional Komunikasi*, vol. 2, no. 1, pp. 184–196, 2018.

[6] S. Masroor and M. S. Husain, “Sarcasm Analysis Using Social Media : A Literature Review,” *IJEMR*, pp. 26–31, 2018.

[7] F. Ardiansyah and A. Kartikadewi, “Sarcasm Detection on Indonesian Politics Tweet Using Multi Labeling Method and Support Vector Machine,” *International Journal of Engineering and Techniques*, vol. 6, no. 3, Jun. 2020, doi: 10.29126/23951303/IJET-V6I3P15.

[8] C. S. Arsya and M. Elsera, “IMPLEMENTASI RANDOM FOREST DALAM MELAKUKAN KLASIFIKASI KATA SARKASME PADA MEDIA SOSIAL FACEBOOK,” *Djtechno: Jurnal Teknologi Informasi*, vol. 4, no. 1, pp. 216–223, Jul. 2023, doi: 10.46576/djtechno.v4i1.3361.

[9] D. Alita, S. Priyanta, and N. Rokhman, “Analysis of Emoticon and Sarcasm Effect on Sentiment Analysis of Indonesian Language on Twitter,” *Journal of Information Systems Engineering and Business Intelligence*, vol. 5, no. 2, p. 100, Oct. 2019, doi: 10.20473/jisebi.5.2.100-109.

[10] R. A. Bagate and R. Suguna, “Different Approaches in Sarcasm Detection: A Survey,” 2020, pp. 425–433. doi: 10.1007/978-3-030-34080-3_48.

[11] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, “CASCADE: Contextual Sarcasm Detection in Online Discussion Forums,” in *Proceedings of the 27th International Conference on Computational Linguistics*, E. M. Bender, L. Derczynski, and P. Isabelle, Eds., Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1837–1848. [Online]. Available: https://aclanthology.org/C18-1156

[12] R. Misra and P. Arora, “Sarcasm Detection using Hybrid Neural Network,” Aug. 2019, doi: 10.13140/RG.2.2.32427.39204.

[13] R. Misra and P. Arora, “Sarcasm detection using news headlines dataset,” *AI Open*, vol. 4, pp. 13–18, 2023, doi: 10.1016/j.aiopen.2023.01.001.

[14] A. C. Wijaya and I. G. A. Wibawa, “ Deteksi Sarkasme dan Ironi pada Twitter dengan Mengunakan Metode CNN,” *Jurnal Nasional Teknologi Informasi dan Aplikasnya*, vol. 1, no. 1, pp. 353–360, 2022, [Online]. Available: https://ojs.unud.ac.id/index.php/jnatia/article/view/92591

[15] N. Jaiswal, “Neural Sarcasm Detection using Conversation Context,” in *Proceedings of the Second Workshop on Figurative Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 77–82. doi: 10.18653/v1/2020.figlang-1.11.

[16] H. Yaghoobian, H. R. Arabnia, and K. Rasheed, “Sarcasm Detection: A Comparative Study,” Jul. 2021.

[17] M. D. Hilmawan, “Deteksi Sarkasme Pada Judul Berita Berbahasa Inggris Menggunakan Algoritme Bidirectional LSTM,” *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, vol. 2, no. 1, pp. 46–51, Feb. 2022, doi: 10.20895/dinda.v2i1.331.

[18] Y. Yunitasari, A. Musdholifah, and A. K. Sari, “Sarcasm Detection For Sentiment Analysis in Indonesian Tweets,” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 13, no. 1, p. 53, Jan. 2019, doi: 10.22146/ijccs.41136.

[19] O. Zalutska *et al.*, “Method for Sentiment Analysis of Ukrainian-Language Reviews in E-Commerce Using RoBERTa Neural Network,” in *International Conference on Computational Linguistics and Intelligent Systems*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258688336

[20] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” Nov. 2015.

[21] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Physica D*, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[22] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, “Understanding the Behaviors of BERT in Ranking,” Apr. 2019.

[23] S. F. Chaerul Haviana, S. Mulyono, and Badie’Ah, “The Effects of Stopwords, Stemming, and Lemmatization on Pre-trained Language Models for Text Classification: A Technical Study,” in *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2023, pp. 521–527. doi: 10.1109/EECSI59885.2023.10295797.

[24] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol Cybern*, vol. 36, no. 4, pp. 193–202, 1980, doi: 10.1007/BF00344251.

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[26] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities.,” *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982, doi: 10.1073/pnas.79.8.2554.

[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[28] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[29] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” Jul. 2019.

AUTHORS

Sheraton Pawestri

Sheraton Pawestri, received a Computer Science Bachelor’s in 2020 and a Master’s in 2023 from the Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia. Currently working as a lecturer in the Department of Informatics, Faculty of Industrial Technology Universitas Ahmad Dahlan Yogyakarta. Her current research interests are Artificial Intelligence (AI), Deep learning, and Natural Language Processing (NLP).

Murinto

Murinto, received his M.Kom from Computer Sciences Universitas Gadjah Mada Yogyakarta in 2024 and Doctor (Dr) from Computer Sciences Universitas Gadjah Mada Yogyakarta in 2021. Currently working as Associate Professor in Department of Informatics, Faculty of Industrial Technology Universitas Ahmad Dahlan Yogyakarta. He is a senior researcher in the college, and his topics of research are Image Processing, Computer Vision, and Artificial Intelligence (AI).

Muhammad Auzan

Muhammad Auzan, received a Bachelor's in Electronics and Instrumentation in 2017 and a Master's in Computer Science (2020) from the Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia. Currently pursuing a Ph.D. at the National Taiwan University of Science and Technology since 2023, he specializes in intelligence control systems, dynamic systems, and robotics. His current research interests are system dynamics, control theory, robotics, and AI development in control systems.