Journal of Information Technology Management
ISSN-OnLine: 2980-7972
University of Tehran

Mushakkal: Detecting Arabic Clickbait Using CNN with Various Optimizers

Razan Alhanaya

Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia. E-mail: razan.mo2021@gmail.com

Deem Alqarawi

Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia. E-mail: Deem.alqarawi5@gmail.com

Batool Alharbi *

*Corresponding author, Department of Information Technology and Communication, Security Forces Hospital, Dammam, Saudi Arabia .E-mail: b10082019@outlook.com

Dina M. Ibrahim

Department of Information Technology, College of Computer, Qassim University, Saudi Arabia. Department of Computers and Control Engineering, Faculty of Engineering, Tanta University, Egypt. E-mail: dina.mahmoud@f-eng.tanta.edu.eg

Abstract

The term "clickbait" refers to content specifically designed to capture readers' attention, often through misleading headlines, leading to frustration among social media users. In this study, titled "Mushakkal," which translates to "variety" in Arabic, we utilized a Convolutional Neural Network (CNN)—a deep learning approach—to detect clickbait within an Arabic dataset. We compared three optimizers: RMSprop, Adam, and Adadelta, evaluating various parameter settings to determine the most effective combination for detecting clickbait in Arabic content. Our findings revealed that the CNN model performed best when both pre-processing and Word2Vec techniques were applied. The Adam optimizer outperformed the others, achieving a Macro-F1 score of 77%. The RMSprop optimizer closely followed, attaining a Macro-F1 score of 76%. In contrast, Adadelta proved to be the least effective for classifying Arabic text.

Keywords: Clickbait Detection, Arabic Dataset, Arabic Clickbait Detection, Deep Learning, Optimizers, CNN

Journal of Information Technology Management, 2024, Vol. 16, Issue 4, pp. 64-78
Published by the University of Tehran, College of Management
doi: https://doi.org/ 10.22059/jitm.2024.99051
Article Type: Research Paper

Received: June 12, 2024
Received in revised form: July 25, 2024
Accepted: September 13, 2024
Published online: October 16, 2024

Introduction

Clickbait refers to headlines that promise information but fail to deliver, designed to entice readers into clicking links by misleading them with false or exaggerated claims. This strategy is often employed to generate financial profit or achieve other objectives from increased clicks (Chen et al., 2015). The idea of clickbait has been extensively adopted and used in many ways to boost the number of viewers for a particular website or Snapchat story as a result of the widespread usage of social media and the ease with which news can be distributed there as well as the ease with which a big number of people can be reached. They are all examples of clickbait, which is centered on deceiving the reader.

The ecosystem of Arabic digital material is not an exception to this pattern. Arabic's distinct language and culture make it more difficult to identify clickbait. Arabic is a highly inflected and morphologically rich language, which makes natural language processing (NLP) tasks, including clickbait detection, more complex compared to languages like English. Moreover, many of the existing models created for other languages are limited in their efficacy due to the lack of high-quality labeled Arabic datasets. Therefore, the development of focused methods for identifying clickbait in Arabic content is urgently needed.

Text processing problems are now best handled by deep learning, and Convolutional Neural Networks (CNNs) in particular. Deep learning models automatically extract complex patterns and representations from the data, in contrast to typical machine learning techniques that depend on features that are explicitly created. CNNs are especially good in NLP for identifying local dependencies in text like important phrases or certain word sequences that make headlines seem appealing. CNNs are quite good at classification tasks like clickbait detection because of their automatic discovery of important features. Moreover, word embeddings like Word2Vec or GloVe, which map words into high-dimensional vector spaces and capture semantic linkages and contextual information crucial for deciphering a headline's meaning, are employed by deep learning models.

To detect clickbait in Arabic text, this research presents Mushakkal, a revolutionary deep learning-based method. The Arabic word "Mushakkal," which means "diversity," refers to the range of optimization strategies used to improve the deep learning model's accuracy. The Mushakkal algorithm is trained on an Arabic dataset to identify and categorize clickbait using CNNs. To increase detection accuracy, the system uses cutting-edge methods like word embeddings and hyperparameter optimization in addition to addressing the linguistic difficulties specific to Arabic. We demonstrate our “Mushakkal” identity in Figure 1.

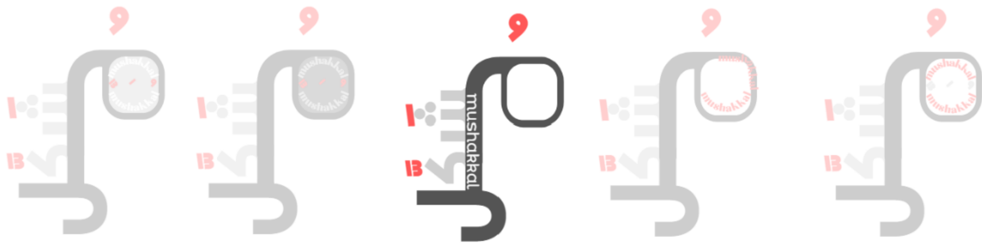

Figure 1. Mushakkal Detection Logo.

Main Contributions

- Developed an Arabic clickbait detection system using CNN trained on an Arabic dataset.
- Implemented a range of optimization techniques to improve model accuracy, under the framework of "Mushakkal."
- Validated the approach through extensive experimental evaluation, demonstrating its effectiveness in clickbait detection.

The rest of this paper is organized as follows: the Related Work section reviews prior research on clickbait detection and the application of deep learning in Arabic text. Following this, the Methodology of Mushakkal section details the development of the Mushakkal system, covering aspects such as the dataset, pre-processing steps, word embeddings, CNN architecture, and optimization techniques. In the Results and Findings section, we present experimental evaluations and a comparative analysis of the results. Finally, the Conclusion summarizes the findings and outlines potential directions for future research.

Literature Review

Both machine learning (ML) and deep learning (DL) have been used in numerous trials to detect clickbait. Regarding machine learning models, they have been employed in several studies, such as Dam et al. (2021), where the researchers investigated the use of Support Vector Machines (SVM) to categorize clickbait in Nepali news. Following testing, they compared its accuracy to that of the Random Forest (RF) model, finding that performance for both variants was nearly the same; however, SVM performed somewhat better than RF classifiers. Additionally, Ahmad et al. (2020) and Putri and Pratomo (2022) employed ML models to identify clickbait and found that applying BERT outperformed the other models, yielding the greatest results. Recently, deep neural network approaches have successfully been applied to many text classification problems, particularly in English text classification. As noted by Alsaleh and Larabi-Marie-Sainte (2021), Convolutional Neural Networks (CNN) are among the most popular models; however, they are not widely applied in Arabic text classification. Furthermore, recent studies have not achieved high classification accuracy due to parameter setting issues. To address this limitation, Alsaleh and Larabi-Marie-Sainte (2021) developed a new hybrid classification model for Arabic text, proposing a genetic algorithm-based CNN. This genetic algorithm is used to optimize the CNN parameters. Their model was tested using two large datasets with different parameters and compared with state-of-the-art studies. The results indicated that classification accuracy improved by 4% to 5%, with the best results achieved using the RMSprop optimizer among three optimizers (Adam, RMSprop, and Adadelta).

On the other hand, when we look into the picture classification world we notice that based on Albayati et al. (2024) and Sahana et al. (2024), the Adam optimizer receives high values, which implies the effectiveness of implementing the Adam optimizer in the CNN model. They compared Adam, RMSprop, Adadelta, and other optimizers to find the most accurate results for handwritten digit recognition in Albayati et al. (2024) and for unleashing human emotions in Sahana et al. (2024). Both conclude that the Adam optimizer receives high values, which made us more determined to include it in Mushakkal.

Kaur et al. (2020) proposed a two-phase model. In their suggested method, the hybrid CNNs-LSTMs model is used in the first phase to identify clickbait. This information is then fed into the second phase, where eight different types of clickbait are classified using the Biterm BTM, including Reasoning, Number, Reaction, Revealing, Shocking/Unbelievable, Hypothesis/Guess, Questionable, and Forward-Referencing. Facebook, Twitter, websites, and dataset 3 (text and picture data) were among the three datasets that were used to collect data from various sources. For Datasets 1, 2, and 3 utilizing GloVe embeddings, the accuracy obtained by the hybrid CNNs-LSTMs model is 95.8%, 89.44%, and 94.21%, respectively.

On the other side, Chawda et al. (2019) mentioned that Region-Based Convolutional Neural Networks (RCNNs) are used to improve categorization. They used the left and right context capture with RCNNs. They also used the RCNNs model to test LSTMs and GRUs. RCNNs captured more context in the system architecture than SVM. Their proposed system uses RCNN with GRU and LSTM for clickbait classification. RCNN + GRU with Word2Vec provided a remarkable accuracy of 0.9776 for clickbait classification.

Rajapaksha (2020) discussed the use of a fusion model for clickbait detection. They provide a clickbait detection algorithm that incorporates information from several texts (post text, news headline, and news item content) to identify the extent to which news material contains clickbait. A fusion model that includes elements like sentiment score, similarity score, and topical similarity of text is the suggested methodology for recognizing clickbait. They made use of an English dataset compiled from Facebook, Fox News, and the BBC. To improve classification outcomes, they tested a variety of architectural alterations, including training and testing with various layers of CNNs, LSTMs, and a combination of both. The three CNN layers combined with a Bidirectional LSTM layer with Word2Vec produced the model that performed the best.

Zheng et al. (2018) suggested a Clickbait Convolutional Neural Network (CBCCN) that considers both the general and unique characteristics of various article genres. The preprocessed headlines are used to train the CBCCN and include segmentation, stop word filtering, and part-of-speech filtering. Word2Vec models and a CNNs model make up the CBCCN model. The input heading is preprocessed as in training on the prediction side. CNN interprets the input text from a variety of angles. Each word in the model has two meanings: a general meaning and a type-related meaning that may be used for various NLP tasks. CBCCN is employed in tasks like computing language similarity, extracting relations and events, designating entities, and recognizing word parts. Their suggested model, CBCCN, has an accuracy of 80%. The maximum headline length is one CBCCN restriction, which might result in information loss in lengthy headlines. A pre-trained Word2Vec model that might help CBCCN better comprehend word meanings is not present.

In addition to CNNs, Shaikh and Annappanavar (2020) applied a CNN algorithm to offer a technique to identify clickbait on online social media. The tactic used is highly focused on textual elements. Word2Vec transforms the data. The results have a high accuracy of 82%, outperforming other ML techniques. The Random Forest was used for comparative analysis.

Arabic clickbait headline detection is a topic on which little study has been done. Al-Sarem et al. (2021) constructed the first Arabic dataset of clickbait headline news to more accurately identify clickbait news on social networks, and it also provided a multi-feature-based approach. The three main parts of the proposed approach are data collection, data preparation, and phases of ML model training and testing. After pre-processing, the dataset that was gathered contained 54,893 Arabic news articles. Of these, 23,981 had headlines that were clickbait. The most important features of this preprocessed dataset were selected using the ANOVA F-test. Then, several ML methods were combined with methods for hyper-parameter tuning to ensure that the best settings were found. Once the ML models had been evaluated, the results showed that the Support Vector Machine (SVM) with the top 10% of ANOVA F-test features (User-based Features [UFs] and Content-based Features [CFs]) had the highest accuracy, with a score of 92.16%.

In addition, ML models were used for an Arabic news dataset by Bsoul et al. (2022). With the help of ML models, this dataset aims to automatically categorize news headlines as "Clickbait" or "Not Clickbait." In total, 3,235 records were acquired, of which 2,652 news reports were deemed to be "Not Clickbait" and 583 to be "Clickbait." The dataset's distribution of classes was unbalanced, with a 1:4 ratio between "Clickbait" and "Not Clickbait." The evaluation was therefore conducted on the original dataset, oversampled dataset, and undersampled dataset to obtain the final dataset required for this study. The use of seven distinct ML models was applied to the three training datasets. By performing a 10-fold cross-validation on the training dataset, model hyperparameters were adjusted. The test dataset findings used to evaluate these models produced a Macro F1-Score of approximately

0.81 for the examined feature combinations, effectively proving the dataset's usefulness. The F1-Score value did not increase as a result of the resampling, according to the findings.

Methodology

Figure 2 illustrates the general methodology for detecting Clickbait using CNN with different optimizers, using Mushakkal's framework for detection. Throughout this section, we review each step in detail and provide examples of both anticipated results and justifications.

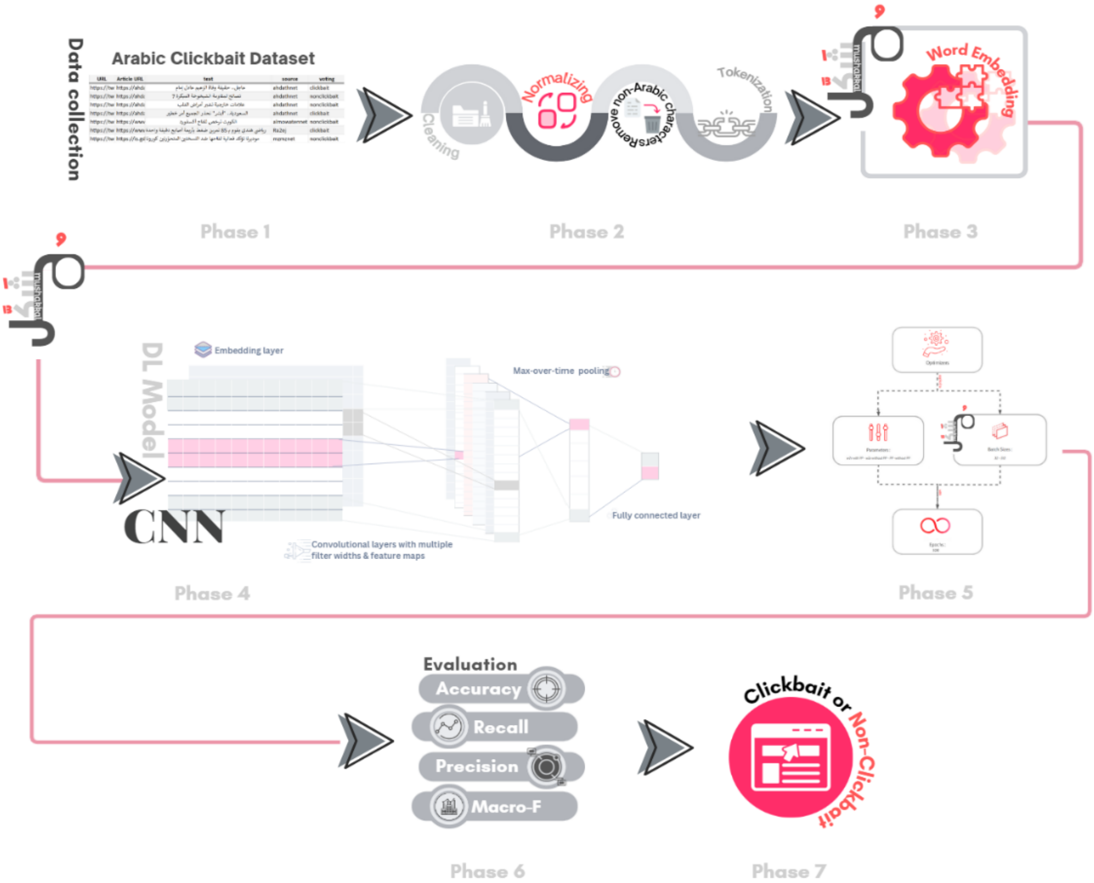

Figure 2. Our proposed Mushakkal Detection Framework

Dataset

Collecting data to train the deep learning (DL) models is the first step in the pipeline for DL models. Training data determines how accurately DL systems can predict the future. The dataset collected by Al-Sarem et al. (2021) is the dataset we're going to utilize because, as far as we know, no other dataset for Arabic clickbait is readily available. This dataset contains 10,564 rows in Arabic that were obtained from Twitter; 7,735 of those rows were classified as non-clickbait, while 2,829 were classified as clickbait, making up an unbalanced dataset. The dataset is composed of five columns (features), as seen in Figure 3, including URL, article URL, content, source, and voting. A description of the characteristics of the dataset can be found in Table 1.

URL	Article URL	text	source	voting
https://tw	https://ahda	عاجل.. حقيقة وفاة الزعيم عادل إمام	ahdathnet	clickbait
https://tw	https://ahda	نصائح لمقاومة الشيخوخة المبكرة 7	ahdathnet	nonclickbait
https://tw	https://ahda	علامات خارجية تشير أمراض القلب	ahdathnet	nonclickbait
https://tw	https://ahda	السعودية.. "أبشر" تحذر الجميع أمر خطير	ahdathnet	clickbait
https://tw	https://wwv	الكويت ترخص للقاح أكسفورد	almowatennet	nonclickbait
https://tw	https://wwv	رياضي هندي يقوم بـ 85 تمرين ضغط بأربعة أصابع دقيقة واحدة	Ra2ej	clickbait
https://tw	https://is.gd	موديرنا تؤكد فعالية لقاحها ضد النسختين المتحوّرتين كورونا	mzmznet	nonclickbait

Figure 3. Sample of Arabic Clickbait Dataset

Table 1. The Description of the Arabic Clickbait Dataset Features

Features	Description
URL	Tweet link
Article URL	Link to the article to which the Tweet refers
text	Tweet content
source	The source who posted the tweet
voting	Labeled of Tweet” Clickbait or non-Clickbait”

Pre-processing

Real-world raw data are frequently incoherent, devoid of identifiable patterns or trends, and incomplete. They most likely contain numerous errors as well. As a result, they go through pre-processing after being gathered to produce a format that is consistent with the DL model. Before classification, preprocessing of the data is required. Before using the data, Mushakkal's model will clean, normalize, tokenize, and eliminate non-Arabic letters.

- Cleanup tasks include changing hashtags to words and removing E-mail addresses, URLs, mentions, punctuation, repetitive characters, stop words, and emojis.
- Normalizing entails changing the hamaza to near letters, deleting diacritics and tatweel characters.
- Removing any non-Arabic characters, including numbers.
- Tokenization entails disassembling the sentence into tokens, which are groups of words.

Because Mushakkal values the Arabic language, it is crucial that libraries with expertise in Arabic natural language processing are used throughout the pre-processing of Arabic data. Therefore, Mushakkal made use of the Pyarabic library (Zerrouki, n.d.) and the Ruqia library (Safi, n.d.) in the pre-processing step.

Word Embedding

Word embedding refers to methods for mapping words or phrases to vectors of real numbers. Using word embedding methods, words are represented as continuous vectors in a small-dimensional space. The most well-liked technique for teaching word embeddings to an external feed-forward neural network is probably the word2vec technology. For Mushakkal's model, we employed the Skip-gram word embedding technique from AraVec (Bakrianoo, n.d.), an open-source Arabic word2vec project.

CNN Model

Due to the processing of text data, CNN is used in a single dimension. The model is given non-linearity by using a variety of layers and activation functions (Ahmed et al., 2023). The CNN model that Mushakkal developed has several layers, including the Fully Connected layer, the GlobalMaxPooling layer, the Embedding layer, and the Convolutional layer with 128 filter sizes and 5 window sizes. Additionally, a Dropout layer with a probability of 0.1 and the activation function for Rectified Linear Units (ReLU) is applied. Because Mushakkal is attempting to categorize input into binary classes, the Sigmoid activation function is utilized for the final layer, which is the output layer.

Hyperparameters and Training

By varying the model parameters, it is possible to change the results significantly. Consequently, Mushakkal tested CNN by using a variety of optimizers and a variety of parameters, as illustrated in Figure 4. The parameters used were three different optimizers with different batch sizes (32,512) to find the optimal model for Mushakkal. Additionally, Mushakkal will experiment with and without these parameters with Word2vec (W2V) and pre-processing (PP) using 500 Epochs. Using random splitting, Mushakkal will divide the dataset into 20% for test data and 80% for training data. In addition, we will use binary-cross entropy as a loss function for binary classification.

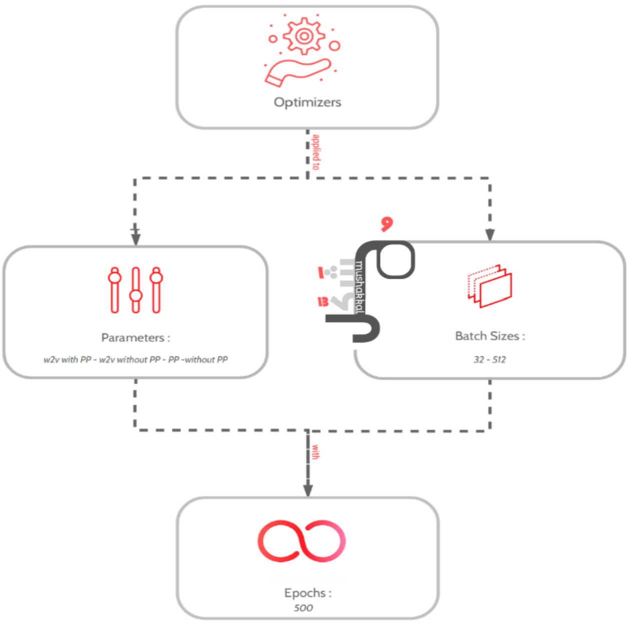

Figure 4. Training Parameters and Hyperparameters to Optimize Mushakkal's Model

Optimizers

According to our knowledge, the Adam optimizer is the leader in text classification problems, but as mentioned by Alsaleh and Larabi-Marie-Sainte (2021), RMSprop outperforms Adam and Adadelta in Arabic text classification. Therefore, in our proposal, Mushakkal's CNN model with each type of optimizer will be tested using the different parameters that we mentioned above. Based on Ruder (2020), Table 2 is a simplified definition of the optimizers we use in this proposal.

Table 2. Description of Optimizers used in Mushakkal Methodology

Optimizer	Description
RMSprop	Is a gradient-based optimization technique used in training neural networks? It uses an adaptive learning rate instead of treating the learning rate as a hyperparameter.
Adam	Is a first-order-gradient-based algorithm of stochastic objective functions, based on adaptive estimates of lower-order moments?
Adadelta	Is a stochastic gradient descent method that is based on adaptive learning rate per dimension to address the continual decay of learning rates throughout training and the need for a manually selected global learning rate

Evaluation

Performance metrics are employed to evaluate the efficacy or quality of the model. These performance indicators let us understand how well the CNN model processed the given data. Performance indicators help assess how well a DL model generalizes on fresh or unstudied data. In our suggested model, we will evaluate the performance of our selected set of techniques using measures including accuracy, precision, recall, and macro F1-score. It's vital to consider the concept of the confusion matrix to comprehend the necessity of employing the performance measurements used for this study. A confusion matrix for a prediction task lists instances of both correct and wrong predictions. Based on Luque et al. (2019), Figure 5 visually depicts a confusion matrix.

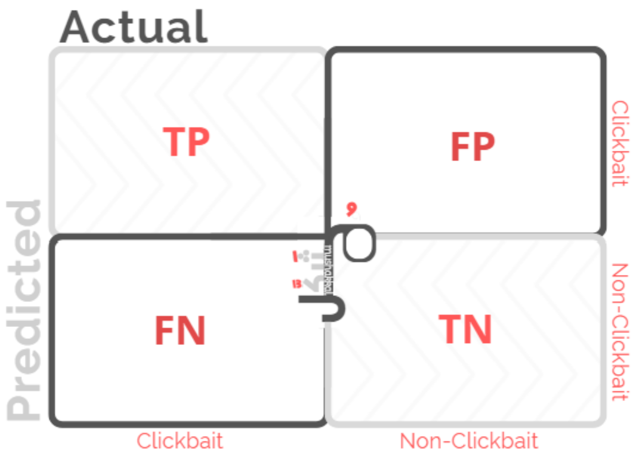

Figure 5. Confusion Matrix for Mushakkal

True Positives (TP): The cases in which the model predicted Clickbait and the actual output was also Clickbait.

True Negatives (TN): The cases in which the model predicted non-Clickbait and the actual output was non-Clickbait.

False Positives (FP): The cases in which the model predicted Clickbait and the actual output was non-Clickbait.

False Negatives (FN): The cases in which the model predicted non-clickbait and the actual output was Clickbait.

The metrics that we will use to evaluate the model are shown in Table 3 based on Czakon (2023) and Allwright (2023).

Table 3. Evaluation Metrics Used in Mushakkal

Metric	Formula	Definition
Accuracy	$Eq.(1) \ = \dfrac{TP+TN}{TP+TN+FP+FN}$	The formula used to determine the percentage of all accurate predictions.
Recall	$Eq.(2) \ = \dfrac{TP}{TP+FN}$	The formula used to determine the percentage of true positive predictions among all possible positive predictions a model may have made.
Precision	$Eq.(3) \ = \dfrac{TP}{TP+Fp}$	The formula used to determine the proportion of accurately predicted actual cases to all positively predicted instances.
Macro-f	$Eq.(4) \ = \dfrac{sum(f1\ scores)}{Number\ of\ classes}$	The formula used calculates the arithmetic mean of the individual class-related f1 score.

Clickbait or Non-Clickbait

After applying the CNN model to the processed data, extracting characteristics and evaluating the results of the Mushakkal model. In this phase, the model will determine whether the headline is Clickbait or non-clickbait according to the final result.

Results

The outcomes of CNN experiments employing optimizers with various parameter settings are displayed in Table 4. Mushakkal will utilize Macro-f as a fair comparison between the results because the dataset we used was uneven. The results from Table 3 were used to create Figure 6 in order to determine the best CNN model by choosing the suitable optimizer and the proper parameters to attain a high Macro-f value.

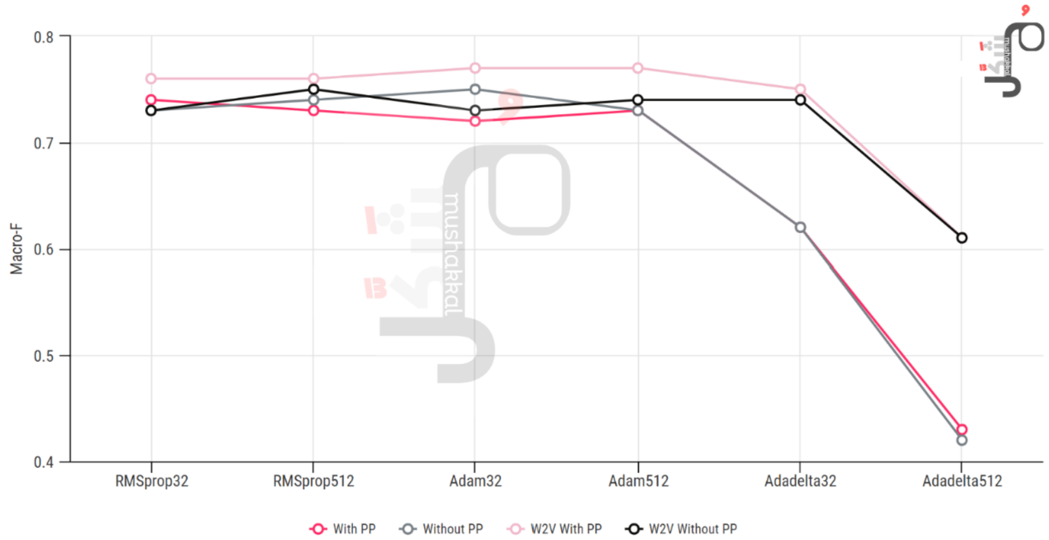

Figure 6. Mushakkal Results and Findings with Optimizers Performance in CNN Model

Table 4. CNN results with different optimizers

Optimizer Type	Batch Size	Scale	With PP	Without PP	W2V With PP	W2V Without PP
RMSprop	32	Accuracy	0.79	0.77	0.83	0.8
		Recall	0.83	0.79	0.93	0.92
		Precision	0.89	0.89	0.85	0.83
		Macro-F	0.74	0.73	0.76	0.73
	512	Accuracy	0.8	0.81	0.82	0.81
		Recall	0.87	0.89	0.92	0.89
		Precision	0.86	0.86	0.85	0.85
		Macro-F	0.73	0.74	0.76	0.75
Adam	32	Accuracy	0.79	0.8	0.83	0.81
		Recall	0.85	0.84	0.92	0.91
		Precision	0.87	0.88	0.85	0.83
		Macro-F	0.72	0.75	**0.77**	0.73
	512	Accuracy	0.8	0.8	0.82	0.81
		Recall	0.87	0.89	0.9	0.91
		Precision	0.86	0.85	0.86	0.84
		Macro-F	0.73	0.73	**0.77**	0.74
Adadelta	32	Accuracy	0.8	0.79	0.82	0.82
		Recall	0.99	0.99	0.94	0.94
		Precision	0.79	0.78	0.84	0.83
		Macro-F	0.62	0.62	0.75	0.74
	512	Accuracy	0.75	0.74 1 0.74 0.42	0.78	0.78
		Recall	1		0.99	0.99
		Precision	0.75		0.77	0.77
		Macro-F	0.43		0.61	0.61

As we can see, Adam optimizer received the highest value of Macro-F equal to 0.77 by using Word2vec with pre-processing in both batch sizes 32 and 512. The performance of the model during training using 32 batch size and 512 batch size are illustrated in Figure 7 and Figure 8, respectively.

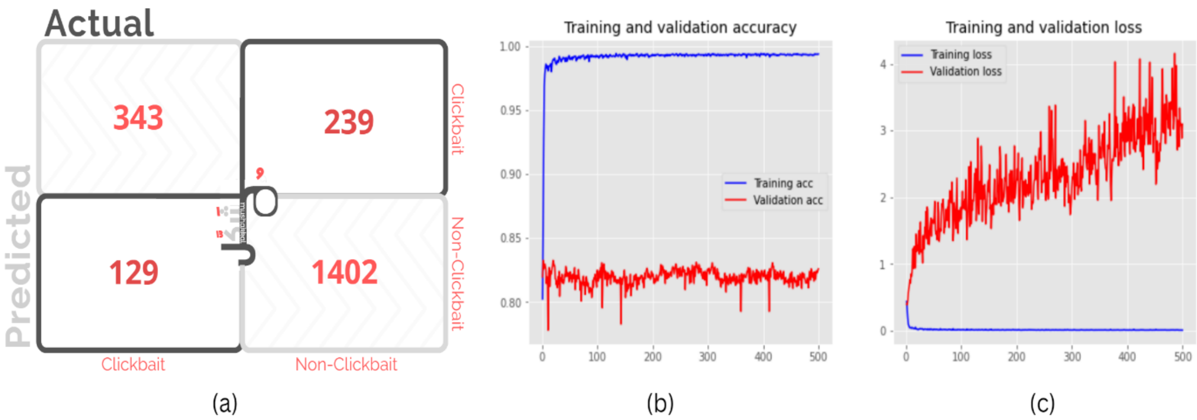

Figure 7. CNN Performance During Training Using Adam Optimizer with Batch 32, (a) Confusion matrix (b) The training and validation accuracy during training (c) The training and validation loss during training

According to the Mushakkal model's confusion matrix, which shows the proportion of correct and incorrect classifications, as shown in the first section of Figure 8, 1745 out of 2113 observations were properly predicted by the model. The validation accuracy of the model during training ranges between 0.77 and 0.83, as seen in the second section of the same figure. The model's validation loss throughout training is shown in the third section. As shown, the validation loss value initially started at 0.5, declined to 0.4 in the first epochs, and then started to move up.

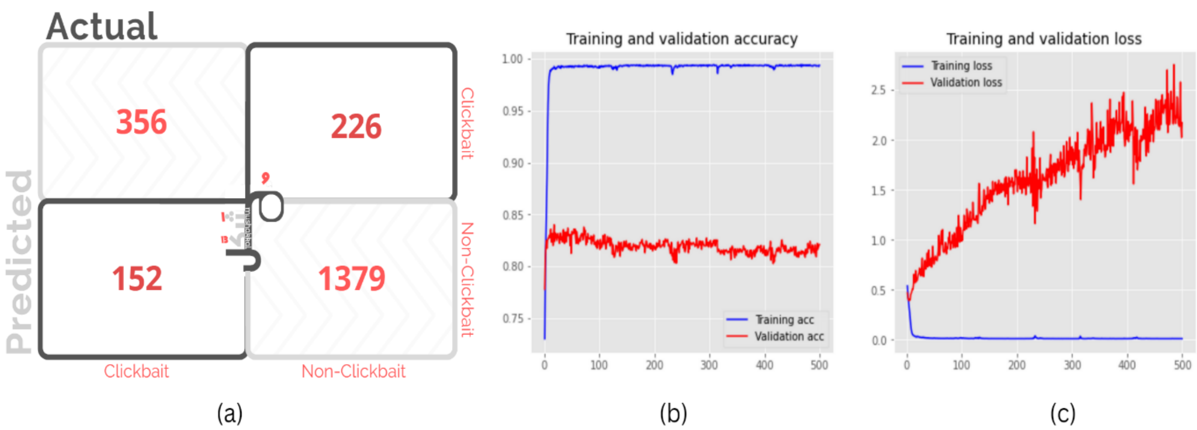

Figure 8. CNN Performance During Training Using Adam Optimizer with Batch 512, (a) Confusion matrix (b) The training and validation accuracy during training (c) The training and validation loss during training

Figure 8 displays the CNN performance when the training period's batch size 512 was used with the Adam optimizer. Furthermore, we note that the model findings were very similar to the model when a batch size of 32 was applied.

Conclusion

In the Mushakkal proposal, we conducted several experiments on the Arabic clickbait dataset using the CNN model. Using various optimizers (RMSprop, Adam, Adadelta) and various parameter settings (different batch sizes, with pre-processing, without pre-processing, word2vec with pre-processing, and word2vec without pre-processing), trials were carried out to determine the best model for CNN to categorize Arabic texts. The CNN model showed its best results when applying both pre-processing and Word2vec. The optimizer Adam also outperformed the rest of the optimizers, obtaining a Macro-F value equal to 77%. In second place comes the optimized RMSprop, which obtained a Macro-F value of 76%. Adadelta was the worst optimizer in Arabic text classification. Future work will involve creating a hybrid model and training it on the Arabic clickbait dataset using the settings that produce the best results.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Ahmad, I., Alqarni, M. A., Almazroi, A. A., & Tariq, A. (2020). Experimental evaluation of clickbait detection using machine learning models. *Intelligent Automation & Soft Computing, 26*(4), 1335–1344.

Ahmed, S. F., Alam, M. S. B., Hassan, M., Rozbu, M. R., Ishtiak, T., Rafa, N., Mofijur, M., Ali, A. B. M. S., & Gandomi, A. H. (2023). Deep learning modelling techniques: Current progress, applications, advantages, and challenges. *Artificial Intelligence Review, 13521–13617*(11). https://doi.org/10.1007/s10462-023-10466-8

Albayati, A. Q., Altaie, S. A. J., Al-Obaydy, W. N. I., & Alkhalid, F. F. (2024). Performance analysis of optimization algorithms for convolutional neural network-based handwritten digit recognition. *IAES International Journal of Artificial Intelligence, 13*(1). https://doi.org/10.11591/ijai.v13.i1.pp563-571

Allwright, S. (2023). Metrics for imbalanced data (simply explained). In *Stephen Allwright*. https://stephenallwright.com/imbalanced-data-metric/

Alsaleh, D., & Larabi-Marie-Sainte, S. (2021). Arabic text classification using convolutional neural network and genetic algorithms. *IEEE Access, 9*, 91670–91685. https://doi.org/10.1109/ACCESS.2021.3091376

Al-Sarem, M., Saeed, F., Al-Mekhlafi, Z. G., Mohammed, B. A., Hadwan, M., Al-Hadhrami, T., Alshammari, M. T., Alreshidi, A., & Alshammari, T. S. (2021). An improved multiple features and machine learning-based approach for detecting clickbait news on social networks. *Applied Sciences, 11*(20), 9487. https://doi.org/10.3390/app11209487

Bakrianoo. (n.d.). GitHub - bakrianoo/aravec: AraVec is a pre-trained distributed word representation (word embedding) open source project which aims to provide the Arabic NLP research community with free to use and powerful word embedding models. In *GitHub*. https://github.com/bakrianoo/aravec

Bsoul, M. A., Qusef, A., & Abu-Soud, S. (2022). Building an optimal dataset for Arabic fake news detection. *Procedia Computer Science, 201*, 665–672. https://doi.org/10.1016/j.procs.2022.03.088

Chawda, S., Patil, A., Singh, A., & Save, A. (2019). A novel approach for clickbait detection. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1318–1321). https://doi.org/10.1109/ICOEI.2019.8862781

Chen, Y., Conroy, N. J., & Rubin, V. L. (2015). Misleading online content: Recognizing clickbait as “false news.” In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection* (pp. 15–19). https://doi.org/10.1145/2823465.2823467

Czakon, J. (2023). Evaluation metrics for binary classification (and when to use them). In *Neptune.ai*. https://neptune.ai/blog/evaluation-metrics-binary-classification

Dam, S. R., Panday, S. P., & Thapa, T. B. (2021). Detecting clickbaits on Nepali news using SVM and RF. In *Proceedings of 9th IOE Graduate Conference, 9*, 140–146.

Kaur, S., Kumar, P., & Kumaraguru, P. (2020). Detecting clickbaits using two-phase hybrid CNN-LSTM biterm model. *Expert Systems with Applications, 151*, 113350. https://doi.org/10.1016/j.eswa.2020.113350

Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition, 91*, 216–231. https://doi.org/10.1016/j.patcog.2019.02.023

Putri, D. U. K., & Pratomo, D. N. (2022). Clickbait detection of Indonesian news headlines using fine-tune bidirectional encoder representations from transformers (BERT). *Inform: Jurnal Ilmiah Bidang Teknologi Informasi Dan Komunikasi, 7*(2), 162–168.

Rajapaksha, R. W. V. P. C. (2020). Clickbait detection using multimodel fusion and transfer learning [Institut polytechnique de Paris]. https://doi.org/10.1016/j.eswa.2023.120537

Ruder, S. (2020). An overview of gradient descent optimization algorithms. In *Ruder.io*. https://www.ruder.io/optimizing-gradient-descent/

Safi, R. (n.d.). Ruqyai/Ruqia-Library: Python library used for Arabic NLP to process, prepare and clean the Arabic text. In *GitHub*. https://github.com/Ruqyai/Ruqia-Library

Sahana, M., Umesh, P., Kodipalli, A., & Rao, T. (2024). EmoCNN: Unleashing human emotions with customized CNN using different optimizers. *Procedia Computer Science, 235*, 1310–1318. https://doi.org/10.1016/J.PROCS.2024.04.124

Shaikh, M. A., & Annappanavar, S. (2020). A comparative approach for clickbait detection using deep learning. In *2020 IEEE Bombay Section Signature Conference (IBSSC)* (pp. 21–24). https://doi.org/10.1109/IBSSC51096.2020.9332172

Zerrouki, T. (n.d.). Pyarabic: An Arabic language library for Python. https://pypi.org/project/PyArabic/#description

Zheng, H.-T., Chen, J.-Y., Yao, X., Sangaiah, A. K., Jiang, Y., & Zhao, C.-Z. (2018). Clickbait convolutional neural network. *Symmetry, 10*(5), 138. https://doi.org/10.3390/sym10050138

Bibliographic information of this paper for citing:

Alhanaya, Razan; Alqarawi, Deem; Alharbi, Batool & Ibrahim, Dina M. (2024). Mushakkal: Detecting Arabic Clickbait Using CNN with Various Optimizers. *Journal of Information Technology Management,* 16 (4), 64-78. https://doi.org/ 10.22059/jitm.2024.99051