Division of Computing Science and Mathematics
Faculty of Natural Sciences
University of Stirling

Human versus Machine: A Multilingual analysis of machine generated text classification.

Suraj Siddharam Jeoor

Dissertation submitted in partial fulfilment for the degree of
Master of Science in Artificial Intelligence

September 2023

Abstract

Nowadays, Generative AI(Artificial Intelligence) bots like ChatGPT, GPT-4, Google Bard(PALM-2) is revolutionizing the market due to its ability to answer almost anything that human mind can imagine. Over the period, they became so sophisticated that it started to get difficult identifying AI-generated content from human. Because of this, for writing, people started to rely on these powerful machines. They started using them to write academic research[13], fake online reviews[15], delusional news and false social media posts or comments[12]. As these models are accessible from any corner of the world and due to its ability to answer in any language, this started to happen everywhere.

To overcome this, several multilingual models were devised that can classify the text which is either written by a human or generated by language models like ChatGPT, GPT-4 etc. On them, number of researches and analysis performed. Most of them were focussed on English language, some limited testing techniques and analysis. Over the period, these classifiers were started to be tested on different languages but those were selective. This analysis is the expansion of previous researches where several classifiers were tested on English and other unexplored languages like Japanese, German and Hindi. For this analysis 7 Models and one perplexity-based method is examined. Out of which 5 models and one perplexity-based method commonly tested on datasets of all mentioned languages.

As there were no widely available datasets for any of the languages discussed in this analysis, fresh datasets had to be developed from scratch. Overall, all 5 models performed well where train datasets and test datasets consisted of multiple topics-oriented text records. Each model's performance starts to deteriorate when trained and tested on the datasets of single topics-oriented text records. This performance worsens when all models were trained with dataset having multiple topics and testing with dataset having single topics. Model like RoBERTa[22] finds challenging to classify texts well which are out of the dataset or having different topic. Hence many of the times machine-generated texts were misclassified as human written. In this case, conversely overall performance of BERT[4] was better on languages like German and English. XLM-RoBERTa[23] and DistilBERT-Multilingual[26,56] performed well on texts provided in Hindi language. Perplexity based method like GPTZero[21] classifies machine generated texts from humans in English language properly which also confirms the usage of watermarking algorithm by Language models.

Two more models pretrained on specific languages like HindiBERT[31] and BERT-Japanese[32],were trained and tested on the dataset of Hindi and Japanese language respectively which further tested with random texts. The models pretrained on one language classifies human written texts from machine one properly in all circumstances where the text is passed through the model in the language by which model has been trained.

Attestation

I understand the nature of plagiarism, and I am aware of the University's policy on this.

I certify that this dissertation reports original work by me during my University project except for the following:

- The code for GPTZero taken from the section[27].

Signature *Suraj Siddharam Jeoor* **Date** 13/09/2023

Acknowledgements

I would like to extend my gratitude to supervisor and entire faculty who helped me a lot and provided right guidance by which completion of dissertation became possible. I would like to take the moment and thank to student learning services that provided guidance to write the dissertation.

Table of Contents

Abstract i
Attestation ii
Acknowledgements iii
Table of Contents iv
List of Figures vi
1 Introduction 1
1.1 Background and Context 1
1.2 Scope and Objectives 2
1.3 Achievements 3
1.4 Overview of Dissertation 3
2 State-of-The-Art 5
3 Methodology 12
3.1 Datasets 12
3.1.1 Multiple topics-oriented dataset 13
3.1.2 Single topic-oriented dataset 14
3.2 Architecture 15
3.2.1 Pre-processing on the Pandas dataset 15
3.2.2 Converting into the Huggingface dataset 15
3.2.3 Tokenization 15
3.2.4 Classification section 15
3.2.5 Evaluation and comparison 16
3.2.6 Models Explanation 16
3.3 Further Details 16
3.3.1 Gathering, Pre-processing and converting into dataset suitable for models 16
3.3.2 Tokenization and other processes 16
3.3.3 Models 17
3.3.3.1 BERT-cased and BERT-uncased: 17
3.3.3.2 DistilBERT-multilingual: 17
3.3.3.3 RoBERTa: 17
3.3.3.4 XLM-RoBERTa: 18
3.3.3.5 HindiBERT: 18
3.3.3.6 BERT base Japanese (character tokenization): 18
3.3.3.7 GPTZero: 18
3.3.4 Model Training and Testing 18
3.4 Test cases, outcomes and analysis 19
3.4.1 Testing for English Language 19
3.4.1.1 Training and testing models with multiple topic-oriented dataset: 19
3.4.1.2 Training and testing models with single topic-oriented dataset 24
3.4.1.3 Training models with Multiple topic and testing on single topic-oriented datasets: 25
3.4.2 Key findings from the testing for English Language 28
3.4.3 Testing for Hindi Language 29
3.4.3.1 Training and Testing the models with multiple topics-oriented dataset: 29
3.4.3.2 Training and Testing models with single topic-oriented datasets: 29
3.4.3.3 Training models with multiple topics and testing with single topic-oriented datasets: 30
3.4.3.4 Analysis of Model pretrained on Hindi Language: 33
3.4.4 Key findings from the testing in Hindi language 35
3.4.5 Testing for Japanese Language 36

3.4.5.1 Training and testing the models with multiple topics: 36
3.4.5.2 Analysis of Model pretrained on Japanese Language: 36
3.4.6 Key findings from the testing in Japanese language: 37
3.4.7 Testing for German Language 38
3.4.7.1 Test case 1 38
3.4.7.2 Test case 2 38
3.5 Discussion: 39
3.5.1 BERT-cased and uncased version 39
3.5.2 DistilBERT 40
3.5.3 RoBERTa 40
3.5.4 XLM-RoBERTa 40
3.5.5 GPTZero 40
4 Conclusion 42
4.1 Summary 42
4.2 Evaluation 42
4.3 Future Work 42
References 44
Appendix 48

List of Figures

Similarly you can automatically generate a list of figures from paragraphs of style *Figure*. To update this after revisions, right-click in the table and choose *Update Field* for the entire table.

Figure 1. The structure of each dataset 13
Figure 2. Structure of Dataset with multiple topics 14
Figure 3. Project Architecture 15
Figure 4. BERT-cased model correctly classifying machine generated text 20
Figure 5. BERT-cased model correctly classifying the statement generated by human. 21
Figure 6. BERT-Uncased model correctly classifies the statement as machine generated. ... 21
Figure 7. Misclassification by DistilBERT model for machine generated statement 22
Figure 8. RoBERTa's correct classification for human generated script which is the part of dataset 22
Figure 9. RoBERTa's correct classification for machine generated data which is also the part of dataset. 23
Figure 10. RoBERTa's Incorrect Classification for external Machine generated script 23
Figure 11. XLM-RoBERTa's classification of human generated text 24
Figure 12. XLM-RoBERTa's classification of machine generated script 24
Figure 13. (a) Batchwise loss of BERT model while training. (b) Batchwise Validation loss. 26
Figure 14. (a) Batchwise train loss of DistilBERT (b)Batchwise validation loss. 27
Figure 15. (a) Batch wise training loss of RoBERTa (b)Batch wise validation loss. 27
Figure 16. (a) Batchwise training loss of XLM-RoBERTa. (b) Batchwise validation loss. 28
Figure 17. (a) Batchwise training loss of BERT-cased (b) Batchwise training loss of BERT-uncased. (c) and (d) are batchwise validation losses for (a) and (b) respectively. 31
Figure 18. (a) Batchwise training loss of DistilBERT-Multilingual (b) Batchwise validation loss of DistilBERT 32
Figure 19. (a) Batchwise training loss of RoBERTa (b)Batchwise validation loss 32
Figure 20. (a) Batchwise training loss of XLM-RoBERTa (b) Batchwise validation loss of XLM-RoBERTa 33
Figure 21. Explainability when machine generated text passed through HindiBERT 34
Figure 22. Explainability when human generated Hindi text passed through HindiBERT 34
Figure 23. (a)Batch wise training loss of HindiBERT (b) Batch wise validation loss of HindiBERT 35
Figure 24. Japanese text correctly predicted as machine generated by pretrained model . 37
Figure 25. Japanese text correctly predicted as the human generated 37

List of Tables

Table 1.	List of total number of records available into the datasets	……11
Table 2.	Results of models when trained and tested on English dataset with multiple topics	……18
Table 3.	Results when all models are trained and tested on single topic-oriented English Language dataset	……24
Table 4.	Results when models are trained on multiple topics-oriented dataset and tested on single topic-oriented dataset for English language	……24
Table 5.	Results for models when trained and tested on multiple topics oriented Hindi language dataset	……28
Table 6.	Results of models when trained and tested on single topics-oriented Hindi dataset	……29
Table 7.	Results of models when trained on multiple topic-oriented dataset and tested on single topic-oriented dataset	……29
Table 8.	Results of HindiBERT when trained and tested on multiple topics-oriented dataset	……32
Table 9.	Results of HindiBERT when trained and tested on single topic-oriented dataset	……33
Table 10.	Results of HindiBERT when trained on multiple topics-oriented dataset and tested on single topic-oriented dataset	……34
Table 11.	Results of all models and method when trained and tested on Japanese language multiple topics-oriented dataset.	……35
Table 12.	Results of Japanese BERT when trained and tested on Japanese language multiple topics-oriented dataset.	……36
Table 13.	Results of all models and a method when trained and tested on German language dataset.	……37
Table 14.	Results of all models and a method when trained and tested on revised dataset of German language.	……38

1 Introduction

1.1 Background and Context

When AI started revolutionizing the market, lots of development started to take place. Deep learning models like convolutional neural network started to tighten its grip. At that time, for Natural Language Processing, other than NLTK (Natural Language Tool Kit), there were not much tools available into the market that would work on Natural Language Processing [1]. Over the period, in Natural language processing, for classification tasks like sentimental analysis, Machine learning algorithms like Naïve Bayes, Latent Dirichlet Allocation had been used [2].

After the emergence of attention-based models like Transformers from the paper written by A Vaswani et. al. in 2017, lots of experimentation started on it due to its ability of Natural Language Generation. These models were also used for other tasks, like language translation, Logical reasoning, text classification, etc. [3] The structure of transformers mainly consists of two sections: Encoder and Decoder. So different types of study have started for both sections. Researchers thought about designing the structure under which, several encoders are connected in series that gave rise to the BERT (Bidirectional Encoder Representation Transformers). The related work was published by J. Devlin et. al. In 2019, where they implemented MLA (Masked Learning Algorithm) [4]. In the same way, A. Radford et. al. started experimentation on other section of Transformer i.e. decoder. They connected those decoders into series and tested the ability of the same to generate the next word when some texts are provided which started to be widely known as GPT (Generative pretrained Transformers) [5]. Due to this, one field started to arise known as Natural Language Generation (NLG).

The first NLG model that emerged was GPT-2 published by J Wu et al., the model which was initially trained in unsupervised way. Several Internet crawled datasets have been provided to this model without labelling. This model was as large as 1.5 Billion parameters, able to answer 3 questions out of 4 correctly for each batch [6]. In order to learn anything fast, humans take less examples to understand but this system would not. Hence Open AI came up with another model GPT-3, an auto-regressive model with 175 billion parameters trains within few shots of demonstration, was able to achieve the certain tasks like translation, question-answering, and cloze tasks. But it also has some limitations for instance, at certain extent this model start repeating itself which adversely impacts on coherence and cohesion.

To cope up with such problems, the models like ChatGPT(GPT-3.5) and GPT-4 came into the market which were memory efficient and have the multilingual capabilities that increased their popularity and reached to maximum number of users which revolutionized the NLG field [8].

As the popularity of these models were growing, several private models started to emerge for example, Grover[9]. Then eventually, models like LLAMA from Facebook[10], PALM-2 from google[11], and so on, started to make their entries into the market. These NLG models have wide range of applications. Out of which, writing blogs, reports, and news articles are predominant. It is because generative AI not only helps to write such content but also, can logically debate with humans about any topic.

Recent natural language generation (NLG) models have improved machine-generated text's diversity, control, and quality significantly. Additional technical difficulties for detecting misuses of NLG models, such as phishing, disinformation, fake product evaluations, academic dishonesty, and toxic spam, arise from the ability to produce distinct, manipulability, human-like language with extraordinary speed and efficiency. To maximize the potential benefits of NLG technology while reducing harm — a key tenant of trustworthy AI — it is imperative to address the danger of abuse[12]. This risk is not only limited to English language, but also to other languages like Russian, Hindi, Japanese etc. as many news articles, blogs, tweets are also generated into other languages.

The influence of NLG models on human civilization has the potential to be incredibly good and revolutionary. Over 1 billion individuals, or an astounding one in three internet users between the ages of 16 and 64, utilized an online translator (might be ChatGPT, GPT-4, LLAMA) in the previous week[12].
Many of the researchers started on ChatGPT and GPT-4 to write the contents. Due to excessive use of such powerful models, they have been banned from certain scientific journals like Springer-nature group and also from the big universities like Oxford[13, 14]. It should be verified whether the content is machine-generated, or human generated before inclusion. Same goes with other languages.
The Another Important factor is writing comments or feedbacks like product reviews, posts, tweets etc. It has been observed that these NLG models are used to write such things as well[15]. Some of the fake reviews are written just to promote the product. For this, chatbots like ChatGPT has been used[16].
Many of the big firms like European union(EU) are discussing about implementing the Laws on machine generated text and the purpose of the usage[17]. But before that, as mentioned, one must check the integrity of the content. To check the same, several methods and algorithms are invented and the work is still on going.
Lots of studies and researches were performed to check whether even the machine generated text detection is possible, where they described how detectors can be fooled[18]. In contrast, other research performed where they mentioned the possibility of detecting the machine-generated text by implementing several algorithms and building the models[19].
Out of which, the perplexity-based method suggested by Hugging Face[20] is effective for the English language on which GPTZero[21] has been built. So far, this method appears to be promising.
Lots of methods for the same have been explored in which different kinds of models like RoBERTa[22], XLM-RoBERTa[23], GLTR[24] etc. and methods like NELA, stylistic has been implemented[25]. However, these approaches were limited to some languages and some models which provides a limited overview for the same and there were some most frequent languages like Hindi on which the research is yet to be performed.

1.2 Scope and Objectives

This research is mainly focused on examination of human-generated and machine-generated text classification done by various models like BERT[4], RoBERTa[22], GPTZero[21], DistilBERT[26], XLM-RoBERTa[23] and their behaviour on various languages like English, Hindi, German and Japanese and comparing models performance. This research is contribution where one can study the behaviour of all mentioned models and they either build the new model or train existing models via findings.

The main objective here is:

1. To examine the human written versus machine generated text classification behavior of models on unexplored languages like Hindi, Japanese and German.
2. To check the impact of text records dedicated to a specific topic or several topics on all models.
3. To test the human versus machine generated classification ability of models pretrained on specific language.

1.3 Achievements

BERT uses two different sorts of architectures: BERT-base-uncased and BERT-base-cased. These models did well overall on the datasets for the German and English languages. Following testing and training, a pipeline for the same has been developed, on which unique content has been made available. These models performed quite well over there. These models, however, did not perform well on datasets for Japanese and Hindi.
Additionally, the BERT models pretrained especially for languages like Hindi and Japanese are employed. On their respective language datasets, they did well.
The models like multilingual DistilBERT performed good on Hindi and Japanese Datasets but it has the worst performance on English and German dataset.
On practically every dataset, RoBERTa has demonstrated strong performance. When the material was taken from the dataset, it did well in the single content testing, but when the content wasn't the part of the same, it began misclassifying.
The XLM-Roberta did well overall on languages including English, Hindi, and Japanese but poorly on German.
In the case of GPTZero it just performed well on all test cases related to English language. It also confirms the usage of watermarking algorithm primarily used by chatbots like GPT-4 and Google BARD. It failed to perform on rest of language datasets.
This experiment revealed that the BERT model can distinguish between human-based content and machine-generated content for the same language if it is trained on a dataset of that language.
If a big dataset is given to XLM-RoBERTa, it can flawlessly detect machine-created text. RoBERTa requires a large amount of training data with a variety of dataset types even if it earns the greatest score on all input datasets.
Regarding perplexity-based method, GPTZero performs well because it is built on the GPT-2 Model and the tokenizer[26]. Through experimentation, it was discovered that the GPTZero can be improved for languages other than English provided GPT-3.5 (ChatGPT) or GPT-4 is used as the base because these models are trained on multiple languages.
Overall, it came to realize that the classification of human generated content and machine generated content is possible if certain techniques are employed and tested properly.

1.4 Overview of Dissertation

As the dataset created from scratch, first the detailed information of datasets will be provided where, starting from the basics for instance number of columns, column details, languages used to top terminologies like multiple topics and single topics will be explained.
Later, there will be a brief overview of architecture where each part of the same is shortly described. Broadly, this architecture is divided in three parts namely: Pre-processing, Model training, and post processing. The sections of architectures like converting into pandas dataset, further converting it into hugging face dataset and tokenization comes under pre-processing section. After this, there is the section called classification which comes under model training section where models like BERT[4], DistilBERT[26], RoBERTa[22] XLM-RoBERTa[23] etc. and methods like GPTZero[21] are trained. After this, in the post processing section, each model's performance is evaluated and the models explainability can be viewed from the same.

After the brief overview of architecture, and steps that comes under pre-processing and post processing explained in detail. In this section each model's details are provided as well.

Post that, there will be test cases, outcomes and analysis. This section is divided by language. In language subsection, there will be test cases where the results are shown depending on whether the train or test dataset are multiple topic-oriented or single topic-oriented text rec-

ords (explained in dataset section). Further, if there exists the model pretrained on a language, there will be separate test case for the same. After testing models in each language, there is key findings where the observation from outcomes for each language is shared. There will be loss analysis section for some test cases.

Then, there is discussion section where key observation of each models throughout all testings of languages is described. After this section there will be the conclusion where overall summary, limitations and future work will be shared.

2 State-of-The-Art

Writing is the most efficient way to convey information and reach people. Any kind of media can be used for the same, for instance: newspapers, articles, journals, blogs, books, etc. This influences people throughout the world. It can inspire, enlighten, and kindle curiosity or it might take people on an emotional journey. As mentioned in early chapters, there are several types of media that are widely used out of which news article is the media that is most preferred by the people to stay updated[34]. Before the Artificial Intelligence (AI) revolution, articles (and several other media depending on the application and domain) used to be written by humans.

When Generative Artificial intelligence came into the market, people began to employ the same for almost all the domains of writings because it reduces time, efforts and cost. Generative AI (Gen AI) systems, such as ChatGPT, can now create text, audio, image, and video content that is remarkably convincing and hard to identify from human-produced material. This creates difficulties for the authenticity and admissibility of the evidence[33].A survey conducted by E. N. Crothers et. al. states that Machine-generated syntax is getting more sophisticated and prevalent, which increases the potential of abuse from phishing, false information, fraud, etc. Powerful models like GPT-3 make it possible to generate high-quality text. Threat modelling reveals significant potential abuses of natural language generation (NLG) models in domains like spam, influence operations, social engineering, and AI authorship[12].
As per the article published by M. R. Grossman et. al., Regarding the copyright of Machine-generated material, fair use of copyrighted training data, and trademark infringement/dilution, Gen AI raises novel intellectual property challenges. While Gen AI may improve access to justice, it also runs the risk of clogging up courts with unjustified, low-quality lawsuits. Given the implications for due process, there are ethical questions about judges and clerks using Gen AI for research or drafting. In the absence of ethics opinions, caution is advised[33].
By keeping all above points in consideration, the detection of machine-generated text is a crucial safeguard against the misuse of NLG models. E. N. Crothers et. al. draw the conclusion that detection systems are essential for reducing the negative effects of NLG models, but there are still considerable research obstacles in creating trustworthy, moral systems[12].
The NLG (Natural Language Generation) system, which produced news stories on the Finnish municipal elections, was the first application of Gen AI seen in the 2017. The survey conducted by Magnus et al. using machine learning model named Valtteri[34], is rated according to 5 criteria: Credibility, likeability, quality, representativeness, and finally, fluency. Out of which, the NLG system received a good score for Credibility. It has been discovered that if the subject is intriguing, people are prepared to put up with some NLG flaws. Overall, it was found to be possible to produce factual news items using NLG systems like Valtteri[34].

Although this test was better, for this survey, first, the humans were involved to differentiate machine generated content from machine one which is time consuming. Second, there is no guarantee of perfect identification after spending that much time though accuracy in that research for human was 70%. From there, researchers started their experimentation using machine learning algorithms, further which becomes the classification problem. Several methods started to invent out of which methods based on stylometry is used in the research to determine the style of the writing. The related paper published in the year 2018 where H. Adorno et. al. designed a model which used stylometry approach, there were many stylometric features to represent the novels in the vector spaced model. They used three types of stylometric features namely phraseology (lexical diversity, mean word length etc.), punctuations (commas, semicolon, colons etc.), and lexical usage (stop word list, a, the, of, in, etc.). They de-

vised the dataset on these features which is feed directly to machine learning models like logistic regressor, LIBSVM (works on the foundation of support vector machine) and LIBLINEAR (works on linear regression). It has been observed that models like Logistic regressor was giving promising results[35].

In contrast, there was research performed which highlights the essentiality of non-stylometric approaches. In this, they asserted that stylometric techniques have limitations in their ability to combat artificially produced or machine generated misinformation. Although stylometry can effectively stop impersonation by detecting text origin, it is unable to discriminate between legitimate language model applications and those that introduce fraudulent information. The experiment performed indicates that Language model-generated (or Natural language Generated) falsified texts are remarkably similar in style to ones that are true. Post this, they conclude by making the following suggestions: The first involves expanding veracity-based benchmarks, which entails adding records with a great deal of variation that represents several kinds of LM (language models) applications to the dataset on which the model requires to be trained. And the second approach is improving non-stylometric methods. There are different approaches to improve non stylometric methods. First, Machine generated text detection and second fake news detection by fact-based checking. In machine generated text detection, this method focusses on transformer-based approach[3] where one can create the transformer-based Language detector that can be highly trained on machine generated fake news. Another one is based on fact-checking approach. The last one is based on detecting the fake news not by the content, but other parameters for instance, users, originating URL, other user's explicit feedback etc[36].

In the pursuit of non-stylometric approach, many studies were done out of which the most promising was the detection of machine generated content based on statistical methods. Considering this method, the model was designed which was published by Gehrman et. al. known as GLTR(Giant Language model Test Room). This model employs three techniques to assess the content's fidelity: first, calculating the likelihood of a target word; second, the word's absolute rank; and third, the entropy of the anticipated distribution. The last test determines whether the detecting system is familiar enough with the previously generated context to be (overly) confident in its future prediction. The first two checks determine whether a generated word is sampled from the top of the distribution. However, this method has a demerit which is also considered as the part of the future work. There are some adversarial schemes to fool the process as this model assumes that systems use biased sampling for generating text. Another potential limitation are samples conditioned on a hidden seed text. Even if one has access to the model, a conditional distribution will seem different[24].

There was one study published by university of Pennsylvania in collaboration with google. They made the broad assumption that humans and machines use distinct sets of presumptions and techniques when it comes to differentiating between content. They looked at three widely used decoding-based strategies: top-k, nucleus sampling, and untruncated random sampling, and found that advances in decoding techniques have been predominantly tuned for deceiving humans. This has the drawback of introducing statistical anomalies that are simple for automated systems to identify. They asserted that as excerpt lengths rise, the discriminator's accuracy does as well. Human experts were used in this project to distinguish between content that was created by machines versus material that was created by people. Despite the wide variations in accuracy, the median accuracy of 74% was considerably lower than their top discriminator. In this study, most participants were able to distinguish between models and automatically created information based on semantic flaws. Human language has the characteristic of dipping in and out of low-likelihood areas. The challenge of imitating the human

rhythm without using poor word selections that are obvious to humans has not yet been overcome by today's generation systems. They recommended three study directions, and from the viewpoint of machine learning, they underlined the need to provide automatic discriminators a better understanding of the real world so that they will be better equipped to identify the kinds of errors that people are likely to notice[37].

To study the characteristics of human language of dipping in and out of low probability regions, a non-stylometric approach is explored creating zero-shot machine detection using probability curve. This model is known as DetectGPT. The negative curvature portions of the model's log probability function are typically occupied by the text collected from Language Models. They developed the curvature-based criterion for determining if a passage is produced by Language Models by utilizing this finding. The machine-generated text detection model created from this research is a zero-shot version in which the source model is used to detect its own samples without any alterations or adjustments. So, no more external data are needed. The model is showing bias for the wording that was utilized. The models that underlie the API, such as GPT-3, do not offer probabilities. The evaluation of the same could be expensive. Future research may examine the usage of watermarking methods along with detection algorithms like DetectGPT to increase the detection robustness. The second is the link between prompting and detection, or more specifically, the probability of creative prompts successfully shielding future model generations from being picked up by current detection techniques[38].

Another research published was related to watermarking algorithm which also plays the vital role in understanding how machine-generated text can be identified. The watermark can be embedded with negligible impact on text quality and can be detected using an efficient open-source algorithm without access to language model API (Application Programming Interface) or parameters. It works by selecting the randomized set of green tokens during sampling. In the research published by J. Kirchenbauer et. al.[39] , they emphasized on 3 main methods of watermarking. The first is the hard-red list, which creates a red list of tokens that are prohibited from appearing, using pseudo-randomness. A preceding token is used to seed the red list generator, making it possible to duplicate the red list later without having access to the complete sequence that was formed. Second is soft watermarking algorithm. The last layer of any language model produces logits at the last layer. These logits get converted using SoftMax function. In this algorithm, rather than completely prohibiting red list tokens, this algorithm adds the constant, which has a microbial value, to the logits of green list tokens. This rule adaptively enforces the watermark in the situations where doing so will have negligible impact on the quality, while almost ignoring the water mark rule in the low entropy case where there is where there is a clear and unique choice of the “best” word. In private watermarking, it uses the algorithm for a random key that is kept secret and hosted behind the API. This research also describes several types of attacks which might disrupt the watermarking algorithm. Attacks like paraphrasing, discreet alterations, tokenization, homoglyph, and generative not only disrupt the watermarking algorithm but also degrades the quality of machine learning model. They concluded leaving behind the future work. There are still a few unanswered issues related to topics like optimal conditions under which certain type of hashing rule are conceivable, effective technique to check for a watermark in a streaming context or in a situation where a small portion of watermarked text is nested inside of a larger portion of the unwatermarked text and the pursuit of straightforward sensitivity bounds for large and small that are more precise than the ones mentioned above[39]. Many of the commercial applications like ChatGPT, GPT-4, powered by Open AI also uses the same algorithm[40, 41].
Another analysis has published from Switzerland related to detection of machine generated text. Their objective was to examine the model's choices and see if any distinctive patterns or traits could be found. This study focuses on internet reviews and conducts two tests using lan-

guage that is produced by paraphrasing original reviews that were written by humans. They are utilizing a transformer-based machine learning model that was polished on DistilBERT[26]. The lighter version of BERT (Bidirectional Encoder Representation Transformers)[4] called DistilBERT was used for the implementation of the pretrained model[42]. Hugging-face transformers pytorch trainer class has been used to fine-tune the pretrained model. By default, they left the hyper parameters. Utilizing the text classification pipeline of hugging face transformers, the inference phase was put into practice. They included SHAP explainer[29,30] in their justification. The thesis made here is that the perplexity-based method is inferior to the machine-learning technique in terms of performance. This investigation was motivated mostly by the fact that people frequently publish false internet reviews that harm businesses using tools like Chat GPT. They also discovered that, unless explicitly requested, ChatGPT is impersonal (no personal pronouns, no statements of feelings, but rather, reporting experience and employing most of a mixture of third tense and passive speech). Second, ChatGPT frequently recurs. Thirdly, ChatGPT reviews are more all-encompassing and "common" in terms of content. The vocabulary in ChatGPT is also much more formal. But the it was limited to only English and social media when it comes to background[43].

As it is seen from above researches, it can be concluded that recent attempts to detect machine generated text were like detection by certain model signatures present in the generated texts output or by using the advantage of watermarking techniques that imprint specific pattern. The research published by Sadasivan et. al. shows that empirically and theoretically, detection is not possible. They demonstrate how paraphrase assaults can defeat any type of detector, including those that use on machine learning models, neural networks, or watermarking. An attacker could pick up a gentle watermarking technique. The researchers assert that an adversary can launch a spoofing assault using human adversaries who provide signals that can be recognized as being water tagged. But they also observed that high performance LM(Language Models) like GPT- 4 would have low entropy output space. So, hackers might prefer smart prompt to invade these LLMS as they are using vulnerable soft watermarking scheme. For example, attackers could input a prompt that starts with “Generate a sentence in active voice and present tense using only the following set of words that I provide...”. If the logits of LLM have low entropy over vocabulary, soft watermarking scheme samples with tokens with high logit score to preserve perplexity. Automated paraphrase strikes can significantly lower the accuracy of several detectors, including zero-shot classifiers, soft watermarking, and detectors based on neural networks[18].

When it comes to prompts, many of the invasive prompts are rejected by Language models like ChatGPT or GPT-4. The research contradictory to Sadasivan et. al. as been performed where it is found that the expanded use of language model has brought attention to the versatility of generative models. But it also exposed some ingrained bias. The rejection is not binary but rather exists on a continuum, according to a manual examination. When given a cue, ChatGPT/GPT-4's bias can be shown in both the opinions it chooses to convey and, in rare instances, in its complete refusal to participate[44]. Secondly, the designing prompts is not that much easy. A study from the Berkley university shows the same. The project uses a design investigation to investigate the difficulty of creating LLM prompts for non-AI experts. The researchers discovered that non-AI experts had trouble coming up with useful prompts since they frequently investigated prompt designs haphazardly rather than methodically. Finding the settings in which these LLMs make mistakes, coming up with prompting tactics to fix them, and then thoroughly evaluating their efficacy are necessary for designing effective prompting strategies. Even for LLM professionals, the challenge of helping prompt engineers, designers, domain experts, and other end users improve an LLM's output is difficult[44]. The main motive of sharing these resources is to show that even if someone tries to invade the machine by ef-

fective prompting, they cannot do the same as it is inefficient and tedious[45]. Besides it is uncertain that Language Models will provide response to every single questions asked[44].

When It comes to paraphrasing attack, team from google research had explored another aspect which was nearly contradictory to Sadasivan et. al.[18] This study presents a model for 11 billion parameter paraphrase generation dubbed DIPPER 3 and shows the vulnerability of existing detectors to paraphrase assaults. DIPPER's outputs can avoid AI-generated text detectors because to two special features: managing output variety and paraphrasing long-form text in context. But they present a straightforward defence that depends on getting semantically similar generations and must be maintained by a language model API provider to make AI-generated text recognition more resilient to paraphrase attacks. Their method scans a database of sequences already generated by the API, looking for sequences that match the candidate text within a predetermined range, given a candidate text. Using a database of 15M generations from a tweaked T5-XXL model, they empirically test their defence and discover that it can identify 80% to 97% of paraphrased generations in a variety of circumstances while only categorizing 1% of human-written sequences as AI-generated[46].

All classifiers mentioned till now in this review were using supervised approach where the independent and dependent variables were labelled. The research proposed by J. Rozen et.al. suggests an unsupervised method for identifying machine-generated text in a sizable document collection. It is based on the finding that writing produced by machines typically contains more repeated longer phrases (super-maximal repeats) than text authored by humans. These super-maximal repeats are first computed across all pages by this method. The ensemble of classifiers is then trained using a subset of documents with repeats in a self-training way.GPT-2 models of various sizes and sampling techniques (nucleus, top-k) are used in experiments to produce text. Results reveal that the method can rank documents with a high degree of precision (80–90%), particularly for top-k sampling. The strategy may be effective for other large language models because the performance loss with larger model sizes is minimal. The method is potentially more reliable because it is unsupervised and does not require labelled data from the same text generator. The balanced training assumption and relying only on perfect repeats as the signal is the main drawbacks. Additional linguistic research may yield more hints. In conclusion, the research provides a unique unsupervised method for identifying machine-generated text in a collection using distributional statistics of repeats, with encouraging preliminary findings. According to this research, It's intriguing to consider repeats as an indicator of artificial text production[47].

Another article in contrast to Sadasivan et. al.[18] introduces a brand-new detector named G3, which can recognize text produced by sophisticated language models like GPT-3.5 and GPT-4. On the newest models, ChatGPT and GPT-4, existing detectors like GPTZero[21] work badly. RoBERTa[22] fine-tuned on synthetic text from T5 serves as the foundation of G3. Over 90% accuracy is achieved across all models and domains. Even when sampling techniques or paraphrase are used to try to avoid detection, G3 maintains detection accuracy. Paraphrasing has the potential to impair watermarking, however G3 is resistant to this. G3 is a supplement to watermarking strategies. Paraphrasing has the potential to impair watermarking, however G3 is resistant to this. G3 is a supplement to watermarking strategies. According to the analysis, G3 performs noticeably better than earlier detectors like GPTZero on out-of-domain data from GPT-3.5, GPT-4, and other sources. The expected simplified supervised context and lack of analysis on long texts are the main drawbacks[48].
The research like Magnus et. al.[33] was performed in the year 2023 where abstracts from research papers are generated from AI and humans. They constructed feature description which distinguishes between AI Generated text and human written text depending on following crite-

ria: 1.Writing style(Token level features(length of word, Part of speech, Function word frequency, Stop word Ratio)) coherence(cosine similarity between abstract sentences),consistency(cosine similarity between title and each sentence in abstract),Argument Logistics(Self Contradiction, redundant and common sense). Following some investigation, they discovered that the text distributions produced by humans and machine learning models differed noticeably. Scientific language that has been generated by a machine, especially abstracts, lacks insightful details and contains too many generalizations. It is consistent with the real world of scientific knowledge. For machine learning model, they were using bart-large-cnn[49] and RoBERTa-base-openai Classifiers[50] for the machine-based classifications[51].

There was a recent paper by X. He et. al. [52] which is nothing, but the analysis of various detection methods and models. That research introduces MGTBench, a benchmark framework to compare effective large language models (LLMs) like ChatGPT with other machine-generated text (MGT) detection techniques. They employ ten detection techniques, including four model-based BERT[4] classifiers and six metric-based ones such as log-likelihood. The LM(Language Model) Detector (BERT) performs the best overall, according to extensive examination on 3 QA(Question and Answers) datasets with human replies and additional LLM-generated answers. According to this analysis, model-based approaches greatly outperform metric-based approaches in the more difficult task of determining the source model (human or one of six LLMs). LM Detector receives a 0.631 F1-score on model attribution compared to metric-based approaches' 0.134-0.255 F1-score. But as per this analysis, it's still hard to attribute source LLM correctly. Although LM Detector is more resistant to adversarial attacks than ChatGPT Detector, it is still susceptible, demonstrating the need for additional study on robust detection. A helpful benchmark for comparing machine text identification techniques against sophisticated LLMs is offered by MGTBench. Extensions are facilitated by modular design. In closing, the research introduces MGTBench to benchmark machine text detection, exposing shortcomings in existing techniques versus contemporary LLMs. The analysis offers guidance for future research to lead to more robust and generalized detection[52].

All above researches were focussed on the languages like English, German, Finnish and Swedish[33,48]. Out of which English was the most predominant language. The study published by Y. Wang et. al. employs multilingual approach. The research introduces M4, a novel, sizable dataset for Machine-generated text identification that is Multi-generator, Multi-domain, and Multilingual(that is the reason the name is M4). Several detection techniques, including RoBERTa, logistic regression using GLTR (Giant Language model Test Room) features, stylistic characteristics, and NELA features are evaluated through experiments using M4. To analyse the detection models, they explored 3 approaches: different detectors across different domain given a specific generator, different detectors across different generative models for specific domain, and the impact on the performance by interactions of different domains and generators in multilingual setting. The findings indicate that it is difficult for detectors to generalize to new examples from various domains or generators. Detectors frequently mistake computer text for human-written material. High performance is achieved for in-domain detection, but cross-domain and cross-generator performance is subpar, particularly for recall. The outputs of BLOOMz appear to be very distinct from those of other generators. Compared to arXiv, Wikipedia text is simpler to read. This research was primarily conducted on 6 Languages namely: English, Urdu, Chinese, Arabic, Russian, and Indonesian. In this research, the RoBERTa detector performs poorly across domains while achieving the best in-domain accuracy, most likely because of overfitting. In a zero-shot environment, the commercial system GPTZero[21,27] is assessed. It performs well on general purpose text but problems with innovative generators and domains. Investigations assess cross-lingual detection as well. When a language not seen

during training is detected, performance suffers. The research finds that machine text detection is still a difficult problem and emphasizes the need for more robust and generalized methods. As a part of their future work, they might expand their analysis on the other languages like Japanese, German and Bulgarian[25].

After going through all these papers, it came under observation that this analysis for the classification of human-generated and machine-generated content can be expanded to languages like Hindi, German and Japanese. This dissertation is the expansion of the same, an attempt to contribute with some intriguing findings and also, it is binary classification problem where the contexts will be identified as either machine-written(LABEL_0 or 0) or human-written(LABEL_1 or 1). The classic BERT model is pre-trained specifically on English model, hence it classifies contexts well which are written in English[4]. But in this analysis, the capability of classification for the BERT model pre-trained on specific languages like Hindi and Japanese has also examined. For this analysis 7 Models and one detection technique are taken into consideration because as per Y. Wang et. al. the classification model's performance is better than classification techniques (for instance Stylometry[18], NELA etc.)[25]. For this analysis, a Novel dataset has been created from Wikipedia dataset[53] which is available on Hugging face for human-written texts, and took the help from bots like ChatGPT, GPT-4, Google BARD etc. for machine generated contents. After the training and evaluation, the model decision for a particular class has been explained by using SHAP Explainer[29, 30]. The training and testing procedure of the model is primarily divided into three categories namely: 1. Training and testing with datasets having multiple topics oriented text records , 2. Training and testing with datasets having Single topic-oriented text records, and 3. Training with multiple topic oriented text records and testing with single topic oriented text records to check the impact of topics on the classification of context which would be either human or machine generated. Post this the Error Analysis is performed, and the conclusion is drawn. After this, the BERT model specifically trained on specified languages are tested. Many of the models like RoBERTa[22] and XLM-RoBERTa[23] performed well when training and testing both have multiple topics. The performance of the models starts to drop significantly when the dataset of single topic has been passed as a training dataset and testing dataset. The models like BERT and classification techniques like GPTZero[21] maintains their performance well on dataset with multiple topics and dataset with single topic for English language. For other languages, GPTZero performed the worst as it is using GPT-2 for base model[21].

3 Methodology

So far, way or the method of the research performed by other organization or institutes has been explored and justified the novelty of this analysis compared to other researches. Now, actual methodology of this analysis will be explained. First, there will be the detailed description of datasets where all terms related to dataset will be explained. Then, there will be the broad overview of the architecture. Afterwards, detailed information related to this architecture will be provided. Then the experiment and results of each models on different languages will be explained. This section is divided according to the language. Hence experiments performed will be explained according to the same.

3.1 Datasets

In this section, the structure of the dataset will be discussed. The datasets are created for the languages like Hindi, English, Japanese, and German. For Hindi, German, and Japanese, when it comes to human generated texts, the sources like Wikipedia, and the dataset of the Wikipedia available on the Hugging face[53] are used and for machine generated script, the chatbot publicly available like ChatGPT is used. If it is prompted ChatGPT to give answers of questions in the desired language, the bot does. By using the software known as BulkGPT, several questions asked to ChatGPT in bulk and their answers are recorded in the excel file automatically[64,65]. When it comes to English language, as the experimentation on ChatGPT were already done in the previous researches, it is decided to experiment on GPT-4. For GPT-4, the dataset of human and machine generated scripts was already available on Hugging Face[28]. A new type of dataset is created out of that dataset[28], which is further used for training and testing the classifiers. All of the datasets are in the form of excel spreadsheets.

The details of the datasets are shown as per the below table:

Datasets	Number of records in train dataset	Number of records in test dataset	Total Number of records
English Language Dataset(Human Generated vs GPT-4)	2000	1600	3600
Hindi Language Data	402	399	801
Japanese Language Data	407	395	802
German Data	400	400	800
Revised German Data	416	416	832
Revised English Data(with single topic)	50	24	74
Revised Hindi Data(with single topic)	50	24	74
total	3725	3258	6983

Table 1. List of total number of text records available into the datasets

In above table, even though the number of records in train is more than the number of records in the test, there are even number of human written and machine generated records in each dataset for different languages.

Each dataset has been designed in the specific structure; the structure is shown into the below figure:

id	title	text	Generated By	labels
1	Chatrapati Shivaji Maharaj	Shivaji Maharaj's coronation took place in a granc	Machine	0
2	Chatrapati Shivaji Maharaj	He built a strong network of hill forts and coastal	Machine	0
3	Chatrapati Shivaji Maharaj	Several forts, statues, and monuments dedicated	Machine	0
4	Chatrapati Shivaji Maharaj	Throughout Maharashtra and other regions, statu	Machine	0
5	Chatrapati Shivaji Maharaj	Shivaji escaped from Panhala by cover of night, ar	Human	1
6	Chatrapati Shivaji Maharaj	Shivaji Maharaj was born on February 19, 1630, ir	Machine	0
7	Chatrapati Shivaji Maharaj	Shahaji was a rebel from brief Mughal service. Sh	Human	1
8	Chatrapati Shivaji Maharaj	Shivaji organized his military forces into different	Machine	0
9	Chatrapati Shivaji Maharaj	Shivaji Maharaj was a prolific builder of forts. He	Machine	0
10	Chatrapati Shivaji Maharaj	The Bijapur sultanate was displeased at their loss	Human	1
11	Chatrapati Shivaji Maharaj	Shivaji had a deep appreciation for literature and	Machine	0
12	Chatrapati Shivaji Maharaj	At the time of Shivaji's birth, power in the Deccan	Human	1
13	Chatrapati Shivaji Maharaj	In 1657 the sultan, or more likely his mother and	Human	1
14	Chatrapati Shivaji Maharaj	In the Treaty of Purandar, signed between Shivaji	Human	1
15	Chatrapati Shivaji Maharaj	In 1674, Shivaji Maharaj crowned himself as the C	Machine	0
16	Chatrapati Shivaji Maharaj	Upon the request of Badi Begum of Bijapur, Aurar	Human	1
17	Chatrapati Shivaji Maharaj	Shivaji Maharaj adhered to a strict code of ethics	Machine	0
18	Chatrapati Shivaji Maharaj	During the bombardment of Panhala, Siddi Jauha	Human	1

Figure 1. The general structure of all datasets (each row of title might vary)

As per the above figure, the entire table is divided into 5 columns:

Id: This Column is kept for the convenience so that number of records can be maintained.

title: This represents the topic on which the text (or statement) is provided.

text: This column contains records of statements of specific topic either generated by human or by machine.

Generated By: This column is categorical where there are two values: Either Human or Machine. This column specifies the source by which respective text records are generated. This is kept for the convenience of the one who is going through the dataset.

labels: The labels column is kept for the machine purpose. For sample text record if the value of label is “0”, then that text is machine generated. If it is “1”, the respective text record is written by human.

Above all columns, the only important columns for models are text and labels.

The dataset of the Wikipedia on Hugging face is available in many languages. As there was a requirement of only 3 languages(i.e. Hindi, Japanese and German), the dataset for the same has been extracted.

For English, as dataset were available, it became simple to create the new dataset out of it.

Broadly, all datasets will be either of the two types as mentioned below:

3.1.1 Multiple topics-oriented dataset

The multiple topics-oriented dataset is defined as the kind of dataset where each text(might be human or machine-generated) varies from each other by the topics. The structure of such dataset is as shown in the below figure.

id	title	text	Generated By	labels
1	Cauvery Wildlife Sanctuary	The Cauvery Wildlife Sanctuary is a protected a	Machine(GPT-4)	0
2	Lachhu Maharaj (musician)	Laxmi Narayan Singh (16 October 1944 - 5 Octo	Machine(GPT-4)	0
3	Maryland Route 318	Maryland Route 318 (MD 318) is a state highw	Machine(GPT-4)	0
4	Suzanne Duchamp	Suzanne Duchamp-Crotti (20 October 1889 – 1	Human	1
5	Hurricane Patricia	Hurricane Patricia was the strongest tropical cy	Human	1
6	George Ronan	Ensign George Ronan was a commissioned offi	Machine(GPT-4)	0
7	Highness	Highness (abbreviation HH, oral address Your H	Human	1
8	Acacia glaucocaesia	Acacia glaucocaesia is a shrub or tree of the ge	Machine(GPT-4)	0
9	Nemili	Nemili is a Taluk in Ranipet district in the Indiar	Machine(GPT-4)	0
10	Ogden Park	Ogden Park, also known as Ogden Skating Park	Human	1
11	Cassandra Lee Morris	Cassandra Lee Morris (born April 19, 1982) is a	Machine(GPT-4)	0
12	We Rule	We Rule was a free-to-play mobile game devel	Human	1
13	Sideshow Bob Roberts	"Sideshow Bob Roberts" is the fifth episode of	Machine(GPT-4)	0
14	1987 NBA Finals	The 1987 NBA Finals was the championship se	Machine(GPT-4)	0
15	Grey-fronted honeyeater	The grey-fronted honeyeater (Ptilotula plumul	Human	1
16	Nanao Castle	was a Muromachi period yamajiro-style Japan	Human	1
17	Joseph Henry Reason	Joseph Henry Reason (March 23, 1905 – July 2(	Human	1
18	Ou Chuliang	Ou Chuliang (; born 26 August 1968) is a retire	Machine(GPT-4)	0
19	Kathleen Williams (politician)	Kathleen Williams (born February 16, 1961) is a	Machine(GPT-4)	0
20	Jeff LeBlanc	Jeff LeBlanc (born February 9, 1986) is a Canad	Machine(GPT-4)	0

Figure 2. Structure of Dataset with multiple topics

In the above figure each topic (or title in the above table) is different from other and the related text records are also different with each other which are either generated from human or machine. Though it appears as random records in above figure for the dataset of languages like Hindi, English and German, each topic has human generated and machine generated text. For instance, if number of text records is 2000 in the dataset, out of the same, the number of topics (or titles) will be 1000, each having human and machine generated text records.

3.1.2 Single topic-oriented dataset

Here, the single-topic-oriented dataset is defined as set of text records related to only one topic though the structure, style will be different and primarily, those text records might be generated by human or machine. The Figure 1 shows the example of single topic dataset. For English language dataset, “Chhatrapati Shivaji Maharaj” topic is preferred related to which texts are available. For Hindi dataset, “Maharana Pratap Singh” topic is chosen.

Datasets with single topic is created for only two languages: Hindi and English.

3.2 Architecture

Before deep diving into the test, models, and results it is important to know the architecture at the broader level through which the models have been trained and tested. By this, the overview of the testing process will be known which further help to get the understanding of the results and explainability.

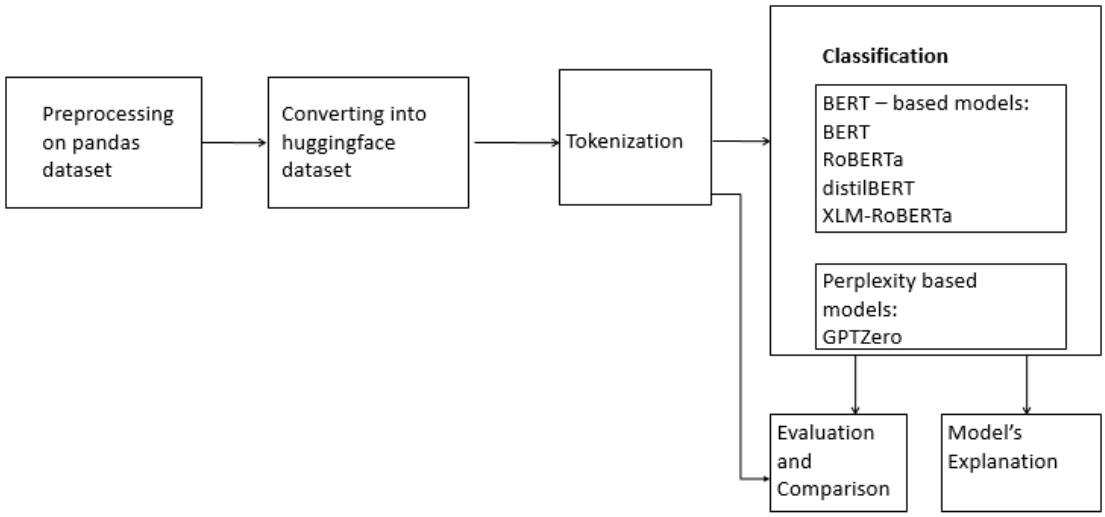


Figure 3. Project Architecture

The Entire architecture is divided into following subsections

3.2.1 Pre-processing on the Pandas dataset

First, this dataset was in the excel spreadsheet form, that data is converted to Pandas data frames. Pandas is nothing but the library that has been used to convert dataset to the suitable form.

3.2.2 Converting into the Huggingface dataset

Then the Pandas dataset is converted into the Huggingface dataset suitable for models used in this dissertation. Basically, this dataset consists of 4 columns namely: ID, Title, text, Generated by, and Label (for model's convenience).

3.2.3 Tokenization

Before tokenization, all columns described in previous section are eliminated except label column which is further included with the columns present in tokenization. In this, basically tokenized numbers and labels by which the model can classify are present(further details will be explained in depth later).

3.2.4 Classification section

In this section, basically all the classifiers and classification techniques are used. The tokens after the tokenization is passed through each model for the sake of training. Then it is further those trained models are evaluated under the section of evaluation and comparison.

3.2.5 Evaluation and comparison

In evaluation and comparison section, after training the models, they are tested. These models are evaluated based on 5 parameters that is: Accuracy, precision, recall, F1-score and the confusion matrix. All trained models are compared on these parameters as well.

For the confusion matrix, scikit-learn library is used and for other parameters, evaluate from Hugging face transformers library is used.

3.2.6 Models Explanation

Under this section, the reason of model's decision for Human-generated (which is also known as LABEL_1) or Machine-generated (which is also known as LABEL_0) is evaluated using SHAP Explainer[29,30]. Here, as all models are running on the GPU(Graphical Processing Unit), first those trained models are brought on CPU(that is on system) and then these models are converted to the pipelines so that any texts can be classified, using pipeline function that comes from the Hugging face transformers library. After that, this pipeline is passed through the explainer function along with the text that might be written by the human or machine. By doing so each text has been assigned a specific SHAP value depending on the model pipeline that is passed through explainer. The SHAP values are either below zero or above zero. The texts having SHAP value below zero are highlighted with blue colour and red if the value is above zero. Then the average SHAP values of the whole paragraph is calculated by explainer. This is useful in error analysis of the model.

3.3 Further Details

In the last section there was a general overview of the Architecture. In this section, the parts used in this architecture will be discussed in detail.

3.3.1 Gathering, Pre-processing and converting into dataset suitable for models

For gathering, as the dataset was available on the hugging face, it is decided to extract the file in the form of comma separated version (CSV) file using the Hugging Face dataset library. Then that data was arranged in the form as per Figure 2. This process is partially completed as it has been done on the texts which were human-generated. For machine generated texts, prompts on random subjects were provided in bulk to ChatGPT using the software known as BulkGPT[64,65], the outputs were extracted in terms of Excel file and arranged in the same dataset where human generated texts were already arranged. The number of human generated texts were kept equal to machine generated to avoid any kind of bias. Further these records shuffle randomly so that any model can become more robust. This has been performed on each set of Dataset.

3.3.2 Tokenization and other processes

Before tokenization, the library called as Sentence piece, which is the part of Hugging face transformers, were imported as it is required by most of the models available on Hugging Face repository. Post that, the excel sheets where the main data is available, uploaded on the Google Colab. These sheets are divided in terms of the type of the datasets for each language: Training and Testing. Google Colab is the online IDE(Integrated Development Environment) provided by Google on which Jupyter Notebooks can be hosted. The Speciality of such IDE is it provides GPU on which the model can run efficiently. After loading datasets as files into Google Colab, those were referred by the source code to extract the data. The Panel Data or Pandas

library is used to convert the excel sheet data into the data frames. These data frames are further converted to Hugging face datasets by using the datasets package that comes from the transformers library[55]. After this, two more packages are imported to hugging face libraries which plays an important role for the tokenization and sequence classification. Then desired model names are passed through the functions originating from these packages. Then after tokenization, the unnecessary columns are removed and the columns got after tokenization along with the column named labels are maintained(most of the times there are input ids which are the encoded ids obtained after tokenization and attention mask to increase the predictability my masking important ids.

3.3.3 Models

Before proceeding with further process, it is important to be aware of the models used in the same. These are following models that have been tested.

3.3.3.1 BERT-cased and BERT-uncased:

These models are taken from the research performed by J. Devlin et. al. [4]. Bidirectional Encoder Representations from Transformers, or BERT, is a brand-new language representation approach. In order to pre-train contextual representations from unlabelled text, it employs a deep bidirectional Transformer encoder. In contrast to earlier models, which were unidirectional, this enables BERT to integrate both left and right context. Masked Language Modelling and Next Sentence Prediction are two brand-new unsupervised prediction tasks used for pretraining. Because of them, BERT can pre-train a deep bidirectional representation. BERT can be fine-tuned on downstream NLP tasks after pre-training by only adding a tiny layer that is specific to those tasks. For cased and uncased, the architecture is same with the slight difference of consideration of capital cased letters. Uncased doesn't considers such letters but cased architecture does[4].

3.3.3.2 DistilBERT-multilingual:

DistilBERT is a compressed and speedier version of BERT that is 40% smaller and 60% faster while retaining 97% of its language understanding capabilities. A smaller student model is trained through knowledge distillation to mimic the actions of a bigger teacher model (BERT). During pre-training, the student employs a triple loss that combines distillation, cosine, and standard language modelling[26]. In multilingual approach, this model is pretrained on languages like English, Spanish, German, Chinese, Arabic and Urdu[56].

3.3.3.3 RoBERTa:

The pretraining process of RoBERTa, a modified version of BERT, has been enhanced to produce better results on downstream NLP tasks. The adjustments consist of training with bigger batches, longer sequences, and more data; removing the target of next sentence prediction; dynamically altering the masking pattern; and training for additional iterations[22].

3.3.3.4 XLM-RoBERTa:

A new multilingual language model called XLM-RoBERTa or XLM-R, was pretrained on 2.5 TB of filtered CommonCrawl data in 100 languages. When more languages are added to a fixed-capacity model, the "curse of multilingually" results in performance degradation, underlining the trade-off between capacity and transferability. Additionally, XLM-R performs well in monolingual situations. Low-resource languages like Swahili and Urdu are where XLM-R excels the most, outperforming earlier models by over 15% in accuracy[23].

3.3.3.5 HindiBERT:

This Machine learning model is produced by L3Cube, which is nothing but the BERT pretrained specifically on Hindi language. It has been pretrained on 1.8 billion Hindi language tokens. As per the findings of the paper published by R. Joshi et. al.[31], This model performs well than the multilingual models available in the market like mBERT[57], IndicBERT[58] and XLM-RoBERTa[23]. In this dissertation the model is tested for the ability of classification when human or machine generated data is provided.

3.3.3.6 BERT base Japanese (character tokenization):

This Model also uses BERT architecture having 12 layers, 768 dimensions of hidden states, and 12 attention heads. This model is trained on nearly 17 million sentences and 3 Billion Japanese Kanji words[32].

3.3.3.7 GPTZero:

This is not the model but this the method where the fidelity of the content is determined by the perplexity score[20]. In general, the perplexity score is determined by the likelihood of the occurring of word provided other vicinity words are available. If this probability is low, the perplexity is high and vice versa. Perplexity is also termed as an entropy. As per the information theory, the more entropy of the paragraph or the content will be, the more information it will convey. And if it conveys more information, that paragraph has the high entropy and perplexity which concludes that content is more likely to be generated by human[20,27,59].

3.3.4 Model Training and Testing

For this training and testing, the function named “AutoModelForSequenceClassification” originated from the transformers library has been imported via which models from the Hugging face repository are downloaded, trained and tested.

Each model’s performance is examined one at a time. Each time, the model is hosted on the GPU(Graphical Processing Unit). Here, AdamW optimizer[60] is used to train the model. Number of Epochs taken were 2 for the same as this much were enough to take the model’s accuracy somewhere between 90% to 100%. For this optimizer the learning rate is 5×10^{-5}. Number of Batches varies according to the number of records into the dataset. For each count, the loss is calculated and recorded. Along with this, the optimizers update the model at each step reducing the loss by back propagation.

The same thing is repeated with testing but instead, not only loss, but the logits were calculated which further turned into probabilities using argmax function. Over here the validation loss is also calculated. Further, the Model is evaluated using confusion matrix.

After model trained properly, it is converted to the pipeline. Meanwhile, SHAP library is imported to examine the model's explainability. The sample of human generated, and machine generated text is taken. After this the "Explainer" function is used through which model pipeline and sample text has been passed. This function first calculates SHAP values for each word and then it calculates the average SHAP value of the entire text. Then the probability is calculated out of these SHAP values[30].

3.4 Test cases, outcomes and analysis

In this section, there will be a detailed overview of test cases. In those cases, the output would be represented and post that the output will be analysed using the output obtained from the SHAP explainer[29,30].

There will be language-wise approach for the overview:

3.4.1 Testing for English Language

This Entire testing and analysis are divided in 3 types. Although analysis in English language already done in the previous research[25], this analysis varies from the same in terms of the topics. In this analysis there will be multiple topics and single topic-oriented dataset. The dataset for multiple topics consists nearly 1000 topics where each topic has human-generated and machine generated content which makes it 2000 number of records for training. Same thing is implemented in test dataset where number of topics were 800 and total number of text records will be 1600. In terms of single topic, as discussed in section 3.1.2, in total, there are 50 records for training and 24 records for testing where records of each dataset are divided fairly for human generated and machine generated texts. The 3 types of analysis are as follows:

3.4.1.1 Training and testing models with multiple topic-oriented dataset:

In this type of analysis, the training data consists of nearly 1000 topics(or titles) each having human-generated and machine generated text and test dataset consists of 800 different topics than train but the structure is same as training.

The results for the same can be shown as follows:

Models/Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
BERT-cased	96.12	99.2	93	96
BERT-uncased	95.62	99.3	91.87	91.45
DistilBERT-multilingual-cased	97.18	99	95.25	97.13
GPTZero	80.5	84.03	75	79
RoBERTa	99.37	99.62	99.12	99.3
XLM-RoBERTa	90.31	98.78	81.62	89.39

Table 2. Results of models when trained and tested on multiple topics-oriented English datasets

As per the above table for this arrangement, RoBERTa performs the best out of the models where each parameter is overall more than 99%. When it comes to BERT model considering upper case letters (BERT-cased), shows significant rise in the accuracy, recall and F1-score com-

pared to the model that doesn't considers upper case letters (uncased version). For cased and uncased version, there is no significant difference in precision. DistilBERT is showing more effective results than XLM-RoBERTa though memory required for DistilBERT is less and both of models are pretrained on multiple languages. GPTZero does not performed well compared to rest of the models. As it appears for all models, classification parameters like Accuracy, precision, Recall and F1-score are high which is sceptical. Hence SHAP Explainer used to do the further analysis of the each model's decision.

Explainability for classic BERT models:

After training both BERT models(both cased and uncased version) and testing, these models were converted to the pipelines and passed through the SHAP explainer through which the random text sample which is either generated by human or machine is passed. And here are the results:

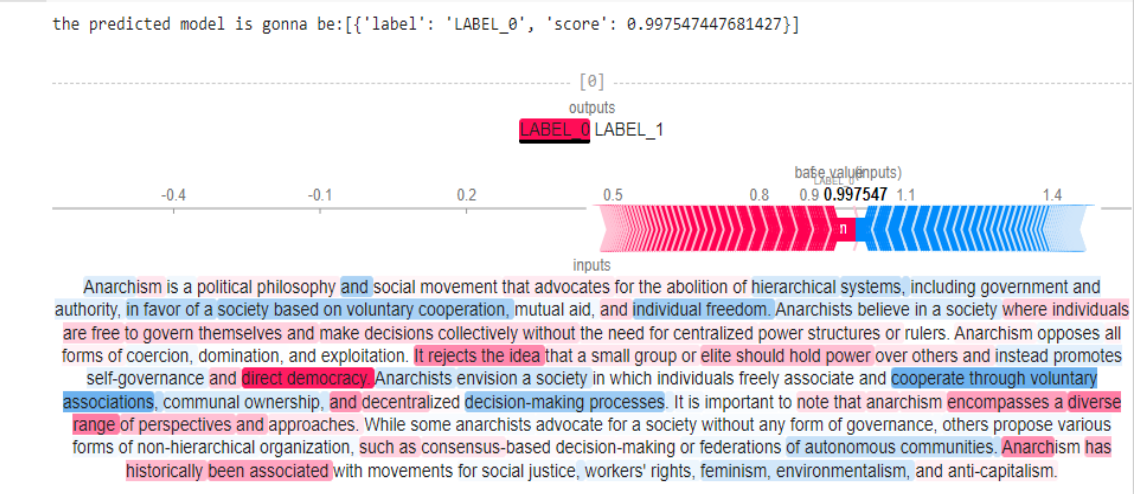


Figure 4. BERT-cased model correctly classifying machine generated text

At the top of the figure, there is small statement which tells us the decision of the model and the overall score. Here, the model's decision was "LABEL_0" which denotes that the paragraph is generated by a machine. In actual scenario, that paragraph was generated by machine. Right below that statement, there is the output where the model's decision is highlighted in magenta colour as "LABEL_0". Right below that, there is the graph that shows the impact of each words from the paragraph. Below that graph, there is the paragraph which is evaluated for the explainability. The blue highlighted words this show negative impact of the words on the model's decision. For such words, the SHAP value is negative. The words highlighted in Magenta colour shows positive impact of words on model's decision.

These representations are a part of SHAP values, which are based on ideas from cooperative game theory and offer a means of equitably allocating each feature's contribution to the prediction among all features. This enables to comprehend the effects of certain features while considering how those features interact with other features in the model[29,30].

In the text of the figure 3, the words highlighted in Magenta are more than the words highlighted in blue. From the human perspective, if one reads the sentences highlighted in magenta colour, it entails the simplicity of those sentences. This same experiment is repeated for human generated statements and here is the result:

Figure 5. BERT-cased model correctly classifying the statement generated by human.

In the above figure, "LABEL_1" is highlighted in magenta colour. The texts highlighted by magenta colour in above paragraph are the texts responsible for the model's decision to classify the paragraph as human generated or "LABEL_1". The above paragraph emphasizes the complexity because of which model classifies it as human generated.

For BERT-uncased the results are pretty much similar, below image is the example of accurate identification of machine generated text.

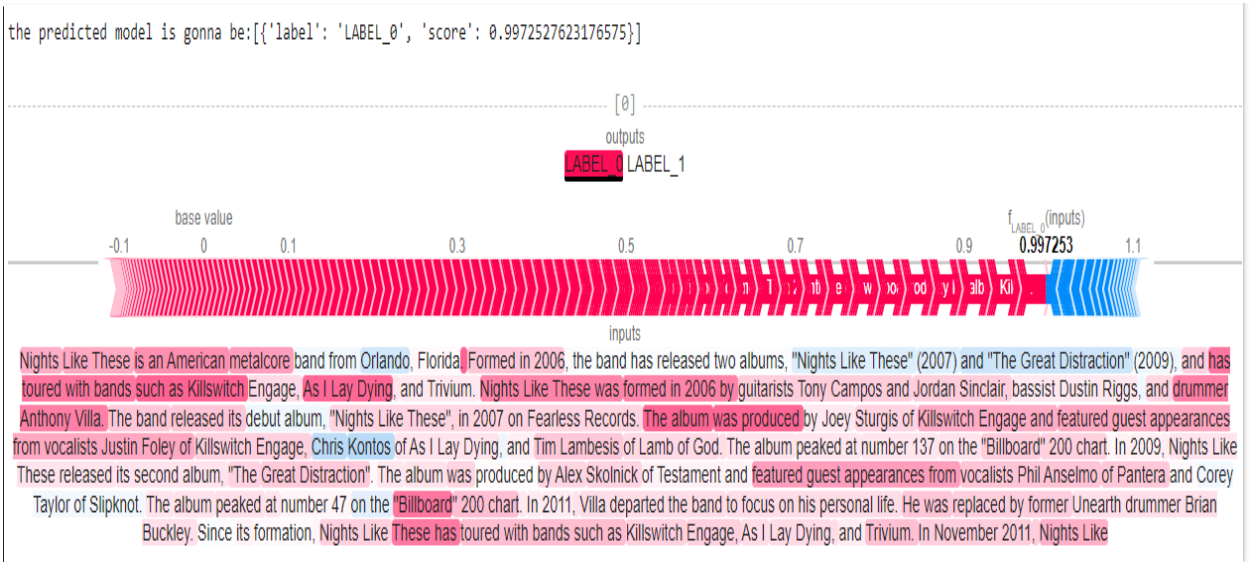


Figure 6. BERT-Uncased model correctly classifies the statement as machine generated.

Explainability for DistilBERT-multilingual model:

Although DistilBERT performs well on the table shown above, it was time to test it in the real-world scenario. For this, the same test of BERT was repeated, and the results are as per the below figure:

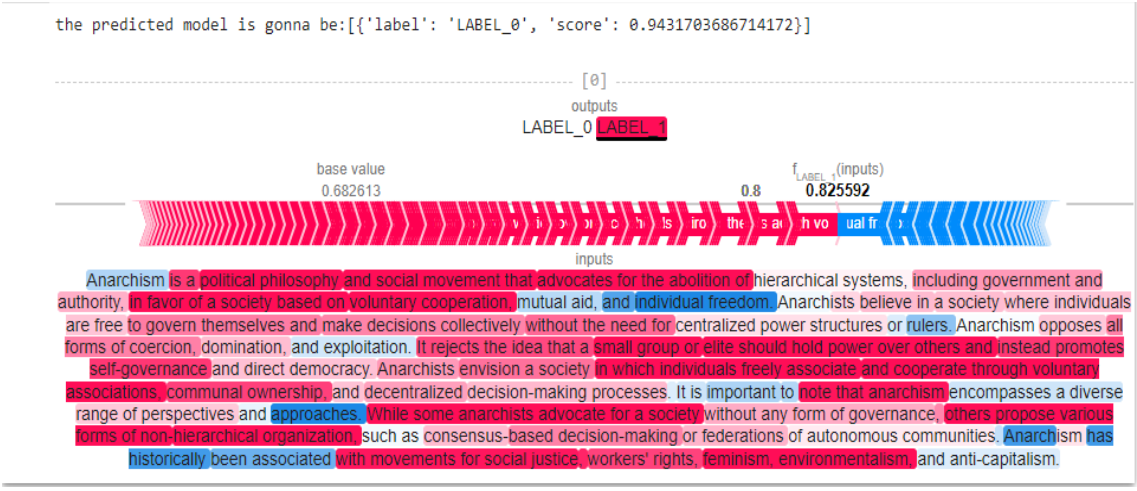


Figure 7. Misclassification by DistilBERT model for machine generated statement

DistilBERT was classifying well for Human Generated data, but it was considering machine generated content as the human. In short, this model was not able to distinguish between human generated paragraph and machine generated paragraph which entails the model is overfitting.

Explainability for RoBERTa model:

Surprisingly for RoBERTa model which was the most accurate model for the dataset provided as per the table 2 has misclassified the texts when it has been tested through the explainer function. When the topic is the part of the dataset through which RoBERTa is trained, the related human generated, and machine generated paragraphs are perfectly classified. Here are the below figures as the part of evidence.

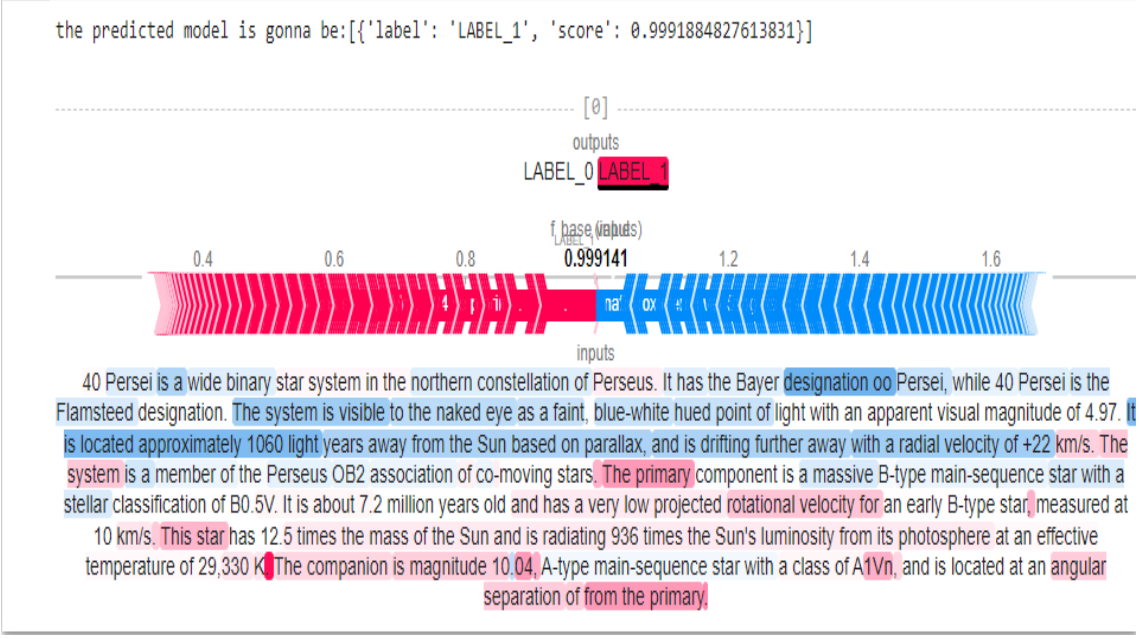


Figure 8. RoBERTa's correct classification for human generated script which is the part of dataset

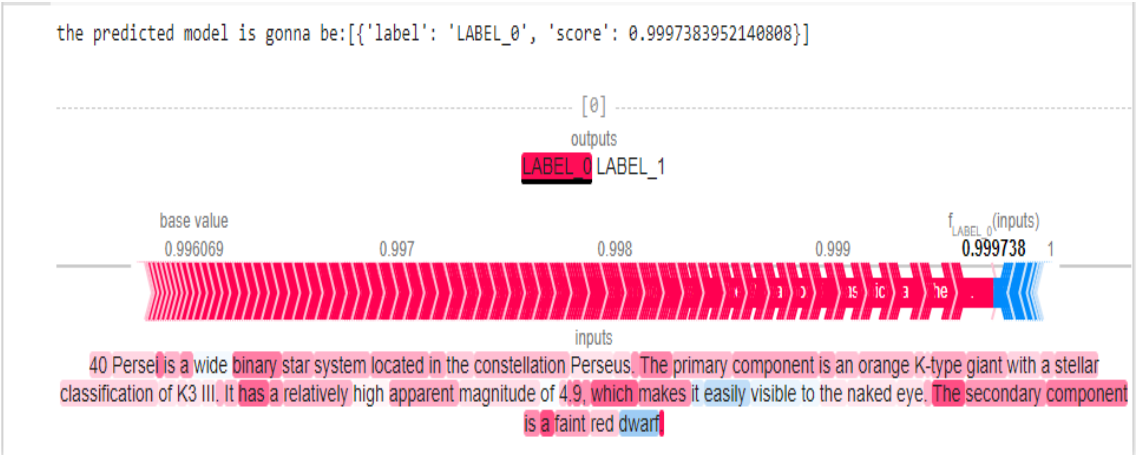


Figure 9. RoBERTa's correct classification for machine generated data which is also the part of dataset.

When the topic is completely new for the RoBERTa, the related paragraphs are misclassified by this model. Although it classifies well for the human generated data, it fails to label machine generated content as the "LABEL_0" which is the label for the same. The Image is shown below as the part of the evidence.

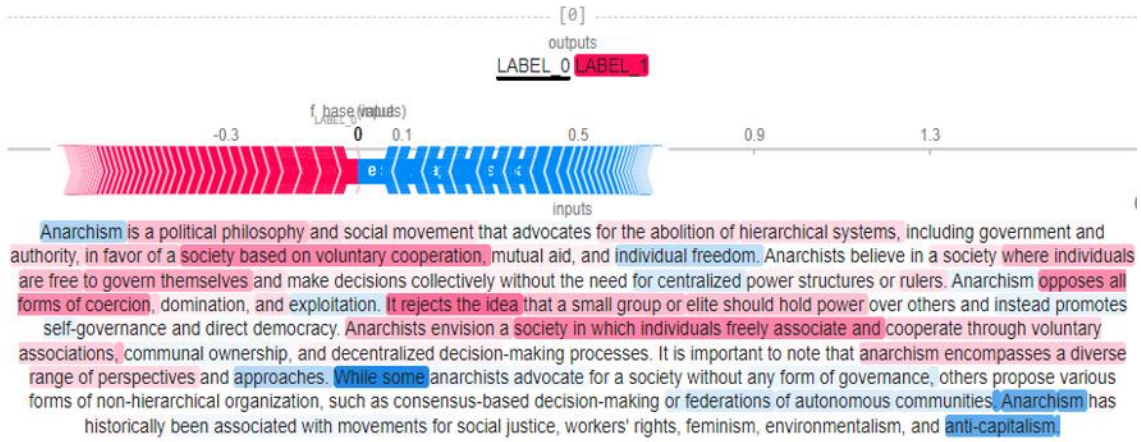


Figure 10. RoBERTa's Incorrect Classification for external Machine generated script

Explainability for XLM-RoBERTa model:

Though the accuracy and other classification performance parameters of XLM-RoBERTa is low compared to the models like RoBERTa and DistilBERT which performed best on the dataset, XLM-RoBERTa successfully classified the human generated and machine generated texts in the real time scenarios unlike RoBERTa and DistilBERT which have failed to do so.

Here are couple of samples where XLM-RoBERTa is correctly classifying.

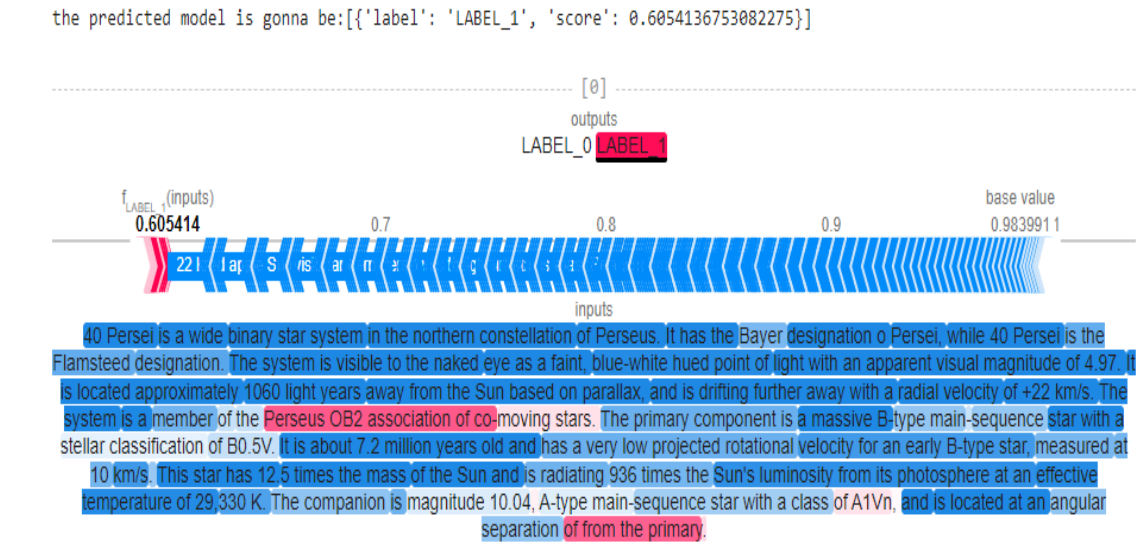


Figure 11. XLM-RoBERTa's classification of human generated text

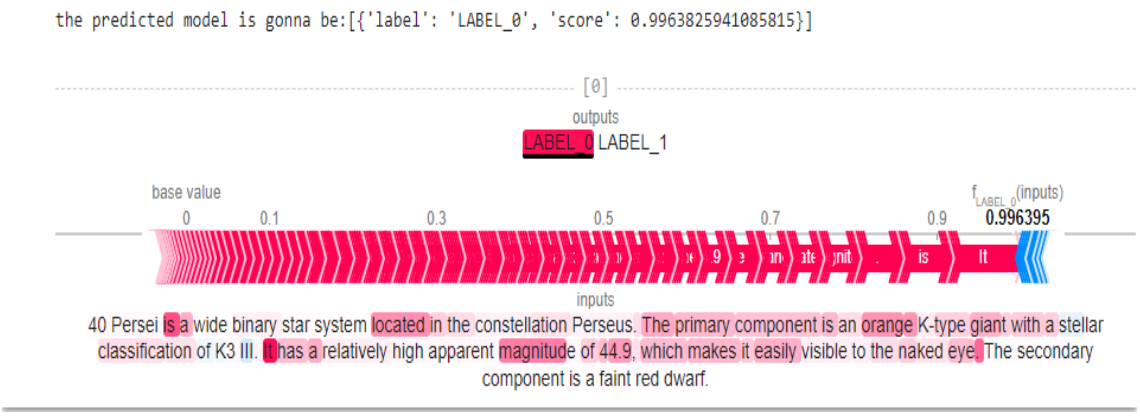


Figure 12. XLM-RoBERTa's classification of machine generated script

The samples shown in above figure are external text samples (i.e. the samples which are not the part of the dataset). XLM-RoBERTa classified those samples correctly. It not only classified these datasets correctly, but it also classified the samples in correct manner which were the part of the main dataset.

As GPTZero is not the model but the perplexity-based technique, explainability is not applicable for the same.

3.4.1.2 Training and testing models with single topic-oriented dataset

Here, the models primitively trained on the dataset having multiple text records of single topic (or title for reference). But the number of text records for human generated is same as the machine generated. For instance, If there are 50 text records of single topic (or title), 25 text records will be human generated and other 25 will be machine generated. The test dataset follows the same structure as train. Machine generated texts are extracted by repetitively asking the same question to ChatGPT as it answers with different set of words for every prompt.

Here is the table for the same:

Models/Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Bert-cased	75	100	45	62.5
Bert-uncased	75	100	45	62.5
Distilbert-multilingual-cased	79.16	100	54	70.58
GPTZero	68	78.57	45.38	57.89
Roberta	45.83	45.83	100	62.85
XLM-Roberta	87.5	100	72	84.2

Table 3. Results when all models are trained and tested on single topic-oriented English Language dataset

When the dataset changed from multiple topic to single topic, all classification parameters for models and method reduced drastically. Out of which RoBERTa's performance is worse than other models and method. XLM-RoBERTa performed the best. Over here surprisingly, DistilBERT performed well compared to BERT which is efficient as DistilBERT is the compressed version of BERT. Performance parameter of GPTZero has also reduced considerably from the previous test case.

3.4.1.3 Training models with Multiple topic and testing on single topic-oriented datasets:

After testing on the single topic, the different test case is created where the dataset of multiple topics (or titles) has been provided as a training and models were tested on the dataset having single topic. Here are the results for the same.

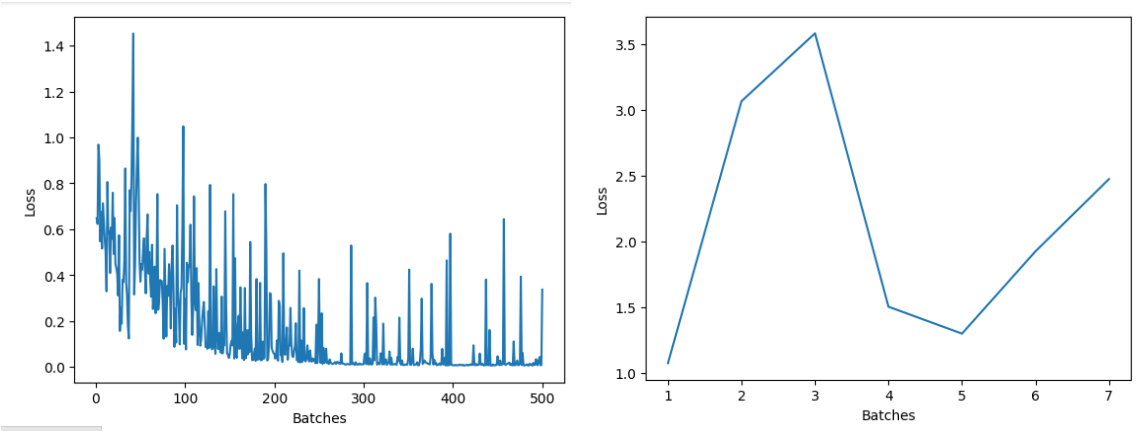
Models/Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Bert-base-uncased	66	100	29	45
Bert-base-cased	66	100	29.16	45.15
Roberta-base	52	0	0	0
Distilbert-base-multilingual-cased	58	100	12.5	22.22
XLM-Roberta-base	52	0	0	0
GPTZero	68	78.57	45.38	57.89

Table 4. Results when models are trained on multiple topics-oriented dataset and tested on single topic-oriented dataset for English language

Compared to other test cases, the performance of all models went down drastically. Especially when it comes to models like RoBERTa and XLM-RoBERTa, they performed the worst out of the other models or methods. Comparatively GPTZero, which relies on perplexity method, performed well with an accuracy as high is 68% and 78.57% precision. When it comes to classification performance of models like BERT-uncased and cased for this test case, there is no

big difference between their parameters. Even there is no difference into the graph of batch-wise training loss which will be shown into the analysis.

Loss analysis for BERT model:

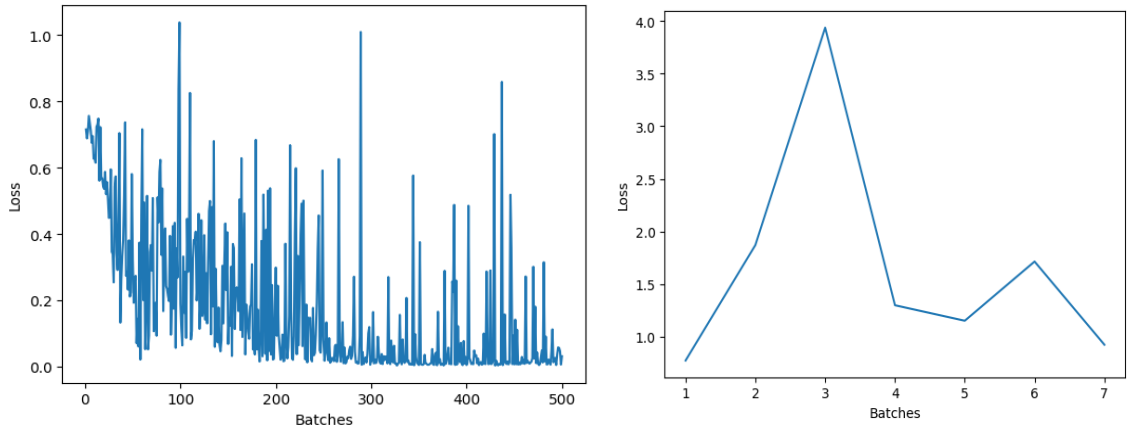


(a) **(b)**

Figure 13. (a) Batchwise loss of BERT model while training. (b) Batchwise Validation loss.

As explained in the section 3.2.5, while training, for each epoch there are certain number of the batches and for each batch, the loss is calculated. In general, this loss is calculated so that in the next batch, the model will be optimized by reducing the loss. But here before optimization process, in the section of training, this loss is stored in a variable having datatype as list. This is plotted with respect to number of batches by using Matplotlib library. It has been done for each model and graphs are plotted. In the case of BERT(both cased and uncased version), in the (a) part of the figure 12 which shows training loss, there are some surges. The highest loss of this model is above 1.4. This highest peak is observed when the number of batches were somewhere around 50. As number of batches increased, there were some spikes but the frequency of those spikes were reduced and the overall trend is going down. In the section (b) which is validation loss, the highest peak was 3.5 and the overall trend is somehow increasing which entails though BERT achieved comparatively significant accuracy, title(topic) does leave the significant impact.

Loss analysis of DistilBERT-multilingual model:

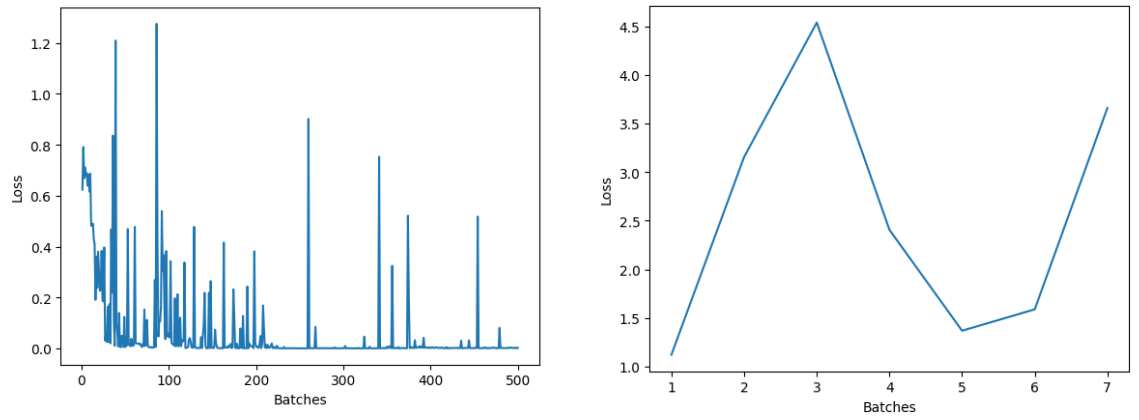


(a) (b)

Figure 14. (a) Batchwise train loss of DistilBERT (b)Batchwise validation loss.

In the section (a) of the figure 13, just like the BERT, there are spikes of high loss though the trend is downgrading and by the 200th batch, the loss is staying approximately zero though there are some surges. The highest training loss ever reached over here is approximately 1.5. Compared to BERT the surges over here are highest which entails that it takes a very long time for this model to be trained. This also emphasizes that variance in the model is large which is responsible for frequent surges. Besides as training data is more, the variations in the data are more which is another reason for that much amount of losses. Into the section (b) of Figure 14, there is the peak of loss at the 3rd batch of the validation data. From there, it goes as low as below 1 which depicts that though this model endures lots of loss surges into the training part, in the validation part it started to show some promising results.

Loss analysis of RoBERTa model:



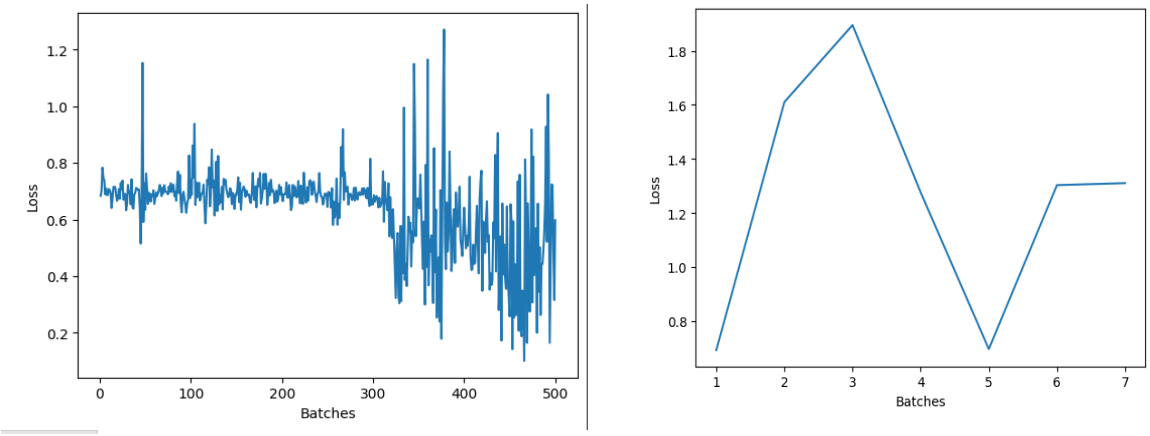
(a) (b)

Figure 15. (a) Batch wise training loss of RoBERTa (b)Batch wise validation loss.

Although this model fared worse in this test situation, there are quite few surges and the batch-wise training loss drops to zero as fast as by 40th or 50th batch which emphasizes that the grasping rate of RoBERTa is comparatively well than BERT and DistilBERT, but when it comes to section (b) of the Figure 14, at the 5th batch, the validation loss was below 1.5. but from that juncture, the loss starts to rise, and the trend became positive which entails that the number of data provided to RoBERTa was not sufficient and some parameters tweaks were also necessary

which will be described in detail into limitation section. On table and in actual sense, RoBERTa did not performed well for this test case.

Loss analysis of XLM-RoBERTa model:



(a) (b)

Figure 16. (a) Batchwise training loss of XLM-RoBERTa. (b) Batchwise validation loss.

Over here, there are some frequent surges. With those frequent surges, the average training loss is remaining constant. Over here, the trend of the graph for Section (a) of figure 15 is neither positive nor negative but neutral. Even In the section (b), initially, the trend was positive but from the 4th batch, the trend starts going down as low as below 0.8. From 5th batch, the trend stayed positive till 6th batch. After that it is neutral.

3.4.2 Key findings from the testing for English Language

In this testing, BERT model (whether cased or uncased version) performed well for overall test cases. Besides the perplexity-based method like GPTZero also performed well. However, models like RoBERTa and XLM-RoBERTa performed worst in 2 out of 3 test cases.

DistilBERT performed well compared to RoBERTa and XLM-RoBERTa for all testcases.

These findings clearly entail that fixed masked algorithm works well which is implemented in BERT models (it will be explained further in discussion session). Besides perplexity-based models like GPTZero were able to classify the paragraphs which were generated by Language Models like GPT-4 and Google BARD which confirms the usage of watermarking algorithm for Language Models from their creators like OpenAI and Google respectively.

AdamW optimizer doesn't works properly for XLM-RoBERTa as the loss were remaining constant and it appears like model was not able to learn the data pattern properly enough to classify machine generated texts from humans.

3.4.3 Testing for Hindi Language

The test cases for Hindi language is same as English but there will be additional test case for the same. The main motive for this additional test case is to check the model's ability to classify human generated text from machine generated which is in Hindi language, when the model is pretrained on that language.

3.4.3.1 Training and Testing the models with multiple topics-oriented dataset:

Just for the information, in Hindi language testing, each Train and test dataset has nearly 200 topics (or titles) that are different from train to test dataset and vice versa. Each topic has human and machine generated text pairs. Below table is the result for this test case:

Models Name	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BERT-Cased	90.82	96.11	85.34	90
BERT-Uncased	48.01	100	0.1	0.9
DistilBERT-Multilingual	99.2	100	98.5	99.2
RoBERTa	79.69	77.97	85.09	81.37
XLM-RoBERTa	98.99	100	98	99.02
GPTZero	47.86	0	0	0

Table 5. Results for models when trained and tested on multiple topics oriented Hindi language dataset

In the above table, XLM-RoBERTa, which has RoBERTa architecture as the foundation but trained on multiple languages in which Hindi is involved[23], and DistilBERT-multilingual performs much better than the rest of the models. Whereas, GPTZero performs the worst. Surprisingly there was huge difference between performance of BERT cased version and uncased version. Due to the time constraints, these models were not tested and explained like the tests done in the English languages on the real-time texts which will be specified into the limitations section as well.

3.4.3.2 Training and Testing models with single topic-oriented datasets:

In this section, the models are trained and tested on the single topic-oriented datasets. The results are shown on next page.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BERT-cased	75	75	100	66
BERT-Uncased	50	50	100	74
DistilBERT-Multilingual	91.66	85.7	100	92.3
GPTZero	50	0	0	0
RoBERTa	50	0	0	0
XLM-RoBERTa	66	60	100	74.99

Table 6. Results of models when trained and tested on single topics-oriented Hindi dataset

When the training for the single dataset performed and tested with another dataset having same format as train, all values of classification parameters of the models reduced drastically but the model like DistilBERT doesn't show that significant dip compared to the previous test case. From the above table, DistilBERT-Multilingual performed well among all the models listed whereas RoBERTa and GPTZero performed the worst. BERT-cased performed comparatively well than its uncased version and XLM-RoBERTa model.

3.4.3.3 Training models with multiple topics and testing with single topic-oriented datasets:

In this section all models undergone through training with the datasets having multiple topics-oriented text records. The results for the same as shown below:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BERT-cased	36	41.86	72	52.94
BERT-Uncased	50	50	100	74
DistilBERT-Multilingual	38	41	56	47.45
GPTZero	50	0	0	0
RoBERTa	50	0	0	0
XLM-RoBERTa	56	53.84	84	65.62

Table 7. Results of models when trained on multiple topic-oriented dataset and tested on single topic-oriented dataset

from the above table, it becomes clear that when the models are trained on the multiple topics-oriented dataset and when it is tested for the single topic, all values of classification parameters (i.e. Accuracy, Precision, Recall and F1-score) reduce drastically which entails the impact of topics on the classification behaviour. In the above table, XLM-RoBERTa performed well compared to other models (or methods) shown above. Whereas models like RoBERTa and GPTZero performed worst.

Loss analysis of BERT

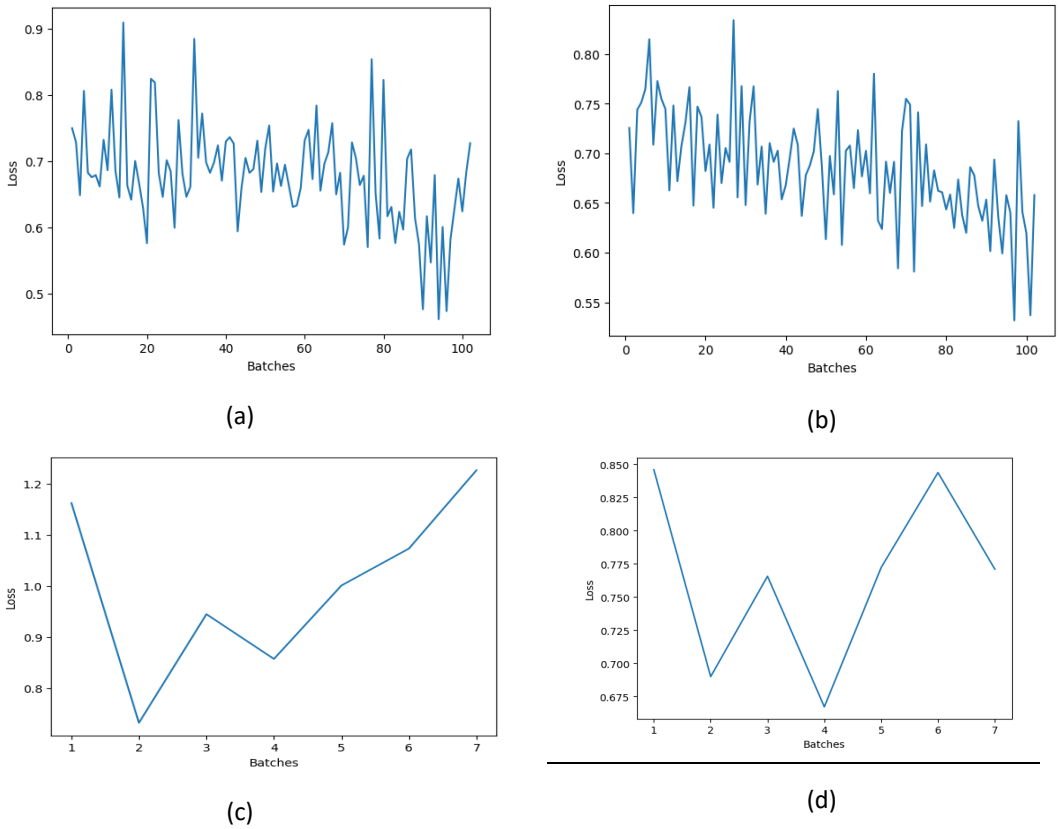
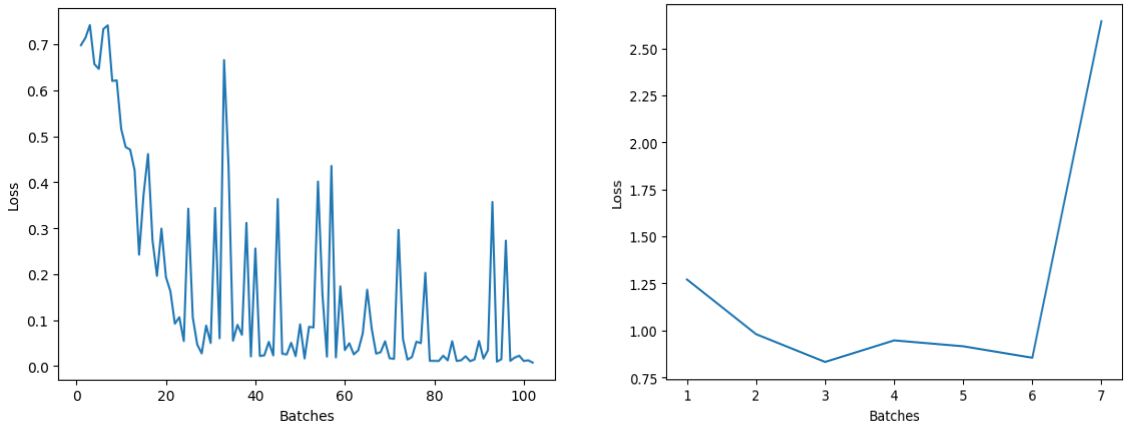


Figure 17. (a) Batchwise training loss of BERT-cased (b) Batchwise training loss of BERT-uncased. (c) and (d) are batchwise validation losses for (a) and (b) respectively.

The training loss for BERT-cased and BERT-uncased is nearly similar which is shown in section (a) and section (b) of the Figure 16 respectively. The trend is moving downward and the rate of movement is very low. When it comes to batchwise validation loss, as shown in section (c) after 4th Batch, the loss for BERT-cased is increasing, compared to BERT-uncased which is shown in section (d) of Figure 16 where from the 6th batch, the trend is going down.

Loss analysis of DistilBERT-multilingual:

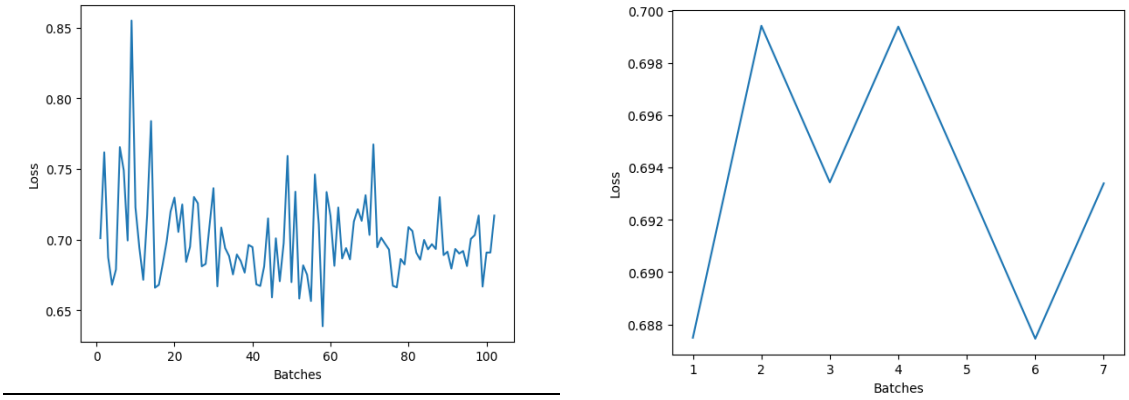


(a) (b)

Figure 18. (a) Batchwise training loss of DistilBERT-Multilingual (b) Batchwise validation loss of Distil-BERT

When it comes to training loss of DistilBERT, thought there are surges, the overall trend goes down roughly from the 10th batch the highest. From that juncture, it stays as low as tending to zero. In the Section (b) of figure 17, the loss from the 6th batch shoots beyond 2.50.

Loss analysis of RoBERTa:

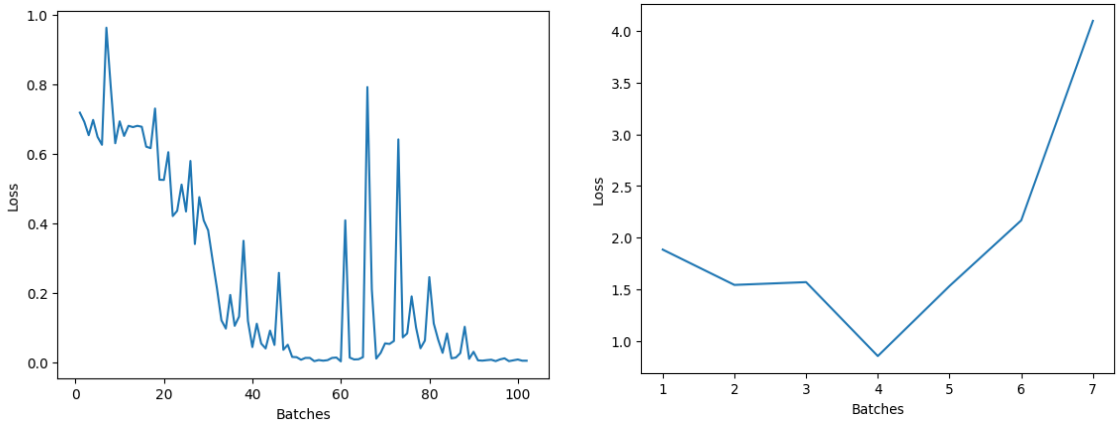


(a) (b)

Figure 19. (a) Batchwise training loss of RoBERTa (b)Batchwise validation loss

The training loss for the RoBERTa is staying somewhere around 0.70 and the trend over here is neutral till the last batch. In the section (b) of the Figure 18 which depicts the graph of validation loss, the loss dips down to as low as below 0.688 on 6th batch from the 4th batch where it was as high as 0.7. From there, the loss starts soaring.

Error analysis of XLM-RoBERTa:



(a) (b)

Figure 20.(a) Batchwise training loss of XLM-RoBERTa (b) Batchwise validation loss of XLM-RoBERTa

As shown in the section (a) of Figure 19, the spikes are low compared to other models. The training loss starts to get low from the 40th batch which is slower than DistilBERT-Multilingual. In the section (b), the validation loss was low on 4th Batch but from there, it starts to rise, and it goes as high as 4.0.

3.4.3.4 Analysis of Model pretrained on Hindi Language:

For this analysis, the model is selected which is pretrained on Hindi language. This Model is known as HindiBERT[31] designed by R. Joshi et. al. trained on nearly 3 billion Hindi tokens. The main motive of this analysis is to check whether the model pretrained on one language classifies human generated and machine generated texts accurately which are written or generated in that language. For Instance, HindiBERT's classification ability will be checked for human and machine generated Hindi language texts. This analysis is the combination of previous analyses which will be performed on this model.

Training and testing HindiBERT with datasets of multiple topics:

The test and analysis done for this model is similar to Section 3.3.3.1. The results are as shown below:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
HindiBERT	98.4	100	97.11	98.5

Table 8. Results of HindiBERT when trained and tested on multiple topics-oriented dataset

For test dataset the model was performing exceptional. Hence this model is checked on real time basis by creating the pipeline. Then the model's decision is explained using SHAP explainer.

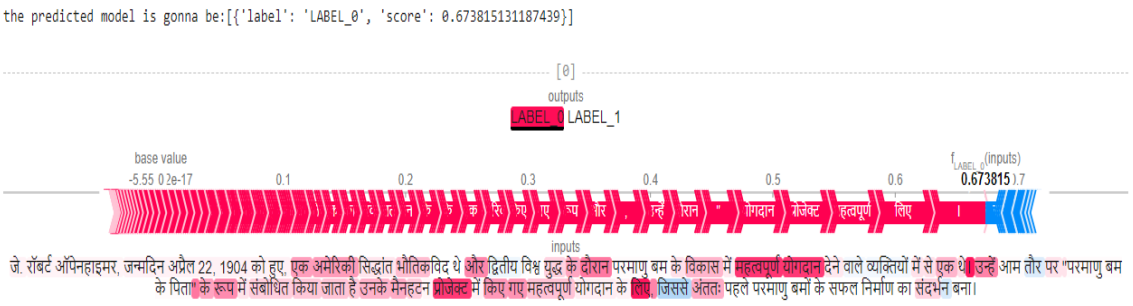


Figure 21. Explainability when machine generated text passed through HindiBERT

When machine generated Hindi text passed through HindiBERT, it classified it properly. Then this model is tested for human generated Hindi text. The results are as per below image.

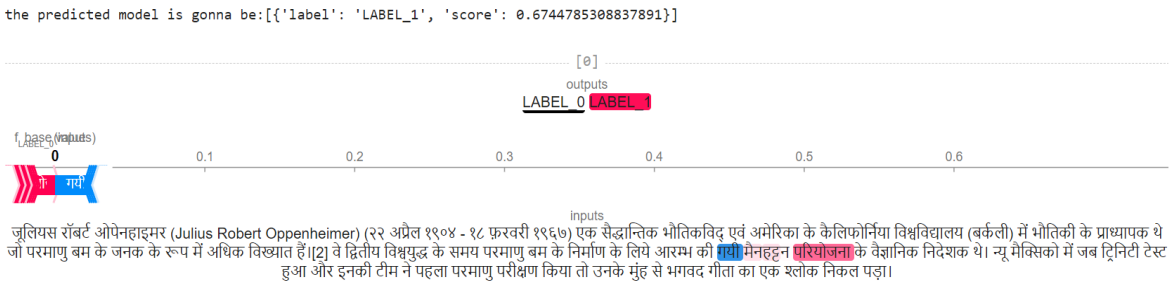


Figure 22. Explainability when human generated Hindi text passed through HindiBERT

From above images, the model perfectly identifies both texts properly. From average guy point of view who knows Hindi will be able to identify human generated and machine generated text. It is because it has been observed that chatbots like ChatGPT, GPT-4 are using English words which are written in Hindi language. The second observation from the human perspective is some Hindi words are used repetitively. The third findings over here is the frequency of conjunctions in machine generated is low than human generated.

Training and testing HindiBERT with datasets of single topic:

When the model is exposed to dataset having single topic (title), all values of classification parameters reduced drastically. But compared to other models, this model has exceptional results

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
HindiBERT	58.33	100	16.66	28.57

Table 9. Results of HindiBERT when trained and tested on single topic-oriented dataset

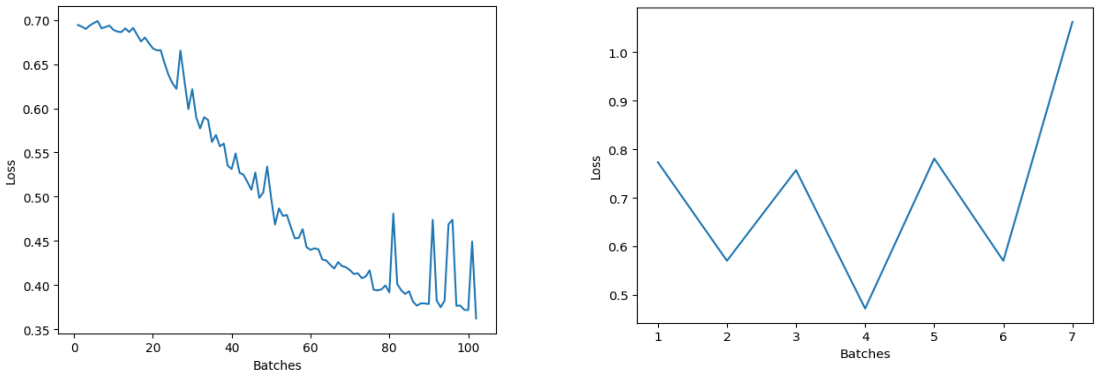
Training HindiBERT with datasets having multiple topics and testing with datasets of single topic:

Just like section 3.3.3.3, HindiBERT has been tested with datasets where each text record is dedicated to only one topic and trained with the dataset of multiple topics. The results are as per the below table.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
HindiBERT	60	58	72	64.28

Table 10. Results of HindiBERT when trained on multiple topics-oriented dataset and tested on single topic-oriented dataset

The exceptional thing over here is classification parameters are much better than previous test case performed on HindiBERT. The error analysis performed on the same and the results are as per the below figures:



(a) (b)

Figure 23. (a)Batch wise training loss of HindiBERT (b) Batch wise validation loss of HindiBERT

In the section (a) of the Figure 22, the loss of HindiBERT reduces gradually and in the end, there are some spikes, the overall trend is downward tending to 0. Into the section (b), for validation loss, though the trend is inverse of the section (a), On train and test dataset the model performed comparatively well than other models for this test case.

3.4.4 Key findings from the testing in Hindi language

Just like the testing for English Language, the topic (or title) leave the significant impact on the models and their parameters.

Many Hindi words generated by machines like ChatGPT, GPT-4 are used repetitively which becomes easy for machine to identify.

Many times, Hindi words generated by machines have the reference of English. For instance, the word project has a specific translation in Hindi but it has been used directly which generally is not observed into the Hindi sentences that is written by human. HindiBERT detects it accurately.

Frequency of morphological conjunctions in human written Hindi text is more than in machine generated texts which is also detected by HindiBERT.

When it comes to other models, XLM-RoBERTa performed well than rest of the models. Though classic BERT models and DistilBERT-Multilingual performed well, they failed to maintain that performance in the test case section 3.4.3.3.

3.4.5 Testing for Japanese Language

After testing for Hindi language, models were trained and tested for Japanese. The results will be analysed over here. In this section, there will be 2 test cases through which the model will be tested. As the dataset for single topic doesn't exists for Japanese, the two test cases are as follows:

3.4.5.1 Training and testing the models with multiple topics:

In this section, all the models were trained and examined on the dataset having multiple topics. Each dataset has 400 different topics (or titles). This dataset has the even distribution for number of human-generated and machine-generated text records. Same is repeated with teste dataset. The results are as per the below table.

Models/Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BERT-cased	98.9	100	98	99
BERT-Uncased	99.49	100	99.014	99.5
DistilBERT-Multilingual	100	100	100	100
GPTZero	48.6	0	0	0
RoBERTa	99.7	100	99.5	99.7
XLM-RoBERTa	99.7	100	99.5	99.7

Table 11. Results of all models and method when trained and tested on Japanese language multiple topics-oriented dataset.

From the above table, DistilBERT performed well than other models whereas GPTZero performed the worst. DistilBERT is giving 100% results in every classification parameter, it appears that model is somehow overfitting. Other than these two, rest of models performed equally well.

3.4.5.2 Analysis of Model pretrained on Japanese Language:

This analysis is like the one preformed on Hindi Language. In this, there is the model called BERT-base-Japanese-char which is pretrained on over 3 billion Japanese kanji letters and 170 thousand Japanese books[32]. After training and testing the model on dataset, here are results:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BERT-base-Japanese-char	100	100	100	100

Table 12. Results of Japanese BERT when trained and tested on Japanese language multiple topics-oriented dataset.

The results appeared to be sceptical because each classification parameter was giving 100% results, hence this trained model tested on random Japanese texts which were generated by human and machine. Further this model explainability is also shown. Here are below results.

Figure 24. Japanese text correctly predicted as machine generated by pretrained model

For human, the results are as per the below figure

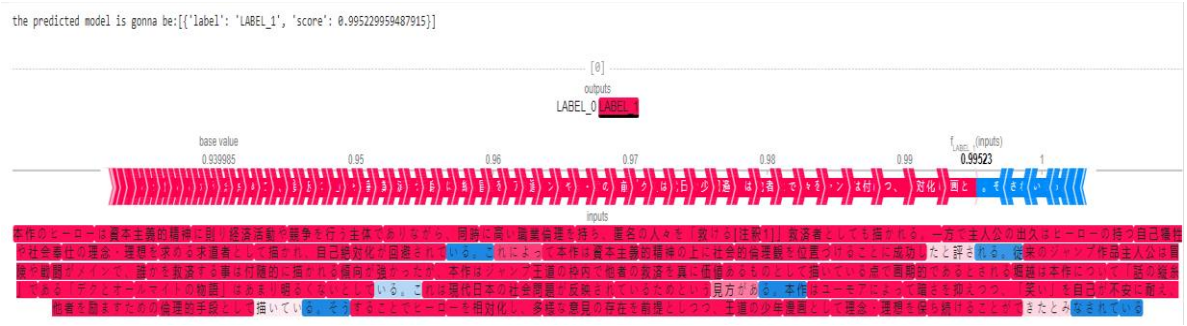


Figure 25. Japanese text correctly predicted as the human generated

3.4.6 Key findings from the testing in Japanese language:

After consulting with few primary point of contact of Japanese language speaking people, it was understood that Japanese speaking people these were the points from which one can identify human generated script from the machine one an probably BERT-Japanese is also using those parameters:

1. Length of sentences: From the datasets and even the random texts that has been provided to the model pipeline, mostly the length of human-generated Japanese text is more than the machine one.
2. Sentence's complexity: From the section 3.4.5.2, the Figure 24 is the output of correct prediction of machine generated text whereas Figure 25 is the same but for human generated text. After careful observation, Japanese text shown in Figure 25 has more complex Kanji[61] words than Figure 24 where they have just used Hiragana[62] and Katakana[63] style frequently to convey message.

3. Usage of pronouns: In the Japanese text of figure 25, the usage of pronouns is more frequent unlike in the figure 24 where they used name of the person more frequently rather than using pronouns.
4. Lexical Resource: The Lexical diversity for the Japanese text shown in figure 25 is more than the lexical diversity for the text shown in figure 24 which became the crucial factor for model to differentiate machine generated texts from humans one.

3.4.7 Testing for German Language

In this section all models (other than HindiBERT and BERT pretrained on Japanese) were went through two cases that varies on the structure of train and test dataset: test case 1 will be the test using train- test dataset pair, each having 400 topics (or titles) with texts written for the same and test case 2 consists of pair having 200 topics where each topic has human and machine generated text.

3.4.7.1 Test case 1

The results for this test case are derived as follows:

Models/Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BERT-cased	94	98.8	88.7	93.5
BERT-Uncased	88.25	91.39	83.5	87.39
DistilBERT-Multilingual	99	99.4	98.4	98.9
GPTZero	52	100	1	3
RoBERTa	98.75	97.79	99.48	98.72
XLM-RoBERTa	99.75	100	99.48	99.74

Table 13. Results of all models and a method when trained and tested on German language dataset.

Here the models which are trained on multiple languages are giving the best results compared to the other models. The worst performing model in this case was GPTZero though when it comes to classification in other languages than English, it performed well.

3.4.7.2 Test case 2

For the test case 2 the dataset provided is the revised version of the dataset provided in test case 1 where the number of topics has been reduced to 200 from 400. In that, each 200 topics has the texts which are generated by human and machine both. Models trained for this arrangement and here are the results:

Models/Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BERT-cased	90.82	96.11	85.34	90
BERT-Uncased	81	93.6	62.3	77.2
DistilBERT-Multilingual	95.6	92.06	100	95.86
GPTZero	49.3	0	0	0
RoBERTa	98.75	97.79	99.48	98.72
XLM-RoBERTa	99.5	100	99.13	99.5

Table 14. Results of all models and a method when trained and tested on revised dataset of German language.

Even though the number of topics reduced, and each topic has human and machine generated text, there is no significant difference in the performance. XLM-RoBERTa performed well compared to other models for this test case.

When it comes to testing this model on real-time basis, surprisingly DistilBERT-Multilingual, XLM-RoBERTa, which are pretrained on multiple datasets, misclassifies the human generated and machine generated text. But both architectures of BERT model and RoBERTa model classifies texts well when they have been converted to pipeline (like section 3.3.3.1, 3.3.3.4, and 3.3.5.2).

3.5 Discussion:

After testing every model and keeping their findings in consideration these are some takeaways that pointed out for each model used in this analysis:

The topics (or title column in the dataset described in section 3.1) has the impact on classification performance of human written and machine generated text.

If classification model especially BERT is pretrained on the specific language, it perfectly classifies human generated text from machine generated for that language without much influence of topics(titles).

For models that are used in almost every test case, here are the main takeaways for the same:

3.5.1 BERT-cased and uncased version

In the instance of BERT-base cased/uncased, these models are pre-trained in an unsupervised manner across the 3 billion words. These phrases come from the corpus of Google Books and Wikipedia[4]. The English language is the main training language for this model and many of the letters in German language is like English. As a result, the model was producing acceptable results in both German and English. Additionally, the model correctly identified each corpus (either human or machine-generated) when it was given to it.

Besides, BERT-cased works on masked language algorithm where 15% of the input corpus is masked and model predicts those word depending on rest of words into the corpus[4]. If models predict words with low probabilities from the input corpus, the entropy of that corpus is high[59], and the high information depicts that text is human written. Besides especially when it comes to English, BERT learns the pattern by which the human written text is classified from

machine generated. But due to time constraints, the datasets were low. If the veracity of dataset increased, one can built the sustainable and robust classifier out of it.

3.5.2 DistilBERT

DistilBERT is a model that has been trained via transfer learning. That is, specified weights or information are transferred from the teacher to the student using a cross-entropy over the soft objectives (teacher probability). It implies that the predetermined weights of the Bert model have been reduced from the big model (teacher) to the tiny model (student)[26,42,43,56]. There is a chance that some of the most significant parameters that would be necessary for the models will be lost during the distillation process. In the event of a multilingual approach, these factors may change according to language.

3.5.3 RoBERTa

RoBERTa is primarily trained on English language. Hence, inability to work on other languages, though the model was giving satisfactory results is an obvious thing. RoBERTa used dynamical masking algorithm where in every epoch, tokens are masked differently[22], unlike BERT which only does it once[4]. Due to dynamic masking the variance increases. As the variance increases, the chances or correct prediction reduced significantly due to inability of learning from the data. As a result, for that model, entropy is increased because of which it misclassifies machine-generated script as human. This is the reason why model's classification performance parameters are all time low when it is trained on the dataset having multiple topics and tested on single topic.

3.5.4 XLM-RoBERTa

The specific dialects, variants, or regionalisms found within a language may be missed by multilingual models like XLM-RoBERTa. They frequently give a broader picture of each language, which might not be appropriate for jobs requiring in-depth linguistic comprehension. As XLM-RoBERTa is pretrained on many languages and it is using dynamic masking approach[23], the variance for such models increases as each time, the words in text are masked dynamically. Above this, due to variations in languages, this variance further soars enormously leading to overfitting.

3.5.5 GPTZero

This architecture relies on the gpt2-large model, which is pretrained on the English language, which explains why it achieves very high accuracy in English and extraordinarily low accuracy in other languages[27]. Because the machine-produced corpus from LLMs (Large Language Models) like GPT-4 and Google Bard has been correctly identified, bard also uses the watermarking process to preserve the copyright on the words that have been generated.

This might fail due to following points:

Language Model Training: Depending on the language, the amount and calibre of training data may differ, which may influence how well the language model performs and how easily the perplexity scores may be understood.

Language Complexity and Grammar: Language complexity and grammar can have an impact on perplexity scores. Languages that have more complicated grammatical structures, morphological variances, or ambiguous word usage may present difficulties for language models and may have higher perplexity scores.

Even though perplexity can be a useful statistic for comparing language models across different languages, it is crucial to take into account the aforementioned considerations and tailor the analysis to the particular language and data at hand. Obtaining accurate and useful perplexity scores across several languages requires comparative analysis and domain-specific evaluation.

4 Conclusion

4.1 Summary

All model's classification behaviour for human written text and writings generated by machines like ChatGPT, GPT-4 and Google BARD, under different languages and circumstances have been examined and analysed. The languages used other than English were Hindi, German and Japanese. The models like BERT-cased, BERT-uncased, RoBERTa, the model pretrained on multiple languages like XLM-RoBERTa, DistilBERT-Multilingual and perplexity-based method like GPTZero have been tested, analysed, and compared with each other. Further, this same classification behaviour of models specifically pre-trained on one language like HindiBERT and BERT-Japanese has been examined on Hindi and Japanese language datasets, respectively. By this, the impact of topics on classifying human and machine generated paragraphs has been analysed. The Robustness of models pretrained on specific language is tested under different situations which emphasizes the ineffectiveness of topics on the classification behaviour of such models.

4.2 Evaluation

Overall, BERT performed well on different languages compared to other models. Despite being first trained on English, BERT was successful with German since the two languages share a lot of letters. Higher entropy and low probability predictions yields human-written text. GPTZero is using GPT-2 as a base model which is primarily trained on English data. Hence the human written sentences in English language classified well from the machine one (for instance ChatGPT, Google BARD etc.) which confirms the usage of watermarking algorithm by this model. Multilingual models like DistilBERT-Multilingual and XLM-RoBERTa classified texts well in almost all languages but failed to maintain performance in different test cases designed for those languages (especially for English and Hindi). RoBERTa classifies human generated text from machine well in the case of German language. When it comes to other language, though it performs well on the datasets, it fails to perform when it is exposed to the texts which are out of the datasets. All these models failed to perform in test case (or situation) where they have been trained on datasets with multiple topics and tested on datasets with single topic which shows the impact of topics on classification of human and machine generated texts. The models pretrained on specific languages classifies human and machine generated text of that language comparatively better than other models irrespective of topics variation. Overall, human written text varies from language models in terms of coherence and cohesion, lexical source, regional grammar, morphology and style of writing.

4.3 Future Work

Due to time constraints some analysis remained that can be further performed into the future. All models used in this analysis were ran on fixed parameters which were type of optimizer, learning rate and number of epochs. This analysis can be performed for those models on datasets used in this dissertation by varying the mentioned parameters. This analysis was limited up to 3 different languages other than English. Hence this can be expanded for different languages as well. Some test cases were not performed for some languages. So that can be expanded. The perplexity-based method like GPTZero uses GPT-2 as the foundation model for tokenization and calculating log likelihood to further calculate perplexity. GPT-2 is trained on only English language. Once GPT-3.5(that is ChatGPT) is open sourced like GPT-2, the algorithm for perplexity derived on Hugging face web site[20], can be implemented for different language by which human generated text can be classified from machine as GPT-3.5 is trained on different languages and it can be used as the base model for GPTZero instead of GPT-2. Further, HindiBERT can be compressed to a small model using the transfer learning approach based on

papers of DistilBERT and Articles available[26,42,43,56]. Further, that model can be trained to classify human generated and machine generated texts in Hindi language.

References

[1] NLTK, "Natural Language Toolkit — NLTK 3.4.4 documentation," *Nltk.org*, 2009. https://www.nltk.org/

[2] P. Joshi, Artificial intelligence with Python : build real-world artificial intelligence applications with Python to intelligently interact with the world around you. Birmingham, Uk: Packt Publishing Ltd, 2017.

[3] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv.org*, 2017. https://arxiv.org/abs/1706.03762

[4] J. Devlin, M.-W. Chang, K. Lee, K. Google, and A. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 2019. Available: https://arxiv.org/pdf/1810.04805

[5] A. Radford, K. Narasimham, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2018. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[7] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., . . . Amodei, D. (2020). Language Models are Few-Shot Learners. ArXiv. /abs/2005.14165

[8] S. Vaughan-Nichols, "GPT-3.5 vs GPT-4: Is ChatGPT Plus worth its subscription fee?," *ZDNET*, Jun. 12, 2023. https://www.zdnet.com/article/gpt-3-5-vs-gpt-4-is-chatgpt-plus-worth-its-subscription-fee/

[9] Y. Rong *et al.*, "Self-Supervised Graph Transformer on Large-Scale Molecular Data," *arXiv.org*, Oct. 28, 2020. https://arxiv.org/abs/2007.02835(accessed Jul. 29, 2023).

[10] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models," *arXiv:2302.13971 [cs]*, Feb. 2023, Available: https://arxiv.org/abs/2302.13971

[11] R. Anil *et al.*, "PaLM 2 Technical Report," *arXiv.org*, May 17, 2023. https://arxiv.org/abs/2305.10403

[12] E. N. Crothers, N. Japkowicz, and H. L. Viktor, "Machine-Generated Text: A Comprehensive Survey of Threat Models and Detection Methods," IEEE Access, vol. 11, pp. 70977–71002, 2023, doi: *https://doi.org/10.1109/ACCESS.2023.3294090*.

[13] V. Sankaran, "Scientific journals ban ChatGPT use by researchers to write studies," *The Independent*, Jan. 27, 2023. https://www.independent.co.uk/tech/chatgpt-ai-journals-ban-author-b2270334.html(accessed Jul. 29, 2023).

[14] "OII | Science journals ban listing of ChatGPT as co-author on papers." https://www.oii.ox.ac.uk/news-events/coverage/science-journals-ban-listing-of-chatgpt-as-co-author-on-papers/#:~:text=Published%20on&text=The%20publishers%20of%20thousands%20of(accessed Jul. 29, 2023).

[15] Ai Reviews, "AI-Written Product Reviews Now Found on Amazon (ChatGPT)," *reviews.ai*, Jun. 19, 2023. https://www.reviews.ai/ai-written-product-reviews-now-found-on-amazon-chatgpt/(accessed Jul. 30, 2023).

[16] A. Sengupta, "Scammers are now using ChatGPT to write Amazon reviews, give 5 stars to products," *India Today*, Apr. 27, 2023.

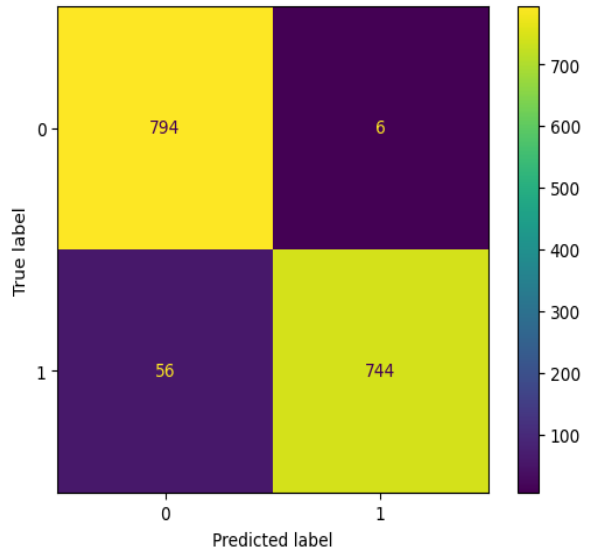
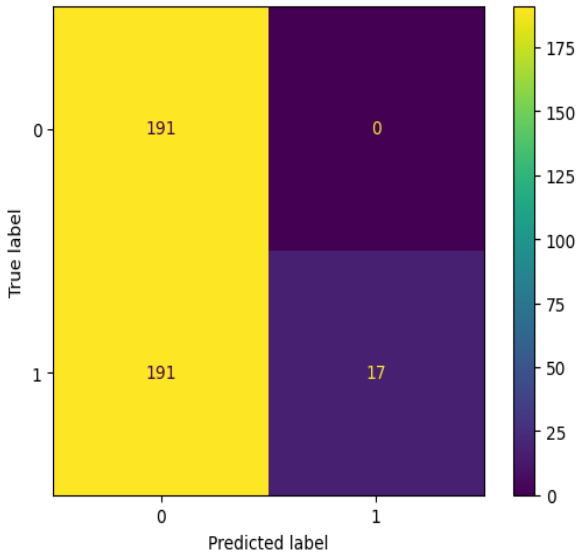
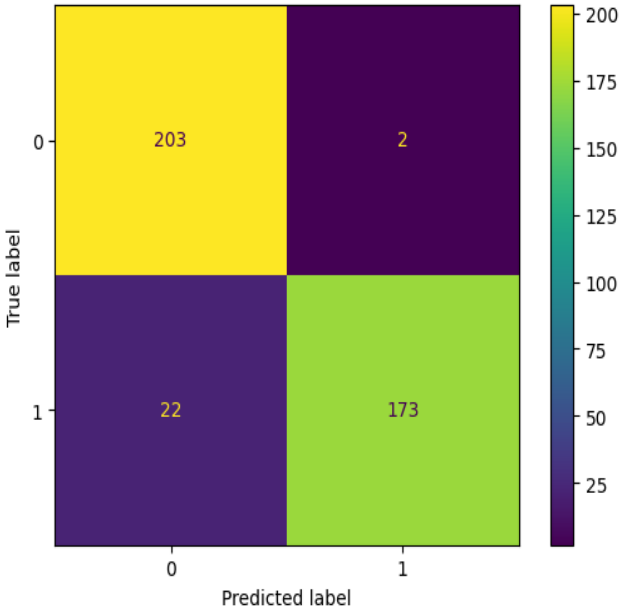
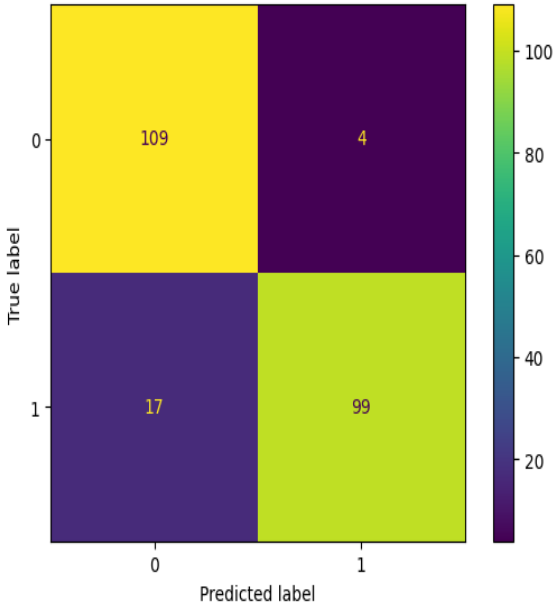
https://www.indiatoday.in/technology/news/story/scammers-are-now-using-chatgpt-to-write-amazon-reviews-give-5-stars-to-products-2365226-2023-04-27(accessed Jul. 30, 2023).
[17]R. Bommasani, K. Klyman, D. Zhang, and P. Liang, "Do Foundation Model Providers Comply with the Draft EU AI Act?," *crfm.stanford.edu*, Jun. 15, 2023. https://crfm.stanford.edu/2023/06/15/eu-ai-act.html
[18]V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, "Can AI-Generated Text be Reliably Detected?," *arXiv:2303.11156 [cs]*, Mar. 2023, Available: https://arxiv.org/abs/2303.11156
[19]K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, "Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense," *arXiv:2303.13408 [cs]*, Mar. 2023, Available: https://arxiv.org/abs/2303.13408
[20]H. F. Community, "Perplexity of fixed-length models," *huggingface.co*. https://huggingface.co/docs/transformers/perplexity
[21]"GPTZero," *gptzero.me*. https://gptzero.me/
[22]Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv.org*, 2019. https://arxiv.org/abs/1907.11692
[23]A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," *arXiv:1911.02116 [cs]*, Apr. 2020, Available: https://arxiv.org/abs/1911.02116
[24]S. Gehrmann, H. Strobelt, and A. M. Rush, "GLTR: Statistical Detection and Visualization of Generated Text," *arXiv:1906.04043 [cs]*, Jun. 2019, Available: https://arxiv.org/abs/1906.04043
[25]Y. Wang *et al.*, "M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection," *arXiv.org*, May 24, 2023. https://arxiv.org/abs/2305.14902 (accessed Jul. 30, 2023).
[26]V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv.org*, 2019. https://arxiv.org/abs/1910.01108
[27]Tayyab, B.U. (2023). *Implementation of DetectGPT and GPTzero in Pytorch could be found here: https://github.com/BurhanUlTayyab/DetectGPT* [online] GitHub. Available at: https://github.com/BurhanUlTayyab/GPTZero[Accessed 31 Jul. 2023].
[28]A. Bhat, "aadityaubhat/GPT-wiki-intro · Datasets at Hugging Face," *huggingface.co*, 2023. https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intrverview (accessed Aug. 01, 2023).
[29]V. Trevisan, "Using SHAP Values to Explain How Your Machine Learning Model Works," *Medium*, Jul. 05, 2022. https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137
[30]"An introduction to explainable AI with Shapley values — SHAP latest documentation," *Readthedocs.io*, 2021. https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html
[31]R. Joshi, "L3Cube-HindBERT and DevBERT: Pre-Trained BERT Transformer models for Devanagari based Hindi and Marathi Languages," *arXiv.org*, Jan. 08, 2023. https://arxiv.org/abs/2211.11418(accessed Aug. 01, 2023).
[32]"cl-tohoku/bert-base-japanese-char · Hugging Face," *huggingface.co*, Jun. 01, 2023. https://huggingface.co/cl-tohoku/bert-base-japanese-char(accessed Aug. 01, 2023).
[33]M. R. Grossman, P. W. Grimm, D. G. Brown, and M. Xu, "The GPTJudge: Justice in a Generative AI World," *Social Science Research Network*, May 23, 2023. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4460184
[34]M. Melin, A. Back, C. Sodergard, M. D. Munezero, L. J. Leppanen, and H. Toivonen, "No Landslide for the Human Journalist - An Empirical Study of Computer-Generated Election

News in Finland," *IEEE Access*, vol. 6, pp. 43356–43367, 2018, doi: https://doi.org/10.1109/access.2018.2861987.
[35]H. M. Gomez Adorno, G. Rios, J. P. Posadas Durán, G. Sidorov, and G. Sierra, "Stylometry-based Approach for Detecting Writing Style Changes in Literary Texts," Computación y Sistemas, vol. 22, no. 1, Mar. 2018, doi: https://doi.org/10.13053/cys-22-1-2882
[36]T. Schuster, R. Schuster, D. J. Shah, and R. Barzilay, "The Limitations of Stylometry for Detecting Machine-Generated Fake News," *Computational Linguistics*, vol. 46, no. 2, pp. 1–18, Mar. 2020, doi: https://doi.org/10.1162/coli_a_00380.
[37]D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, "Automatic Detection of Generated Text is Easiest when Humans are Fooled," *arXiv.org*, May 07, 2020. https://arxiv.org/abs/1911.00650
[38]E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature," *arXiv:2301.11305 [cs]*, no. 2, Jan. 2023, Available: https://arxiv.org/abs/2301.11305
[39]J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A Watermark for Large Language Models," *arXiv.org*, Jun. 06, 2023. https://arxiv.org/abs/2301.10226
[40]S. Aaronson, "My AI Safety Lecture for UT Effective Altruism," *Shtetl-Optimized*, Nov. 29, 2022. https://scottaaronson.blog/?p=6823
[41]R. Montti, "How The ChatGPT Watermark Works And Why It Could Be Defeated," *Search Engine Journal*, Dec. 30, 2022. https://www.searchenginejournal.com/chatgpt-watermark/475366/#close
[42]S. Sučik, "Learn how to make BERT smaller and faster," *Rasa*, Aug. 08, 2019. https://rasa.com/blog/compressing-bert-for-faster-prediction-2/(accessed Aug. 10, 2023).
[43]S. Mitrović, D. Andreoletti, and O. Ayoub, "ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-generated Text," *arxiv.org*, vol. 1, Jan. 2023, doi: https://doi.org/10.48550/arXiv.2301.13852
[44]M. Reuter and W. Schulze, "I'm Afraid I Can't Do That: Predicting Prompt Refusal in Black-Box Generative Language Models," *arXiv.org*, Jun. 14, 2023. https://arxiv.org/abs/2306.03423(accessed Aug. 11, 2023).
[45]J.D. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang, "Why Johnny Can't Prompt:
M. Lewis *et al.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," *arXiv:1910.13461 [cs, stat]*, Oct. 2019, Available: https://arxiv.org/abs/1910.1346
[46]K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, "Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense," *arXiv:2303.13408 [cs]*, Mar. 2023, Available: https://arxiv.org/abs/2303.13408
[47]M. Gallé, J. Rozen, G. Kruszewski, and H. Elsahar, "Unsupervised and Distributional Detection of Machine-Generated Text," *arXiv.org*, Nov. 04, 2021. https://arxiv.org/abs/2111.02878
[48]H. Zhan, X. He, Q. Xu, Y. Wu, and P. Stenetorp, "G3Detector: General GPT-Generated Text Detector," *arxiv*, vol. 1, May 2023, doi: https://doi.org/10.48550/arxiv.2305.12680
[49]M. Lewis *et al.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," *arXiv:1910.13461 [cs, stat]*, Oct. 2019, Available: https://arxiv.org/abs/1910.1346
[50]I. Solaiman *et al.*, "Release Strategies and the Social Impacts of Language Models," *arXiv:1908.09203 [cs]*, Nov. 2019, Available: https://arxiv.org/abs/1908.09203
[51]Y. Ma, J. Liu, and F. Yi, "Is This Abstract Generated by AI? A Research for the Gap between AI-generated Scientific Text and Human-written Scientific Text," *arXiv.org*, Jan. 24, 2023. https://arxiv.org/abs/2301.10416

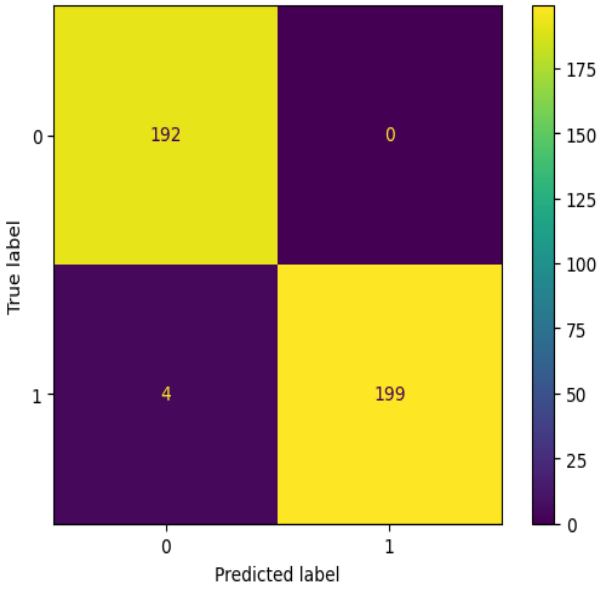
[52]X. He, X. Shen, Z. Chen, M. Backes, and Y. Zhang, “MGTBench: Benchmarking Machine-Generated Text Detection,” *arXiv:2303.14822 [cs]*, Mar. 2023, Available: https://arxiv.org/abs/2303.14822
[53]Graelo, “graelo/wikipedia · Datasets at Hugging Face,” *huggingface.co*, Aug. 21, 2023. https://huggingface.co/datasets/graelo/wikipedia(accessed Aug. 22, 2023).
[54]B. D. Horne, W. Dron, S. Khedr, and S. Adali, “Assessing the News Landscape,” *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW ’18*, 2018, doi: https://doi.org/10.1145/3184558.3186987
[55]H. face Community, “Introduction - Hugging Face NLP Course,” *huggingface.co*. https://huggingface.co/learn/nlp-course/chapter5/1?fw=pt(accessed Aug. 26, 2023).
[56]“distilbert-base-multilingual-cased · Hugging Face,” *huggingface.co*, Jun. 01, 2023. https://huggingface.co/distilbert-base-multilingual-cased(accessed Aug. 26, 2023).
[57]J. Libovický, R. Rosa, and A. Fraser, “How Language-Neutral is Multilingual BERT?,” *arXiv.org*, Nov. 08, 2019. https://arxiv.org/abs/1911.03310(accessed Aug. 26, 2023).
[58]D. Kakwani *et al.*, “IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages,” *ACLWeb*, Nov. 01, 2020. https://aclanthology.org/2020.findings-emnlp.445/
[59]C. Huyen, “Evaluation Metrics for Language Modeling,” *The Gradient*, Oct. 19, 2019. https://thegradient.pub/understanding-evaluation-metrics-for-language-models/
[60]“AdamW — PyTorch 1.10.0 documentation,” *pytorch.org*. https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html
[61]“Kanji - character | Easy Japanese,” *NHK WORLD-JAPAN*. https://www.nhk.or.jp/lesson/en/letters/kanji.html(accessed Sep. 01, 2023).
[62]“Katakana - alphabet | Easy Japanese,” *NHK WORLD-JAPAN*. https://www.nhk.or.jp/lesson/en/letters/katakana.html
[63]“Hiragana - alphabet | Easy Japanese,” *NHK WORLD-JAPAN*. https://www.nhk.or.jp/lesson/en/letters/hiragana.html
[64]BulkGPT, “Introducing BulkGPT - The fastest way to bring your ChatGPT workflow to mass production.,” www.youtube.com. https://youtu.be/osivRn3ZvvI(accessed Sep. 08, 2023).
[65]“BulkGPT - Batch Processing Of ChatGPT Requests,” *bulkgpt.ai*. https://bulkgpt.ai/(accessed Sep. 08, 2023).

Appendix 1

Due to the bulkiness, confusion matrices were not added into the main content. Hence as a part of appendix, confusion matrices for each model trained and tested on each language's Multiple topic-oriented datasets(please infer the column "Number of records in test dataset" for each language's total number of records in the Table 1 of section 3.1 :Datasets) has been shown below:

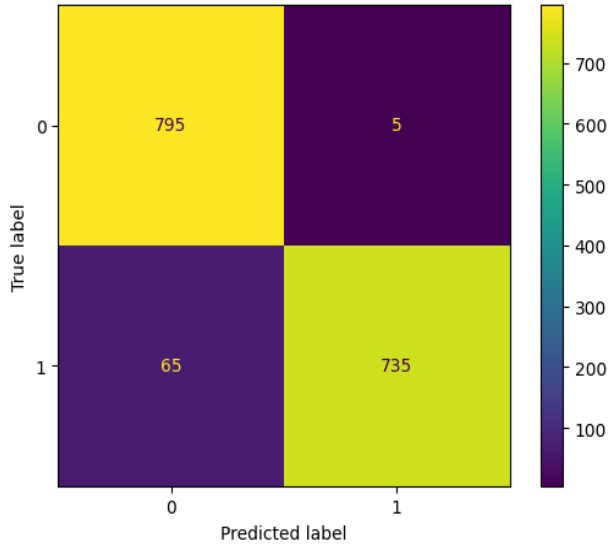
<table>
<tr><th colspan="2">Model: Bert-cased</th></tr>
<tr><td>English
 
</td><td>Hindi
 
</td></tr>
<tr><td>German
 
</td><td>German-revised
 
</td></tr>
<tr><td colspan="2"></td></tr>
</table>

Japanese

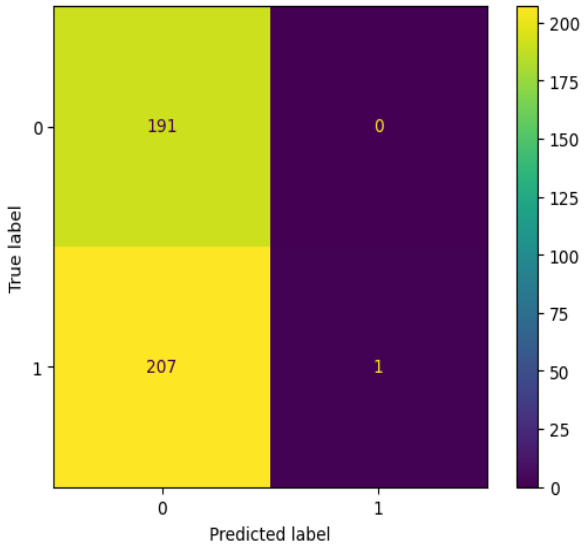


Model:Bert-uncased

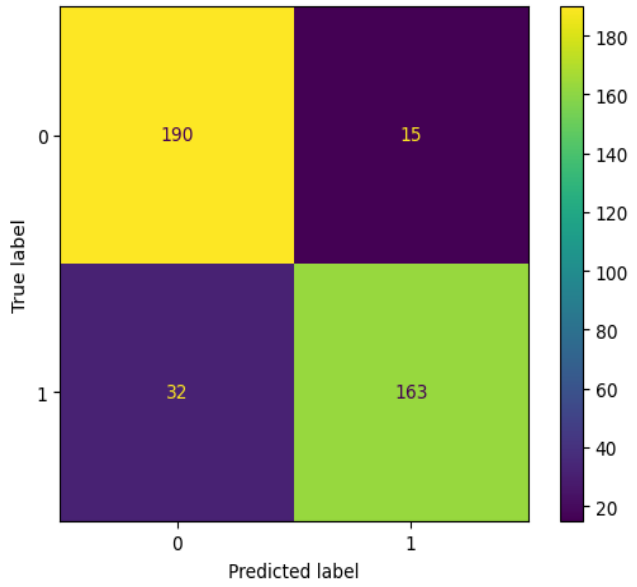
English



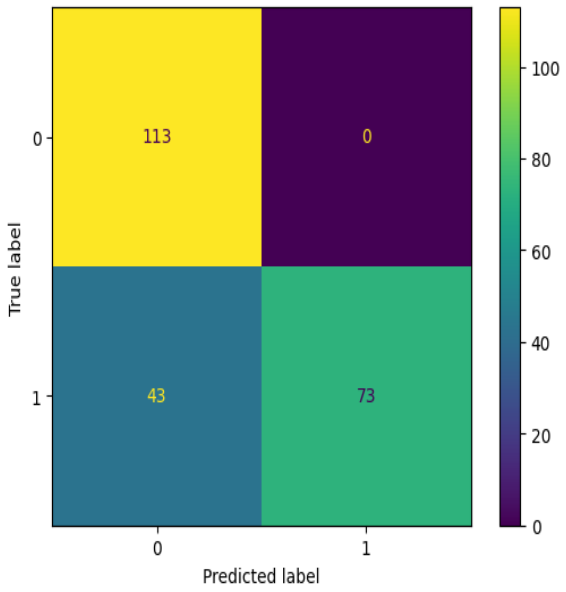
Hindi



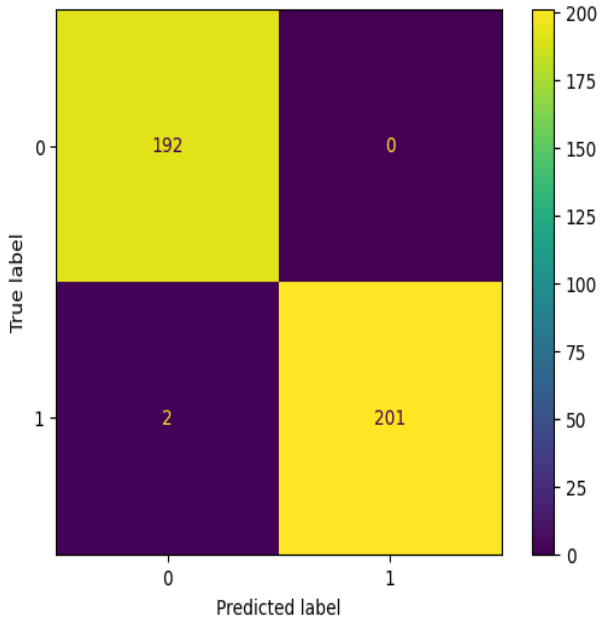
German



German-revised

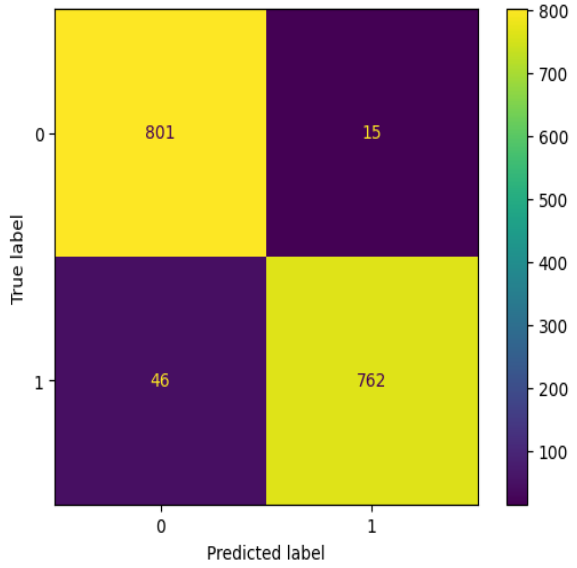


Japanese

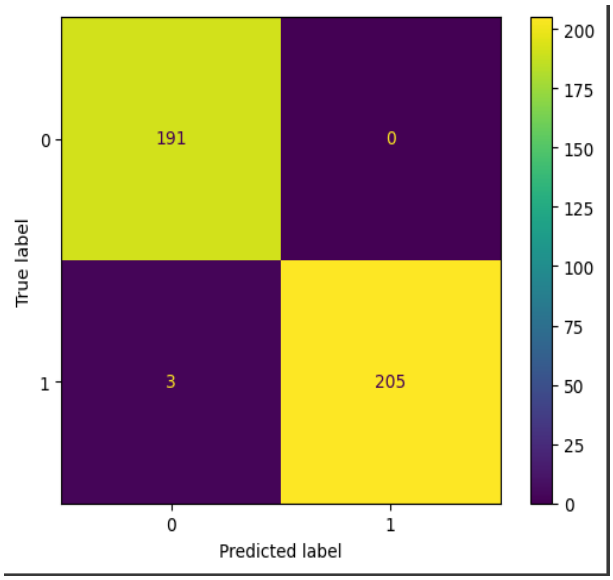


Model: distilbert-multilingual

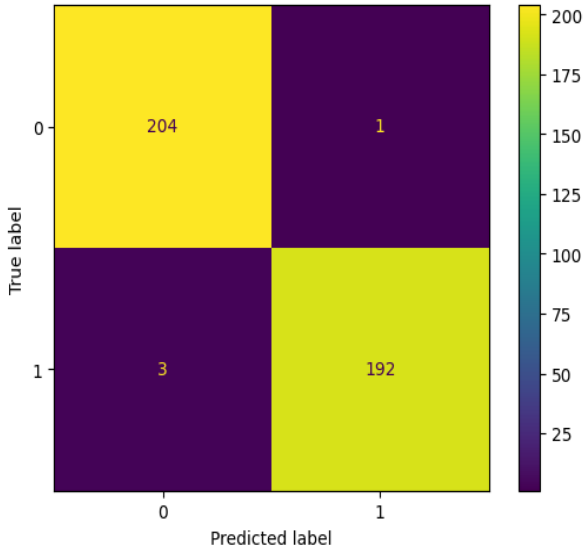
English



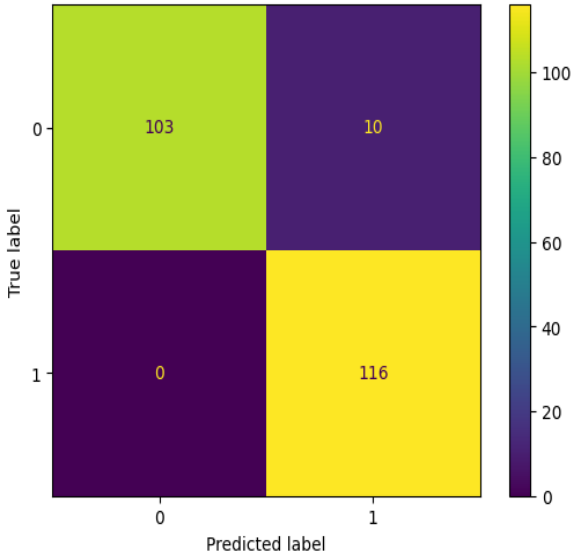
Hindi



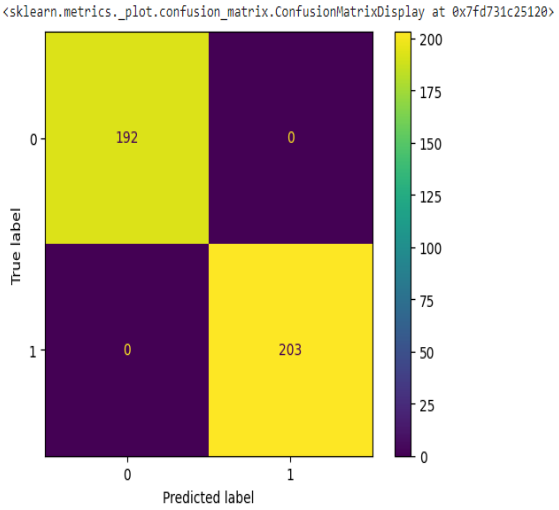
German



German-revised

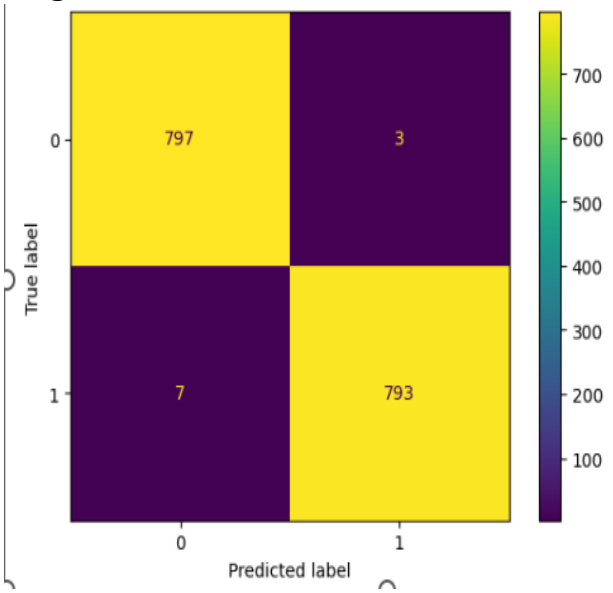


Japanese

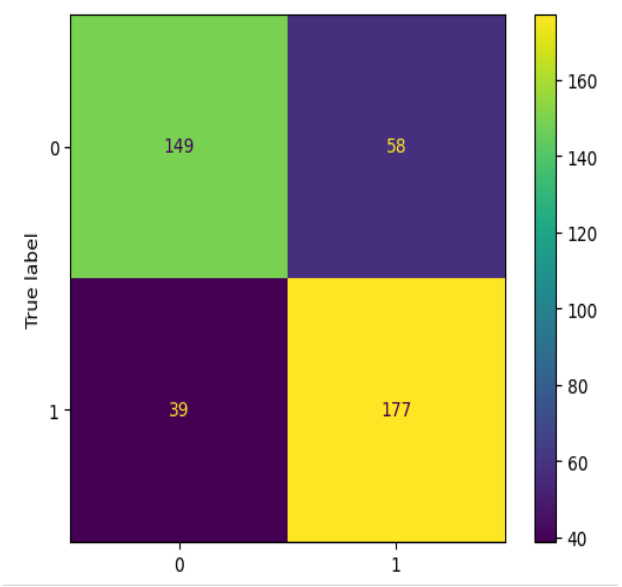

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fd731c25120>
192
0
0
203
0
1
True label
0
1
Predicted label
200
175
150
125
100
75
50
25
0

Model: RoBERTA

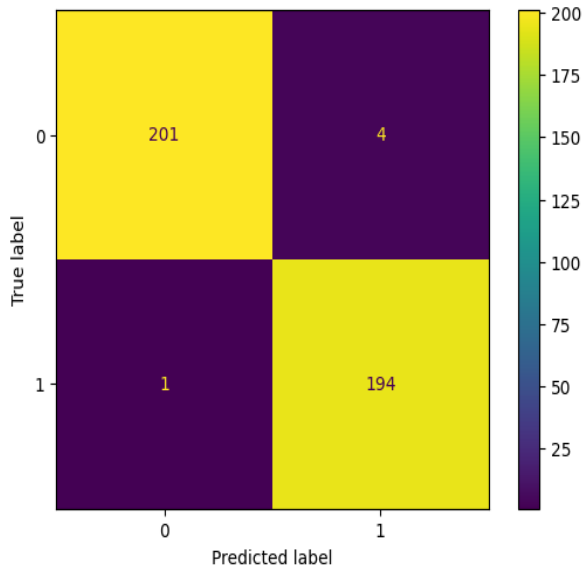
English



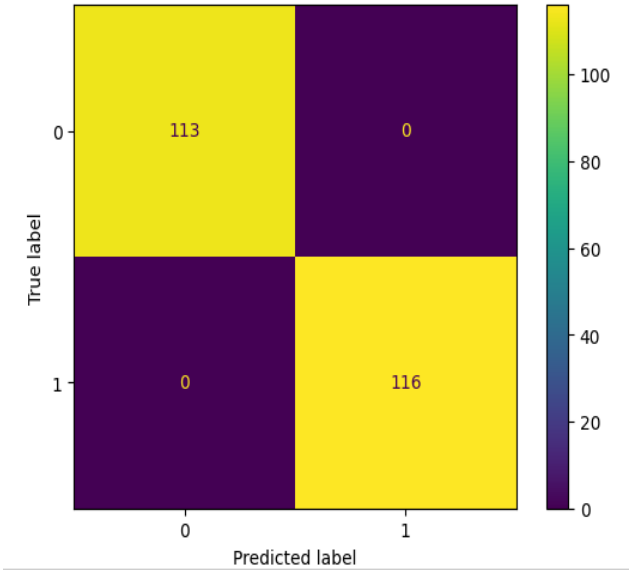
Hindi



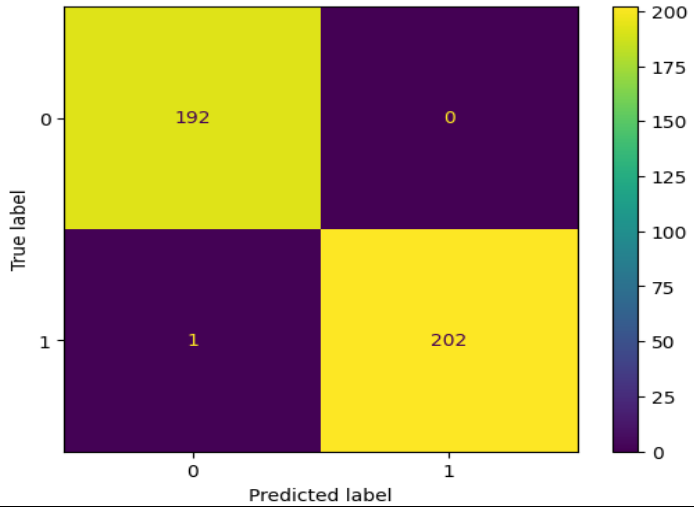
German



German-revised

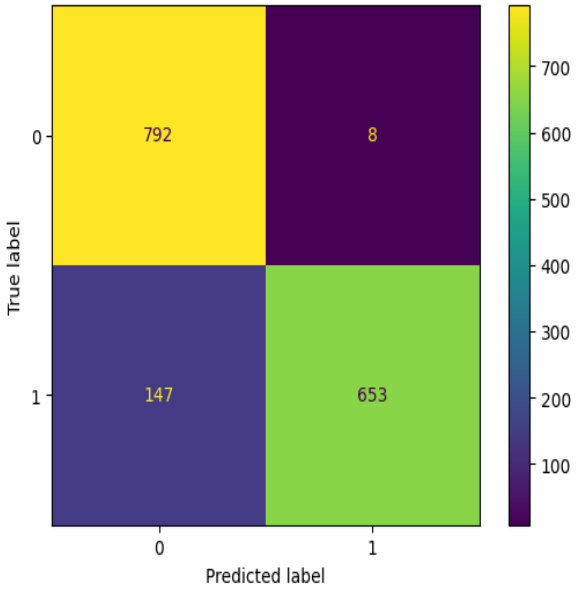


Japanese

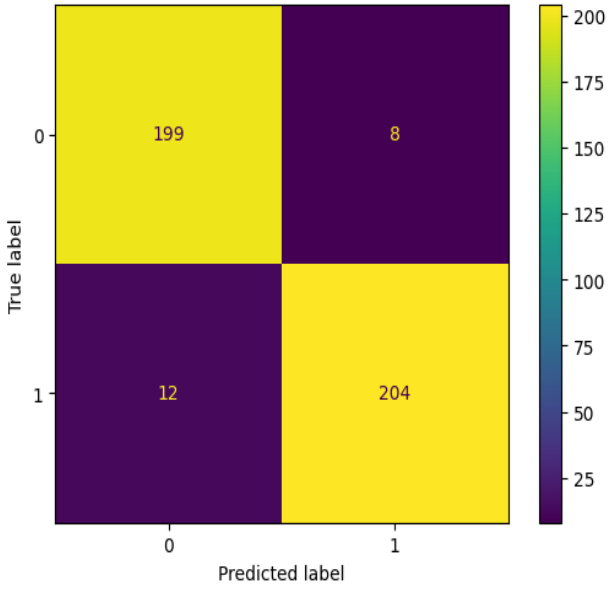


Model: XLM-RoBERTA

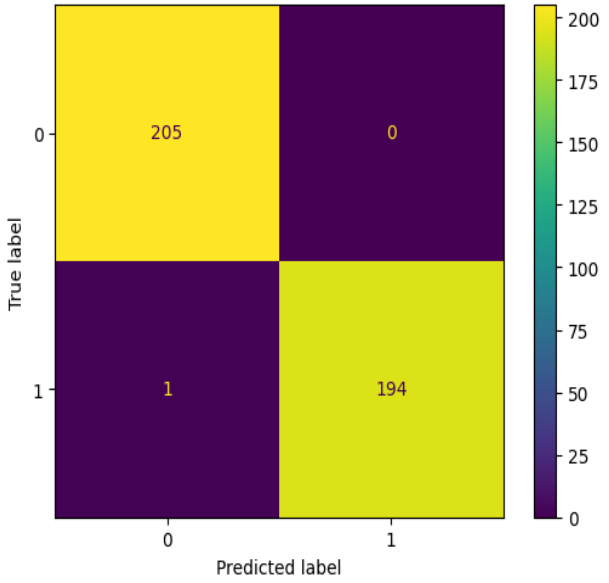
English



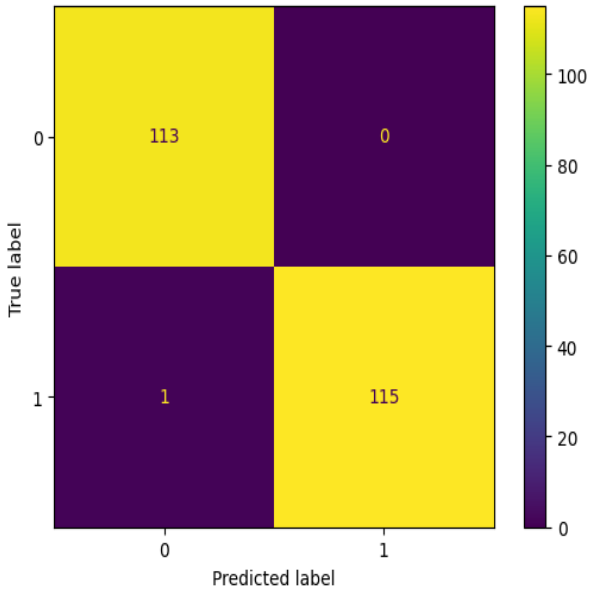
Hindi



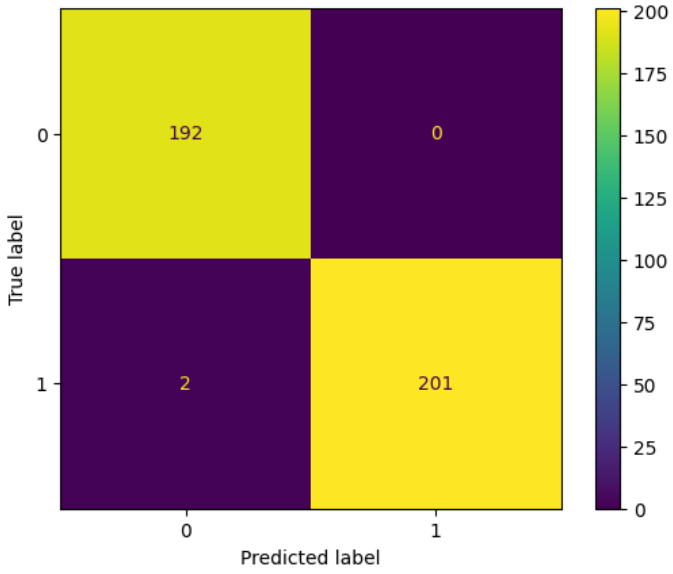
German



German-revised

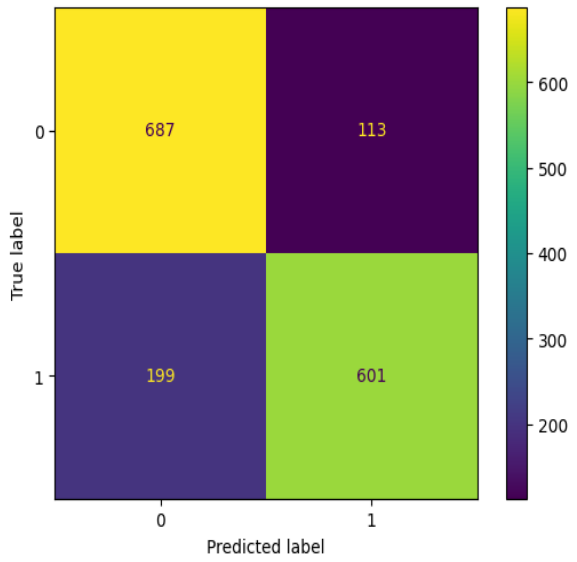


Japanese

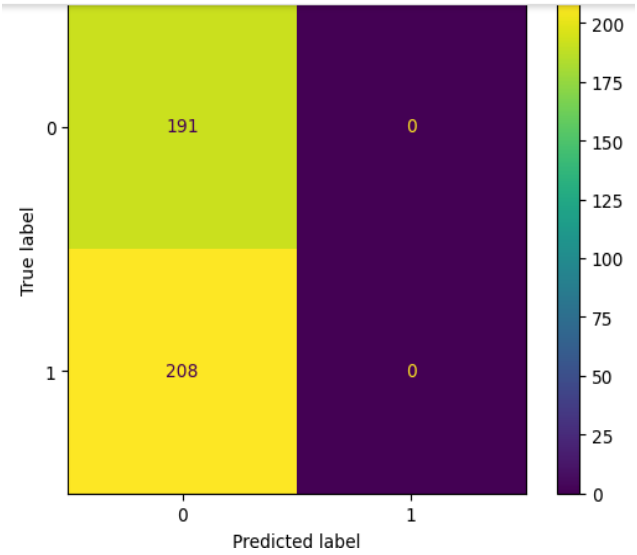


GPTZero

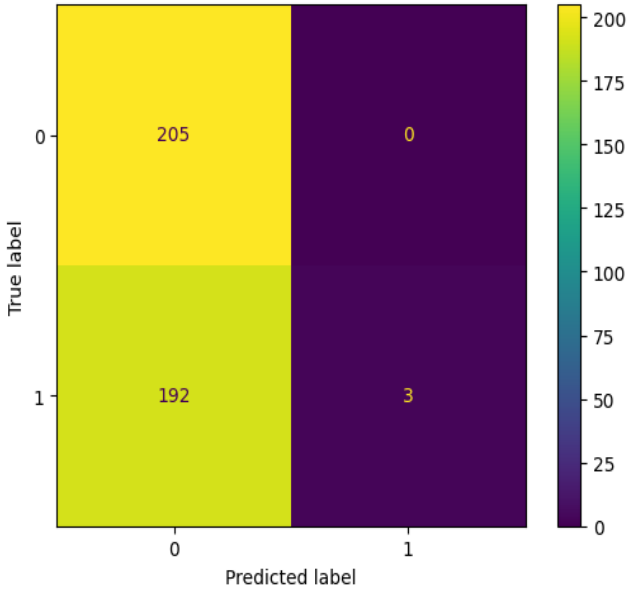
English



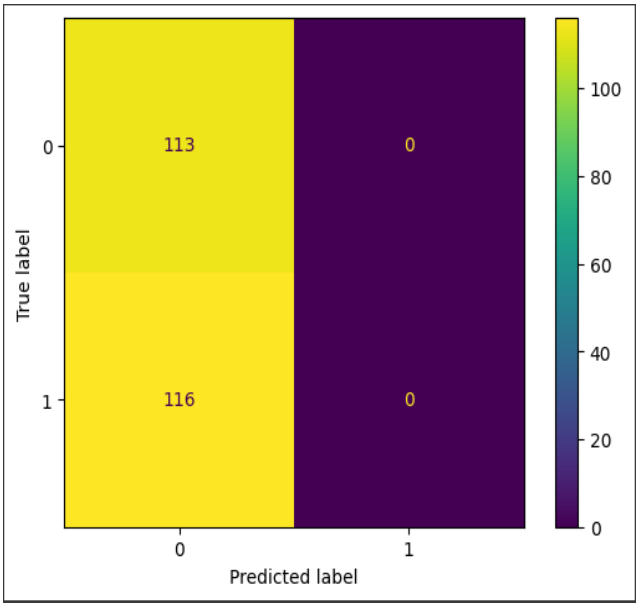
Hindi

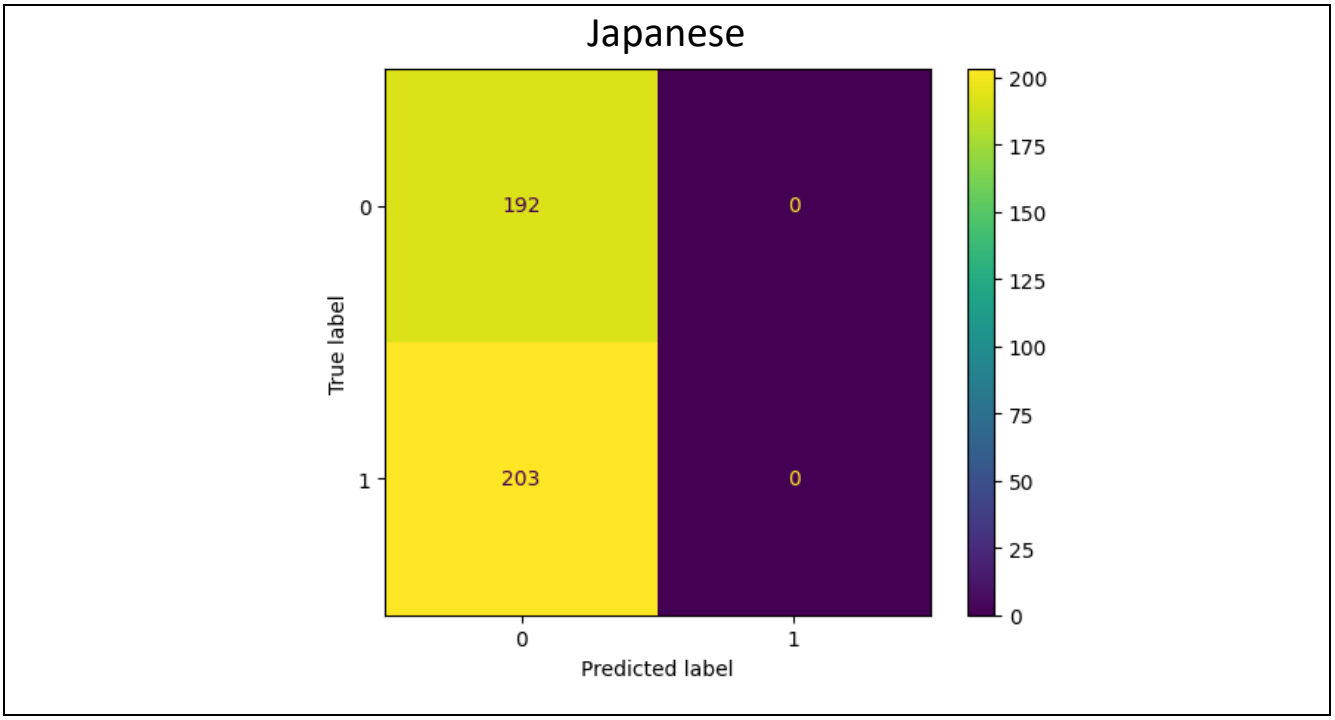


German



German-revised




Japanese
192
0
203
0
True label
Predicted label
0
1
0
1
200
175
150
125
100
75
50
25
0

Appendix 2

While running related programs on the repository, please make sure all of programs are running on Google Colab IDE (Integrated Development Environment). It is because it becomes convenient to upload dataset directly on colab and give their paths into the code as the source dataset.