International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Text Marker -Text Evaluation Tool for Marking AI and Real Knowledge-based Entries and Responses

1 D.Sreya, 2 K.Srivalli Varshini, 3 K. Ashrith, 4 Asst. Prof.K.Priyanka

Bachelors in Computer Science and Engineering AIML, Geethanjali College of Engineering and Technology, Hyderabad, Telangana, India

4 Assistant Professor, Computer Science, Geethanjali College of Engineering and Technology, Hyderabad, Telangana, India

---

Abstract - In today's world, because of how quickly artificial intelligence writing tools are getting better, it's now hard to tell whether something was written by AI or a person. This makes things complicated for students writing papers, for checking if information is true, and for making sure work is original. A system for identifying AI versus human writing has been built to automatically sort text into one of these two groups. It's built around a RoBERTa transformer model – this is the main part that decides what something is, and it's very good at grasping the meanings of words and how language works. People using it can type or copy and paste text into a webpage, and the system will analyze it immediately. It's been created using Python's many options and put on Railway(deployment tools) so anyone can get to it online easily. It reliably and quickly says whether text is AI or human, and how sure it is of its answer, which is helpful for school, research and looking at content in general.

Keywords: Artificial Intelligence, RoBERTa, Natural Language Processing, Text Classification, AI-Generated Text.

1. INTRODUCTION

In recent times, Artificial Intelligence (AI) has advanced considerably, with major developments in processing and generating human language. Applications such as ChatGPT, Gemini, and other AI writing tools are at the forefront of artificial intelligence, allowing users to produce a wide range of documents such as essays, reports, articles, emails, and correspondence in almost no time. Despite the fact that these AI tools fundamentally improve the time and convenience of content creation, they have also brought about a significant barrier to the detection of the authorship of a text. In fact, this problem is of utmost concern in academic circles, online publication, the world of research, and professional communication where the roles of originality, the attributed source, and authenticity are considered paramount.

Thanks to the introduction of AI-based writing, it has now become a huge burden to analytically uncover whether text is or not. Is it a work of a man or one worked on by a machine? AI-generated text may be rather accurate grammatically, logically arranged, and meaningful in context, which leads to its close imitation of human writing. Therefore, smart systems that can enable the following are on the rise. recognizing automatically the source of the written text. Such systems might be of great assistance in content moderation, analysis in relation to plagiarism, academic assessment, and maintaining faith in the digital. communication environment.

The above issue is desired to be solved by the project AI vs Human Text Detection System, which will develop a machine learning-driven project that classifies the provided text into two categories: AI-written and written by a person. Its mechanism provides its users with the capability of typing/pasting text into a text box using a very simple and user-friendly web interface. When a text is submitted, it is run through the processing and analysis of the underlying classification model, and the results are immediately provided to the user. The system is therefore user-friendly, interactive, and can be used in real time.

RoBERTa (Robustly Optimized BERT Approach) is on the 2nd core of this project because it is extremely powerful. A language model built on a transformer that is highly utilized in text classification tasks. This is why I have opted to use one of the reasons. Its powerful context-filling processing is RoBERTa. it works well, and that it works, in natural language. exceedingly adept in a range of NLP exercises. In this project, the model is the following: is set up to detect linguistic patterns, sentence structure, contextual relations, and features of style that differentiate. Texts written by humans using AI.

To simplify it and make it implementable, the whole system is developed in Python, some libraries which include Transformers, PyTorch, Pandas, Scikit-learn, and above all, the project is hosted on Railway to make the application accessible through a Web interface. Besides just being a classifier, the project also comes with a number of very nice output features, such as prediction confidence, visualization of the workflow, logs, and keeping track of the history, which enhance transparency and make it easier for the user.

2. RELATED WORK

As artificial intelligence continues to evolve quickly in the production of text, such researchers and software engineers have been seeking how best to distinguish between machine-generated text and human-created text. Previously, such common machine learning models as Naive Bayes, SVM, Random Forest, and Logistic Regression were typically used in such a task. These techniques employed human control over the choice of the features such as the frequency of words, length of sentences, grammar, and style of writing. It should also be kept in mind that despite these models being beneficial for simple classification purposes, they hardly managed to grasp deep contextual knowledge and complex language structures. Along with the rise of Natural Language Processing (NLP), deep learning and transformer-based models have been utilized for text classification. For instance, models such as BERT, RoBERTa, and DistilBERT are widely used because they effectively capture semantic relationships, contextual information, and sentence-level patterns.

Many papers give evidence that transformer architectures surpass traditional machine learning techniques in the task of recognizing AI-generated texts due to their superior ability to learn contextual representations. Several detection systems nowadays, to some extent, rely on comparisons of outputs produced by multiple modern AI language models, such as ChatGPT and similar systems, to study differences in writing style repetition, coherence, and predictability. Nevertheless, quite a few of them are either complicated, very expensive computationally, or not very feasible for deployment to the real-time environment. For this project, RoBERTa has been selected as the primary model in light of the fact that it is well known for being able to understand contexts thoroughly, performing very well in text classification tasks of the binary type, and being a very stable model.

Our proposal aims at offering a simple, user-friendly, and working tool that is capable of detecting AI vs. human text live.

3. PROBLEM STATEMENT

The fast growth of AI has led to the invention of. Very advanced text generation programs, like Small language models ChatGPT, Gemini, and other large language models. These AI generators of text are able to produce coherent, grammatically. in a matter of accurate, and contextually relevant text. seconds. Although the technology has various benefits, it also comes with a significant negative feature, namely how to determine AIgenerated. text of human-written text? Oftentimes, AI material is so, as is the case with human writing, which is subject to manual detection. nearly impossible.

This problem has left a very strong impact on school, university, and research work, publishing on the internet, and professional communication regions, where the requirement of original, authentic work is very high. Students might use AI-generated assignments, authors might believe that it is okay to use AI-written articles, and one might think that the web is flooded with machine-created text. In a world where there is no efficient content detection method, it would be very tough to keep the principles of justice, trust, and lucidity in written communication. Current content detection methods suffer from low efficiency, poor user-friendliness, or are unsuitable for real-time use. There are mechanisms that are dependent solely on. dumb keyword analysis or superficial statistical characteristics, which are certainly not sufficient to locate the latest AI-created text. Therefore, a very intelligent and efficient system is needed that is able to analyze text automatically and ascertain its origin. This proposal will be designed to develop a credible and advanced text classifier (computer-based) which accepts user text as input and output whether the text came out human or AI-generated. through the use of a state-of-the-art NLPbased model.

4. METHODOLOGY

The design of the system is based on NLP (Natural Language). Text processing) and deep learning models to process text. and define the relation between it being human or AI. model. The various major steps involved in the project are different. Similar to preparing databases, training the tokenization of the text. the model, predictions, interpretation of the outcomes, and deployment. When it is very important, every step is very significant. resorts to creating a really useful and very simple-to-use. text categorization system. Detailed methods are given below.

4.1 Dataset Collection

Before anything else, the project needs to have good training and testing of the dataset can be done using a dataset that is available. model. The data in this project was composed of two primary. columns: Input textual content will be in a text column, and the column created refers to the corresponding class label:

- 0 Human-written text
- 1 AI-generated text

The dataset has human as well as AI-generated samples, which makes it possible to use this data for a binary text classification task. One of the features of the datasets demonstrated that the data has mostly human-written samples as compared to AI-generated ones.

This is the foundation of the whole detection system and this dataset. It is used as the primary tool for training the model to be able to detect writing. language flow, styles, and writing patterns variations between human and AIgenerated text.

4.2 Data Preprocessing

Preprocessing of data provides an excellent initial step in training models by improving the quality of the data and getting rid of the superfluous. inconsistencies. It plays a major role in the overall process; well-organised and clean data always gets superior results. using machine learning algorithms.

The preprocessing steps done in this project are:

- Deleting the empty or empty values.
- Converting the text data to the string type.
- Getting rid of the repeating text samples .
- Preparing the data for model input .

These preprocessing activities guarantee that the dataset is reliable and can be trained and tokenized. Cleaning the input data not only assists the model in learning important stuff. patterns in a more efficient way, but it also decreases noise during. classification.

4.3 Train-Test Data Splitting

The data undergoes preprocessing after which it is split into the training and learning on test sets in the train- test split method. In this case, 90% of the data is consumed in the training project, with the rest 10% to be used as a test.

A stratified division is predetermined in order to maintain the shares of AI-generated and human-written. text classes of the training and testing dataset. The tactic will be effective in making the model analysis just. And the testing process will be an actual reflection of the model. Would in reality do. The training dataset helps the model to learn and perceive the differences between the two kinds of text. The testing data will, however, be used to determine the level of performance of the. The learned model works on hidden inputs.

4.4 Tokenization and Text Encoding

The system assigns numerical representations. The tokenizing converts text to numbers. RoBERTa conversion is done by the tokenizer. It is similar to RoBERTa. model's training patterns. This is done by breaking the words into. subunits. The subunits are numbered. This allows models to deal with input data. Each token gets a unique identifier. The system allocates numerically. representations.

The process of tokenizing has the following steps:

- Break down the text into smaller parts known as tokens.
- Thereupon, encode each of these tokens into a series of integers, called input IDs.
- In a separate tensor, we denote which tokens must be attended to and which ones should not (such as padding tokens) in a matrix that we refer to as attention masks.
- Truncation and padding can be used to achieve the need that sequences in a batch are of equal length.
- A sequence length is set to a maximum of 256 characters to make the text standard encoded so that the model can operate with the text.
- The concept of tokenization gives the model a better understanding of the meaning and structure of the text.

4.5 RoBERTa-Based Classification Model

The classification in the system has the model as its principal component. It is trained on RoBERTa (Robustly Optimized BERT Approach), a transformer model that is intended to process higher-level natural language understanding tasks.

- Previously used to deal with sequence classification using ROBERTa with two output labels:
  - 0 - human-written
  - 1 - AI-generated.
- It excels at recognizing sentence context, semantic meaning, language flow, writing style, and patterns in text.
- This precision enables the system to reliably determine the origin of the text, whether it is produced by a human or an AI tool.
- RoBERTa is selected as it can consistently detect these features in the text of the real world.
- And it has been performance tested with a variety of writing samples showing consistent results across. styles and formats.

4.6 Model Training Process

The RoBERTa model is trained using the preprocessed set of data and the Hugging Face Trainer API. During the training, pattern analysis of datasets familiarizes the model with the notion of detecting differences between AI-generated and human-written text.

The training structure adopted in the project is:

- Epochs: 2
- Batch size: 16
- Learning rate: 2 x10^{-5}
- Weight decay: 0.01

Loading the training dataset, Tokenized text is given as input to the RoBERTa model. Predictions are compared to the actual labels. Loss is computed. Model weights are revised towards improved performance.

The model can be able to identify the various types of text classes increasingly with a higher level of accuracy through learning over and over.

4.7 Evaluation Metrics

After the training period, a group of typical classification metrics is tested on the model in order to test its performance. functions. These measures are some of the instruments through which we gauge the capability of the system to clearly differentiate. the text written by AI and by human beings.

The criteria of evaluation in this project are:

- Accuracy
- Precision
- Recall
- F1-score

These metrics are important, as they provide a balanced opinion as to how the model works, especially in cases where the dataset is small. even somewhat undeveloped. The use of multiple measures of evaluation will ensure the model is not measured based on these. on just one measure of performance.

Metric	Description
Accuracy	Shows how correctly the model classifies all input samples.
Precision	Indicates how many of the texts predicted as AI are actually AI-generated.
Recall	Measures the model ability to identify all actual AI-generated texts.
F1- score	Combines precision and recall into a single metric to evaluate model performance effectively.

4.8 Performance Visualization

A number of graphical analysis methods have been applied to this project to aid in understanding the behavior and performance of the model better. The charts and plots provide a graphical method of evaluating the extent to which the model has learnt, as well as the failure points.

4.9 Prediction System

Once the model has been trained and saved, a prediction function is generated to categorize the new input text of users. The prediction system works in real time and follows these steps:

- User types or copies some text to the input box.
- RoBERTa tokeniser tokenises the text.
- The tokenised input is sent through the trained model.
- The model predicts the class label

The system displays:

- AI Generated
- Human Written
- Prediction Confidence

This prediction mechanism will make the project be a practical and interactive application, and not an extension training model.

4.10 User Interface and Deployment

A web-based interface was developed to enable users to interact with the system to ensure that the system is simplified. application. These features are part of the interface:

- Text input area
- Prediction button
- Result display
- Logs
- Workflow visualization
- History tracking

Fig.4.10.1 Workflow

With the assistance of the railway, the project is hosted on the web, which allows users to connect to the system using their browsers without having to install the application on their machines. The application becomes more user-friendly and open to all with its deployment.

4.11 Tools and Technologies

The tools and technologies that are used to implement the project are:

- Python - for overall development
- Pandas - for dataset handling
- Scikit-learn - for preprocessing and evaluation
- PyTorch - for deep learning model support
- Transformers (Hugging Face) - for RoBERTa model and tokenizer ● Matplotlib / Seaborn - for visualization
- Railway - for deployment

These technologies are jointly used to create an all inclusive, effective AI vs. human text classification system.

5. SYSTEM ARCHITECTURE

The architecture of the system is designed to allow the proposed AI vs Human Text Detection System to discover whether a text is written by a person or an AI program based on the text typed by a person. The entire system is hosted on Railway that not only enables online access but also allows users to engage with the app through a web-based User Interface.

The starting point of the architecture is the Preprocessing Module, where the input text is cleaned and prepared for analysis. Following the pre-processing step, the text is delivered to the Tokenization Module that transforms the textual information into a format comprehensible by the machine which fits the model. Then, the processed tokens are given. to the RoBERTa Model which indeed works as the major classification engine of the system. Taking into account the contextual and linguistic characteristics, the model feeds on the content of the input and sends the result to the. Prediction Module.

The Prediction Module creates the ultimate classification result and dispatches it to the User Interface, where the user will see the result. Meanwhile, the forecasting output is stored in History Storage to use later, whereas the details of the system operation are logged in the Logs Module that is used to monitor and track the predictions. This architecture forms the basis of the clean, streamlined and user-friendly nature of the text classification real time workflow.

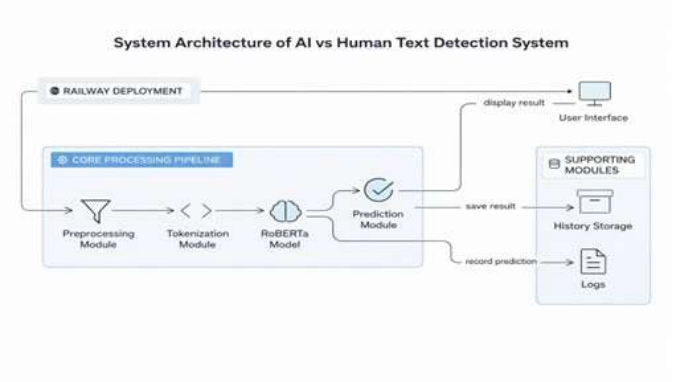


Fig. 5.1 System Architecture

6. RESULTS AND DISCUSSION

6.1 Training Performance

The RoBERTa model of the prepared dataset was trained as a chance to test the. performance of the proposed AI vs Human Text Detection system. The model could classify text as the training progressed using contextual patterns, sentence structures, and linguistic features as a means of analyzing text.

The model became steadily more predictive, with fewer losses in terms of value, performance-wise, as training progressed. In the beginning, the model produced less accurate predictions

due to randomly produced weights. initialised. Nevertheless, the model was able to learn with further training to distinguish the origin of the text appropriately, whether that was a product of an AI system or a human writer.

A training and validation loss curve analysis indicates that the model has been able to learn effectively and reduce overfitting greatly. Changes in the performance of the model across epochs are a sign of success in the model. identification of meaningful patterns in the data.

6.2 Model Behaviour Analysis

After training, the model was successful in identifying differences in the writing styles of AI-generated and human-written text. The key behaviours, which we identified are:

Pattern recognition: The model can identify the organised and repetitive ones, which are game changers of the AI-generated texts. Contextual comprehension: The model analyzes the sentence flow and logic. coherence of the given text.

Language style detection: The model is able to differentiate between the different natural human writing styles and the more standardized AI-generated responses. The model obscures significant issues of linguistic structure and issues of classification out of the deep levels of linguistics.

6.3 Performance Metrics

The performance of the model is evaluated using standard classification metrics:

Confusion Matrix

The confusion matrix indicates that most samples are correctly classified, with **1603 human-written and 1298 AI-generated texts** identified accurately. Only a few misclassifications are observed, indicating high accuracy and strong model performance.

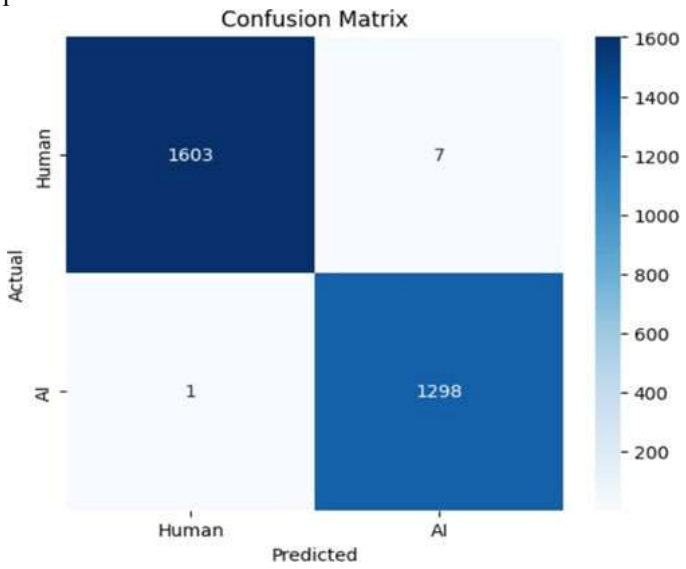


Fig.6.3.1 Confusion Matrix

Precision Recall Curve:

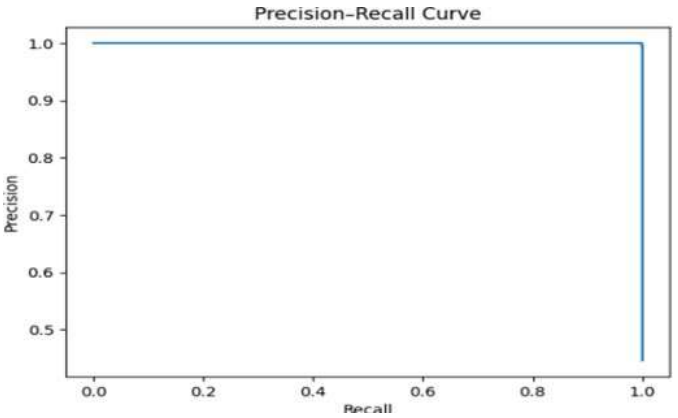


Fig.6.3.2 Precision Recall Curve

The Precision-Recall curve helps assess the model's performance in identifying AI-generated content, especially in cases of class imbalance. In the plotted graph, the curve appears near the top right area, which indicates high precision and recall. This means the model performs accurately with very few misclassifications.

F1 Score per Epoch

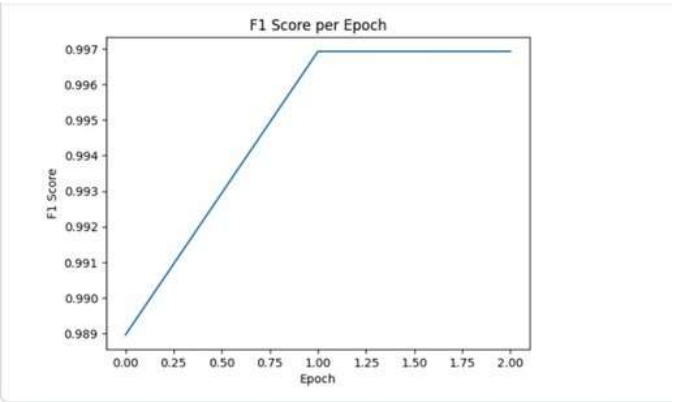


Fig.6.3.3 F1 Score per Epoch

The F1 Score per Epoch graph shows the improvement of the model during training. From the graph, the F1- score increases from 0.989 to 0.997, indicating better performance after training. The stable high value in later epochs shows that the model has learned effectively and provides accurate classification results

Class-wise Performance Chart

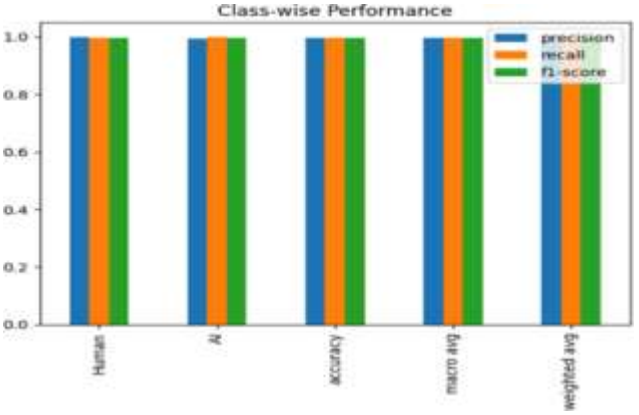


Fig.6.3.4 Class-wise Performance Chart

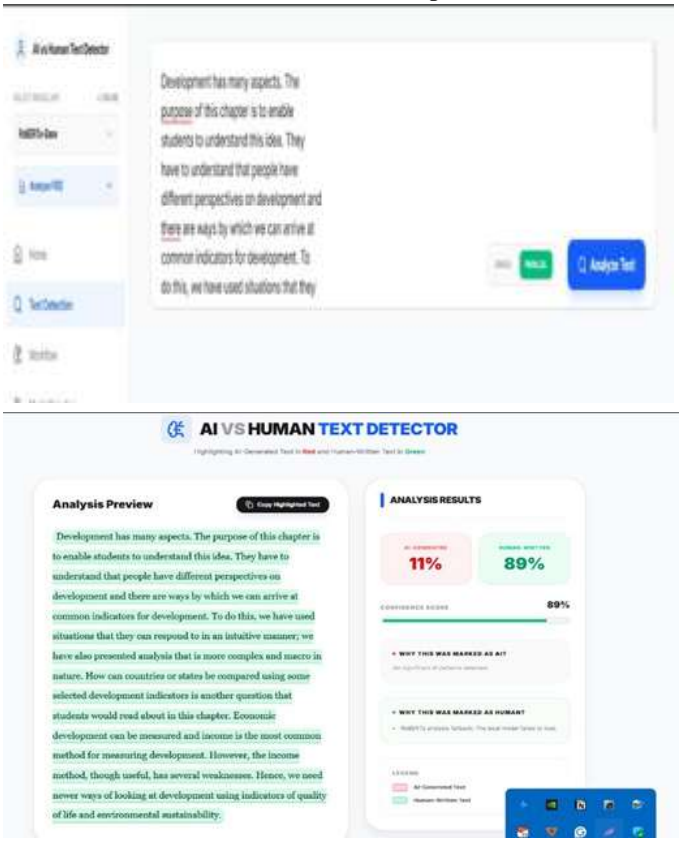
The Class-wise Performance Chart shows the performance of the model for both **Human-written and AI-generated text** classes. The plotted results indicate that precision, recall, and F1-score all reach values approaching 1.0.

6.4 Visualization Results

To understand model performance, we used a confusion matrix, ROC curve with AUC, a precision-recall curve, training vs validation loss over epochs, F1 score per epoch, class-wise accuracy chart, confidence distribution across predictions, and error analysis of misclassified samples.

If the model misclassifies a sample, the error chart shows exactly which one it was. That assists in identifying trends. in mistakes. The loss curve is observed to give the training when the training takes place. stopped improving. Large AUC implies that the model can. draws the line between classes. Confidence levels are shown for each prediction. This affords a clear sight of where accuracy drops.

Final Prediction Output



6.4.1 Final Prediction Output

The figure above displays the last output screen of the created AI v.s. Human Text Detection System. Once the user submits the input text, the system goes through preprocessing, model analysis, aggregation of results, and final classification steps. In this case, the system takes into account the. text AI-generated with the probabilities of AI = 11% and Human = 89%. Also, the interface marks the text that is being analyzed and shows the steps of execution, thus making the prediction transparent and approachable. use

6.5 Discussion

The model works well at telling apart AI-generated text from ROBERTa-based text classifier on human-written text. underpinning, and precision remain firm. Real-time classification with confidence scores makes it helpful in checking homework and content filters, as well as research. Work.

Nonetheless, there is a gap - not all writing styles are represented in the data. primarily complex AI results. When AI and humans blur, misclassification is possible. Smaller datasets, more varied, may take up the next steps. tests, and experimenting a number of models at the same time to sharpen results.

7. ADVANTAGES

The AI vs Human Text Detection System that we are proposing. compares with the traditional manual in various ways. verification of contents. On the other human judgment. hand, is not only time-consuming, but may also be. parametric, but the proposed system is a fast and AIgenerated text detector code to check whether a text is AIgenerated or not. human-written. In addition to that, the model on the basis of. RoBERTa transformer will allow the system to understand contextually the meaning, the constructions, and linguistic habits of constructions. far more efficient than either simple keyword-based or rulebased methods.Secondly, the system has the ability to predict in real time, which implies that the users will only be required to paste the text. post onto the interface and receive the results of classification. immediately. This attribute renders the product highly functional. and convenient to students, researchers, and content. verifiers. Moreover, the system reveals the. confidence in prediction, thereby enhancing the transparency. and helping the users to better interpret the result.

In addition to offering the key functionalities, the project has become a successful one in its sphere by contributing. capabilities such as workflow visualization, logs, and history. tracking. As such, the system gets to be more. interactive and informative. In the case of the deployed application. Single-user online with Railway, does not have to go through a complicated installation and can easily. access it. In short, the system that we are proposing is an exceedingly efficient, intelligent, and convenient method of. authenticating modern text.

8. APPLICATIONS

The methods researched in this paper have applications that go much further. are not just out of text classification and are flexible to. different real-life situations, where tracing the authorship of a piece of writing is critical. An example that can be given is that of in the Such a system in the education field. teachers and educational establishments in identifying if students have composed their own assignments, essays, or reports, or if these are produced by AI tools. This is one way of upholding academic honesty, along with encouraging students' creativity. In content publishing and journalism, the instrument may be the verifiability of the authors of many articles, blogs, and written fragments. information available on the web, prior to publication. This will do good in restoring the openness as well as reliability of internet media houses.

For research and academic writing, the tool may work as a helper to educators and review committees in that it assists in finding out if the paper or research report being submitted includes parts that are AI-generated. This study is also relevant to job hiring and workplace communication, where entities may be interested in whether the written answers, claims, or papers come from an individual's creativity or are automatically produced.

Besides this, the system has an application in supporting content moderation, digital forensics, as well as online platform surveillance, where it has become essential to tell apart contents produced by a machine and genuine human communication.

9. FUTURE SCOPE

Despite the fact that the proposed AI vs Human Text Detection system is able to produce results very close to the point of classifying text, there is a number of improvements that can be made in future work for increasing the degree of its accuracy and the range of its applicability. For instance, one of the options is to employ more advanced and larger transformer models capable of grasping the subtleties of human language better and consequently performing classification more accurately. Additionally, training on the system is necessary. Various datasets of different styles are represented by large and diverse datasets. writing, human as well as AI-generated ones, to cover a wide range of possible variations of text. Apart from discriminating between human-generated and AIgenerated. text, the system can be further developed to detect the specified AI model that generated the content, and contribute to. additional capabilities to the system.

The employment of more than one model and the integration of the models. Also, possible outputs in relation can be viewed as a way of. increasing the accuracy and reliability of prediction. Moreover, fitting the system with real-time interactive APIs and the fact that it can be incorporated into browser extensions. will make it a handy tool to use frequently and to it. in primary and secondary school and studies. On the other hand, It could also be the goal of future research to make the system more. explainable, e.g., by giving in detail. reasons why a suggestion was given, therefore. helping users to reason about the logic of the classification.

These characteristics will place the system in a better position. resiliency, increased scalability, and more usability in real-world use cases.

10. CONCLUSION

The production of the AI vs Human Text Detection System. indicates a significant move towards tackling the growing. The problem of determining variations in AI-generated and those made by people in the digital world. Using a transformer model with the help of RoBERTa, the suggested system is capable of automating the textual input processing and grouping according to the corresponding category using contextual and linguistic patterns. As compared to the manual inspection. or rule-based techniques, this is a technique that provides more. wise and reliable solution

This paper presents a good example of how Natural Deep Learning and Language Processing (NLP). The approaches may be effective to address the. issues of plagiarism, cheating, and digital communication. The system is also not impractical. but also easy to use, with users having the option to input. Get the prediction results instantly through an online platform using texts. platform. Concisely, the paper indicates that there was a need for. AI text recognition for up-to-date use, and at the same time. time, it lays down a sound foundation about the. development of better text classification, transparency, and mass content verification systems.

11. REFERENCES

1. Yinhan Liu, Myle Ott, Naman Goyal, et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” arXiv preprint arXiv:1907.11692, 2019.

2. Eric Mitchell, Yoonjoo Lee, Alexander Khazatsky, et al., “DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature,” ICML, 2023.

3. Hugging Face, “Transformers Documentation,” Available: https://huggingface.co/docs/

4. Railway, “Railway Deployment Platform Documentation,” Available: https://docs.railway.com/.

5. Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush, “GLTR: Statistical Detection and Visualization of Generated Text,” ACL, 2019.

6. Alec Radford, Jeffrey Wu, Rewon Child, et al., “Language Models are Unsupervised Multitask Learners,” OpenAI, 2019.
7. Ashish Vaswani, Noam Shazeer, Niki Parmar, et al., “Attention Is All You Need,” NeurIPS, 2017.

8. Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, “DistilBERT: A Distilled Version of BERT,” arXiv:1910.01108, 2019.

9. Vinu Sankar Sadasivan, Aounon Kumar, Sshreyas Shetty, et al., “Can AI-Generated Text be Reliably Detected?” arXiv:2303.11156, 2023.

10. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Tountanova, BERT: Pre-training of Deep Bidirectional Transformers in Language Understanding. Proceedings of NAACL-HIT, 2019.