Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 10, No. 04, December 2024.

NOVEL MULTI-CHANNEL DEEP LEARNING MODEL FOR ARABIC NEWS CLASSIFICATION

Imad Jamaleddyn[1], Rachid El Ayachi[1] and Mohamed Biniz[2]

(Received: 4-Jul.-2024, Revised: 11-Aug.-2024 and 26-Aug.-2024, Accepted: 31-Aug.-2024)

ABSTRACT

In the era of digital journalism, the classification of Arabic news presents a significant challenge due to the complex nature of the language and the vast diversity of content. This study introduces a novel multi-channel deep-learning model, CLGNet, designed to enhance the accuracy of Arabic-news categorization. By integrating Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), the proposed model effectively processes and classifies Arabic-text data. Extensive experiments were conducted on multiple datasets, including CNN, BBC and OSAC, where the model achieved outstanding accuracy and robustness, outperforming existing methods. The findings underscore the effectiveness of our hybrid model in addressing the challenges of Arabic-text classification and its potential applications in automated news categorization systems.

KEYWORDS

Convolutional neural networks (CNNs), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs), Word embedding, Arabic-text classification.

1. INTRODUCTION

In recent times, digital data has become indispensable in every aspect of modern life, significantly impacting the quality of organizations and financial environments. The focus on data and information quality has intensified in both business and academic realms [1]. The news industry holds particular significance as a purveyor of quality information, essential for the success of any news organization [2].

However, like many other industries, journalism has been profoundly affected by the digital revolution, with online news consumption surpassing print media in recent years. Blogs, social media and online-only newspapers have become increasingly important sources of news, particularly for younger demographics. According to the latest Digital News Report from the Reuters Institute, while a percentage of 32% of the Flemish population still relies on print media, a staggering 78% now obtains news online. Online news consumption has remained stable over the past five years, while newspaper consumption has experienced a significant decline [3].

Consequently, scientists and researchers have endeavored to automate the news-classification process, prompting the development of strategies to identify similarities among news articles and categorize them automatically. This involves categorizing unknown texts to extract their meaning. Formally, this task involves assigning categories to a set of texts represented by x = x1, x2..., xn, labeled with values from a set of categories represented by l = l1, l2..., ln. A classification model is trained using a training dataset, establishing a relationship between features and class labels. The unidentified category of a text can then be determined using this trained model. However, manually completing this task for unannotated documents is labor-intensive and time-consuming. Text classification has found success in numerous fields, including sentiment analysis, information retrieval, spam-mail detection and more [4].

Deep-learning algorithms, particularly in Natural Language Processing (NLP) and text categorization, have gained prominence for their ability to enhance efficiency in maintenance and refurbishment projects. These algorithms, such as Artificial Neural Networks (ANNs), Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep Belief

1. I. Jamaleddyn and R. El Ayachi are with Department of Computer Science, Faculty of Sciences and Technics, Moulay Slimane University, Beni Mellal, Morocco. Emails: i.jamaleddyn@gmail.com and rachid.elayachi@usms.ma
2. M. Biniz is with Department of Mathematics and Computer Science, Polydisciplinary Faculty, Moulay Slimane University, Beni Mellal, Morocco. Email: mohamedbiniz@gmail.com

Networks (DBN) and Long Short-Term Memory (LSTM), employ multi-stage, non-linear processing models to abstractly represent data. Recent advancements in text processing, machine learning, training datasets and processing power have bolstered the field's progress [5].

Given the limited resources for Arabic-text classification and the burgeoning amount of Arabic data on the internet, there's a pressing need for systems to manage this significant volume of data. This work proposes a hybrid approach to automate Arabic-news categorization, integrating LSTM, CNNs and gated recurrent units (GRUs) to train a classifier across various news datasets. Natural-language processing (NLP) techniques are employed for processing Arabic texts. The subsequent sections of this article detail the analysis of text classification using deep-learning models, the comprehensive strategy and methodology, discussion and experimental results on popular datasets and finally, conclusions and future-research perspectives.

2. LITERATURE REVIEW

Machine learning and deep learning have significantly advanced natural-language processing (NLP) across various applications, including machine translation [6]-[7], entity resolution [8]-[10], sentiment analysis [11]-[12], question-answering systems [13]-14] and Arabic-news classification [15]-[16].

In recent years, the field of Arabic-news classification has seen considerable advancements, yet it remains fraught with challenges that impede the efficacy of these systems. The primary issue at the heart of this research is the inherent complexity of the Arabic language, characterized by its rich morphology, diverse dialects and unstructured nature. This linguistic complexity poses significant hurdles in natural-language processing (NLP) [17], particularly in the domain of feature extraction and model training. Additionally, there is a notable research gap concerning the evaluation and enhancement of Arabic-specific models, in the context of fake-news detection—a critical area given the rise of misinformation on social-media platforms. The following literature review delves into the current methodologies and innovations aimed at overcoming these challenges, exploring various approaches to feature augmentation, supervised-learning and deep-learning techniques in Arabic NLP.

To address feature sparsity, Zhang et al. proposed augmenting features in non-negative matrix factorization with term (T) and word (W) sets [18]. They introduced clustering indicators to enhance the text-word clustering process, achieving significant improvements over Word2vec and Character-level CNN in testing on diverse datasets, including Twitter sports.

Ameur et al. explored supervised-learning approaches for Arabic-text categorization, integrating static, dynamic and fine-tuned word embeddings with RNNs and CNNs [19]. Their models, tested on the OSAC dataset, demonstrated remarkable effectiveness and performance enhancements. By amalgamating Convolutional Neural Networks and Bidirectional General Recurrent Units, their methods significantly out- performed standard CNN and RNN models, boosting the F-score for Arabic text categorization by 98.61 percentage points.

Bdeir and Ibrahim delved into Arabic-tweet classification using two primary deep-learning approaches, CNN and RNN approaches [20]. Leveraging the Twitter API, they amassed 160,870 Arabic tweets spanning various topics, such as criminal accidents, entertainment, sports and technology. The dataset was divided into 90% (144K tweets) for training and validation and 10% (16K tweets) for testing. Despite feature sparsity in short texts, they found that all deep-learning models performed comparably, achieving a macro-F1 accuracy of 90.1%. CNN, RNN-LSTM and RNN-GRU approaches yielded results ranging from 92.71% to 92.95%.

Hassanein et al. [21] presented a model specifically designed to enhance feature selection for Arabic text classification. They address the challenge posed by the unstructured nature of Arabic text, which complicates machine processing. The authors proposed a multi-step feature-selection approach utilizing the Al-Khaleej-2004 corpus, which encompasses a broad range of news categories. Their methodology begins with pre-processing to extract highly weighted terms that represent the documents' content. This is followed by several steps aimed at refining the feature set. The effectiveness of their feature selection method is evaluated using four classifiers: Naïve Bayes (NB), Decision Tree, CART and KNN classifiers. The study, conducted using WEKA and MATLAB, compares the classifiers based on precision, recall, F-measure and accuracy. Among the classifiers tested, CART classifier demonstrated the best performance, while KNN classifier was found to be the

least effective.

Al Qadi et al. [22] presented a scalable approach for automatically tagging Arabic-news articles using shallow learning techniques. The study focused on the creation and utilization of two extensive datasets derived from various Arabic-news portals. The first dataset comprises 90,000 single-labeled articles spanning four domains: Business, Middle East, Technology and Sports. The second, larger dataset contains over 290,000 articles with multiple tags. These datasets have been made freely available to the Arabic computational linguistics research community, facilitating further research in the field. To validate the effectiveness of the datasets, the authors implemented ten different shallow learning classifiers. Additionally, an ensemble model was developed to combine the top-performing classifiers using a majority-voting mechanism. The classifiers demonstrated strong performance on the first dataset, with accuracy ranging from 87.7% (AdaBoost) to 97.9% (SVM). Analysis of the misclassified articles highlighted the limitations of single-label classification and underscored the importance of adopting a multi-label approach for improved accuracy.

Tahseen et al. [23] proposed the use of deep-learning techniques for detecting Arabic fake news, utilizing a dataset called AraNews, which includes articles from diverse fields, such as politics, economy, culture and sports. In their study, the authors introduced a Hybrid Deep Neural Network designed to enhance detection accuracy. This network combines the strengths of Text-Convolutional Neural Networks (Text-CNNs) and Long Short-Term Memory (LSTM) architectures. Specifically, Text-CNN is employed to extract relevant features, while LSTM handles the long-term dependencies within the text sequences. The hybrid model's performance was evaluated against the individual performances of Text-CNN and LSTM architectures. The results demonstrated that the Hybrid Deep Neural Network achieved superior accuracy, with a score of 0.914, compared to 0.859 for Text-CNN and 0.878 for LSTM.

In recent years, the detection of fake news has become increasingly important due to the rapid spread of misinformation on social-media platforms. While significant research has focused on English-language fake-news detection, there has been a notable gap in addressing this issue for the Arabic language. A significant contribution to filling this gap was by developing a large and diverse Arabic fake news dataset and employing advanced transformer-based models for classification [24], which utilize eight state-of- the-art Arabic contextualized embedding models, including AraBERT and QaribBERT, to achieve a remarkable accuracy exceeding 98%. This work not only highlights the challenges inherent in Arabic natural-language processing, such as variations in dialects and the complex structure of the language, but also demonstrates the effectiveness of transformer models in tackling these challenges. The study provides a comprehensive comparison with other fake news detection systems, underscoring the robustness of these models in accurately identifying fake news in Arabic. This research serves as a foundational reference for subsequent studies aiming to improve the accuracy and reliability of fake-news detection in non-English languages, particularly in the context of Arabic.

Despite the proliferation of studies focusing on English-language data, there is a conspicuous research gap concerning the evaluation of Arabic BERT models in the context of fake-news detection. To addresses this gap, rigorously evaluating and comparing the performance of various Arabic BERT models have been carried out by using the recently published CT23-dataset [24], which comprises a diverse array of Arabic tweets. This research is pivotal in advancing the understanding of how Arabic BERT models can be effectively leveraged to combat misinformation in the Arabic language. The study thoroughly assesses the performance of five prominent models—"Arabic Base BERT," "AraBERTV2.0," "CamelBERT MSA," "ArBERT," and "MarBERT"—revealing that "AraBERTV2.0" outperforms the others with a remarkable accuracy rate of 96%. These findings not only highlight the potential of Arabic BERT models in addressing the challenge of fake-news detection, but also provide valuable insights into the disparities among existing models, offering pathways to further enhance the precision of fake-news detection in Arabic. This work contributes significantly to the broader discourse on fake-news detection and underscores the necessity of continuous evaluation and adaptation of detection methods to keep pace with the evolving tactics of misinformation.

3. HYBRID APPROACH ARCHITECTURE

Designing an effective classifier entails a series of crucial steps, including text pre-processing and model training using deep-learning algorithms. The pre-processing of texts involves employing various approaches from the field of natural language processing (NLP), while the training phase focuses on fitting the model to pre-processed texts. In this study, a combination of NLP techniques was utilized to clean the datasets and extract the features. Subsequently, Convolutional Neural Networks (CNNs) [25], Gated Recurrent Units (GRUs) [26] and Long Short-Term Memory (LSTM) [26] were chosen for training the model.

3.1 Approaches and Proposed Model

The architecture of our hybrid approach is depicted in Figure 1, focusing on the development of a categorization system capable of classifying unclassified texts. A pre-processed sentence serves as the input to the model. Subsequently, the word vector is obtained through the process of word embedding. The model comprises several key components, including the pre-processing phase and the training phase, each of which will be detailed in the following paragraphs.

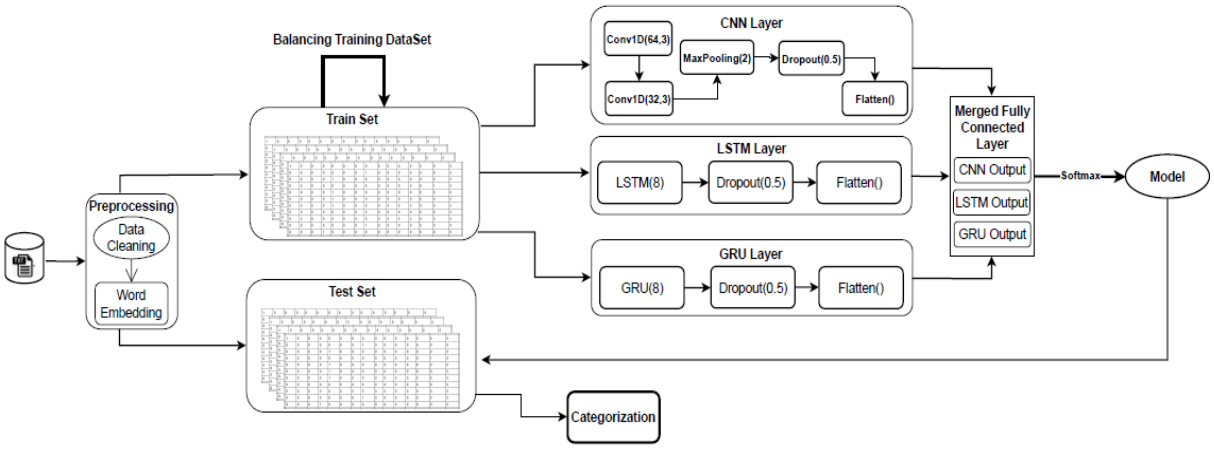

Figure 1. The hybrid architecture of the proposed multi-channel model CLGNet.

3.2 Derivation Patterns and Pre-processing

In Arabic, the majority of terms are derived from roots using standard patterns, enabling the derivation of various word forms, including verbs, adjectives, nouns and adverbs. The pattern associated with a word determines its properties, such as number (plural | singular), gender (feminine | masculine) and tense (present, past and future) [27].

Given the challenges of utilizing Arabic text directly in the training step to construct the model, a pre-processing step becomes necessary prior to training. This pre-processing step comprises two main components: text cleaning and feature extraction.

- **Text Cleaning**

The cleaning process of our datasets initiates with the removal of diacritics (tashkeel) from the text, followed by eliminating non-Arabic letters or numerals. Stemming normalization is then applied to replace various versions of the same word with its normalized or root form. Next, stop words (common words devoid of useful information) are removed, along with any unnecessary letters or symbols. Finally, the text undergoes tokenization, wherein it is segmented into its constituent words and phrases.

- **Feature Extraction**

Feature extraction aims to extract the most pertinent information from a dataset and represent it in a machine-readable format. One-hot encoding [28] is a prevalent method for describing categorical variables as binary vectors. In machine learning, categorical data like words or integers, is often transformed into a numerical representation suitable for input into machine-learning models. One-hot encoding entails generating a new binary feature for each distinct category within a feature. For

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 10, No. 04, December 2024.

example, if a feature has three distinct categories (A, B and C), three additional binary features—one for each category—are created. The original feature is then replaced by these three new binary features, with the value corresponding to the original category set to 1 and all other values set to 0.

3.3 Training and Model Selection

Following the completion of the pre-processing step, a layer of embedding was generated effectively using the pre-processed raw texts. This layer serves as input during the training phase of designing classifiers, which is further segmented into various parts. The initial step involves splitting the pre-processed data into training and test sub-sets, facilitated by the train/test split methodology.

A common strategy for evaluating the efficacy of a machine-learning model is the train/test split. Here, the model is trained using both the training set and the test set. The training set aids in determining the correlations between inputs and outputs, while the test set evaluates the model's ability to generalize to new inputs. In this research, balancing the datasets was crucial to enhancing the model's efficiency and performance. To achieve this, a percentage of 50% of the datasets was allocated for training and 25% for validation, with the remaining 25% being reserved for testing the classifier's effectiveness.

Balancing the dataset is vital in deep learning to prevent the model from becoming overly specialized, a condition known as overfitting. Overfitting can occur when there is a disproportionate representation of one type of data over the other. Common practices to address this imbalance include over-sampling the minority group and under-sampling the majority group, thereby redistributing the data in the training set to enhance the model's ability to generalize to new data.

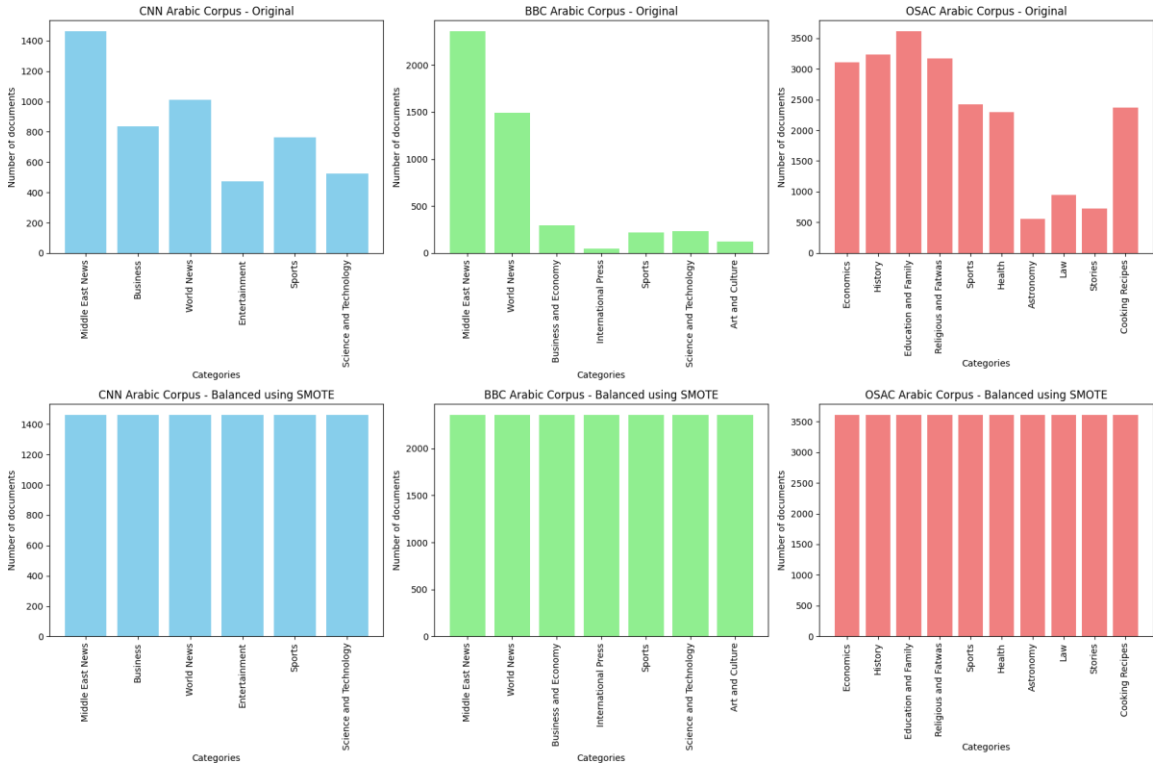

Figure 2. Class distribution of the datasets before and after balancing.

This paper utilizes SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance within our datasets. SMOTE is an effective strategy for generating synthetic samples of the minority class to match the number of data points in the dominant class. By creating new, plausible data points between existing samples, SMOTE enhances the representation of the minority class, which helps correct biases that often arise in machine-learning models trained on imbalanced datasets.

The significance of employing SMOTE lies in its ability to improve model performance and fairness. By balancing the dataset, SMOTE ensures that the model learns effectively about the minority class, leading to more accurate and reliable predictions. This is particularly important in applications where

the minority class is crucial, but underrepresented, such as medical diagnosis or fraud detection, ensuring that all classes are treated equitably and improving the overall decision-making accuracy.

Figure 2 illustrates the class distribution of three of the datasets that we used: the CNN Arabic corpus, the BBC Arabic corpus and the OSAC Arabic corpus. The figure reveals the class imbalance inherent in each dataset.

After dividing and evenly distributing the datasets, the data was input into three distinct deep-learning models: Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs) and Gated Recurrent Units (GRUs). The CNN model utilized two convolutional layers with 64 and 32 filters, respectively and a kernel size of 3. This setup allowed the CNN to detect hierarchical features within the text. Following the convolutional layers, a MaxPooling layer with a pool size of 2 was applied to down- sample and reduce dimensionality, while a dropout layer with a 0.5 probability was used to mitigate overfitting. The output from these layers was then flattened into a one-dimensional vector.

The LSTM and GRU models, each with a single layer of 8 units, were designed to handle sequential dependencies in the text. Both models included a dropout layer with a 0.5 probability after their respective recurrent layers to prevent overfitting. The outputs of the LSTM and GRU layers were also flattened into one-dimensional vectors, preparing them for integration with the CNN features.

The final step involved merging the outputs from all three models. The flattened feature vectors from the CNN, LSTM and GRU models were concatenated into a single, comprehensive feature vector. This integrated feature vector was then used to generate predictions through a final prediction model, often involving additional dense layers and a softmax activation function to output class probabilities. This hybrid approach allowed the model to leverage both local features and sequential patterns, enhancing its overall predictive performance.

Our model is mathematically expressed as follows:

Let's assume that the input to the model is a sequence of word embedding $x = x_1, x_2, \ldots, x_t$

3.3.1 CNNs

The output of the convolution operation is given in Equation 1:

$$h = relu(W_c x + b_c) \tag{1}$$

The max-pooling operation can be represented as shown in Equation 2:

$$h_{pool} = Max(h) \tag{2}$$

Finally, the results of the fully connected layer can be illustrated as in Equation 3:

$$y_{cnn} = W_f h + b_f \tag{3}$$

where, W_c is the weight matrix for the convolution operation, b_c is the bias term, W_t is the weight matrix for the fully connected layer and b_f is the bias term.

3.3.2 LSTM

At each time step t, the LSTM updates its hidden state h_t and its memory cell C_t. The hidden state h_t is used to make a prediction, while the memory cell C_t is used to retain information over time, as described in Equation 4.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)\ f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)\ o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$

where W_i, U_i, b_i, W_f, U_f, b_f, W_o, U_o ,and b_o are the parameters of the LSTM, x_t is the input at time step t, h_{t-1} is the hidden state at time step t 1 and σ is the sigmoid activation function. And i_t is the update gate, f_t is the forget gate, and o_t is the output gate.

$$C_t = f_t C_{t-1} + i_t \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$

where C_{t-1} is the memory cell at time step *t-1* and *tanh* is the hyperbolic tangent activation function as represented in Equation 5.

The hidden state h_t is updated as shown in Equation 6.

$$h_t = o_t \tanh(C_t) \tag{6}$$

Finally, the output passing through a fully connected layer can be represented as in Equation 7:

$$y_{lstm} = W_y h_t + b_y \tag{7}$$

where W_y and b_y are the parameters of the fully connected layer.

3.3.3 GRU

At each time step t, the GRU computes the updated hidden state h_t as shown in Equation 8:

$$\begin{aligned} z_t &= \sigma(W_z[h_{t-1} + x_t\ ] + bz) \\ rt &= \sigma(Wr[ht-1 + xt\ ] + br \\ h'_t &= tanh(W_h[r_t h_{t-1} + x_t\ ] + bh) \\ h_t &= (1 - z_t)h_{t-1} + z_t h'_t \end{aligned} \tag{8}$$

where W_z, W_r, W_h, b_z, b_r and b_h are learnable parameters of the GRU. σ is the sigmoid function used to produce a value between 0 and 1, while the tanh function maps its input to the range (-1, 1).

Finally, the final hidden state ht can be used to make predictions by passing it through a fully connected layer, as shown in Equation 9.

$$y_{gru} = W_y h_t + b_y \tag{9}$$

where W_y and b_y are learnable parameters and y is the predicted probability distribution over the classes. The final output of our model is presented in Equation 10, obtained from Equations 3, 7 and 9:

$$Y = [y_{cnn}, y_{lstm}, y_{gru}] \tag{10}$$

4. EXPERIMENTAL RESULTS AND DISCUSSION

This paragraph will delineate the achieved results both before and after balancing our datasets, followed by a discussion based on the proposed approach. Additionally, it will entail a comparison between our proposed model and other models utilizing the same datasets. We employed 50% of the datasets for training purposes, 25% for validation and reserved the remaining 25% for testing. Furthermore, our model was trained using an embedding layer size of 250, with a batch size of 512 and 10 epochs and compiled with an Adam optimizer. All experiments were conducted on a personal computer with the following specifications: an Intel(R) Core (TM) i7-8650U processor, 32 GB RAM and a frequency of 2.11 GHz.

The following datasets were utilized in both the training and testing phases of developing our model.

4.1 Datasets

To implement and train the models in this research, it is necessary to collect Arabic-news documents to use them for these purposes. These datasets were chosen due to their diversity, relevance and availability, allowing for comprehensive experimentation and evaluation of the proposed models. By leveraging these datasets, this research aims to develop a robust Arabic-news classification system that can effectively categorize news articles across different domains and topics. This research utilized three popular datasets:

4.1.1 CNN Arabic Corpus

CNN Arabic corpus is collected from CNN Arabic website cnnarabic.com. The corpus includes 5,070 text documents. Each text document belongs to 1 of 6 categories (Business 836, Entertainments 474, Middle East News 1462, Science and Technology 526, Sports 762 and World News 1010). The corpus contains 2,241,348 (2.2M) words and 144,460 district keywords after stop words removal [29].

4.1.2 BBC Arabic Corpus

BBC Arabic corpus is collected from BBC Arabic website bbcarabic.com. The corpus includes 4,763 text documents. Each text document belongs to 1 of 7 categories (Middle East News 2356, World

News 1489, Business and Economy 296, Sports 219, International Press 49, Science and Technology 232 and Art and Culture 122). The corpus contains 1,860,786 (1.8M) words and 106,733 district keywords after stop-word removal [29].

Table 1. CNN Arabic corpus dataset.

Categories	Number of documents
Middle East	1462
Business	836
World	1010
Entertainment	474
Sport	762
SciTech	526
All	5070

Table 2. BBC Arabic corpus dataset.

Categories	Number of documents
Middle East	2356
Business & Economy	296
World	1489
International Press	49
Sport	219
Science & Technology	232
Art & Culture	122
All	4763

4.1.3 OSAC Arabic Corpus

OSAC Arabic corpus is collected from multiple websites. The corpus includes 22,429 text documents. Each text document belongs to 1 of 10 categories (Economics, History, Education and Family, Religious and Fatwas, Sports, Health, Astronomy, Law, Stories and Cooking Recipes). The corpus contains about 18,183,511 (18M) words and 449,600 district keywords after stop words removal [29].

Table 3. OSAC Arabic corpus dataset.

Categories	Number of documents
Economics	3102
History	3233
Education & Family	3608
Religious and Fatwas	3171
Sports	2419
Health	2296
Astronomy	557
Low	944
Stories	726
Cooking Recipes	2373
All	22,429

4.2 Evaluation Metrics

The performance of a model can be effectively assessed by applying evaluation metrics to estimate its capabilities and construct the optimal model based on these criteria. During the research for this work, Accuracy, Precision, Recall and F1-score were employed [30].

In essence, the Accuracy metric gauges the proportion of correctly predicted cases out of the total anticipated cases. Precision is determined by dividing the number of correctly predicted positive models by the total number of models predicted as positive. Similarly, Recall measures the proportion of correctly labeled models (positively classed models). The F1-score, calculated using the harmonic mean of Recall and Precision, provides a balanced assessment of a model's performance.

Additionally, to evaluate a model's accuracy in classifying samples as either positive or negative,

statisticians commonly use the area under the receiver operating characteristic curve (AUC-ROC). Moreover, dissimilarities between expected and realized values can be quantified using techniques, such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE). Lastly, the Matthews Correlation Coefficient (MCC) is utilized to evaluate accuracy while accounting for false positives.

4.3 Results and Analysis

This paper presents the findings of several experimental studies conducted to comprehensively assess the model across all dimensions. The study commenced with a comparative analysis of evaluation metrics before and after dataset balancing. Following this, overfitting graphs were generated to scrutinize the stability of the model. Subsequently, the hybrid method was tested against approaches grounded in deep-learning algorithms. Finally, a comparison was drawn with existing literature to evaluate and validate the proposed approach.

4.3.1 Comparison of Evaluation Metrics

This experiment entails comparing the outcomes achieved by our model for each dataset against those of three distinct feature-extraction approaches. The ensuing tables present the results obtained.

Table 4. Performance evaluation of the proposed model for CNN dataset.

Dataset	Accuracy	Precision	Recall	F1-score	AUC-ROC	MSE	RMSE	MAE	MCC
Balanced	94.825	94.903	95.052	94.977	0.995	0.015	0.122	0.023	0.932
Unbalanced	93.195	92.601	93.186	92.827	0.994	0.018	0.133	0.028	0.915

Table 5. Performance evaluation of the proposed model for BBC dataset.

Dataset	Accuracy	Precision	Recall	F1-score	AUC-ROC	MSE	RMSE	MAE	MCC
Balanced	90.346	90.028	79.539	83.698	0.982	0.023	0.150	0.030	0.848
Unbalanced	88.247	84.082	75.618	78.954	0.981	0.026	0.159	0.040	0.815

Table 6. Performance evaluation of the proposed model for OSAC dataset.

Dataset	Accuracy	Precision	Recall	F1-score	AUC-ROC	MSE	RMSE	MAE	MCC
Balanced	99.356	99.146	99.882	99.514	1.000	0.001	0.038	0.0024	0.989
Unbalanced	99.086	98.856	98.711	98.782	1.000	0.001	0.037	0.0026	0.989

The model's performance was assessed on both balanced and unbalanced datasets. Tables 4, 5 and 6 illustrate the results, demonstrating notably high performance, particularly for the balanced dataset. Specifically, the model achieved accuracies of 94.57%, 90.34% and 99.08% for the CNN, BBC and OSAC datasets, respectively. The precision values were 94.12%, 90.02% and 99.04%, while the recall values were 94.50%, 79.53% and 98.88% and the F1-scores were 94.30%, 83.69% and 98.93%, respectively, for the CNN, BBC and OSAC datasets.

Moreover, the model exhibited high AUC-ROC values of 0.995 for the CNN dataset, 0.982 for BBC and 1.000 for the OSAC dataset, indicating its excellent ability to distinguish between positive and negative classes. Additionally, the MSE values were 0.015, 0.026 and 0.001 for the CNN, BBC and OSAC datasets, respectively. The RMSE values were 0.122, 0.150 and 0.038 and the MAE values were 0.023, 0.030 and 0.0024, respectively. These metrics collectively suggest that the model's predictions were close to the actual class probabilities.

Furthermore, the MCC values were 0.932, 0.848 and 0.989 for the CNN, BBC and OSAC datasets, respectively, indicating the model's strong ability to correctly classify texts.

The confusion matrix in Figure 3 for the CLGNet model on the CNN dataset demonstrates its high performance across various categories. The model accurately predicts the classes with minimal errors. This high level of performance indicates the model's robustness and reliability, making it well-suited for real-world applications in news categorization and content analysis.

The confusion matrix in Figure 3 for the CLGNet model on the BBC dataset demonstrates the model's performance across various categories. The model effectively categorizes the text data, though there

are some misclassifications. This high level of performance indicates the model's robustness and highlights areas where it can be improved for even better accuracy.

The confusion matrix in Figure 3 for the CLGNet model on the OSAC dataset highlights the model's exceptional performance. This high accuracy and near-perfect precision and recall demonstrate the model's robustness in correctly categorizing the text data across diverse categories. Such performance indicates the model's potential for real-world applications in text-classification tasks, providing reliable and accurate results.

Figure 3. Confusion matrices for CLGNet using three datasets.

An error-analysis phase was conducted using the confusion matrices generated for each dataset, as shown in Table 7. These matrices provide insight into the types of errors made by the model, including false positives (FP), false negatives (FN), true positives (TP) and true negatives (TN). By analyzing the confusion matrices, we identified specific classes where the model consistently misclassified instances, particularly in cases where the classes were similar in context or had overlapping features.

Table 7. Summary of errors in confusion matrices for CNN, BBC and OSAC datasets.

Dataset	Class	True Positives	False Positives	False Negatives	Error analysis
CNN	Middle East	346	17	19	Strong classification for dominant classes
	Business	199	12	10	Confusion with World and Middle East
	World	240	13	12	Misclassification with Middle East and Business
	Entertainment	113	7	5	Struggled with classification
	Sport	181	9	9	Occasional confusion with SciTech
	SciTech	122	6	9	Some misclassification with Sport and World
BBC	Middle East	535	35	45	Consistent misclassification with World and Art
	Business & Economy	64	13	10	Confusion with Middle East and World
	World	317	39	40	Confusion with Middle East
	Int. Press	8	11	4	Some misclassification with World and Business
	Sport	46	18	14	Good classification with minor errors
	Sci & Tech	42	18	12	Some confusion with Art
	Art & Culture	20	17	7	Issues with classification, misclassified as Middle East
OSAC	Economics	774	2	1	Strong classification
	History	806	1	0	Some misclassification
	Education & Family	900	0	1	Good performance
	Religious & Fatwas	789	1	0	Strong classification
	Sports	601	0	0	Few misclassifications
	Health	573	1	0	Some misclassification with adjacent categories
	Astronomy	138	1	1	Struggled with classification
	Law	234	1	1	Issues with misclassification
	Stories	174	0	1	Some errors in classification
	Cooking Recipes	593	0	0	Strong classification, few errors

In Table 7 CNN dataset, the model showed strong performance in identifying dominant classes, like Middle East (TP: 346), but struggled with smaller, more ambiguous categories, like Entertainment, where there was a higher incidence of false negatives (e.g., 2 misclassified as Middle East and 1 as Business). Similarly, in the BBC dataset, Middle East and World were often confused due to their contextual similarities, resulting in a notable number of false positives (e.g. 28 misclassified as Middle East). The OSAC dataset presented a more diverse set of classes, with Economics (TP: 774) and

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 10, No. 04, December 2024.

Cooking Recipes (TP: 593) being well distinguished, while smaller classes, like Astronomy and Law, showed more misclassifications (FPs and FNs). These findings suggest that the model performs well in distinguishing between major categories, but struggles with less represented or more contextually similar classes. This discrepancy suggests the need for further refinement of the model, particularly through targeted training on misclassified examples and the use of more sophisticated feature extraction methods to better differentiate between classes with overlapping features.

4.3.2 Interpretation of Training and Validation Graphs

In this experiment, we generated plots depicting the training and validation accuracy and loss values to thoroughly analyze our model's performance on the three datasets. The results are illustrated in the following graphs.

In Figures 4, 5 and 6, our model demonstrated convergence to low values in both training and validation loss curves. This convergence indicates that the model found weights that minimized the difference between predicted and actual class probabilities, as evidenced by the loss converging to near-zero values. Such convergence is crucial for effective model training and signifies that the model has learned a robust representation of the input data.

Furthermore, in addition to loss convergence, the accuracy curve also converged to values close to one. This convergence suggests that the model's representations successfully captured the fundamental patterns within the data, resulting in correct predictions. The high convergence of accuracy indicates that the model generalized well to the validation data, a significant indicator of model success.

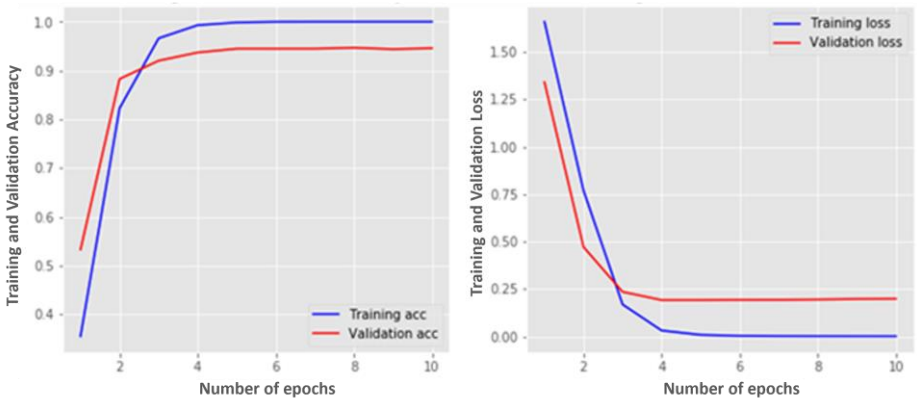

Figure 4. Graph of accuracy/loss training and validation for balanced CNN dataset.

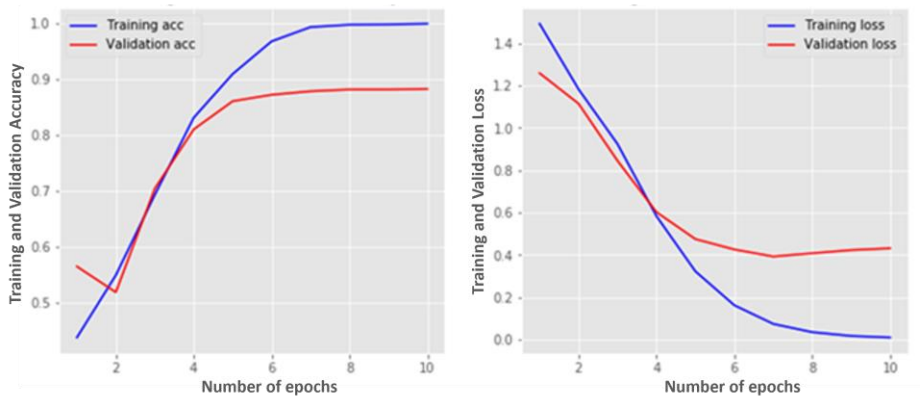

Figure 5. Graph of accuracy/loss training and validation for balanced BBC dataset.

Overall, our model demonstrated the ability to learn from training data and generalize effectively to validation data. By recognizing data patterns and generating reliable predictions, the model proves suitable for text-categorization tasks.

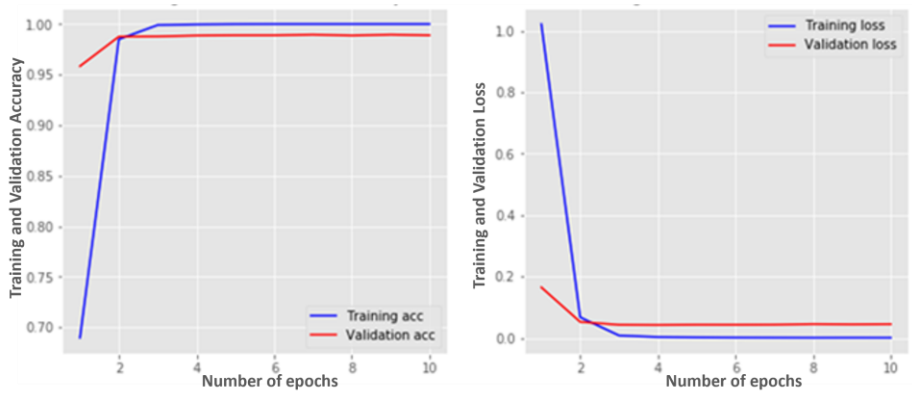

Figure 6. Graph of accuracy/loss training and validation for balanced OSAC dataset.

4.3.3 Comparison with Simple Deep-learning Layers

In this investigation, for comparative purposes, the model was segmented into three distinct components: CNN, LSTM and GRU layers. Each of these layers within our architecture was individually executed with identical parameters as the proposed model. This approach aimed to evaluate their respective performances and compare them against the performance of our proposed method.

Table 8. Comparative study of different deep-learning algorithms against the proposed model for CNN dataset.

Method	Accuracy	F1-score	Execution time (s)	Parameters
Conv1D	90.138	87.190	87.93	12,601,862
LTSM	92.110	91.443	121.24	12,532,294
GRU	93.589	92.878	95.58	12,530,246
CLGNet	94.825	94.977	669.45	12,664,390

Table 9. Comparative study of different deep-learning algorithms against the proposed model for BBC dataset.

Method	Accuracy	F1-score	Execution time (s)	Parameters
Conv1D	90.138	87.190	87.93	12,601,862
LTSM	92.110	91.443	121.24	12,532,294
GRU	93.589	92.878	95.58	12,530,246
CLGNet	94.825	94.977	669.45	12,664,390

Table 10. Comparative study of different deep-learning algorithms against the proposed model for OSAC dataset.

Method	Accuracy	F1-score	Execution time (s)	Parameters
Conv1D	90.138	87.190	87.93	12,601,862
LTSM	92.110	91.443	121.24	12,532,294
GRU	93.589	92.878	95.58	12,530,246
CLGNet	94.825	94.977	669.45	12,664,390

Tables 8, 9 and 10 reveal that the proposed CLGNet model demonstrates superior accuracy and F1-score performance compared to other methods. Despite having longer execution times and larger numbers of parameters, the performance of our proposed CLGNet model remains acceptable.

In summary, the findings suggest that CLGNet outperforms other methods in terms of accuracy and F1-score, even though it may have longer execution times and a larger number of parameters compared to GRU.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 10, No. 04, December 2024.

4.3.4 Comparative Study with Literature

To evaluate the proposed approach, a comparison with the literature is carried out in this context, highlighting diverse methods and results in Arabic-text classification. [31] explored the use of the ImpCHI method combined with an SVM classifier on a dataset of 5,070 Arabic documents, demonstrating superior performance with an F-measure of 90.50% when 900 features were selected. On the other hand, [32] compared traditional vectorization methods, like BOW and TF-IDF, employing classifiers, such as RL, SVM and ANN, to provide a comprehensive evaluation of their effectiveness. Meanwhile, [33] proposed the ArCAR system, a novel deep-learning approach for Arabic-text recognition and classification, achieving an accuracy of 97.76% on the Alarabiya-balance dataset, showcasing the potential of deep learning in this domain. [19] also leveraged deep learning, combining CNN and RNN models with various word embeddings, reporting high performance on the OSAC dataset, thus validating the effectiveness of hybrid architectures in capturing contextual dependencies. Finally, [34] conducted an empirical study comparing five classifiers (SVM, DT, RF, KNN and LR) using different feature-vectorization methods, finding that SVM and LR consistently outperformed others, especially when feature-vectorization techniques were applied, highlighting the stability of these classifiers across different datasets. These studies collectively emphasize the evolving landscape of Arabic-text classification, where both traditional and deep-learning methods are continuously refined to enhance accuracy and contextual understanding.

Table 11 provides a comparison of the performance of various models applied to three datasets: CNN, BBC and OSAC. The performance measure used is the F1-score, a commonly used metric for evaluating the accuracy of a classification model. The table shows that the proposed model outperforms the previous models and the other compared approaches in all datasets, with the largest improvement seen in all of them.

Table 11. Arabic-text categorization on the CNN, BBC and OSAC datasets: A comparative evaluation of recent works.

Dataset	Method	F1-score
CNN	Approach [31]	90.50
	Approach [32]	93.71
	Proposed Model	94.30
BBC	Approach [33]	69.62
	Proposed Model	83.69
OSAC	Approach [19]	98.61
	Approach [34]	98.91
	Proposed Model	98.93

To further understand the performance of our proposed model, it is important to consider the unique strengths of each component model and how they contribute to the overall effectiveness of the hybrid approach. The Convolutional Neural Network (CNN) component excels at identifying local patterns and spatial hierarchies within the text, which are crucial for recognizing key phrases that strongly indicate specific news categories. This ability to capture localized features allows CNNs to effectively process and distinguish between different types of news content. On the other hand, the Long Short-Term Memory (LSTM) network is adept at retaining long-term dependencies within sequences, making it particularly valuable for understanding the broader context of sentences where meaning may be derived from distant words or phrases. This capability enhances the model's understanding of nuanced information in longer texts, which is essential for accurate classification. Complementing these strengths, the Gated Recurrent Unit (GRU) offers a streamlined approach to handling sequential data, providing a balance between maintaining performance and optimizing computational efficiency. The GRU's simplified architecture allows the model to process large volumes of text data more quickly, without sacrificing the ability to capture necessary sequential relationships. By integrating these models, the hybrid approach leverages the CNN ability to extract critical features, the LSTM's contextual understanding and the GRU's efficiency, resulting in a robust and highly accurate Arabic news classification system.

5. CONCLUSION

Arabic-news classification presents a significant challenge in natural-language processing (NLP), requiring the categorization of news articles into predefined categories, such as politics, sports and entertainment. This task is complex due to the intricate nature of the Arabic language and the vast dimensionality of text data. This study addresses these challenges by introducing a multi-channel deep learning model, CLGNet, specifically designed for Arabic-text categorization. The model effectively analyzes and categorizes Arabic-news articles by employing data cleaning, word embedding, dataset balancing and deep-learning techniques including CNNs, LSTM and GRU. On multiple benchmark datasets, including CNN, BBC and OSAC, the model achieves impressive accuracies of 94.57%, 90.34% and 99.08%, respectively, along with F1-scores of 94.30%, 83.69% and 98.93%. Experimental results confirm the effectiveness of the proposed model, showcasing high evaluation metrics, such as Accuracy, Precision, Recall, F1-score, AUC-ROC, MSE and MCC. Our model significantly outperforms other state-of-the-art techniques in analyzing Arabic-news data, demonstrating its capability to overcome text-classification challenges and offering a promising solution for various NLP applications in Arabic. Future work aims to further enhance the model by integrating new methodologies and applying them to big-data scenarios. Additionally, the model will be evaluated on a wider range of Arabic datasets to validate its robustness and applicability across diverse linguistic domains.

REFERENCES

[1] Y. Timmerman and A. Bronselaer, "Measuring Data Quality in Information Systems Research," Decision Support Systems, vol. 126, p. 113138, DOI: 10.1016/j.dss.2019.113138, Nov. 2019.

[2] C. Porlezza, "Accuracy in Journalism," Oxford Research Encyclopedia of Communication, DOI: 10.1093/acrefore/9780190228613.013.773, Oxford University Press, Mar. 2019.

[3] N. Newman, R. Fletcher, A. Schulz, S. Andi, C. T. Robertson and R. K. Nielsen, Reuters Institute Digital News Report 2021, Reuters Institute for the Study of Journalism, pp. 1-164, 10th Edn, [Online], Available:https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2021-06/Digital_News_Report_2021_FINAL.pdf, 2021.

[4] I. Ahmad, F. AlQurashi and R. Mehmood, "Machine and Deep Learning Methods with Manual and Automatic Labelling for News Classification in Bangla Language," arXiv: 2210.10903, DOI: 10.48550/arXiv.2210.10903, 2022.

[5] R. Indrakumari, T. Poongodi and K. Singh, "Introduction to Deep Learning," in Book: Advanced Deep Learning for Engineers, pp. 1–22, DOI: 10.1007/978-3-030-66519-7_1, Springer International Publishing, 2021.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," Proc. of the 31st Conference on Neural Information Processing Systems (NIPS 2017), pp. 1-11, Long Beach, CA, USA, 2017.

[7] H. Hassan et al., "Achieving Human Parity on Automatic Chinese to English News Translation," arXiv: 1803.05567, DOI: 10.48550/arXiv.1803.05567, 2018.

[8] M. Jabrane, I. Hafidi and Y. Rochd, "An Improved Active Machine Learning Query Strategy for Entity Matching Problem," Proc. of Advances in Machine Intelligence and Computer Science Applications (ICMICSA 2022), pp. 317–327, Springer Nature Switzerland, 2023.

[9] J. Mourad, T. Hiba, R. Yassir and H. Imad, "ERABQS: Entity Resolution Based on Active Machine Learning and Balancing Query Strategy," Journal of Intelligent Information Systems, DOI: 10.1007/s10844-024-00853-0, Mar. 2024.

[10] M. Jabrane, H. Tabbaa, A. Hadri and I. Hafidi, "Enhancing Entity Resolution with a Hybrid Active Machine Learning Framework: Strategies for Optimal Learning in Sparse Datasets," Information Systems, vol. 125, p. 102410, Nov. 2024.

[11] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. of NAACL-HLT 2019, pp. 4171–4186, Minneapolis, Minnesota, June 2 - June 7, 2019.

[12] X. Liu, P. He, W. Chen and J. Gao, "Multi-task Deep Neural Networks for Natural Language Understanding," Proc. of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4487–4496, Florence, Italy, July 28 - August 2, 2019.

[13] P. Rajpurkar, J. Zhang, K. Lopyrev and P. Liang, "Squad: 100, 000+ Questions for Machine Comprehension of Text," Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing, pp. 2383–2392, Austin, Texas, USA, 2016.

[14] G. Lample and A. Conneau, "Cross-lingual Language Model Pre-training," Proc. of the 33rd Conf. on Neural Information Processing Systems (NeurIPS 2019), pp. 1-11, Vancouver, Canada, 2019.

[15] K. M. Fouad, S. F. Sabbeh and W. Medhat, "Arabic Fake News Detection Using Deep Learning," Computers, Materials & Continua, vol. 71, no. 2, pp. 3647–3665, 2022.

[16] M. Azzeh, A. Qusef and O. Alabboushi, "Arabic Fake News Detection in Social Media Context Using Word Embeddings and Pre-trained Transformers," Arabian Journal for Science and Engineering, DOI: 10.1007/s13369-024-08959-x, Apr. 2024.

[17] M. M. Abdelsamie, S. S. Azab and H. A. Hefny, "A Comprehensive Review on Arabic Offensive Language and Hate Speech Detection on Social Media: Methods, Challenges and Solutions," Social Network Analysis and Mining, vol. 14, p. 111, DOI: 10.1007/s13278-024-01258-1, May 2024.

[18] L. Zhang, W. Jiang and Z. Zhao, "Short-text Feature Expansion and Classification Based on Non-negative Matrix Factorization," Proc. of Machine Learning for Cyber Security(ML4CS 2020), pp. 347–362, DOI: 10.1007/978-3-030-62463-7_32, Springer International Publishing, 2020.

[19] M. S. H. Ameur, R. Belkebir and A. Guessoum, "Robust Arabic Text Categorization by Combining Convolutional and Recurrent Neural Networks," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 19, no. 5, Article no. 66, July 2020.

[20] A. M. Bdeir and F. Ibrahim, "A Framework for Arabic Tweets Multi-label Classification Using Word Embedding and Neural Networks Algorithms," Proc. of the 2020 2nd Int. Conf. on Big Data Engineering (BDE' 2020), pp. 105-112, DOI: 10.1145/3404512.340452, ACM, May 2020.

[21] A. Hassanein and M. Nour, "A Proposed Model of Selecting Features for Classifying Arabic Text," Jordanian J. of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 275-290, Dec. 2019.

[22] L. Qadi, H. Rifai, S. Obaid and A. Elnagar, "A Scalable Shallow Learning Approach for Tagging Arabic News Articles," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 3, pp. 263-280, 2020.

[23] T. A. Wotaifi and B. N. Dhannoon, "An Effective Hybrid Deep Neural Network for Arabic Fake News Detection," Baghdad Science Journal, vol. 20, no. 4, DOI: 10.21123/bsj.2023.7427, Jan. 2023.

[24] A. B. Nassif, A. Elnagar, O. Elgendy and Y. Afadar, "Arabic Fake News Detection Based on Deep Contextualized Embedding Models," Neural Computing and Applications, vol. 34, pp. 16019–16032, May 2022.

[25] R. Romero, P. Celard, J. Sorribes-Fdez, A. Seara Vieira, E. Iglesias and L. Borrajo, "MobyDeep: A Lightweight CNN Architecture to Configure Models for Text Classification," Knowledge-based Systems, vol. 257, p. 109914, DOI: 10.1016/j.knosys.2022.109914, Dec. 2022.

[26] A. Alqahtani, H. Ullah Khan, S. Alsubai, M. Sha, A. Almadhor, T. Iqbal and S. Abbas, "An Efficient Approach for Textual Data Classification Using Deep Learning," Frontiers in Computational Neuroscience, vol. 16, DOI: 10.3389/fncom.2022.992296, Sept. 2022.

[27] A. Awajan, "Arabic Text Pre-processing for the Natural Language Processing Applications," Arab Gulf Journal of Scientific Research, vol. 25, no. 4, pp. 179–189, 2007.

[28] Y. Sun et al., "Modifying the One-hot Encoding Technique Can Enhance the Adversarial Robustness of the Visual Model for Symbol Recognition," Expert Systems with Applications, vol. 250, p. 123751, DOI: 10.1016/j.eswa.2024.123751, Sept. 2024.

[29] N. Alalyani and S. Larabi, "NADA: New Arabic Dataset for Text Classification," Int. Journal of Advanced Computer Science and Applications, vol. 9, no. 9, DOI: 10.14569/IJACSA.2018.090928, 2018.

[30] M. Hossin and M. N. Sulaiman, "A Review on Evaluation Metrics for Data Classification Evaluations," Int. Journal of Data Mining & Knowledge Management Process, vol. 5, no. 2, pp. 01–11, DOI : 10.5121/ijdkp.2015.5201, 2015.

[31] S. Bahassine, A. Madani, M. Al-Sarem and M. Kissi, "Feature Selection Using an Improved Chi-square for Arabic Text Classification," Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 2, pp. 225–231, Feb. 2020.

[32] I. Jamaleddyn and M. Biniz, "Contribution to Arabic Text Classification Using Machine Learning Techniques," Proc. of Business Intelligence (CBI 2021), pp. 18–32, DOI: 10.1007/978-3-030-76508-8_2, Springer, 2021.

[33] A. Y. Muaad, H. Jayappa, M. A. Al-antari and S. Lee, "ArCAR: A Novel Deep Learning Computer-aided Recognition for Character-level Arabic Text Representation and Recognition," Algorithms, vol. 14, no. 7, p. 216, DOI: 10.3390/a14070216, July 2021.

[34] T. Sabri, O. E. Beggar and M. Kissi, "Comparative Study of Arabic Text Classification Using Feature Vectorization Methods," Procedia Computer Science, vol. 198, pp. 269–275, DOI: 10.1016/j.procs.2021.12.239, 2022.

ملخص البحث:

في عصر الصّحافة الرّقمية، يمثل تصنيف الأخبار باللّغة العربية تحدّياً مهمّاً إلى الطّبيعة المعقّدة للُّغة والتّنوُّع الكبير في المحتوى.

تعرض هذه الورقة نموذجاً مبتكراً متعدّد القنوات يستند إلى التّعلُّم العميق، مصمَّماً من أجل تحسين دِقّة تصْنيف الأخبار باللّغة العربية. ويعمل النّموذج المقترح بفاعلية على معالجة وتصنيف بيانات نَصّية باللّغة العربية.

لقد تمّ إجراء تجارب مكثَّفة على مجموعةٍ من مجموعات البيانات CNN و BBC و OSAC، حيث تمكّن النّموذج المقترح من تحقيق دقّةٍ عاليةٍ ومتانةٍ معتبرةٍ، متفوّقاً بذلك على بعض الطّرق الواردة في أدبيات الموضوع.

وتؤكّد النتائج الّتي تمّ الحصول عليها فاعلية نموذجنا الهجين (CNN، و LSTM، و GRU) في معالجة التّحدّيات المرتبطة بتصْنيف النّصوص باللّغة العربية، بالإضافة إلى إمكانية استخدامه في الأنظمة الأوتوماتيكية لتصْنيف الأخبار.