International Journal of Engineering and Management Research
e-ISSN: 2250-0758 | p-ISSN: 2394-6962
https://ijemr.vandanapublications.com

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

Online Candidate Selection System for Elections

Meregngnage H.E.[1], Fernando J.A.T.C.[2], Fernando W.K.[3] and Fernando N.M.R.M.[4]
[1]Undergraduate, Department of Computer Science & Software Engineering, SLIIT, SRI LANKA
[2]Undergraduate, Department of Information Technology, SLIIT, SRI LANKA
[3]Undergraduate, Department of Computer Science & Software Engineering, SLIIT, SRI LANKA
[4]Undergraduate, Department of Information Technology, SLIIT, SRI LANKA

[1]Corresponding Author: hansakaeranda99@gmail.com

Received: 14-07-2023 Revised: 01-08-2023 Accepted: 23-08-2023

ABSTRACT

This research proposes an innovative online platform for political parties in Sri Lanka to enhance the candidate selection process. The platform incorporates features such as sentiment analysis, background checks, aptitude tests, ranking system, and analysis of candidates' promises and activities. Developed using advanced technologies, it aims to ensure transparency, efficiency, and accessibility for all eligible candidates, ultimately contributing to a more democratic election.

Keywords-- Political, Candidates, Sentiment Analysis

I. INTRODUCTION

The election candidate selection system serves as a critical mechanism to identify competent and trustworthy individuals who can lead and represent the interests of the electorate. Unfortunately, in many countries, including Sri Lanka, the prevalence of false reputations and misleading practices has led to the election of candidates who fail to deliver on their promises, contributing to economic crises and overall societal decline. This research paper presents an advanced candidate selection system that addresses these issues by employing cutting-edge technologies to ensure the integrity of the selection process and hold candidates accountable for their actions. To establish a robust selection process, the proposed system initiates by conducting comprehensive background checks on prospective candidates. This involves examining web articles and online sources to uncover any previous instances of misconduct. Additionally, the system verifies the certifications and qualifications provided by the applicants. Candidates with a history of bad conduct or insufficient qualifications are automatically rejected, safeguarding the electoral process from potentially unfit candidates. Moreover, the system employs sentiment analysis techniques to analyze comments posted by voters on candidates' social media accounts. By categorizing these comments as positive or negative based on the sentiments expressed, the system gauges the popularity of candidates among the electorate. This sentiment analysis-driven approach enables a fair evaluation of candidates' public image and can help identify those who are genuinely trusted by the voters. Furthermore, the system goes beyond assessing public sentiment by scrutinizing candidates' previous electoral promises. By extracting and validating voice records of promises made during past elections, the system establishes a baseline for evaluating candidates' consistency and credibility. It then compares the promises made in the current election cycle with those from the past, enabling voters to assess the likelihood of candidates fulfilling their commitments. To provide a comprehensive overview of candidates' performances, the system monitors candidates' social and political services during previous election periods. Using advanced image processing techniques, the system analyzes Education, Construction developments, Elderly care, Environment and health and social service related services. Leveraging machine learning models, the system categorizes these activities and generates statistical reports, offering voters a comprehensive and data-driven evaluation of candidates' work. The proposed election candidate selection system presents an innovative and comprehensive approach to address the issues of misleading practices and lack of accountability in the electoral process. By incorporating background checks, sentiment analysis, comparison of promises, and analysis of social and political services, this system ensures a more transparent and efficient selection of candidates. The implementation of such a system has the potential to restore trust among voters and promote a more responsible political culture.

II. RELATED WORK/ LITERATURE REVIEW

The selection of political candidates plays a crucial role in ensuring competent and trustworthy leadership, as well as maintaining public trust in the democratic process. However, the prevalence of false reputations and misleading practices has been a persistent issue in many countries,

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

including Sri Lanka, leading to the election of candidates who fail to fulfill their promises and contribute to societal decline [1]. To address these challenges, researchers and scholars have explored various technologies and methodologies to enhance the candidate selection process.

One notable approach in candidate selection involves the use of sentiment analysis, which is a technique used to analyze and categorize sentiments expressed in texts, such as social media posts and comments. In the context of political candidate selection, sentiment analysis can provide insights into public opinion and evaluate the popularity and trustworthiness of candidates among the electorate [6]. The work of Bae and Lee [6] demonstrated the application of sentiment analysis on Twitter audiences, measuring the positive or negative influence of popular Twitter users. Their findings highlight the potential of sentiment analysis in assessing public sentiment towards political candidates.

Background verification of candidates is another critical aspect of the selection process to ensure the integrity and suitability of potential candidates. Machine learning and fuzzy matching techniques have been employed to automate the candidate background verification process [3]. Suman and Kushwah [3] presented a candidate background verification approach that utilizes machine learning and fuzzy matching to analyze and validate the information provided by candidates. By automatically cross-referencing online sources and uncovering instances of misconduct, this approach enhances the screening process and helps identify potentially unfit candidates.

Moreover, the evaluation of candidates' promises and activities is essential in providing voters with a comprehensive understanding of their credibility and performance. Leveraging natural language processing and machine learning techniques, researchers have explored the analysis of electoral promises made by candidates [4]. Németh [4] conducted a scoping review on the use of natural language processing in research on political polarization and highlighted the potential of this approach in assessing candidates' consistency and credibility based on their promises.

To analyze candidates' social and political services, advanced technologies such as image processing and machine learning have been utilized. These techniques enable the automatic extraction and categorization of candidates' activities, providing data-driven insights into their work [13]. For instance, the work of Singh et al. [13] demonstrated the use of deep residual learning for image recognition, which can be adapted to analyze candidates' social and political service-related images. By employing similar techniques, the proposed candidate selection system can generate statistical reports, enabling voters to evaluate candidates' past performances objectively.

In summary, the existing literature has explored various technologies and methodologies to enhance the candidate selection process in political elections. The application of sentiment analysis, background verification using machine learning and fuzzy matching, analysis of electoral promises, and analysis of candidates' social and political services can provide valuable insights into the suitability, credibility, and performance of candidates. By integrating these approaches into an innovative online platform, the proposed system aims to ensure transparency, efficiency, and accessibility for all eligible candidates in Sri Lanka, ultimately contributing to a more democratic election.

III. METHODOLOGY

To implement the proposed candidate selection system, a multi-faceted methodology combining advanced technologies and data analysis techniques will be employed. This section outlines the key steps involved in developing and implementing the innovative online platform for political parties in Sri Lanka.

A. Data Gathering for Dataset Training

Gathering data for the dataset training of the proposed online platform for political parties in Sri Lanka involved a meticulous and multi-faceted approach. The initial step was to compile a diverse collection of historical election data, including information about past candidates, their election results, and campaign promises. To incorporate the aspect of sentiment analysis, social media platforms and news articles were crawled by the research team to collect textual data reflecting public opinions and sentiments towards political candidates and parties.

Also, gathering data on campaign promises involved the compilation of speeches, public addresses, and official statements made by candidates during their election campaigns. These promises were then categorized and evaluated based on their feasibility and potential impact on society. Furthermore, image data gathering for the dataset training of the proposed online platform for political parties in Sri Lanka involved a comprehensive approach. Photographs from public rallies, campaign events, and official candidate portraits were collected from various sources, including social media platforms and official campaign websites. Metadata such as location, date, and event details were also recorded to provide context to the images. Efforts were made to ensure the diversity of the dataset by including images representing different parties, candidates, demographics, and geographic regions.

B. System Overview

1. Doing Background Check of the Candidate using Sentiment Analysis of News Reports

For doing the background check of the election candidates, “distilbert-base-uncased-finetuned-sst-2-english” model is used. BERT stands for Bidirectional Encoder Representations from Transformers. BERT is designed to

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. In candidate selection process BERT model is used for analyze the sentiment of the news regarding election candidates.

In the start of the pretrain process some modules need to be imported. They bring specific functionalities that may need during data preparation, model training, evaluation, and result analysis. Those modules are illustrated in Table1

Module	Functionality
glob	File and directory search using pattern matching
torch	Deep learning framework for building neural networks
json	Handling JSON data
os	Interacting with the operating system
numpy	Numerical computations and array operations
pandas	Data manipulation and analysis with tabular data
seaborn	Data visualization
matplotlib.pyplot	Data plotting and visualization
huggingface_hub	Interacting with Hugging Face's model hub
sklearn	Machine learning algorithms and tools

Dataset was uploaded as the csv file format and using glob.glob() function glob module search for csv files. Using pd.read.csv() function reads the csv files and concatenates all the cvs files using pd.concat() function. Then the load_dataset() function searches for CSV files in a specific directory and reads them into a pandas DataFrame. It selects the 'sentiment' and 'text' columns, drops any rows with missing values, and returns the processed DataFrame. The code then maps the sentiment labels in the DataFrame to numerical values using a predefined dictionary. It creates separate arrays for sentiments and news text from the DataFrame.

Next, a pre-trained DistilBERT model for sequence classification is initialized using the specified model_card. The model is configured with a specific number of labels and the ability to handle input size mismatches during training. A DistilBERT tokenizer is also initialized using the model_card and model is moved to the GPU for potential GPU acceleration if available.

Before perform sentiment analysis customization is done to model's output layer to match the number of sentiment classes in the task, which, in this case, is 3 (positive, negative, and neutral). When creating a custom fully connected layer using the output dimension of the DistilBERT model, the output dimension refers to the size of the hidden state representations generated by the DistilBERT model. This dimension can be accessed through the hidden_size attribute of the DistilBERT model.

For example, let's consider a scenario where the hidden_size of DistilBERT model is 768. This means that each hidden state representation has a dimensionality of 768. To create a custom fully connected layer, can use the torch.nn.Linear module provided by PyTorch. This module allows to define a linear transformation from the input to the output dimension. In this case, it would set the input size of the linear layer to 768 (matching the hidden_size of the DistilBERT model) and specify the desired number of units for the output layer. For example, if want the output layer to have 256 units, it would set the output size of the linear layer to 256.

```python
import torch
from transformers import DistilBertForSequenceClassification

# Load the pre-trained DistilBERT model
model = DistilBertForSequenceClassification.from_pretrained(model_card, num_labels=3)

# Access the output dimension (hidden_size) of the DistilBERT model
output_dimension = model.config.hidden_size

# Define the number of units for the custom fully connected layer
custom_units = 256

# Create a custom fully connected layer
custom_fc_layer = torch.nn.Linear(output_dimension, custom_units)

# Set the activation function if needed
activation = torch.nn.ReLU()

# Example forward pass through the custom layer
output = custom_fc_layer(input)
output = activation(output)

# Print the output shape
print(output.shape)
```

Figure 1

2. Analyze the Sentiment of Social Media Comments to Determine the Popularity of Candidates

An extensive methodology has been developed for implementing a sentiment analysis system using the RoBERTa transformer model, specifically tailored for analyzing sentiment in Facebook-like comments. In order to address the limitations associated with web scraping from social media platforms, which are both prohibited and illegal, a simulated Facebook clone environment has been created.

In this sentiment analysis system, RoBERTa plays a crucial role in comprehending and analyzing the sentiment expressed in Facebook-like comments. RoBERTa is a transformer-based model that utilizes a deep neural network architecture. It learns from the training data, which consists of a large corpus of labeled comments and their associated sentiments. During the training phase, RoBERTa processes the comments and captures the intricate relationships between words, phrases, and the overall sentiment expressed. Through multiple iterations of training, the model refines its internal parameters and adjusts its representation of sentiment-related features.

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

To initiate the analysis, a dataset is generated by gathering comments from the simulated Facebook clone along with their corresponding sentiment labels. This dataset serves as the foundation for training and evaluating the sentiment analysis model. By incorporating diverse comments from various sources into the clone, a wide range of sentiments expressed by users can be captured in an ethical and controlled manner.

The dataset undergoes a preprocessing stage to ensure compatibility with the RoBERTa transformer model. This involves essential steps such as tokenization, encoding, and formatting. Tokenization breaks down the comments into smaller units, such as words or subwords, while encoding assigns numerical IDs to each token. Additionally, special tokens, attention masks, and padding or truncation techniques are employed to ensure consistent sequence lengths. These preprocessing steps enable the transformation of textual data into numerical representations that can be effectively processed by the RoBERTa model.

Following data preprocessing, the sentiment analysis model is trained using the dataset. The dataset is divided into training and testing sets to evaluate the model's performance on unseen data. This division allows for an assessment of the model's ability to make accurate predictions and generalize beyond the training data. By allocating a portion of the dataset for testing, the effectiveness of the model in real-world scenarios can be measured.

During the training phase, the model is exposed to the training dataset through multiple epochs. Each epoch consists of batches of data, which are passed through the model for forward and backward propagation. The RoBERTa model learns from the input data and adjusts its internal parameters to enhance its predictive capabilities. This iterative process enables the model to capture patterns, contextual cues, and underlying sentiment indicators present in the training data.

To facilitate the training process, the Trainer class from the transformers library is utilized. This class manages various training-related tasks, such as loss function computation, parameter updates through gradient descent, and monitoring of training progress. Key metrics such as accuracy and loss are logged at regular intervals, providing insights into the model's performance and convergence over time.

Upon completion of the training phase, the model's performance is evaluated on the testing dataset to obtain an unbiased assessment of its ability to generalize to unseen data. Metrics such as accuracy and F1 score, which combines precision and recall, offer a comprehensive measure of the model's performance. These metrics provide insights into the model's strengths and weaknesses, guiding potential improvements.

Once satisfactory performance is achieved, the trained model is saved for future use. The model parameters, representing the acquired knowledge during training, are stored in a designated directory. This facilitates easy retrieval and deployment of the sentiment analysis model whenever necessary. The saved model becomes a valuable asset that can be integrated into various applications, systems, or platforms for sentiment analysis within the simulated Facebook clone environment.

During the deployment phase, the sentiment analysis model is applied to new, unseen comments within the Facebook clone. An inference process is established, involving the preprocessing of raw comments and their passage through the trained model. Preprocessing includes tokenization, encoding, and generation of input tensors. The preprocessed comments are then fed into the model, which produces predicted sentiment labels based on the output logits. These sentiment labels indicate the predicted sentiment of each comment, offering valuable insights into the sentiments expressed by users within the Facebook clone environment. Fig 1. shows a better understandment of this sentiment analysis system.

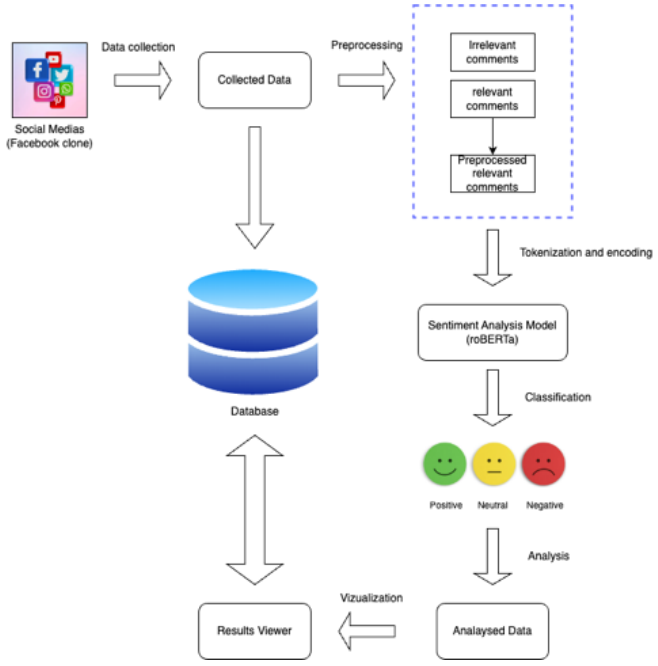


Figure 2: Sentiment analysis using social media comments

The sentiment analysis system developed through this methodology holds significant potential for various applications, allowing businesses to gain insights into customer opinions, identify emerging trends, and make data-driven decisions based on the sentiments expressed within the simulated Facebook clone environment.

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

3. Calculate the Similarity between Two Speeches of a Candidate

The methodology employed in this research aims to leverage advanced machine learning algorithms to calculate the similarity between speeches made by political candidates. The objective is to monitor candidates from previous elections, extract their voice records, convert them into text, and compare their political promises with those made in the current election period. This section provides a comprehensive and detailed description of the methodology used, which includes speech-to-text conversion, promise identification, embedding generation, similarity calculation, analysis, and visualization.

Speech-to-Text Conversion:

The initial step in the methodology involves the conversion of voice records of politicians' speeches from previous and current election periods into text format. To achieve this, a powerful speech recognition library called SpeechRecognition is utilized. This library interfaces with various speech recognition engines and provides a convenient way to transcribe spoken language into textual form. In this research, the Whisper model from the Hugging Face library is employed for speech recognition. Specifically, the WhisperForConditionalGeneration model is utilized, which is trained to recognize and transcribe speeches in Sinhala language and generate text outputs.

Promise Identification:

Once the speeches are transcribed into textual form, the subsequent step is to identify the political promises made by the candidates. This phase incorporates the application of natural language processing (NLP) techniques, which enable the analysis of the transcribed text to extract the essential components, including the promises made by the candidates. Several NLP techniques are employed, including sentiment analysis, named entity recognition (NER), and topic modeling. Sentiment analysis assists in determining the sentiment expressed in the speeches (e.g., positive, negative, or neutral), while NER helps identify relevant entities such as key political issues or specific promises. Topic modeling techniques aid in identifying the main themes or topics addressed in the speeches, providing additional context for promise identification.

Embedding Generation:

To facilitate effective comparison and similarity calculation between speeches, embeddings are generated for each speech. Embeddings are high-dimensional numerical representations of text or speech data that capture the semantic meaning and contextual information. In this research, the PyAudio library is utilized to extract audio features, and the pyannote.audio library is employed for the generation of speech embeddings. The pyannote.audio library provides a collection of pre-trained models that enable the extraction of high-quality speech embeddings. In particular, the model used in this research is the SentenceTransformer, which is based on the MiniLM-L6 architecture. The SentenceTransformer model generates fixed-length vectors that capture the semantic meaning of the speeches, allowing for efficient similarity calculations.

Similarity Calculation:

After generating the embeddings for the speeches, cosine similarity is employed as the metric for calculating the similarity between them. Cosine similarity measures the cosine of the angle between two vectors and provides a similarity score ranging from 0 to 1, where 1 indicates identical speeches and 0 indicates completely dissimilar speeches. The sklearn.metrics.pairwise.cosine_similarity function is utilized to calculate the cosine similarity between the embeddings. By comparing the embeddings of speeches from previous and current election periods, the similarity scores provide a quantitative measure of how closely related the promises made by candidates are over time.

Analysis and Visualization:

The final step of the methodology involves the analysis and visualization of the similarity scores obtained from the cosine similarity calculations. Various techniques are employed to gain insights into the similarity between speeches. Firstly, the embeddings can be visualized by plotting them using the Matplotlib library, which provides a clear representation of the changes or similarities in the speech patterns over time. Additionally, similarity matrices can be constructed to provide a comprehensive overview of the similarity between all pairs of speeches. These matrices can be visualized using heatmaps or other visualization techniques, allowing for a deeper understanding of the relationships and patterns within the speeches.

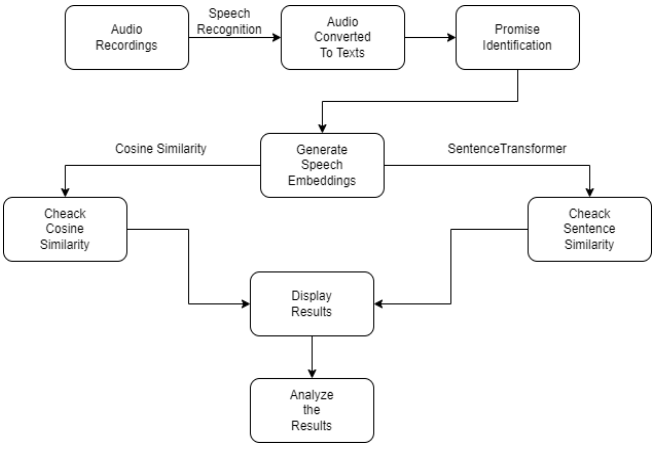


Figure 3: Audio Similarity Calculation process flowchart

By implementing this methodology, the monitoring component of the "Online Candidate Selection System for Elections" can effectively analyze and compare political speeches. The utilization of machine learning algorithms, combined with speech-to-text conversion, promise

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

identification, embedding generation, and similarity calculation, contributes to the system's ability to provide valuable insights into the consistency or changes in political promises made by candidates over time. Through the analysis and visualization of similarity scores, patterns in political rhetoric and the evolution of promises can be identified, enabling voters and decision-makers to make more informed choices.

The rigorous methodology outlined above serves as a robust foundation for the implementation of the monitoring component, ensuring transparency, accountability, and informed decision-making in the electoral process. By leveraging cutting-edge machine learning techniques, this research provides a comprehensive framework for evaluating the similarity of political speeches and contributes to the overall goal of promoting fair and democratic elections.

4. Identify and Categorize Social and Welfare Activities of Candidates using image Processing

In this research paper, we present a methodology for analyzing the activities of election candidates based on images. The objective is to develop a machine learning model that can classify images into different categories, including construction projects, environment-related activities, education initiatives, elderly care efforts, and health and social services. We utilize a supervised learning technique, specifically Keras ResNet101V2, to train the model for accurate image classification. To begin, we collect a dataset of candidate images showcasing their various activities.

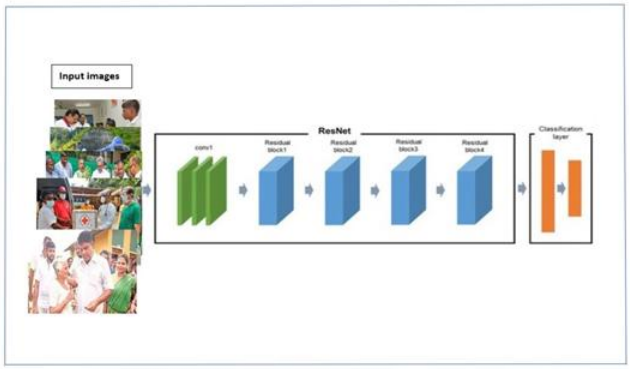


Figure 4

The dataset is annotated with corresponding activity labels, ensuring a supervised learning approach. Each image is preprocessed to ensure uniformity in size and format, resizing them to 224x224 pixels. The model architecture consists of a pre-trained ResNet101V2 convolutional neural network, followed by additional layers for classification. The ResNet101V2 serves as a feature extractor, capturing the key features of the input images. The global average pooling layer is then applied to reduce the dimensionality of the extracted features. Dropout layers are incorporated to prevent overfitting, and dense layers with varying sizes are employed to gradually transform the extracted features into the desired output classes. The model is trained using a suitable optimization algorithm, such as stochastic gradient descent (SGD) or Adam, to minimize the categorical cross-entropy loss. The training process involves feeding the labeled images through the network iteratively, adjusting the model's weights to improve its predictions. We utilize a batch size of 32 and train the model for a predetermined number of epochs. Once the model is trained, we evaluate its performance using a held-out test set. The trained model is used to classify the images in the test set, and the predictions are compared to the ground truth labels. We compute various evaluation metrics, including precision, recall, and F1-score, for each activity category. Additionally, we generate a classification report to provide an overall assessment of the model's performance. To assess the generalization capability of the model, we also consider the accuracy metric, which represents the overall correctness of the predictions across all categories. This methodology utilizes a machine learning model based on Keras ResNet101V2 to classify images of election candidates' activities into different categories. The model's performance is evaluated using precision, recall, F1-score, and accuracy metrics. By accurately categorizing the activities, we can gain insights into candidates' focus areas and contributions, facilitating a more informed evaluation of their suitability for election.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 resnet101v2 (Functional)    (None, None, None, 2048)  42626560

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 256)               131328

 dropout_2 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 5)                 1285

=================================================================
Total params: 43,808,261
```

Figure 5

IV. RESULTS

The pretrained DistilBERT model was tested on the test data gathered by news reports. For each news text in the news variable, encode it using the tokenizer

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

(tokenizer.encode(news_)) and calculate the length of the resulting tokenized sequence. The lengths of all tokenized sequences are stored in the token_lengths list. It use seaborn's distplot() function to create a histogram that visualizes the distribution of token lengths. The token_lengths list is passed as the input to the distplot() function. X-axis of the histogram was represented by 'Token Count' and y-axis was represented by 'Density'. Using plt.xlim([0, 256]) x-axis plot was restricted the range to show token counts up to 256. Finally the plot was displayed using plt.show().

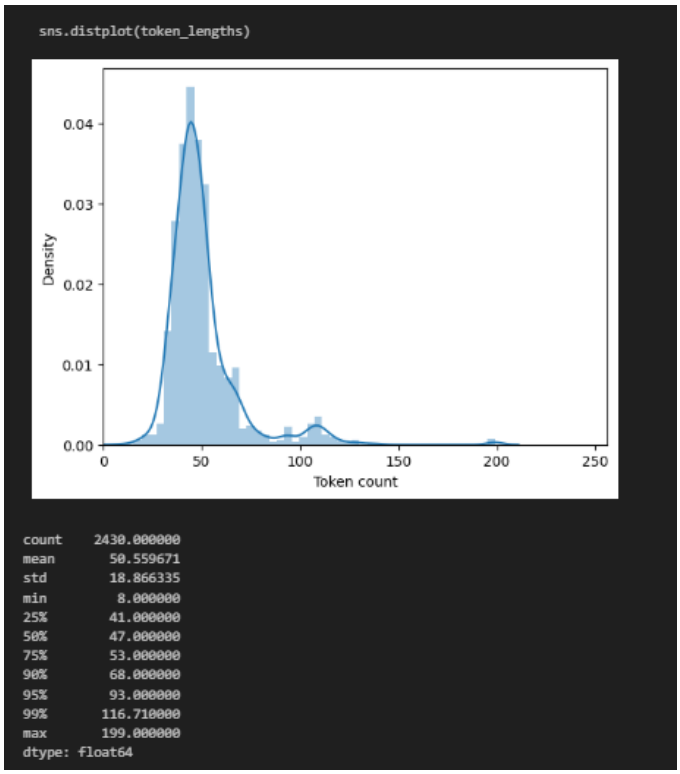


Figure 6

Also the system outputs the training progress and metrics recorded during the training process. Each dictionary entry corresponds to a specific training step or epoch and contains information such as the loss value and learning rate at that point. Loss value is the value of the training loss at that particular step or epoch and learning rate is the learning rate used during that step or epoch. The output shows the training progress over multiple epochs. For each epoch, can see the loss and learning rate values at different stages of the epoch. The final output dictionary summarizes the entire training process, including the total runtime, average samples and steps processed per second, final loss value, and the epoch number. This information is helpful for monitoring the training process, assessing convergence, and evaluating the model's performance.

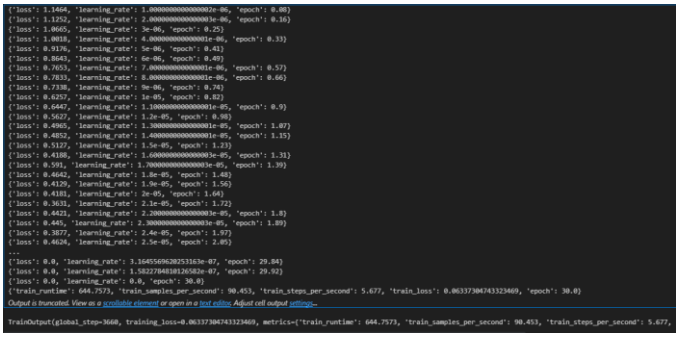
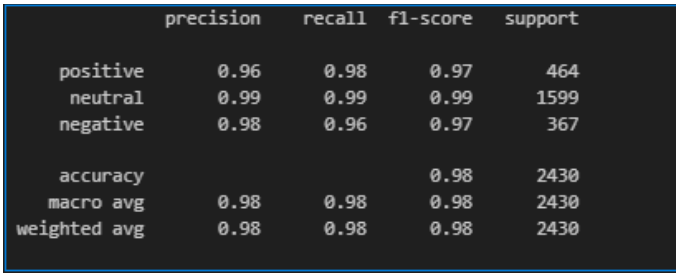


Figure 7

Not only that system performs inference on news data using a loaded model and generates a classification report. It defines a function process_raw_news() that processes raw news text by tokenizing it using the tokenizer and creating an encoding with a maximum length of 150 tokens. The inference_on_news() function takes a news text, processes it using process_raw_news(), and performs inference using the loaded model. It passes the input_ids and attention_mask tensors to the model and retrieves the output logits. The logits are converted to numpy arrays and the predicted class label is obtained by finding the index of the maximum value. The predicted class label is then converted back to its original label using the class_dict_reverse dictionary.

The system creates two lists, P and Y, for storing the predicted labels and the true labels, respectively. It iterates over the news list and calls inference_on_news() to get the predicted label for each news text. The predicted labels are stored in P and the true labels are stored in Y. Finally it uses scikit-learn's classification_report() function to generate a classification report based on the true labels (Y) and predicted labels (P). The classification report provides metrics such as precision, recall, and F1-score for each class (positive, neutral, and negative), as well as overall accuracy, macro average, and weighted average metrics.



	precision	recall	f1-score	support
positive	0.96	0.98	0.97	464
neutral	0.99	0.99	0.99	1599
negative	0.98	0.96	0.97	367
accuracy			0.98	2430
macro avg	0.98	0.98	0.98	2430
weighted avg	0.98	0.98	0.98	2430

Figure 8

When generating confusion matrix it compares the true labels with the predicted labels obtained from the previous step. It normalizes the matrix to represent the percentage of predictions for each class. The code then

creates a heatmap using the seaborn library, where each cell in the heatmap corresponds to the percentage of predictions for a specific combination of true and predicted labels. The heatmap is annotated with the values inside each cell and displayed with appropriate labels and a title. The resulting visualization provides an overview of the model's performance by showing the distribution of correct and incorrect predictions across different classes.

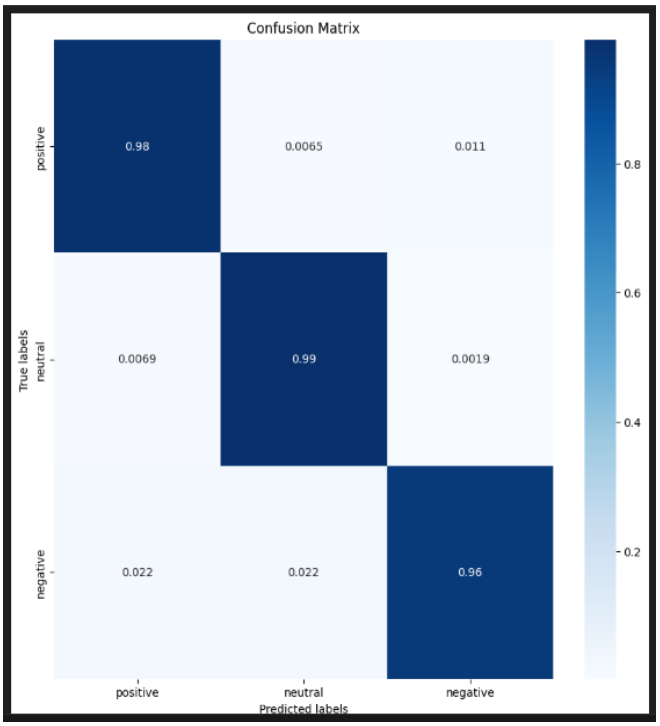


Figure 9

The sentiment analysis system provides a comprehensive set of statistical results and outputs that enable us to evaluate its performance and gain insights into the dataset. One important aspect is the analysis of token lengths, which allows us to understand the complexity and structure of the comments. By calculating the length of tokens for each comment using the RoBERTa tokenizer, we obtain valuable information about the composition of the dataset.

To visualize the distribution of token lengths, we plot a distribution graph. This graph shows the frequency of different token lengths in the dataset, offering an overview of the comment lengths and their distribution. The x-axis represents the token count, while the y-axis represents the frequency of comments. This analysis helps us understand the variability in comment lengths and provides insights into the dataset's overall composition.

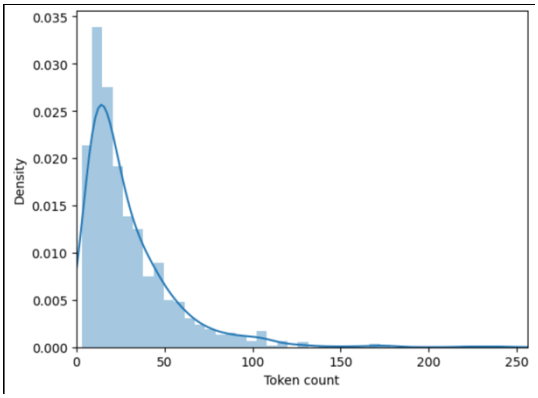


Figure 10: Distribution graph of Token lengths

During the training process, the system provides updates on the loss and learning rate at different epochs. The loss value indicates how well the model is performing in terms of minimizing the difference between predicted and actual sentiment labels. By monitoring the loss, we can assess the model's convergence and its ability to capture sentiment information effectively.

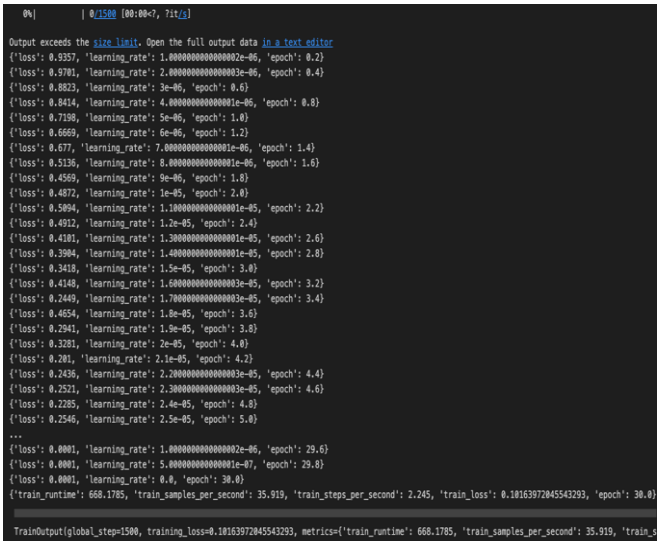


Figure 11: Training process output

- The training process consists of 30 epochs (iterations over the training dataset).
- The training starts with a relatively high loss value of 0.9357 and gradually decreases over the epochs.
- The learning rate, which determines the step size for parameter updates, starts at 1.0000000000000002e-06 and increases slightly in each epoch.
- The loss continues to decrease, indicating that the model is improving its performance and fitting the training data better.

- The loss values progressively decrease, reaching a minimum value of 0.0001 in the later epochs.
- The training process completes with a final loss of 0.10163972045543293.
- The training runtime is approximately 668.1785 seconds, with an average of 35.919 samples per second and 2.245 steps per second.

Overall, these results suggest that the sentiment analysis model undergoes effective training, continuously improving its performance as the training progresses. The decreasing loss indicates that the model is successfully learning patterns and making accurate predictions on the training data.

The learning rate represents the rate at which the model adjusts its parameters during training. By tracking the learning rate, we can understand the model's adaptability and its ability to optimize performance over time. The updates on loss and learning rate at each epoch give us a comprehensive view of the training progress and allow us to analyze the model's performance at different stages.

In addition to loss and learning rate updates, the system also reports training metrics at each epoch. These metrics include various performance indicators that provide insights into the model's training process. By examining these metrics, such as accuracy, precision, recall, and F1 score, we can assess the model's performance in classifying sentiments accurately.

Finally, at the end of training, the system provides a summary of the training process, including the total number of steps, training loss, runtime, and other performance metrics. This summary offers a comprehensive overview of the training performance and enables us to evaluate the efficiency of the sentiment analysis model.

From the plot embeddings following results are shown:

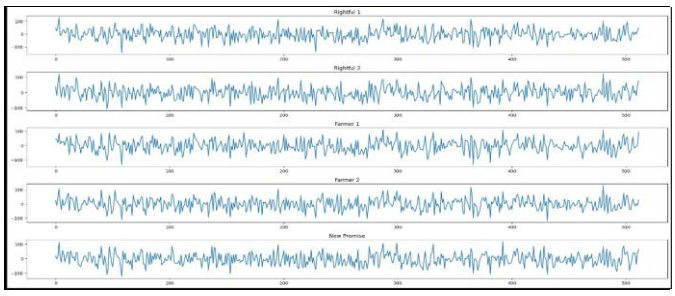


Figure 12: Plot of the embeddings of sample records displayed separately

Here all the sentence embeddings are show separately at first. The plots of individual speech embeddings reveal the patterns and variations present in the speeches.

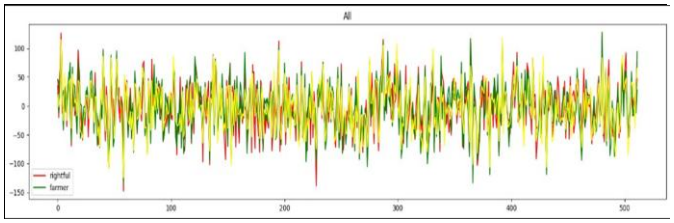


Figure 13: Plot of the embeddings of sample records displayed altogether, by manually annoting similar colors to similar voice records

The combined plot of all embeddings provides an overview of the similarities and differences between speeches, with different colors representing different speech categories.

	r1	r2	f1	f2	n1
r1	1.000000	0.799298	0.842400	0.850907	0.848873
r2	0.799298	1.000000	0.778198	0.778697	0.809464
f1	0.842400	0.778198	1.000000	0.821966	0.849671
f2	0.850907	0.778697	0.821966	1.000000	0.862467
n1	0.848873	0.809464	0.849671	0.862467	1.000000

Figure 14: Cosine similarities.

The cosine similarity scores between embeddings show the degree of similarity or dissimilarity between pairs of speeches. These scores help quantify the audio-based similarity between speeches.

Test Cases:

1)

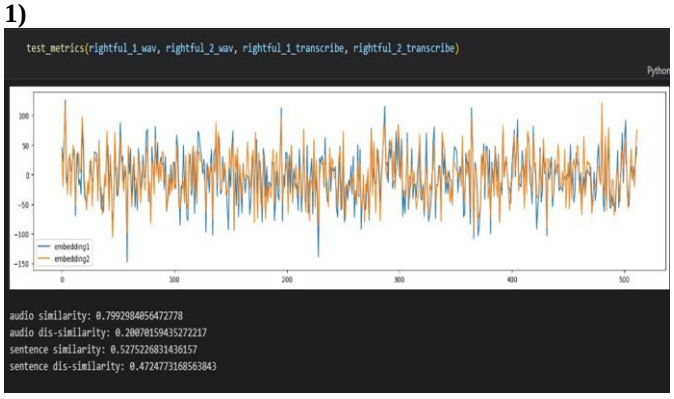


Figure 15: Test Case 1

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

2)

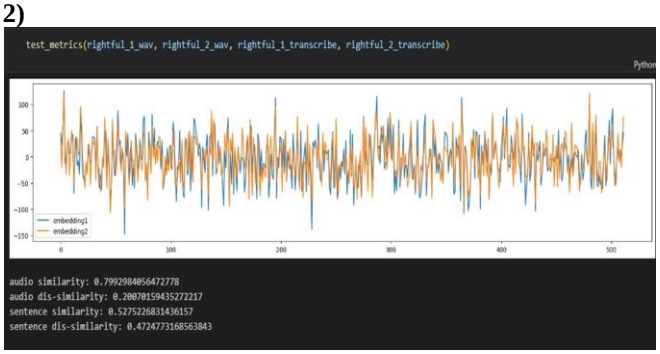


Figure 16: Test Case 2

3)

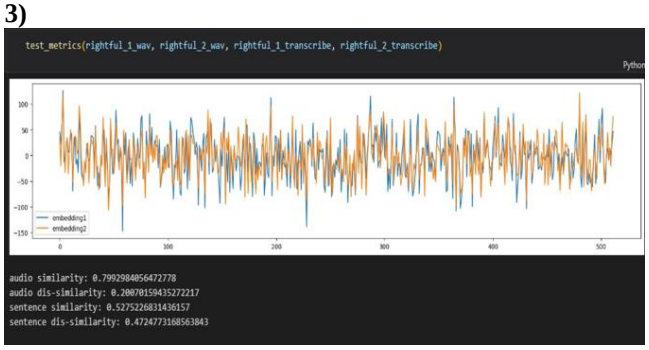


Figure 17: Test Case 3

In each case, the results include both audio similarity and dissimilarity scores, as well as sentence similarity and dissimilarity scores. These scores indicate the degree of similarity or dissimilarity between the corresponding speeches based on their audio embeddings and sentence embeddings. The higher the similarity score, the more similar the speeches are, while a higher dissimilarity score indicates greater differences between the speeches.

By analyzing the audio and sentence embeddings and comparing the speeches using different similarity metrics, the code enables a quantitative and qualitative evaluation of the promises made by candidates from previous elections. These findings contribute to the candidate selection process in the "Online Candidate Selection System for Elections" by providing insights into the consistency, reliability, and novelty of political promises.

The classification report obtained from the evaluation of the trained model provides valuable insights into its performance and the accuracy of the activity categorization. The report reveals precision, recall, and F1-score metrics for each activity category, as well as the overall accuracy of the model. The precision metric measures the proportion of correctly predicted instances within a specific category, indicating the model's ability to avoid false positives.

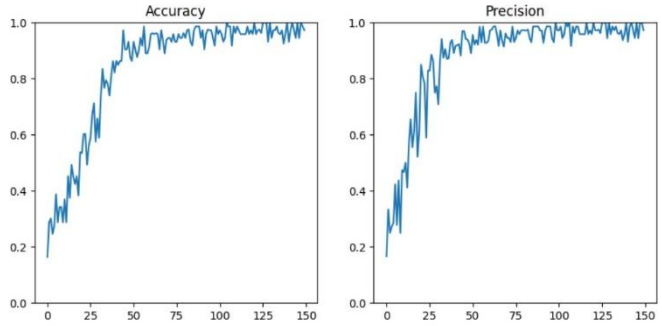


Figure 18

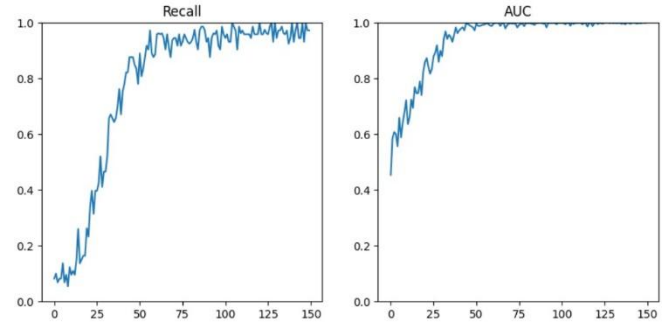
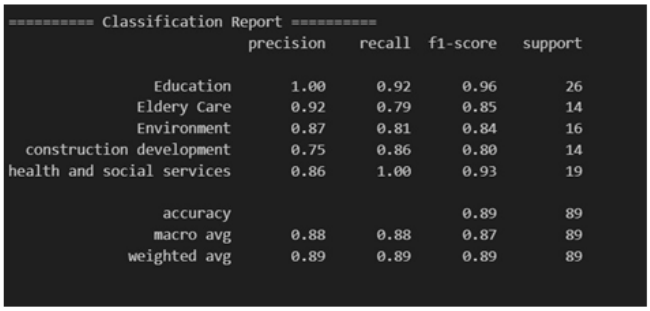


Figure 19

In our study, the precision scores for the different categories are quite high, demonstrating the model's capability to accurately classify candidate activities. For instance, the "Education" category achieves a precision score of 1.00, indicating that all predicted instances in this category are correct. The "Elderly Care" category also shows a high precision score of 0.92, implying that the model effectively identifies candidate activities related to this domain. The recall metric represents the proportion of true positive instances that are correctly identified by the model within each category. A high recall score indicates that the model effectively captures the relevant activities associated with a particular category. In our research, the recall scores for most categories are relatively high, demonstrating the model's ability to identify a significant number of activities accurately. For instance, the "Health and Social Services" category achieves a recall score of 1.00, indicating that the model correctly identifies all instances related to this category. The F1-score is the harmonic mean of precision and recall, providing an overall measure of the model's accuracy in categorizing activities. A high F1-score implies a balance between precision and recall, suggesting that the model performs well in classifying activities across all categories. In our study, the F1-scores for most categories are commendable, ranging from 0.80 to 0.96. This indicates

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

that the model achieves a good trade-off between precision and recall, ensuring accurate and consistent activity classification.



========== Classification Report ==========

	precision	recall	f1-score	support
Education	1.00	0.92	0.96	26
Eldery Care	0.92	0.79	0.85	14
Environment	0.87	0.81	0.84	16
construction development	0.75	0.86	0.80	14
health and social services	0.86	1.00	0.93	19
accuracy			0.89	89
macro avg	0.88	0.88	0.87	89
weighted avg	0.89	0.89	0.89	89

Figure 20

The overall accuracy of the model is reported as 0.89, representing the percentage of correctly predicted instances across all categories. This demonstrates that the model performs well in categorizing the activities of election candidates based on images. The high accuracy score indicates that the model is reliable and robust in its predictions, providing valuable insights into the focus areas and contributions of the candidates. The evaluation results reveal that the trained model successfully categorizes election candidates' activities with high precision, recall, and F1-scores. The model's overall accuracy of 0.89 further emphasizes its effectiveness in accurately identifying and classifying various activity categories. These results highlight the potential of the proposed methodology for analyzing and understanding the activities of election candidates, contributing to a more comprehensive evaluation of their suitability for public office.

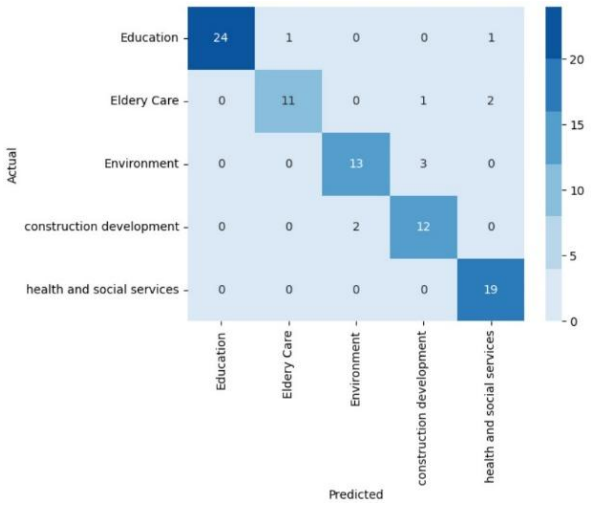


Figure 21

V. CONCLUSION

In conclusion, the proposed online platform for political parties in Sri Lanka offers a comprehensive solution to improve the candidate selection process. By incorporating advanced technologies such as sentiment analysis, background checks, aptitude tests, and promise analysis, the system aims to address the issues of false reputations, misleading practices, and lack of accountability. It provides benefits such as ensuring the selection of competent and trustworthy candidates through background checks, enabling voters to make informed decisions based on genuine trust, and promoting transparency and accountability through the evaluation of candidates' promises and past performances.

Overall, the implementation of this platform has the potential to enhance the integrity and democratic nature of elections in Sri Lanka. Through its features and advanced technologies, the system aims to restore trust among voters by selecting candidates based on merit and credibility. By providing voters with reliable information and data-driven evaluations, it empowers them to make informed choices and holds candidates accountable for their actions and promises. The proposed platform represents a step forward in creating a transparent, efficient, and responsible political culture, ultimately contributing to the overall well-being of society in Sri Lanka.

REFERENCES

[1] Achmad Maududie, Windi Eka Yulia Retnani & Muhamat Abdul Rohim. *An approach of web scraping on news website based on regular expression*.

[2] Xianyong Yi, Rongge Zheng, Aoyu Wang, Hao Qin & Yufeng Chen. *Design and implementation of Word2Vec parallel algorithm based on HPC*.

[3] Himanshu Suman & Anurag Singh Kushwaha. (2020). *Candidate background verification using machine learning and fuzzy matching*.

[4] Renáta Németh. (2022). *A scoping review on the use of natural language processing in research on political polarization: Trends and research prospects*.

[5] Y. Liu et al. Roberta: A robustly optimized Bert pretraining approach. *arXiv.org*. Available at: https://arxiv.org/abs/1907.11692. (Accessed May 22, 2023).

[6] Y. Bae & H. Lee. (2012). Sentiment analysis of Twitter audiences: Measuring the positive or negative influence of popular twitterers. *JASIST*, *63*(12), 2521-2535.

[7] A Restaurant recommender system based on sentiment analysis. Elham Asani a, Hamed Vahdat-Nejad a, Javad Sadri b a Faculty of Electrical and

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.18

Computer Engineering, University of Birjand, Birjand, Iran b Computer Science and Software Engineering Department, Concordia University, Montreal, Quebec, Canada

[8] Sentiment Analysis on Twitter Data Varsha Sahayak Vijaya Shete Apashabi Pathan BE (IT) BE (IT) ME (Computer) Department of Information Technology, Savitribai Phule Pune University, Pune, India.

[9] TWEETEVAL: Unified Benchmark and Comparative Evaluation for Tweet Classification Francesco Barbieri Jose Camacho-Collados† Leonardo Neves Luis Espinosa-Anke† Snap Inc., Santa Monica, CA 90405, USA †School of Computer Science and Informatics, Cardiff University, United Kingdom.

[10] Candidate selection procedures for the European elections Employee Candidate Selection Systems Based on Web Umniy Salamah, Muhammad Prima Permana Faculty of Computer Science, Universitas Mercu Buana, Jakarta, Indonesia.

[11] Machine Learning-Based Sentiment Analysis for Twitter Accounts Ali Hasan 1, Sana Moin 1, Ahmad Karim 2 and Shahaboddin Shamshirband Machine Learning-Based Sentiment Analysis for Twitter Accounts Ali Hasan 1, Sana Moin 1, Ahmad Karim 2 and Shahaboddin Shamshirband Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, VietnamReceived: 16 January 2018; Accepted: 24 February 2018; Published: 27 February 2018

[12] An Experiment In Candidate Selection Katherine Casey Abou Bakarr Kamara Niccoló Meriggi Working Paper 26160 http://www.nber.org/papers/w26160 NATIONAL BUREAU OF ECONOMIC RESEARCH 1050 Massachusetts Avenue Cambridge, MA 02138 August 2019, Revised July 2021.

[13] He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770-778.

[14] H. Singh, N. Helian, R. Adams & Y. Sun. (2022). Sentiment analysis using BLSTM-ResNet on textual images. *International Joint Conference on Neural Networks (IJCNN), Padua, Italy*, pp. 1-8. DOI: 10.1109/IJCNN55064.2022.9892883.

[15] C. Kochgaven, P. Mishra & S. Shitole. (2021). Detecting presence of covid-19 with resnet-18 using pytorch. *International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India*, pp. 1-6. DOI: 10.1109/ICCICT50803.2021.9510085.

[16] A. Michele, V. Colin & D. D. Santika. (2019). Mobilenet convolutional neural networks and support vector machines for palmprint recognition. *Procedia Comput. Sci., 157*, 110–117. DOI: 10.1016/j.procs.2019.08.147.

[17] N. Gouda & J. Amudha. (2020). Skin cancer classification using ResNet. *IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India*, pp. 536-541. DOI: 10.1109/ICCCA49541.2020.9250855.

[18] H. Jabnouni, I. Arfaoui, M. A. Cherni, M. Bouchouicha & M. Sayadi. (2022). ResNet-50 based fire and smoke images classification. *6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sfax, Tunisia*, pp. 1-6. DOI: 10.1109/ATSIP55956.2022.9805875.

[19] O. Sudana, I. P. A. Bayupati, & D. G. Yudiana. (2020). Classification of maturity level of the mangosteen using the convolutional neural network (CNN) method. *Int. J. Adv. Sci. Technol., 135*, 37-48. DOI: 10.33832/IJAST.2020.135.04.

[20] S. Karthikeyan et al. (2020). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computing and Applications, 32*(11), 1-24.