Explainable Detection of AI-Generated Product Reviews for Moderation Support



Osman, Sufyan

This 25/26: SCC.300: Third Year Project is submitted for the degree of
BSc Computer Science

24 April 2026

School of Computing and Communications

I certify that the material contained in this dissertation is my own work and does not contain unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation. Regarding the electronically submitted work, I consent to this being stored electronically and copied for assessment purposes, including the school's use of plagiarism detection systems in order to check the integrity of assessed work. I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Sufyan Osman 24 April 2026

Abstract

Online platforms are being challenged by the growing trend of artificial intelligence (AI)-generated reviews, which can mislead consumers and reduce customer confidence online. Modern natural language processing models were shown to accurately identify whether text is artificially produced; however, most of these systems operate as opaque “black boxes”. As such, this study investigates whether a review detection system can be incorporated into more transparent workflows for moderation purposes.

Three models were developed for comparison: a baseline using TF-IDF combined with logistic regression, and two transformer-based classifiers, DistilBERT and RoBERTa. Both the baseline and transformer models were tested on a balanced dataset of 40,526 product reviews containing both human-generated content and GPT-2-generated examples. To provide greater insight into the models’ decision-making processes, LIME (Local Interpretable Model-agnostic Explanations) was used alongside the transformer models. The model output was then translated into simple English through the development of a triage-style reporting process.

Both transformer models performed better than the baseline model, with DistilBERT achieving the best trade-off between high accuracy and low false positives on human-written reviews. In addition, further analysis showed that DistilBERT was also more consistent when reviewing disagreement cases and was not simply relying on review length as an easier alternative signal. However, the triage results also showed that high-confidence false positives still occurred in several cases, indicating that the system cannot be used independently of human moderators. In conclusion, although the results suggest that explainability can make even highly accurate classifiers much easier to understand and therefore more useful for moderation support, they also clearly show that human judgement remains necessary, and that caution is needed when applying these findings to other domains.

Table of Contents

1 Introduction *6*

1.1 Natural Language Processing's Role in Text Classification **6**

1.2 The NLP Limitations and Explainability **6**

1.3 Aims and Objectives **7**

1.4 Report Structure **7**

2 Background *8*

2.1 Challenges Detecting AI-text **8**

2.2 Methods to Detect AI-Text **8**

2.3 Using Explainable AI for moderation **9**

3 Methodology *11*

3.1 Dataset **11**

3.1.1 Dataset Selected and Limitation 11

3.1.2 Exploratory Analysis 11

3.1.3 Data splitting Strategy 12

3.2 Model Selection **12**

3.2.1 Baseline Model: TF-IDF + Logistic Regression 13

3.2.2 Transformer Model 1: DistilBERT 13

3.2.3 Transformer Model 2: RoBERTa 13

3.3 Text Representation and Tokenisation **13**

3.4 Training Configuration **13**

3.5 Evaluation Methodology **14**

3.6 LIME **14**

3.7 Loss Function **15**

3.8 Optimiser **15**

4 Development *16*

4.1 Technology Stack **16**

4.2 End-to-end system flow **16**

4.3 Dataset and Preprocessing **17**

4.4 Baseline Pipeline **17**

4.5 Transformer Pipeline **17**

4.6 Inference and supporting analysis scripts **18**

4.7 Explainability layer and Moderator-facing outputs **19**

4.8 Reproducibility and Practical Implementation **20**

4.9 Chapter summary **20**

5 Testing *21*
5.1 Testing strategy **21**
5.2 Test data characteristics and truncation analysis **21**
5.3 Testing Setup **21**
5.4 Supplementary Diagnostic Testing **22**
5.5 Explainability Testing: **22**
6 Results & Evaluation *24*
6.1 Overall Performance Comparison **24**
6.2 Error Analysis and Model Behaviour **26**
6.3 Explainability and moderation usefulness **28**
6.4 Summary **29**
7 Discussion & Conclusion *30*
7.1 Discussion **30**
7.2 Limitations and future work **31**
7.3 Conclusion **31**
References *32*
Appendix *35*
Project Proposal **35**
Ethics Approval for Dataset **40**

1 Introduction

Trust in e-commerce is upheld by consumer feedback, as consumers rely on honest reviews to make decisions. Research confirms that consumer-written online reviews strongly sway purchase intentions. For instance, a 2024 report discovered that although bad reviews reduce the likelihood of purchase, good ratings greatly enhance consumer trust and purchase likelihood (Ahn & Park, 2024).

However, the rise in popularity of Large Language Models (LLMs) has enabled the mass production of synthetic product reviews, which threaten the credibility of marketing and e-commerce by flooding platforms with misleading reviews (Meng, et al., 2025). It has been found that review fraud no longer requires paid farms as it can now be automated. This means the creation of fake reviews through artificial intelligence and bots is also possible, and such AI-generated reviews can be made more cheaply than hiring humans, thanks to advances in natural language processing (Özaydın, 2025). This was also seen in a 2023 report by the Australian Government which notes that LLMs "may also be used to increase the volume and sophistication of fake reviews online". In practice, this could produce vast numbers of realistic looking reviews that are harder to distinguish from genuine customer feedback (Digital Platform Regulators Forum, 2023).

1.1 Natural Language Processing's Role in Text Classification

The current approach is to use natural language processing (NLP) to classify the text; this is done by using supervised machine learning (ML) which trains a model using a dataset with the intention of finding patterns within it. Currently, transformer models are the main approach as context is used in decision making which leads to better performance (Mir, et al., 2023).

1.2 The NLP Limitations and Explainability

While NLP classifiers can accurately assign a probability score to a review, they fail to provide a justification for their decision. In moderation environments, this lack of transparency prevents human moderators from trusting the system's output (Moradi & Samwald, 2021). This is a limitation as it becomes difficult to trust the decisions of high-performance models when its inner decision-making logic remains hidden. Given this, if a system flags a verified customer review as AI without explanation, the moderator risks alienating a legitimate user. For instance, one study demonstrated that giving users transparent criteria and an explanation for moderation decisions helped build proper user trust in AI models and even reduced the spread of misinformation (Palmeira Ferraz, et al., 2024). Therefore, there is a clear need for systems that provide evidence alongside their predictions. Simpler and more interpretable approaches also exist, but they often do not match the performance of modern transformer-based models.

This study proposes that Explainable AI (XAI) should be used along with current methods for review moderation, as this will combine the accuracy of deep learning with the interpretability of traditional methods. The approach applied in this study is Local Interpretable Model-agnostic Explanations (LIME). LIME identifies which words were the most influential in the

model's decision by creating a map of feature importance that can be presented to a human user.

1.3 Aims and Objectives

The current detection systems have limitations as discussed. Given this, the aim is to develop a system that prioritises explainability to support moderation teams.

Objectives:

- Establish a baseline model to identify a minimum expected performance for detecting AI-generated text.
- Develop a transformer-based classifier that can accurately separate each class.
- Integrate an explainability layer to dissect the model's decisions and provide insights about each word.
- Create a report that translates model outputs into an actionable guidance document for a moderator.
- Critically evaluate system performance on generalisation and any linguistic issues.

1.4 Report Structure

The overall structure of this report has been organised into seven chapters. Chapter 1 introduces the problem of reviews generated by Artificial Intelligence (AI), identifies the motivation for conducting this research, and states the aims and objectives of the study. Chapter 2 reviews the literature on relevant issues associated with detecting and interpreting AI text and moderating AI-generated content. Chapter 3 details the methodology employed during the study, specifically identifying the dataset, how the model was selected, and the training set-up and evaluation strategy. Chapter 4 describes how the system was developed, detailing the implementation of the classifiers, explainability layer, and moderator-facing tools. Chapter 5 identifies the testing process and the procedures used to evaluate model behaviour. Chapter 6 provides an overview of the results of the models, evaluating them based on performance, error behaviour, and moderation usefulness. Finally, Chapter 7 discusses the wider implications of the findings of the study, outlines the limitations of the study, and concludes with suggestions for future research. References are located towards the end of the report, as well as an appendix which contains supplementary material where relevant.

2 Background

This chapter summarises and discusses the key studies that relate to detecting AI-generated reviews and making such systems more transparent in moderation environments. The literature does not point towards a single best approach for this problem. Instead, it highlights a common trade-off: methods that are easier to interpret usually fail to match the performance of modern transformer models, while higher-performing models are harder to justify when they are wrong. Therefore, the present study is both a summary of prior work and an explanation of the competing factors which motivate the design of this study.

2.1 Challenges Detecting AI-text

AI-generated reviews are becoming harder to detect as transformer-based language models are performing with greater fluency and coherence. Transformers are the key architecture behind modern LLMs, which makes generated text harder to distinguish from genuine writing (Vaswani, et al., 2017).

A common finding in research on machine-generated text is that detection is not a problem that is solved once and for all but rather treated as a moving target. Generation methods such as decoding strategies can be altered and optimised to fool humans into thinking it was not computer-generated text. Additionally, detection performance depends on the length of available text as detectors are less accurate on shorter excerpts (Ippolito, et al., 2020). It is also common for the "generator distribution" to change significantly across model families, updates, and how it was prompted which can break detectors trained on older data. This is typically called out-of-distribution (OOD) or domain-shift generalisation failure in AI-text detectors (Borile and Abrate, 2025). Additionally, it was also found that AI-text detectors can degrade using paraphrasing, so if the same review is written in a different style (even if still AI-written), it can change the detector's outcome (Nicks, et al., 2024).

This is relevant as strong benchmark results do not guarantee good "real world" results. Even if an AI-text detector can successfully detect text from a controlled database, it can still fail in detecting a new model which generates text with different prompt strategies. AI-text detection should therefore be understood as both a classification problem and a robustness problem (How accurate is the detector? – What are the quality and types of errors produced by the detector?).

2.2 Methods to Detect AI-Text

Early research shows that people have been faking reviews long before LLMs grew in popularity the problem was discussed under terms such as opinion spam and deceptive reviews. Opinion spam was widespread and came in many different forms, requiring tailored detection techniques rather than generic "spam" methods (Jindal & Liu, 2008). Researchers also created labelled datasets to demonstrate deceptive reviews can be made to sound authentic, with human judges performing at near chance (Ott, et al., 2011). Although these findings predate the growth of LLMs, they remain important because they show that realistic deception is difficult for humans to detect, whether it is written by a person or by AI.

One solution called ‘Stylometric Analysis’ relies upon cues such as sentence length, n-grams, punctuation patterns, etc. (Stamatatos, 2009). Researchers have also explored using syntactic stylometry as an alternative to lexical features (Feng, et al., 2012). The main advantage of stylometry is that it is transparent in explaining the features of text; however, older research suggests that structure alone is not enough to tell truth from deception (Yoo and Gretzel, 2009). Given the growth of LLMs in recent years they can easily mimic human-like surface patterns therefore making classic stylometry less reliable (Ippolito et al., 2020).

A second approach to this is using supervised machine learning with sparse text representations such as TF-IDF. Classical information retrieval research suggests that TF-IDF-style grew in popularity for measuring query–document similarity and became a strong baseline text representation (Salton & Buckley, 1988). While this isn’t just one-off evidence as surveys of text categorisation show the effectiveness and portability across document classification tasks (Sebastiani, 2002).

Another approach is using transformer-based classifiers as they enable contextual token representations allowing the meaning of a word to change based on surrounding words (Vaswani, et al., 2017). This also introduced BERT as it established the transfer learning paradigm where a pre-trained encoder can be fine-tuned to fit a classification task (Devlin, et al., 2019). These have been adopted widely as they achieve incredibly high in-domain performance on their testing and training data. However, studies suggest that transformers can be brittle under OOD conditions as the model can pick up spurious correlations and degrade generalisation (Borile and Abrate, 2025).

This study also sits alongside AI-text detectors such as GPTZero and DetectGPT. However, these tools are mainly designed to detect AI-generated text, whereas this project focuses more specifically on how detection and explanation can be combined in a moderation-oriented workflow.

Overall, the literature seems to consistently suggest that there is a trade-off as methods with higher interpretability fail to match the performance of an in-domain transformer. However, high-performing models may be harder to justify as they generalise poorly on OOD tasks. This motivates the present study, which tests whether a strong transformer-based detector can be paired with an explanation layer to make its outputs more usable in a moderation context.

2.3 Using Explainable AI for moderation

In a moderation environment, detection systems serve as decision-support tools for moderators rather than pure prediction engines. Algorithmic moderation research suggests that at scale, platforms combine automated filtering with human review, and people lack belief in the legitimacy of the platform if decisions are opaque or contestability is limited (Gorwa, et al., 2020). Explanations can influence how the decision is perceived and hence how users react to the enforcement. For example, a study on posts removed from Reddit along with a reason provided found when removals were accompanied by a clear reason, users are more likely to adjust behaviour to reduce odds of removal in the future (Jhaver, et al., 2019). Therefore,

supporting the view that explanations are a big part of moderation to prevent rule breaking on platforms.

Research on automation bias suggests that people can over-trust decision aids leading to complacency and systematic errors when the explanation is wrong (Parasuraman & Manzey, 2010). However, in XAI literature, it was found that explanations improve performance of a team when the model is correct but can also lead to harm when the AI is wrong (Bansal, et al., 2021). Therefore, solidifying that explanations must not be blindly relied upon.

LIME is a widely used post-hoc, model-agnostic local explanation method. It was proposed as a method for explaining individual predictions by learning an interpretable surrogate model in the neighbourhood of a specific instance, using perturbations of the input and feature weighting to approximate local decision behaviour (Ribeiro, et al., 2016). In text classification, LIME highlights the words that most influenced the decision, making the output far easier for non-technical users to interpret. For this reason, it will be used in this study.

LIME was a reasonable choice here for practical reasons. It is not constrained to a specific neural structure (unlike Integrated Gradients), nor is it simply an output of the model (unlike Attention Visualisation); therefore, it was a lightweight and practical method for inspecting classifier decisions within a moderation setting.

Finally, others suggest that an explainable layer on a black box model is not sufficient in high-stake scenarios. Models should be interpretable by design rather than relying on post-hoc explanations that may be misleading (Rudin, 2019). In the case of review moderation, this study directly challenges the methods I plan to use. However, when using review moderation there is not always an easy choice: While simpler models are possibly easier to explain, they may lack the necessary performance needed to accomplish the increasing complexity of AI-generated reviews. Therefore, while I do not treat LIME as a replacement for true interpretability; I use it here as a practical trade-off which will allow the human moderator to better understand what is going on with a stronger (and therefore less interpretable) classifier.

3 Methodology

The aim of this chapter is to explain my design decisions behind each model and the dataset used in this study. I will also include details on the training setup as I aim to build three comparative classifiers: a baseline model and two transformer-based models. In addition, I explain how model performance was evaluated, including error analysis and checks for robustness indicators related to moderation. Finally, I will outline how the interpretability was assessed and how evaluation prioritised risks such as false positives, where genuine users may be incorrectly punished.

3.1 Dataset

3.1.1 Dataset Selected and Limitation

All models trained in this study were trained using the HuggingFace dataset "theArijitDas/Fake-Reviews-Dataset", which contains 40,526 labelled product reviews with each record including category, rating, text and label fields (theArijitDas, n.d.). This is a processed and cleaned version of Kaggle's "Fake Review Dataset", with an additional 94 AI-generated records added by the maintainer (Mexwell, n.d.). The dataset is binary encoded as 0 = human written and 1 = computer generated and is balanced as close to a 50/50 split for each class. In this study only the text and label are used for training and evaluation, and the other fields are retained for context but are not used as model inputs.

Academic articles show this dataset was produced by (Salminen, et al., 2022) and used GPT-2 to create the final "computer-generated" class. In the data creating process, human reviews were drawn from the publicly available Amazon Review Data dataset (Ni, 2018). The authors generated 2,000 GPT-2 reviews per product category, producing 20,000 AI-generated reviews in total which was paired with 20,000 human-written reviews to form a balanced dataset of 40,000 reviews where it was finally released publicly via OSF (Salminen, et al., 2022).

Although theArijitDas/Fake-Reviews-Dataset is larger and class balanced the computer-generated class is produced using GPT-2, so any models I train using this dataset will learn artefacts specific to GPT-2. This may cause it to fail at generalising to newer LLMs or different prompting styles. The dataset also only contains reviews from Amazon in the human-written class which may not cover the diversity of real moderation environments. However, the dataset remains well suited to the current study as it provides a large balanced dataset that can be used as a benchmark for comparing baseline and transformer classifiers under the same conditions. This is important as it ensures that analyses of error patterns, false-positives and explanation quality can be done under a controlled setting, so findings are interpreted as evidence about the model behaviour. Regardless while evaluating the models, I will continue to factor in the dataset's limitations.

3.1.2 Exploratory Analysis

Prior to training any models, I performed an exploratory analysis to check whether class balance and review length might affect the results. This was important as classifiers tend to

learn shortcuts, such as associating one class with shorter or longer text, rather than learning more meaningful language patterns.

The study found that there was an average difference in length between the two types of review (AI-labelled; mean = 305 characters) and those written by humans (mean = 397 characters). Most of the reviewed texts were brief, as evidenced by the median number of tokens being 48. Thus, if the maximum token count is set at 128, nearly all reviewed samples will contain their entire text. However, truncation did not affect both classes equally: 19.88% of reviews exceeded 128 tokens, with truncation occurring more often in human-written reviews than AI-generated ones. This introduced a possible risk that the models could be influenced by review length or lose slightly more information from the human class. In the evaluation, a length bias analysis is included for this reason.

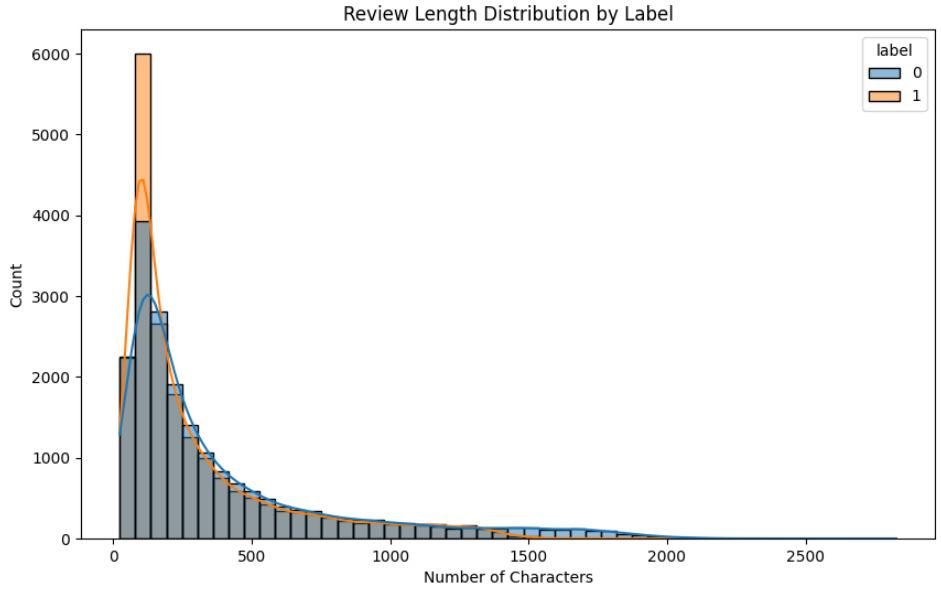


Figure 1 Review Distribution by Label. Human Review (label 0) has a wider spread and higher mean compared to AI-generated reviews (label 1).

3.1.3 Data splitting Strategy

I employed a standard hold-out split for evaluation with 70% training / 10% validation / 20% testing, using a fixed random seed (42) so it can be reproduced. The transformer split yielded 28,368 training reviews, 4,053 validation reviews, and 8,105 test reviews. The validation set was only used in checkpoint selection during fine-tuning.

3.2 Model Selection

This study frames AI-generated review detection as a binary text classification task. Given a review predict if it is human written (0) or AI-generated (1). I developed three models that increase in complexity allowing me to examine the trade-off between accuracy and explainability.

3.2.1 Baseline Model: TF-IDF + Logistic Regression

Although stylometric approaches are relevant in the literature. The baseline used a TF-IDF paired with Logistic Regression. The TF-IDF uses an English stop-word removal and a fixed dimensionality of 5,000 features to prevent overfitting and speed up training.

TF-IDF + Logistic Regression was chosen as it is computationally efficient and widely used as a reference in text classification. It may also be partially interpretable as the learned coefficients can be inspected to identify the most discriminative words in each class.

3.2.2 Transformer Model 1: DistilBERT

DistilBERT is a transformer-based language model that was derived from BERT via knowledge distillation, where a "student" model is trained to reproduce behaviour of a larger "teacher" model. Therefore, DistilBERT retains a lot of BERT's contextual understanding while doing it with fewer parameters, making it more efficient. DistilBERT was chosen because it offers a useful balance between contextual modelling performance and computational efficiency, making it a realistic candidate for moderation-oriented settings.

In this project DistilBERT was fine-tuned for binary sequence classification and was chosen due to its efficiency making it a realistic candidate for moderation environments.

3.2.3 Transformer Model 2: RoBERTa

RoBERTa is closely related to BERT but trained under a more robust pre-training regime which has been shown to improve downstream performance across many NLP tasks. In this case, I fine-tuned RoBERTa-base using binary sequence classification setup. RoBERTa serves as a great comparator due to its increased complexity. This will allow me to assess if an increase in model capacity will affect performance or error patterns on the same data.

3.3 Text Representation and Tokenisation

For the baseline, I used TF-IDF with a fixed maximum vocabulary size of 5,000 features, as this improves training speeds and efficiency.

For transformer models, I used fixed-length tokenisation with a maximum input length of 128 tokens. This was selected as the median length is only 198 characters (40-50 tokens), so most reviews will fit within this limit. Therefore, keeping the time complexity cost low enough to train on most consumer hardware, whilst being able to utilise most records in this dataset. However, when interpreting model performance, it should be noted that predictions are based on the first 128 tokens of each review.

3.4 Training Configuration

The transformer models were fine-tuned for binary classification (cross-entropy over two classes), with weight decay to help reduce overfitting. Training was run for a small number of epochs (3) with conservative learning rates (around 1e-5 to 2e-5) to retain the benefits of pretraining. During fine-tuning, model performance was monitored on the validation split after each epoch. The checkpoint with the highest validation accuracy was then selected as the final model.

Rather than focusing on the optimiser internals, I prioritised a stable and consistent fine-tuning setup across both models that fits the available compute. Complete hyperparameters and execution details are provided in the Development chapter/appendix.

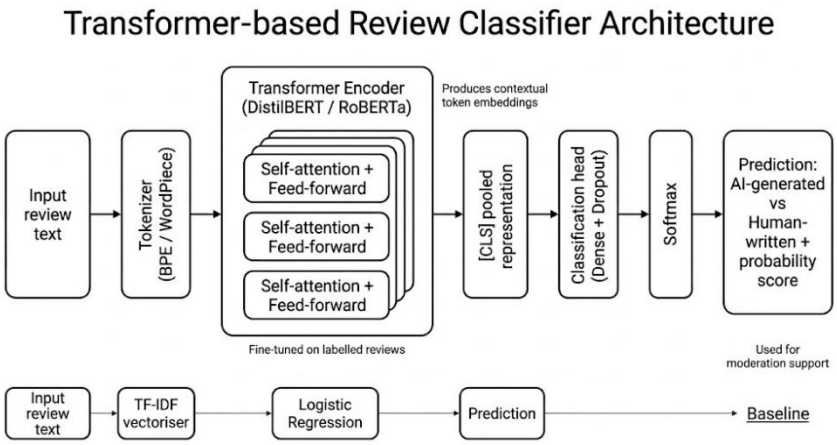


Figure 2 Review Classifier Architecture

3.5 Evaluation Methodology

False positives in moderation are seen as more serious because they directly punish legitimate people. Therefore, I do not solely look at accuracy when evaluating the models. I also report: F1-score, per-class precision/recall/F1, macro-averaged precision/recall/F1, weighted-average precision/recall/F1 and visualisations such as confusion matrices.

Aside from metrics, I also created targeted analyses:

1. Length Bias Analysis - I tested if the model is using review length to classify reviews rather than the linguistic patterns.
2. Model Disagreement Analysis - I identified each sample in the full 8,105 test set, where the two models made different predictions, and recorded which model was correct.
3. LIME Disagreement – Takes a specific disagreement case and runs LIME on both models to show the words each model focuses on when making opposing predictions and outputs as an HTML file for viewing.
4. Top Discriminative words – extracts the top 10 highest-weighted words for each class from the logistic regression coefficients and reveals what signals the baseline relied upon when making its decision.

3.6 LIME

To make predictions usable to non-technical moderators, I integrated LIME to provide local, post-hoc explanations. For each review, LIME can highlight the tokens that influenced the model in making its decision. The goal is to improve decision making; therefore, an explanation

should provide evidence to help a reviewer understand why a review was flagged and support qualitative inspection of uncertain or incorrect cases.

Explanations are therefore used in two ways: (i) error analysis and inspection of ambiguous cases, and (ii) generation of moderator-facing summaries in later stages of the system. The following section 4 will provide details of the development of this methodology.

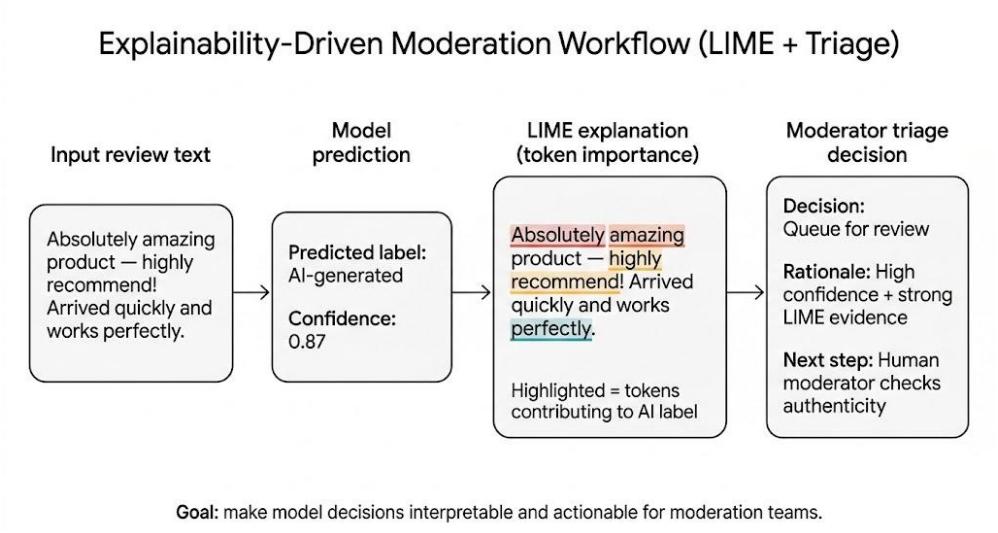


Figure 3 Proposed Moderation Workflow

3.7 Loss Function

The transformer models were trained using cross-entropy loss, which is standard for supervised classification tasks. This was appropriate here because the task involves predicting one of two classes: human-written or AI-generated.

3.8 Optimiser

DistilBERT and RoBERTa were fine-tuned using AdamW, a widely used optimiser for transformer-based language models due to its stability and effectiveness during fine-tuning (Loshchilov and Hutter, 2019).

4 Development

The aim of this chapter is to focus on the engineering of the end-to-end pipeline (including the baseline classifier, the transformer-based models, the explainability layer and the moderator-facing tools). The system was built as an operational prototype, capable of classifying product reviews as either human-written or AI-generated and provided explanations and a triage output.

4.1 Technology Stack

The system was implemented in Python 3, with PyTorch used as the main machine learning framework. The Hugging Face Transformers and Datasets libraries were used to load trained models, tokenizers and the review dataset. The baseline model used scikit-learn, with the dataset being handled using pandas and NumPy, and visualisations were produced by matplotlib and seaborn. Post-hoc local explanations of how the model made its decisions were generated using LIME. The entire development process was conducted within a standard Python virtual environment.

For portability of the project across all possible hardware configurations, I provided the transformer script with the ability to check if CUDA or CPU execution is available on the machine. Based on this, it adjusts the machine that it runs on and the batch-size. I can then use smaller batch-sizes to iteratively develop my work locally on my personal laptop while also developing full transformer training on Lancaster University's HEX Cluster using an NVIDIA GPU and larger batch-sizes. This reduces the amount of time needed to manually configure the environment so I can easily reuse the codebase.

4.2 End-to-end system flow

The system is an end-to-end pipeline to detect whether customer reviews are generated using AI. The process begins by loading a labelled review dataset from HuggingFace. Each model independently preprocesses and tokenises the text, branching the system into separate classification paths as follows: a TF-IDF plus logistic regression baseline and two fine-tuned transformers (DistilBERT and RoBERTa) each operating separately. Following training of the models, the two transformer models are used as input into scripts that perform evaluation to create confusion matrices, identify disagreements between DistilBERT and RoBERTa, and analyse length-based biases in the predictions made by DistilBERT. The baseline is independent, reporting its own accuracy and identifying the most important words for each class. Lastly, an explainability and reporting layer surrounding the DistilBERT model will use LIME to produce HTML formatted explanations at the word level, and plain-English style summaries of each individual review along with a triage report categorising reviews based on their level of risk. An additional script will apply LIME to both transformer models to evaluate the explanations produced by each model on disagreement cases, producing separate HTML explanations for DistilBERT and RoBERTa.

The project was distributed among several different files (scripts) that were all separate from each other and were used for performing data analysis, model training, evaluating performance, generating explanations of predictions, and generating reports. This development approach had

the benefit of creating a modular pipeline which is much easier to test, run in isolation, or evaluate against others.

4.3 Dataset and Preprocessing

All models used the ArijitDas/Fake-Reviews-Dataset (loaded directly from HuggingFace) as it provides a single data source for review text and labels for all classifiers. This dataset has several features, however, only 'review text' and 'labels' are used to train the models. To prepare the raw data for processing, the data is first shuffled with a fixed random seed (42) and then split into training, validation, and test sets. For the transformer models this resulted in 28,368 training examples, 4,053 validation examples, and 8,105 test examples. There are two preprocessing methods available depending on the type of model being used. For the baseline method, review text is first converted into sparse TF-IDF vectors and then any words classified as stop words by the English language dictionary are removed. Each review text will have a limited number of unique terms (vocabulary cap = 5000). The review text for the transformer models was first tokenised, using the appropriate pre-trained tokenizer for the selected model. Tokenisation included padding and truncation of review text. For both DistilBERT and RoBERTa, I used a maximum input length of 128 tokens. My primary concern when deciding how long to allow the input to be was memory usage. As attention complexity increases quadratically with sequence length, it was necessary to keep input lengths as low as possible to enable me to process large volumes of reviews.

4.4 Baseline Pipeline

The Baseline Model was implemented in BaselineClassifier.py, which used the scikit-learn package. Using `TfidfVectorizer(stop_words='english', max_features=5000)`, I converted all the reviews into sparse numerical representations. Then, I trained a logistic regression classifier on the resultant feature matrix. This created a simple and robust reference point, which would be both quick to train and to understand. Unlike the transformers, the baseline did not require a GPU and could be trained easily on typical hardware.

In addition, I made a second script called Top10Words.py to analyse the learned Logistic Regression coefficients, and to obtain the top N discriminative words for each class. I found it to be helpful to compare against the black box nature of the transformer models since, unlike the transformer models, in the baseline each feature's weight was directly inspectable. This will allow me to evaluate if the baseline is picking up on reasonable lexical signals or if it is looking at random features. Additionally, this provides a simple way for me to interpret the results without having to use another library to explain.

4.5 Transformer Pipeline

The two transformer-based classifiers were implemented using HuggingFace Transformers and PyTorch. The first model was `distilbert-base-uncased`, and the second was `roberta-base`. Both were loaded using `AutoModelForSequenceClassification` with `num_labels=2`, allowing them to operate as binary classifiers over the human-written and AI-generated review classes.

For both models, tokenization was handled using the corresponding pretrained tokenizer with `padding="max_length", truncation=True, and max_length=128`. DistilBERT therefore used WordPiece tokenisation and RoBERTa used byte-pair encoding. Fine-tuning was performed by using the Hugging Face Trainer API. Both models were trained for a total of three epochs with a weight decay of 0.01, as well as saving the model's state at the completion of every epoch. Additionally, best model loading was also enabled upon completion of the training process. The learning rate was established at 2e-5 for DistilBERT and 1e-5 for RoBERTa. Accuracy was used as the metric for selecting the best checkpoint. A validation set comprising 10% of the data (4,053 examples) was used as the `eval_dataset` during training, allowing the Trainer to monitor performance and select the best checkpoint after each epoch without exposing the test set. The test set was reserved solely for final evaluation.

The results of the training for DistilBERT and RoBERTa are stored in `distilbert_results/` and `roberta_results/` respectively. Each checkpoint contains model weights, optimiser state, scheduler state, RNG state, training arguments, and configuration files. Classification reports were also saved as a JSON file so that I could use them again during my report for final evaluation metrics. This was a good way to separate the heavy training phase from subsequent analysis (so I wouldn't have to retrain all models) and allow me to review model outputs without retraining all the models from the start.

4.6 Inference and supporting analysis scripts

A consistent inference procedure is applied across all post-training evaluation scripts. Inference is as follows:

(1) Load in the saved model and tokenizer

(2) Tokenize the input review, adding truncation and padding to a fixed maximum length.

(3) Perform one forward pass of the model while disabling gradient tracking.

(4) The predicted class is then extracted by applying argmax over the output logits.

(5) SoftMax is applied to the logits to obtain the class probability distribution and extract the probability value, if a confidence score is required.

Writing the inference scripts in this way made it easy to compare how models are performing and eliminated potential inconsistencies that may have existed when comparing model performance during training versus evaluation.

I also developed several tools to analyse model performance in more detail. For example, cm.py was used to compare side-by-side confusion matrices for the transformer models, while compare.py identified cases where DistilBERT and RoBERTa made different predictions on the same test example and stored them in model_disagreements.csv. The purpose of this analysis was to identify cases where a model may have relied on a shortcut, such as review length, and to further examine how the shortcut used affected accuracy. These tools moved the

project beyond reporting average accuracy alone and instead provided insight into behaviours that are important in a moderation setting, such as boundary cases and shortcut learning.

Model disagreements were treated as useful diagnostic data. When DistilBERT and RoBERTa produced different predictions for the same review, these cases were treated as potentially ambiguous or difficult examples for further inspection. This provided a good basis for qualitative analysis than only relying upon summary metrics which helps identify cases where model confidence should be interpreted more cautiously and where human oversight may be warranted.

4.7 Explainability layer and Moderator-facing outputs

One of the main parts of developing the product was building an Explainability Layer with LIME. I applied this layer in three different ways. First, DistilBERTLIME.py generates an individual HTML file which explains a model's decision by showing which words in a review had the biggest influence on the model. Secondly, ContradictionLIME.py generates two separate LIME explanations for the same disagreement case, one for DistilBERT and one for RoBERTa, providing me with a basis for qualitative comparison. Finally, ModeratorReport.py uses the LIME output along with the classifier prediction to create a moderator-style report in natural language.

From an implementation standpoint, LIME is based on perturbing the input text to approximate the model's local decision boundary. A prediction wrapper is then used to obtain the model's predictions for those perturbed samples. Next, a local linear surrogate model is fitted to determine which words in the input text are most influential in the prediction. In ModeratorReport.py, I used 500 perturbation samples and took the top five most influential words identified by LIME. The top five most relevant to the predicted class were then included in the moderator report. When run as a standalone program, LIME uses 5,000 perturbation samples. For the deep-dive disagreement cases, I used six features so that slightly more information could be shown when comparing the two models' side by side. The results are saved as HTML files, allowing the user to view and navigate them outside of the command line.

Explanations give valuable insight into how each prediction was made, but I also built moderator-facing reports. The plain-English summaries created in ModeratorReport.py include the predicted class, confidence level, explanation, and recommendation. TriageReport.py complements this by applying the classifier in batch across 100 test samples, assigning each review a risk level based on prediction confidence, and saving the results to a CSV file. If the AI-predicted class has a confidence score of 0.7 or greater, the review is labelled as high risk. If the human-written class has a confidence score of 0.7 or greater, the review is labelled as low risk. Reviews with any confidence score below 0.7 are labelled as ambiguous and flagged for human review instead of being processed automatically. This creates a clear human-in-the-loop process for dealing with ambiguous cases. From a practical standpoint, this allows the project to operate more as a moderation support system rather than a fully automated classification system.

4.8 Reproducibility and Practical Implementation

To support reproducibility, consistent random seeds were applied throughout the data splitting process. `Train_Test_Split()` was called with `random_state = 42` for all baseline models, while my transformer scripts used a consistent (shuffled) index range created using seed = 42. As such, a checkpoint was written out after each epoch, and the best model was automatically selected upon completion of training. As a result, I was able to run the development process multiple times under identical conditions, with all model outputs preserved.

During this project there were also many practical trade-offs made. My choice of maximum input token length (128) and limited number of training epochs were both made based on keeping the cost of training as low as possible on the available hardware. I chose to implement dynamic batch-size selection, which allowed for local experimentation on a personal laptop while large-scale experiments could utilise the GPU hardware on the HEX cluster. While DistilBERT was significantly smaller and lighter weight than RoBERTa, it was much easier to fine-tune repeatedly during development. Therefore, RoBERTa represented a more powerful, but computationally expensive alternative model for comparison. Additionally, the LIME predictor wrappers in ContradictionLIME.py did not explicitly enforce the 128-token limit used during training, meaning LIME perturbations were processed at the tokenizer's default maximum length.

Upon reviewing, there were some limitations in the implementation. Specifically, while both the baseline and transformer scripts utilised the same 20% test partitioning size, they did not share the exact same data splitting procedure. This means the actual test examples differ between the baseline (80/20 split) and the transformer evaluations (70/10/20 split), so direct comparisons between the two should be treated with caution. Another limitation of this implementation is that no explicit random seed was specified for the LIME explainer. Therefore, its explanations will have been slightly non-deterministic. These issues do not prevent the successful implementation of the models; however, it requires caution when making comparisons and viewing this as a prototype rather than a production-ready moderation tool.

4.9 Chapter summary

In conclusion, the development process produced a complete review detection prototype system for AI-generated reviews. The development of the system was completed as a modular Python-based pipeline that contained three different classifiers. Alongside the classifiers, there were numerous analytical scripts, a LIME-based explainability layer, and reporting output for moderators. This structure allowed the system to do more than just make predictions as the system also produced interpretative results, such as "the most influential words" "triage classifications," and "natural language guidance." These design choices were the foundation of the testing and evaluation that is covered in the following chapter.

5 Testing

In this chapter, I will describe my methods for evaluating the models built and the metrics used to assess their performance. While the primary goal of testing is to evaluate each model's prediction accuracy, it also aims to determine if they are behaving in a manner consistent with that of an actual moderation environment.

To achieve these goals, I evaluated the models at three different levels: using formal statistical techniques to analyse their performance on data withheld during training, conducting additional diagnostics, including analysing model disagreement and assessing length biases, and performing applied testing of the models' ability to provide explanations for their predictions and to triage.

5.1 Testing strategy

The primary method for testing within the scope of this project was empirical and not software engineering based. I did not develop an automated unit-test suite nor a continuous-integration pipeline. Correctness was determined through manual execution of scripts, review of saved files, reports and visualisations.

While both were run using the same dataset and used the same 20% test partition size, the transformer model was run using a fixed index-based slice (after shuffling via HuggingFace using seed = 42) whereas the baseline was run using sklearn's train_test_split() function (with a specified random state of 42). Thus, the actual test sets that each method produced will be slightly different because they use two different types of random number generators.

5.2 Test data characteristics and truncation analysis

The 128-token limit was applied to both transformers. Therefore, I calculated how frequently reviews would be truncated at this point. Approximately 19.88 percent of all reviews were longer than 128 tokens. Truncations occurred in human-written reviews (i.e., 22.43%) more frequently than in AI-generated reviews (i.e., 17.33%). When evaluating results, it is useful to note that some loss of information is slightly more likely to occur from a review authored by a human due to the greater frequency of truncation. Nonetheless, as approximately 80% of the sample size did not exceed the 128-token limit, the chosen sequence length effectively included most samples.

5.3 Testing Setup

Baseline was tested through TF-IDF with the English stop-words removal and the vocabulary capped at 5,000 features, followed by logistic regression using the default parameters from scikit-learn. Baseline's primary measure of performance reported was accuracy.

Both transformers were run on the 8,105-example split of held out data. DistilBERT was trained with a learning rate of 2e-5 for 3 epochs with a batch size of 8.

RoBERTa was trained with a learning rate of 1e-5 for 3 epochs with a batch size of 16. For all training runs that used the transformer models, the best checkpoint based on validation set accuracy was selected.

When evaluating the model, I reported accuracy along with each class's precision, recall, and F1 score. These metrics are particularly relevant when evaluating a moderation system as false positives can be detrimental.

5.4 Supplementary Diagnostic Testing

As well as the primary report classifications, I performed some smaller diagnostics.

Confusion Matrix Script (cm.py): creates side by side confusion matrices for DistilBERT and RoBERTa. The confusion matrix script is best viewed as an internal tool to help understand how the models performed after training. It uses 500 examples from the testing data. This helps evaluate the models as confusion matrices identify the true/false positives and negatives.

Disagreement Analysis Script (compare.py): runs a comparison on the test set. This was done to determine all instances in which both DistilBERT and RoBERTa gave differing results for the same input. These comparisons were saved to model_disagreements.csv for further examination.

Length Analysis (length_analysis.py): testing for potential "shortcuts" was carried out using length_analysis.py. This analysed distribution of review lengths across True Positives, False Positives, True Negatives, and False Negatives. Similar to the Confusion Matrix Script, this only uses samples from the testing data and will help determine if any correlations are made between review length and the classification given.

Moderator-Style Triage Workflow Test: tests a Moderator-style triage workflow using TriageReport.py. This categorised 100 samples into one of three categories: High Risk, Low Risk, or Ambiguous, based upon confidence. A confidence value of 0.70 was chosen as the threshold. This was used to assess if the output of the classifier could be easily translated into a simple human-in-the-loop moderation assistant.

5.5 Explainability Testing:

As explainability is a central part of this project scripts were built to test the LIME integration directly.

To generate an explanation for any single review, a standalone Python script DistilBERTLIME.py was built. It uses num_features = 5 and LIME's default parameters of 5,000 perturbations, and a default kernel bandwidth of 25, it returns an HTML file to show how much weight each word carried in predicting the classification result for an input in DistilBERT.

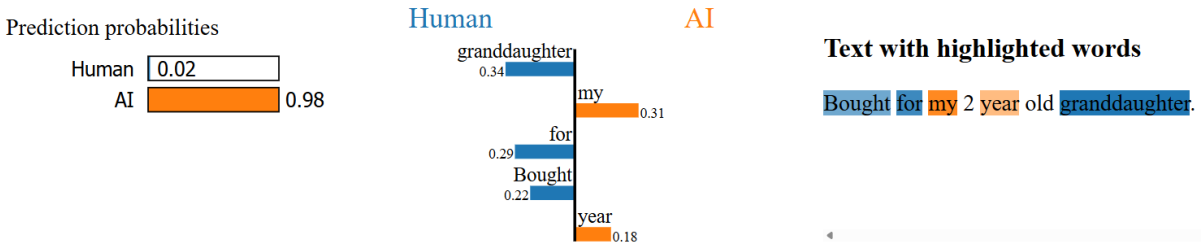


Figure 4 DistilBERTLIME.py HTML Output Example

Another script was built to test LIME on both models to further understand why disagreements occurred between DistilBERT and RoBERTa. The objective was to see what features each model emphasised when they came up with different classifications for the same review. This returns two HTML files which allows you to compare why the two models classified the same review differently.

Lastly, the ModeratorReport.py module integrated the DistilBERT model's predictions with a LIME explanation to create a moderator-style report. The resulting explanation showed the top five contributing features to the predicted category inside a simple English explanation/recommendation.

6 Results & Evaluation

6.1 Overall Performance Comparison

The strongest overall result of this study was that in-distribution the task can be solved well; however, the type of model used does matter since the best model will depend upon how you define what constitutes the "worst" or "least desirable" mistake. Comparatively speaking, the headline numbers showed that both types of transformers were significantly better than the baseline (TF-IDF + LR). Specifically, the baseline achieved an accuracy level of 86.44% and a macro F1 of .86 on its respective test set. By comparison, the DistilBERT model achieved 97.91%, while the RoBERTa model achieved 96.87%. This difference is sufficient to demonstrate that the contextual transformer representations are not simply improving marginally over the baseline; rather, they are representing information that the sparse baseline could not. However, the baseline remains a valuable tool as it demonstrates that the data is not completely trivial and that a simple lexical weighting method would get a high accuracy. This suggests that there are significant stylistic cues in the data rather than merely minor semantic distinctions.

Most of the baseline's highest weighted words were consistent with a particular interpretation of the data. For example, "problem," "reason," "couple" and "materials" were all strong indicators for the AI class. On the other hand, the presence of "actually," "instead," "maybe," and "quite" pushed the predictions toward the Human class. It is worth noting that the baseline appears to have picked up differences in register rather than simply identifying one group appears random or artificial. The baseline can distinguish between two types of registers, specifically: A more general type (i.e., the AI) which relies upon evaluation and less narrative, and another type (the Human) that contains a higher frequency of hedges, narrative phrases, and self-correction. However, the baseline works because the dataset appears to contain recurring verbal habits, but it cannot model those habits in context. That helps explain why the transformers move so far ahead.

The main result from these comparisons is not DistilBERT scoring better on accuracy, rather it is showing which model performed better at moderating. Good moderation performance is clearly demonstrated as DistilBERT has an overall Human Recall of 0.9653 compared to 0.9386 for RoBERTa. While the results of both moderation models show a difference in Human Recall, that difference may have been less critical if this were a typical benchmark test or research paper. However, the actual number of human reviews that fall under AI reviews due to a reduction in Human Recall can be very large; therefore, I will consider the difference in Human Recall to be significantly more important than other tests.

As a result, when looking at the error profiles based upon those numbers, the results become very clear. The Baseline incorrectly identifies approximately 13% of all human generated reviews as AI-generated, whereas RoBERTa incorrectly identifies approximately 6.1% of all human generated reviews as AI-generated. Similarly, DistilBERT incorrectly identifies approximately 3.5% of all human generated reviews as AI-generated. Therefore, even though RoBERTa produces impressive numbers such as a 99.84% Human Precision and a 99.85% AI

Recall (which are quite high), the overall balance of the moderation system that is most important is the ability to detect AI while minimising the penalty on true humans.

Model	**Accuracy**	**Macro F1**	**Human Precision**	**Human Recall**	**AI Precision**	**AI Recall**
Baseline	86.44%	0.86	0.86	0.87	0.87	0.85
DistilBERT	97.91%	0.9791	0.9926	0.9653	0.9665	0.9929
RoBERTa	96.87%	0.9686	0.9984	0.9386	0.9424	0.9985

Table 1 Performance Overview

6.2 Error Analysis and Model Behaviour

The confusion matrix and disagreement analysis provide a very clear comparison to determine how significantly different the two transformer models are. Again, on the 500-sample confusion matrix, DistilBERT performed best at an accuracy of 97.60% compared to 96.60% for RoBERTa. Additionally, on the full test dataset of 8,105 reviews, there were only 244 disagreements between the two models. This represents only 3.01 percent of the total test data. The most valuable insight in this regard is contained within that 3%. In the 244 disagreements, DistilBERT was correct in 194 instances while RoBERTa was only correct in 50. Therefore, when faced with truly ambiguous decisions, DistilBERT makes accurate predictions approximately four times more than RoBERTa. This increases the significance of the headline one point difference; it's not merely a better overall model but also a model that performs well under conditions of ambiguity.

The differences in the models can be seen through their disagreements. The most common pattern is that: actual Human, DistilBERT flags Human, RoBERTa flags AI. It has been observed that RoBERTa has been more sceptical about short, easy-to-write and conversational reviews such as *"love the fit. needed for work. this pair is my back up pair"*. That is important to note since it gives a clue on what kind of a moderation risk there will be as colloquial human written reviews are going to have a higher chance to get flagged. On the other hand, while DistilBERT is not perfect either it fails to flag AI written reviews when RoBERTa identifies them – (i.e., *"GREAT LIGHTS - VERY HAPPY AND VERY TAKENING - WELL WELL"*). Overall, the biggest difference remains: RoBERTa is the more aggressive detector for AI reviews as it captures nearly all of them. Whereas DistilBERT is the more moderate detector as it penalises far fewer real humans.

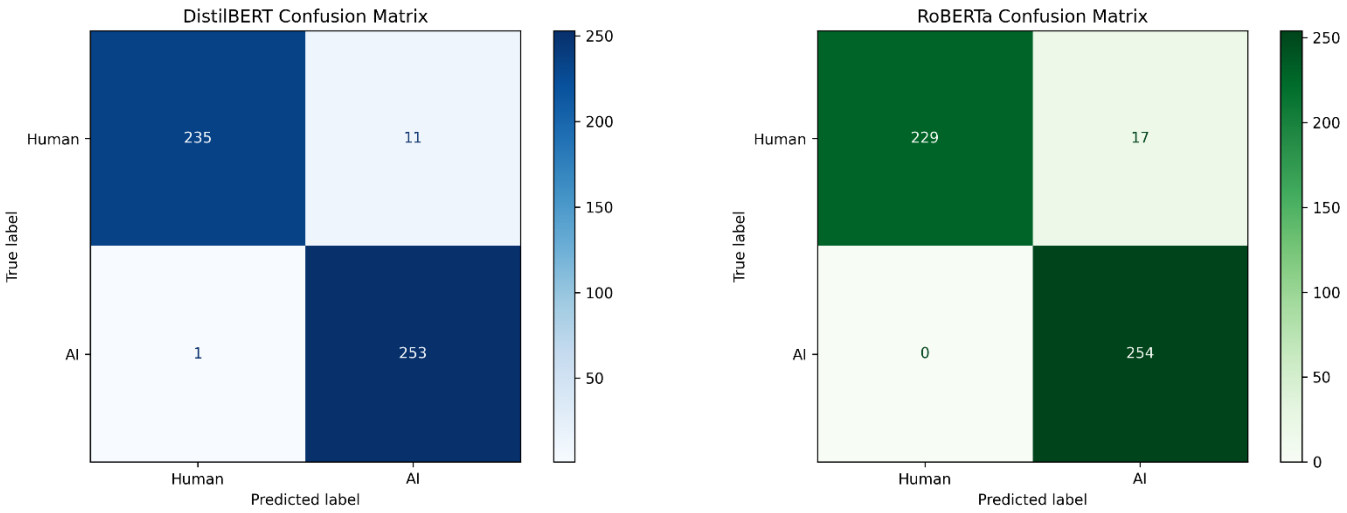


Figure 5 Confusion Matrices

Furthermore, length analysis was performed on the 500-sample subset to determine if the model was simply using review length in its predictions. This is due to the dataset demonstrating that human reviewers write substantially more (on average) than computer-generated reviews. In fact, for each sample reviewed by human reviewers and evaluated with DistilBERT, the average character count was 424, whereas the average character count for samples created with AI reviewers was 268. Consequently, if the model relied upon a superficial cue and determined the author of a given review based solely on the number of characters contained within the review, then I would have expected that character count to be highly correlated with predictions. However, when I ran a Pearson correlation between predicted authorship and character count, the results were a relatively modest -0.222. This result is significant because it suggests that DistilBERT was not primarily labelling reviews by review length alone. While length did appear to increase vulnerability for some human-written reviews, at least to some extent, it is not the primary decision-making factor. Furthermore, I found that all correct predictions have a near confidence score of 1.0 whereas incorrect ones were spread between ~0.5–0.95 it shows the model is not just occasionally wrong but is less certain when it is wrong, which is a positive sign about its calibration.

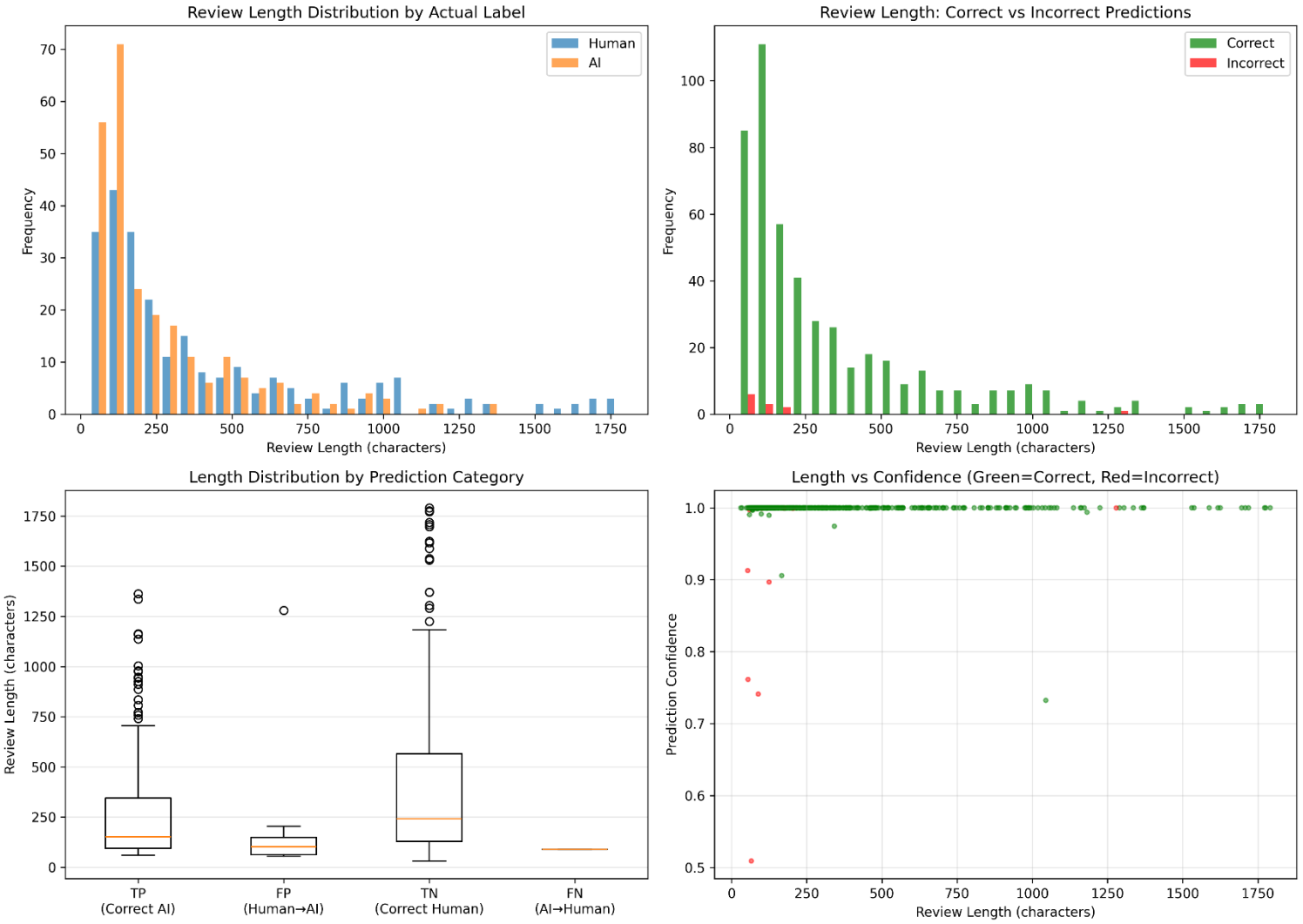


Figure 6 Review Length Analysis

6.3 Explainability and moderation usefulness

The explainability and triage outputs add another layer to the evaluation, because they give us some insight as to whether the system is useful beyond accuracy. The best example is the LIME explanation of the false positive "Bought for my 2 year old granddaughter." DistilBERT classified this review at an approximately 98.5% AI confidence level, yet the local explanation was not dominated by what would be considered an "AI sounding" word. Instead, the local explanation revealed a mix of cues: "granddaughter", "bought" moved away from the AI class, whereas "my", "year" moved towards it. This indicates that the model's error is not easy to describe using a common-sense story like "it identified marketing language". Rather, the model has learned a fragile boundary locally where ordinary personal phrasing can still combine to make a highly confident wrong decision. Thus, LIME adds value not because it proves the model is correct but how non-intuitive a wrong decision can be.

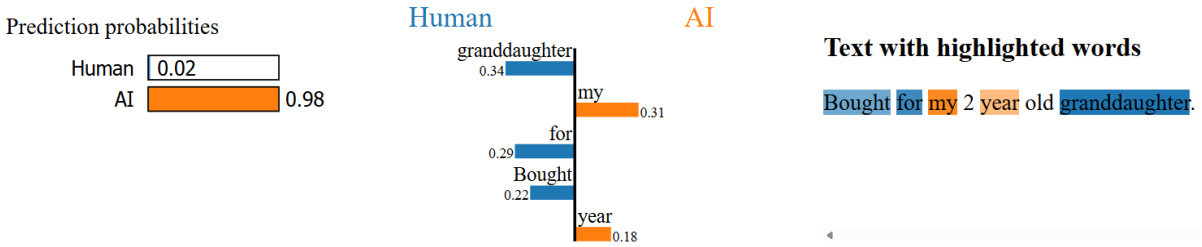


Figure 4 DistilBERTLIME.py HTML Output Example

The triage layer made it clear how serious these moderation issues are. In the 100-sample test batch, the system's accuracy was 95%. However, when you examine each of the five incorrect classifications, four were false positives. Of those four false positives, three were classified as High Risk, with confidences over 89%, and two of them had confidences greater than 99.9% and 99.7% respectively. This is the first example where we have seen that, while a model may perform well overall, its behaviour can still be very unsafe. A model can be wrong with high confidence in exactly the way that would lead to unfair treatment by moderators. The fact that there is an Ambiguous classification for the remaining sample provides some hope, since the threshold caught one of the low-confidence misclassifications. However, the overall lesson is much harsher: while thresholds help at the edges, they do nothing to address the major issue of high-confidence false positives.

Triage Category	Count	Correct	Incorrect
High Risk (Predicted AI)	51	48	3
Low Risk (Predicted Human)	48	47	1
Ambiguous (Manual Review)	1	0	1
Total	100	95	5

Table 2 Triage Report

6.4 Summary

The findings from this study suggest an intermediate conclusion. This study developed a well-defined benchmark detection tool for use on selected datasets. Among the models considered, DistilBERT was the stronger choice, as it balanced high precision with fewer false positives than RoBERTa. The additional evaluation tools also strengthened this conclusion. The disagreement analysis showed that DistilBERT performed better than RoBERTa on the most difficult cases, while the length analysis showed that its stronger performance was not simply the result of following a simple or trivial shortcut. The LIME and triage outputs also demonstrated why human intervention is still necessary, even when the overall metric values appear positive. Therefore, the main value added by this study was not simply “a model which achieves 97.9 percent”, but rather a clearer understanding of what each number means in a moderation context.

7 Discussion & Conclusion

7.1 Discussion

The main value of this project was not just to develop a highly effective classifier, but to provide clarity about what "highly effective" means when moderating content. Both transformer models were significantly better than the TF-IDF + Logistic Regression baseline using the same dataset; however, DistilBERT emerged as the best general option due to its high accuracy. Furthermore, compared to RoBERTa, it had a much smaller false positive rate. This is important since the worst possible error in this case is not failing to identify a fake review but rather identifying a real user as a fake. Therefore, the results suggest that the choice of machine learning model should not be based on performance alone, but also on the consequences of each error.

The results of this study were useful in addressing the central aim of the research, which was to develop an explainable detection system designed to support human moderators rather than replace them. The goal of establishing a baseline was met: the logistic regression model served as a useful starting point for measuring performance and illustrated that there are recurring stylistic patterns present in the dataset. However, this also demonstrated the limitations of using shallow models, or methods whose primary strength is their interpretability. Although the baseline model was relatively interpretable, it still demonstrated a tendency to make errors when classifying human-generated content. This represents an important trade-off and directly reflects a key theme in the background literature: increased transparency does not necessarily result in a more effective moderation system. This is consistent with earlier work on stylometric and surface-level approaches, which are often more interpretable but may struggle to capture the complexity of deceptive or AI-generated text (Stamatatos, 2009; Feng et al., 2012; Ippolito et al., 2020).

An important contribution of this work is that it aligns closely with Rudin's critique of post-hoc explanations (Rudin, 2019). While I am not disproving the argument, parts of my results do help show that, when the stakes are high, it is preferable to use or develop models that are understandable by design rather than trying to explain them after they have already been created. At the same time, my results also show why a completely interpretable-by-design approach is difficult in this area. Although the simplest model I developed (the baseline) was the most transparent, it also performed the worst and would have incorrectly penalised many human-written reviews compared with DistilBERT. Therefore, the real question is not simply "interpretable versus black box", but rather "what level of harm is acceptable?". In this study, LIME represents a practical middle ground: rather than claiming to reveal the 'truth' of a prediction, it helps expose the local evidence behind each decision, giving moderators more interpretable support than an unexplained score alone.

The significance of the explanation layer lies in the fact that it does not make the model trustworthy on its own but instead makes its mistakes more transparent. The LIME example of a false positive is significant because it illustrates how far from intuitive the model's local boundary can be. In moderation terms, this promotes contestability by giving a moderator the

opportunity to question what the model decided and what led it to make that decision. While this is much less than true interpretability, it is still better than blind faith.

This study should be viewed as an example of a controlled benchmark rather than proof that a real-world system for detecting fake AI-generated content exists. While the data used for training and testing the model is very large and has been made to be balanced, this makes a good basis for comparing models fairly; however, it only represents a single boundary between Amazon human reviewers and GPT-2 generated text. Due to this, the model's high in-distribution performance does not indicate whether the model would also perform well when applied to new forms of Large Language Models (LLMs), other prompts, or on additional platforms. Furthermore, previous studies have suggested that AI-text detection systems are subject to distribution shifts (Borile and Abrate, 2025). Therefore, the most conservative way to interpret these results is that they measure behaviour well under controlled conditions yet leave open the much harder task of demonstrating real world applicability.

7.2 Limitations and future work

Although both approaches used a seed of 42, they were not based on identical data splits. The baseline used `sklearn.train_test_split`, whereas the transformer models used Hugging Face dataset shuffling and selection. Due to this, the baseline and transformer models were likely trained and tested on different partitions of the data, so direct numerical comparisons between them should be treated with caution. Finally, due to limited time for testing, I did not evaluate how well the detector performs on newer generative models such as GPT-5. Since the synthetic class in the dataset was generated using GPT-2, it is unclear whether the detector would generalise to more recent models or to different prompting strategies.

Future work would involve testing it on reviews created using newer language generation models (such as GPT-5), and across a broader range of prompt styles than were used here, to determine whether the present findings remain valid across a wider variety of possible distributions. Additionally, it would be worth exploring how LIME compares with other methods for explaining model behaviour, including SHAP and integrated gradients, to see whether these provide a more accurate or stable representation of model decisions. Finally, a better experimental design could use reviews from multiple sources.

7.3 Conclusion

In conclusion, this study achieved all its main goals. It provided baseline results and developed two classifiers using transformer-based technology to be compared against each other. In addition, an explanation layer was added to both models and these model outputs were translated into a moderator-oriented triage workflow. DistilBERT is the strongest overall choice between the models tested for detecting harmful false positives (that is, it produces fewer harmful false positives than RoBERTa) while also providing strong benchmark performance. Although the project demonstrated the capabilities of models, it also demonstrated how easily they can fail or produce false information when over-relied upon.

References

Ahn, H. and Park, E., 2024. The impact of consumers' sustainable electronic-word-of-mouth in purchasing sustainable mobility: an analysis from online review comments of e-commerce. *Research in Transportation Business & Management*, [e-journal] 52, article 101086. doi:10.1016/j.rtbm.2023.101086.

Bansal, G., Wu, T., Zhou, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M.T. and Weld, D.S., 2021. Does the whole exceed its parts? The effect of AI explanations on complementary team performance. In: Association for Computing Machinery, *CHI Conference on Human Factors in Computing Systems*. Online virtual conference (originally Yokohama, Japan), 8-13 May 2021. New York, NY: ACM, pp.1-16. doi:10.1145/3411764.3445717.

Borile, C. and Abrate, C., 2025. How to generalize the detection of AI-generated text: confounding neurons. In: Association for Computational Linguistics, *Findings of the Association for Computational Linguistics: EMNLP 2025*. Suzhou, China, 4-9 November 2025. Suzhou, China: Association for Computational Linguistics, pp.25461-25476. doi:10.18653/v1/2025.findings-emnlp.1388.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In: Association for Computational Linguistics, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota, USA, 2-7 June 2019. Minneapolis, Minnesota: Association for Computational Linguistics, pp.4171-4186. doi:10.18653/v1/N19-1423.

Digital Platform Regulators Forum, 2023. *Working paper 2: examination of technology - large language models*. [online] Available at: <https://dp-reg.gov.au/publications/working-paper-2-examination-technology-large-language-models>.

Feng, S., Banerjee, R. and Choi, Y., 2012. Syntactic stylometry for deception detection. In: Association for Computational Linguistics, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea, 8-14 July 2012. Jeju Island, Korea: Association for Computational Linguistics, pp.171-175.

Gorwa, R., Binns, R. and Katzenbach, C., 2020. Algorithmic content moderation: technical and political challenges in the automation of platform governance. *Big Data & Society*, [e-journal] 7(1), article 2053951719897945. doi:10.1177/2053951719897945.

Ippolito, D., Duckworth, D., Callison-Burch, C. and Eck, D., 2020. Automatic detection of generated text is easiest when humans are fooled. In: Association for Computational Linguistics, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online, 5-10 July 2020. Online: Association for Computational Linguistics, pp.1808-1822. doi:10.18653/v1/2020.acl-main.164.

Jhaver, S., Bruckman, A. and Gilbert, E., 2019. Does transparency in moderation really matter? User behavior after content removal explanations on Reddit. *Proceedings of the ACM on Human-Computer Interaction*, [e-journal] 3(CSCW), pp.1-27. doi:10.1145/3359252.

Jindal, N. and Liu, B., 2008. Opinion spam and analysis. In: Association for Computing Machinery, *Proceedings of the 2008 International Conference on Web Search and Data Mining*. Palo Alto, California, USA, 11-12 February 2008. New York, NY: ACM, pp.219-230. doi:10.1145/1341531.1341560.

Loshchilov, I. and Hutter, F., 2019. Decoupled weight decay regularization. In: *7th International Conference on Learning Representations (ICLR 2019)*. New Orleans, LA, USA, 6-9 May 2019. New Orleans, LA: ICLR.

Meng, W., Harvey, J., Goulding, J., Carter, C.J., Lukinova, E., Smith, A., Frobisher, P., Forrest, M. and Nica-Avram, G., 2025. Large language models as 'hidden persuaders': fake product reviews are indistinguishable to humans and machines. arXiv preprint, arXiv:2506.13313. [online] Available at: <https://arxiv.org/abs/2506.13313>.

Mexwell, n.d. *Fake Reviews Dataset*. [dataset] Available at: <https://www.kaggle.com/datasets/mexwell/fake-reviews-dataset/data> .

Mir, A.Q., Khan, F.Y. and Chishti, M.A., 2023. Online fake review detection using supervised machine learning and BERT model. arXiv preprint, arXiv:2301.03225. [online] Available at: <https://arxiv.org/abs/2301.03225> .

Moradi, M. and Samwald, M., 2021. Post-hoc explanation of black-box classifiers using confident itemsets. *Expert Systems with Applications*, [e-journal] 165, article 113941. doi:10.1016/j.eswa.2020.113941.

Ni, J., 2018. *Amazon Review Data (2018)*. [dataset] Available at: <https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/>.

Nicks, C., Mitchell, E., Rafailov, R., Sharma, A., Manning, C.D., Finn, C. and Ermon, S., 2024. Language model detectors are easily optimized against. In: *Twelfth International Conference on Learning Representations (ICLR 2024)*. Vienna, Austria, 7-11 May 2024. Vienna: ICLR.

Ott, M., Choi, Y., Cardie, C. and Hancock, J.T., 2011. Finding deceptive opinion spam by any stretch of the imagination. In: Association for Computational Linguistics, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA, 19-24 June 2011. Portland, Oregon: Association for Computational Linguistics, pp.309-319.

Özaydın, H., 2025. Fake reviews and ratings undermining consumer trust. In: Özkaynar, K. and Abbasoğlu, Ş., eds. *Consumer, Marketing, AI: Dark Sides and Ethics*. [online] Available at: <https://doi.org/10.58830/ozgur.pub710.c3028> .

Palmeira Ferraz, T., Dias Duarte, C.H., Ribeiro, M.F., Goes Braga Takayanagi, G., Alcoforado, A., de Deus Lopes, R. and Susi, M., 2024. Explainable AI to mitigate the lack of

transparency and legitimacy in internet moderation. *Estudos Avançados*, [e-journal] 38(111), pp.381-405. doi:10.1590/s0103-4014.202438111.020.

Parasuraman, R. and Manzey, D.H., 2010. Complacency and bias in human use of automation: an attentional integration. *Human Factors*, [e-journal] 52(3), pp.381-410. doi:10.1177/0018720810376055.

Ribeiro, M.T., Singh, S. and Guestrin, C., 2016. 'Why should I trust you?': explaining the predictions of any classifier. In: Association for Computational Linguistics, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. San Diego, California, USA, 12-17 June 2016. San Diego, California: Association for Computational Linguistics, pp.97-101. doi:10.18653/v1/N16-3020.

Rudin, C., 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, [e-journal] 1(5), pp.206-215. doi:10.1038/s42256-019-0048-x.

Salminen, J., Kandpal, C., Kamel, A.M., Jung, S.-G. and Jansen, B.J., 2022. Creating and detecting fake reviews of online products. *Journal of Retailing and Consumer Services*, [e-journal] 64, article 102771. doi:10.1016/j.jretconser.2021.102771.

Salton, G. and Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), pp.513-523. doi:10.1016/0306-4573(88)90021-0.

Sebastiani, F., 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), pp.1-47. doi:10.1145/505282.505283.

Stamatatos, E., 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3), pp.538-556. doi:10.1002/asi.21001.

theArijitDas, n.d. *theArijitDas/Fake-Reviews-Dataset*. [dataset] Available at: <https://huggingface.co/datasets/theArijitDas/Fake-Reviews-Dataset> .

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. and Garnett, R., eds. *Advances in Neural Information Processing Systems 30*. Long Beach, California, USA, 4-9 December 2017. Red Hook, NY: Curran Associates, Inc., pp.5998-6008.

Yoo, K.-H. and Gretzel, U., 2009. Comparison of deceptive and truthful travel reviews. In: Höpken, W., Gretzel, U. and Law, R., eds. *Information and Communication Technologies in Tourism 2009*. Vienna: Springer, pp.37-47. doi:10.1007/978-3-211-93971-0_4.

Appendix

Project Proposal

Sufyan Osman

25/26: SCC.300: Third Year Project

SCC.300: Third Year Project Proposal

Abstract

In this project I plan to investigate the growing problem of AI-generated product and service reviews. This has major implications for consumer trust and can contribute to deceptive marketing. As large language models (LLM) become more accessible, review-based websites are increasingly vulnerable to manipulation that can mislead buyers and affect consumer trust. This project will therefore focus on detection approaches which provide clear and actionable explanations to a platform's trust team to ensure that steps can be taken to flag and remove LLM-generated reviews. The outcomes are expected to improve our understanding of which linguistic signals are most likely to be generated by LLMs and used in reviews.

1. Introduction

Studies show that one of the main factors affecting consumer purchase decision-making is the average score of a product's online reviews (Zhu et al., 2012). At the same time, the rapid growth of large language models has led to a surge in AI-generated review content, as even novice users can now use these tools to produce convincing fake reviews that may be mistaken for genuine customer feedback (Xylogiannopoulos et al., 2024).

The problem is likely to worsen as these tools become more accessible. In 2023 alone, 4.4% of the 30.2 million reviews submitted to TripAdvisor were identified as fraudulent and removed (Xylogiannopoulos et al., 2024). In response to growing concern, regulators have introduced stronger rules explicitly targeting fake reviews and placing greater responsibility on businesses for the content hosted on their platforms (Ungoed-Thomas, 2025).

The proposed project will therefore investigate explainable detection methods for AI-generated product and service reviews, with the objective of producing detection outputs that are both accurate and interpretable for moderation teams. Rather than only returning a binary decision, the project will examine how model predictions can be supported by explanations that highlight the linguistic patterns influencing the result.

2. Background

Previous work on deceptive and spam reviews provides a useful foundation for this project while also showing the important gaps that this study intends to address.

Survey papers reveal decades of work on using linguistic signals to detect spam, highlighting that the development of review detection algorithms remains an active research area (Wu et al., 2025; Crawford et al., 2015). More recent work has begun to focus specifically on AI-generated review

detection, with studies arguing that explainability is an important requirement because moderation teams need actionable reasons rather than opaque system outputs (Chernyaeva et al., 2025).

Explainable AI methods have matured and are now widely discussed in relation to natural language processing classifiers. Techniques such as SHAP have been used to show which features or words push a model towards a particular decision, and prior work suggests that explanations can help analysts build trust in a system while also identifying its weaknesses, although these explanations must still be assessed carefully before deployment (Lundberg and Lee, 2017).

Existing work therefore shows strong interest in AI-text detection, growing concern around the moderation of generated content, and increasing emphasis on interpretability. However, there is still limited work that focuses specifically on how detection outputs can be communicated in a way that is directly useful for practical moderation workflows. This project will address that gap by developing and evaluating a review detection pipeline that combines classification performance with interpretable, moderator-facing outputs.

3. The Proposed Project

The aim of this project is to compare different approaches to build a reliable AI-generated review detector that flags suspicious product/service reviews and explains why each review is suspicious, so the platform's trust teams can triage faster and make better decisions.

Objectives

- Develop and test different methods of classifying online customer reviews to detect if the review was generated by artificial intelligence (AI) or written by a human.
- Implement a baseline machine learning classifier using TF-IDF features and logistic regression.
- Test how well fine-tuned versions of transformers perform in comparison to the baseline model.
- Integrate an explainability method so that model predictions can be supported with clear and interpretable reasoning.
- Develop a simple moderator-facing output format that translates model predictions and explanation features into plain-English summaries for triage.

3.1 Methodology

This project uses a comparative methodology focused on text classification. Only review text will be used because of the limited availability of datasets containing both account metadata and corresponding AI-generated reviews. However, sufficient labelled review-text datasets do exist, which makes supervised learning possible and allows the study to focus directly on whether review text itself is human-written or AI-generated.

The project begins by researching labelled datasets that contain a mix of review content written by humans and generated by artificial intelligence (AI). The dataset will then be cleaned and prepared for modelling before being divided into training and testing data. A baseline classifier will first be implemented using TF-IDF features with logistic regression. After this, transformer-based models will be trained and evaluated for the same task to compare their performance against the baseline approach.

To improve the interpretability of model outputs for moderators, the project will incorporate a low-resource post-hoc explainability component. Keywords and phrases are extracted from each review which were used by the model to make its decision that lead to a flagged review. These outputs can be easily transformed into a few sentence summaries for moderators so they can understand both how accurate the model was and why it made those decisions.

The quantitative evaluation of the system is based on established classification metrics (accuracy, precision, recall, and F1-score). The study will also analyse a confusion matrix and disagreements between models, as well as several examples of output, to assess whether the explanations produced by the model can help understand its behaviour. Therefore, this project will measure both the predictive value of the system's performance and whether it supports moderation or triage decisions.

4. Programme of Work

Project Setup and Research

- Revisit the papers referenced in this proposal and summarise the relevant methods (AI detectors, linguistic features, explainable AI (XAI))
- Decide on the datasets to be used and submit the appropriate ethics documentation.

Data Assembly and Preprocessing

- Download and standardise datasets into a single schema.
- Download LLM-generated and human-written text for the text classifier dataset.
- Create splits for training and testing data.

Baseline Models

- Develop a TF-IDF + Logistic Regression model.
- Evaluate the model on testing data.

Transformer Model Development

- Implement and fine-tune transformer-based classifiers for the same task.
- Compare their performance against the baseline model.
- Record and analyse differences in prediction behaviour.

Explainability and Moderator-Facing Outputs

- Develop and include an explainability methodology that will be used to identify features that influence predictions.
- Develop explanation outputs for reviews.
- Create a triage-style output that can be easily understood in plain English by moderators.

Testing and Evaluation

- Evaluate each model with appropriate performance metrics and confusion matrices.
- Analyse selected misclassification and disagreement cases.
- Determine how much the explanation outputs improve interpretability and usefulness for moderation.

Write-up and Finalisation

- Collate findings and discussions.
- Address limitations, potential practical applications, and areas for further development.
- Finalise the study including any appendix.

The project schedule can be seen in this Gantt chart:

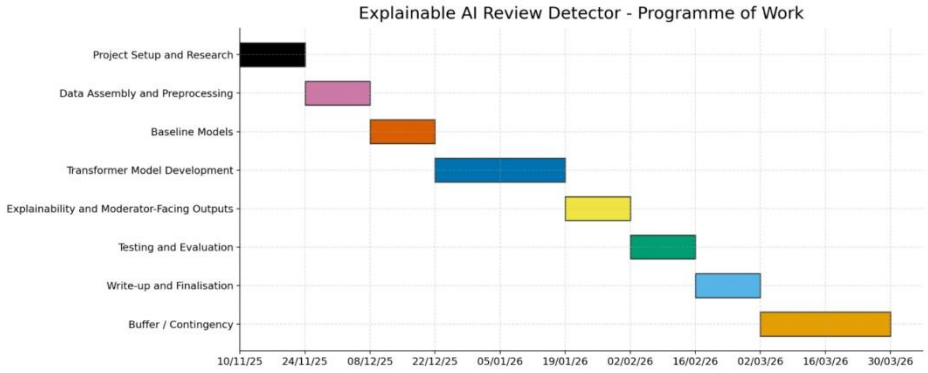


Figure 1 Gantt Chart

5. Resources Required

This project requires minimal resources, primarily access to a GPU-capable machine or cloud-based GPU services. These will be used to train the models and run the explanation computations. The university will provide these resources.

References

Chernyaeva, O., Lee, O.-K. D. and Hong, T., 2025. From fake to AI-generated: Leveraging information manipulation theory and explainable AI for robust detection of manipulated reviews. Decision Support Systems, Volume 197, pp. 114-522.

Crawford, M. et al., 2015. Survey of review spam detection using machine learning techniques. Journal of Big Data, 2(1), p. 23.

Lundberg, S. and Lee, S.-I., 2017. A Unified Approach to Interpreting Model Predictions. Volume 05.

Ungoed-Thomas, J., 2025. UK bans £2.2bn 'sneaky' fees and fake reviews for online products. [Online].

Wu, J. et al., 2025. A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions. Computational Linguistics - Association for Computational Linguistics, 51(1), pp. 275-338.

Xylogiannopoulos, K. F., Xanthopoulos, P., Karampelas, P. and Bakamitsos, G. A., 2024. ChatGPT paraphrased product reviews can confuse consumers and undermine their trust in genuine reviews. Can you tell the difference? Information Processing and Management, 61(6).

Zhu, L., Zhang, W. and Yanchun, Z., 2012. Research on the influence of online reviews on internet consumer purchasing decision. 2012 International Conference on Management of e-Commerce and E-Government, pp. 38-41.

Ethics Approval for Dataset

Outlook

SCC UG Ethics Application - Outcome Notification- Submission

From SCC UG Ethics <scc-ug-ethics@lancaster.ac.uk>
Date Mon 11/17/2025 5:10 PM
To Osman, Sufyan (Student)
Cc SCC UG Ethics <scc-ug-ethics@lancaster.ac.uk>

Dear Sufyan Osman,

Your UG Ethics application for the project titled *"An Explainable Approach to AI-Generated Text Detection for Product Review Moderation"* has been reviewed by *Tharindu Ranasinghe*.

Based on the information provided, **they have APPROVED the application**.

You can now start the study.

Additional Instructions:

- If the design of your study changes, you will need to submit another application.
- Please attach a copy of this email to the Appendix section of your TYP report
- Supervisor comments (if any): ""

Thank you.

Regards,

SCC Ethics Coordinator

--Application Contents--

Basic Information

Student Name: Sufyan Osman

Supervisor/Convenor: Tharindu Ranasinghe

Course/Module Context

Is for TYP research (SCC.300)? Yes

Course name: Not Shown/NA

Module details: SCC.300 - TYP

Is group project? NA

Group Details

Size of group: Not shown/NA

Group member names:

Not shown/NA

Group member email ids:

Not shown/NA

Project Basic Information

Project Title: An Explainable Approach to AI-Generated Text Detection for Product Review Moderation

Project Abstract:

I will be building a text classifier to detect whether a product review is AI generated or human written along with a plain English explanation for the model's outcome. This will be used by moderation teams to triage faster and make better decisions.

Study Aims/Description:

This study aims to train and evaluate a novel, explainable detection system for LLM-generated product reviews using a processed Fake Reviews Dataset (originally ~20,000 AI generated and ~20,000 real reviews) which can be found here https://huggingface.co/datasets/theArijitDas/Fake-Reviews-Dataset.

Does project involve a study/evaluation? Yes

Primary scope for participant type: Secondary data (e.g. you will use data previously generated by humans for other purposes like but not limited to posts from social media)

Human participants in secondary scope: Not Shown/NA

GDPR Compliance: Acknowledged. Confirm evidence in documentation.

Human Participants - 1

Basic information

Participant description:

No new human participants will be recruited for this stage — I am using secondary data only. The dataset is the processed Fake Reviews Dataset (≈20,000 labelled fake, 20,000 labelled real reviews) augmented with 94 AI-generated examples; labels are 0 = human, 1 = computer-generated. The dataset contains no participant contact details or sensitive personal identifiers.

Participant numbers [min,max]: [1, 1]

Participant age range: ["26-30 years","31-65 years"]

Participant recruitment approach:

I will not recruit human participants for this stage — I am using a public secondary data only. I will obtain the processed Fake Reviews Dataset from its Hugging Face repository (theArijitDas/Fake-Reviews-Dataset) under the Apache-2.0 licence.

Human Participants - 2

Special category participants:

- My study does not require participants that definitely satisfy any of the above criteria

Checks for deception, trickery, undue stress and debrief:

- No. The study does not include activities that can cause stress or anxiety or involve physical contact.

- No. The study does NOT provide opportunity for timely and appropriate debriefing.
- No. The study does not involve deception or covert methods (e.g. withholding essential information or intentional misinformation)

Human Participants - 3

Consent and Privileged Data

Use of privileged data:

- No. My study does not collect data which falls in any of the above categories.

Ability to provide informed consent: Yes. Data is acquired from secondary services and participants have agreed that their data can be used for research purposes

Third-party consent: No, the study does not require such permission

---

Non-Human Participants

Non-human participants as primary scope of study?No

Non-human participants as secondary scope? No

Non-human participants details:

Risks

Indicated risk profile:

- No. I have carried out risk assessment and there is no risk to any individuals due to the nature of study or location of the study

---

Data Handling

What type of data is collected?

I will be using public secondary data only. No new participant questionnaires or personal data will be collected in this stage. https://huggingface.co/datasets/theArijitDas/Fake-Reviews-Dataset

Use of digital surveys: My study does not involve computer-based surveys or questionnaires

Data storage strategy:

I will be using public secondary data only. No new participant questionnaires or personal data will be collected in this stage. https://huggingface.co/datasets/theArijitDas/Fake-Reviews-Dataset

---

Anonymity and Confidentiality

Applicable situations:

- Participation in study is anonymous
- Plans for anonymisation ensure that participant cannot be identified indirectly
- Participants have been promised confidentiality
- Data processing will ensure participant confidentiality is maintained

Anonymity and confidentiality plans:

I will be using public secondary data only. No new participant questionnaires or personal data will be collected in this stage. https://huggingface.co/datasets/theArijitDas/Fake-Reviews-Dataset

---

--End Contents--

Observations (if any): Exception granted on basis of automated processing of publicly available dataset