Turkish AI-Generated Review Detection

Halil Erkan
Computer Engineering
TOBB ETU
Ankara, Türkiye
herkan@etu.edu.tr

Zeynep Meriç Aşık
Artificial Intelligence Engineering
TOBB ETU
Ankara, Türkiye
zeynepmericasik@etu.edu.tr

***Abstract*—The increase of AI-generated content, poses new challenges in distinguishing between human and AI-generated texts. This study focuses on the detection of machine-generated reviews in Turkish, leveraging classical machine learning algorithms as a baseline while also implementing two novel frameworks. It is aimed to compare the effectiveness of traditional models with these different approaches, addressing the unique linguistic features of the Turkish language. Initial results indicate that integrating language-specific adaptations significantly enhances the detection accuracy, offering promising directions for further research in AI-generated content identification in Turkish language.**

***Index Terms*—NLP, AI-generated text detection, Turing Bench, Adversarial Training**

I. Introduction

In the evolving landscape of text generation, the distinction between human and machine-generated content is becoming increasingly blurred. This has significant implications, particularly in areas like consumer reviews where authenticity impacts consumer trust and business reputation. To address this challenge, our research focuses on the detection of AI-generated reviews in the Turkish language, a linguistic area that is underrepresented in current literature.

For our baseline, we utilize classical machine learning algorithms—Support Vector Machines (SVM) and Naive Bayes—employing TF-IDF vectorization, and Linear Regression with n-gram vectorization. These methods have proven effective in various text classification tasks so with the two frameworks that is going to be applied to the Turkish data, the aim is to check whether they will work better than these well-established methods in the field of Natural Language Processing.

In addition to these traditional approaches, the two cutting-edge frameworks that is decided to be tested are yet to be explained. The first, as detailed in ”TuringBench: A Benchmark Environment for Turing Test in the Age of Neural Text Generation” [1], offers a comprehensive suite of tests designed to challenge the capabilities of text generation models under diverse conditions. The second framework, titled ”Enhancing Machine-Generated Text Detection: Adversarial Fine-Tuning of Pre-Trained Language Models,” [2] describes an innovative approach involving adversarial fine-tuning of language models to improve detection accuracy.

The goal of this study is not merely to apply these frameworks but to adapt and optimize them for the Turkish context. By doing so, it is aimed to contribute to the broader discourse on machine-generated text detection, offering insights and methodologies that could be adapted for other languages and settings.

II. Related Work

In this field of Natural Language Processing, there is a notable lack of prior research focusing on the Turkish language. Despite these challenges, examining existing work on the topic—whether language-specific or not—can provide valuable insights and clarify the steps needed to advance this study.

Data cleaning and preprocessing are pivotal to the success of Natural Language Processing (NLP) experiments, as they significantly influence the quality and effectiveness of the models developed. The article from Towards Data Science on data preprocessing in NLP outlines several key steps essential for preparing data for NLP tasks [3]. These steps include tokenization, the process of splitting text into meaningful elements like words or phrases; normalization, which involves converting all text to a uniform case and removing punctuation; and removing stop words that are frequent in language but carry little semantic importance. Applying these techniques to both human-written and AI-generated Turkish datasets will be crucial in reducing noise and standardizing the input for further processing. This methods in this case may help the model to generalize the model, even though there is no proof whether these methods are actually helpful in the field.

In the development of NLP systems, particularly those aimed at classifying text such as detecting AI-generated reviews, it is crucial to establish robust baseline models and effective vectorization techniques. This section continues with reviews of foundational machine learning models and text vectorization strategies relevant to the task of detecting AI-generated text. The most popular models utilized for this case is the Support Vector Machines (SVM) and Naïve Bayes classifier [4]–[6].

To train machine learning models on text data, the text must first be converted into a numerical format. This process is known as vectorization, and several techniques are pivotal for transforming raw text into trainable vectors such as Term Frequency-Inverse Document Frequency (TF-IDF) and N-gram Vectorization [7], [8].

Incorporating SVM and Naive Bayes as baseline models, along with sophisticated vectorization techniques like TF-IDF and N-gram, provides a strong foundation for developing an NLP system to detect AI-generated text. These methods ensure that the system is not only equipped to handle the linguistic complexities inherent in natural language but also attuned to the subtle distinctions between human and machine-generated texts.

As the NLP field has evolved, advanced models like BERT, RoBERTa, XLNet, and GPT-2 have been increasingly recommended for text classification tasks due to their ability to understand the context and nuances of language more effectively than traditional models. Devlin et al. [9] demonstrated that BERT, a transformer-based model pre-trained on large corpora, achieves state-of-the-art results in various NLP tasks, including text classification. Liu et al. [10] extended this work with RoBERTa, which refined BERT's training approach, further improving performance in downstream tasks. Yang et al. [11] introduced XLNet, which leverages permutation-based training to capture bidirectional context, outperforming previous models like BERT in several benchmarks. Additionally, Radford et al. [12] developed GPT-2, a transformer model optimized for generative tasks, which also demonstrated strong performance in text classification due to its ability to generate and understand coherent and contextually relevant text. These models are particularly effective in handling complex sentence structures and have shown to outperform traditional methods like SVM and Naive Bayes in text classification, making them suitable for detecting AI-generated text in multiple languages, including Turkish. Moreover, Güneş et al. [13] introduced BERTurk, a BERT-based model specifically trained on Turkish corpora, which has achieved state-of-the-art results in Turkish NLP tasks. The use of such models in this study could provide a more robust detection mechanism due to their deep learning capabilities and their ability to leverage contextual embeddings.

In the context of imbalanced datasets, standard metrics like accuracy may not always provide a true picture of model performance, especially when the data is skewed towards one class. Therefore, other metrics such as Precision, Recall, and the F1 Score are often more indicative of the model's effectiveness in classifying both classes accurately [14].

The method of baseline-modeling is suggested in a survey by Wu et al. [15] dedicated to AI-generated text detection. This survey also covers various detection techniques including watermarking, zero-shot methods, fine-tuning, adversarial learning, and human-assisted methods. The survey emphasizes the need for robust detectors to mitigate the misuse of LLMs, highlighting challenges such as out-of-distribution problems and data ambiguity, and suggests future research directions to enhance detection capabilities and responsible AI governance.

There was mentioned two detection methods: Black-Box and White-Box detection. Black-box detection involves using models to detect AI-generated text without any knowledge of the internal workings of the text generation model. It treats the generator as a black box and relies solely on the input-output behaviour for detection. This study uses pre-trained models like BERT, RoBERTa, GPT-2, XLNET, Electra, and DistilBERT without modifying their architectures or training processes as a first step which is categorized as black-box detection. These models analyse the generated text based on learned patterns and statistical properties. White-box detection, on the other hand, involves understanding and utilizing the internal mechanics of the text generation model. This method can exploit specific features or weaknesses of the generator. If the insights are incorporated into the generation mechanisms of the models or adjusted detection strategies based on the generator's architecture, that would be engaging in white-box detection. In the second part, this study utilizes adversarial training to create texts that probe specific characteristics of the generation model would fall under white-box detection.

As mentioned by Guerrero et al. [16], adversarial attacks are creating problems for current detection models. The study identifies gaps in current research, including the need for more robust models against adversarial attacks and the optimization of detectors for low-resource settings. It emphasizes the importance of combining traditional machine learning models like Naive Bayes and SVM with advanced neural models for effective detection. That is why adversarial training is a vital addition to common models that are utilized on this area of research. The methodologies explored above are discussed throughout the Methodology section.

III. Methodology

A. Baseline Models

The baseline models employed are SVM and Naïve Bayes using TF-IDF vectorization, along with Logistic Regression which utilizes N-gram vectorization. Optimal parameters for these models have not yet been determined. Preliminary testing indicates that the best performance is achieved when unigram, bigram, and trigram vectorizations are used concurrently. This may be due to the similarity in wording of complaints, regardless of the author. However, large language models like GPT often generate unique phrases spanning 2-3 words, which may not typically be used in manual complaint submissions. This phenomenon occurs irrespective of the model being instructed to produce outputs in daily, aggressive, or formal language tones, potentially explaining the superior efficacy of the combined uni-, bi-, and tri-gram approach.

Each model—SVM, Naïve Bayes, and Logistic Regression—also has its unique parameters that require fine-tuning. A dedicated run for hyperparameter optimization is necessary, though it has not yet been conducted. The outcomes from the preliminary tests and their comparative analysis are discussed in the Results section.

B. Frameworks (TuringBench/Adverserial)

Upon accurate evaluation of the machine learning models, a separate experiment will be conducted using specialized frameworks designed to detect AI-generated text. These frameworks, however, are typically trained to identify AI-written content primarily in English and are not specifically tailored for user complaint reviews. The objective is to adapt these models to effectively operate on the uniquely assembled dataset for this study, which focuses on a specific use case.

The first, as detailed in ”TuringBench: A Benchmark Environment for Turing Test in the Age of Neural Text Generation” [1], offers a comprehensive suite of tests designed to challenge the capabilities of text generation models under diverse conditions. The second framework, titled ”Enhancing Machine-Generated Text Detection: Adversarial Fine-Tuning of Pre-Trained Language Models,” [2] describes an innovative approach involving adversarial fine-tuning of language models to improve detection accuracy. The aim of this study is using the methods proposed in the TuringBench article and enhance the performance of this framework using Adversarial Training method. An examination of the models and their operational mechanisms is presented below.

TuringBench is a comprehensive benchmark environment designed to evaluate the capability of various models to distinguish between human-written and AI-generated texts. This framework is particularly relevant given the advancements in generative language models that produce text almost indistinguishable from that written by humans.

TuringBench includes a dataset of 200K samples comprising both human and AI-generated texts across 20 different labels. These labels represent various generative models and a human label, encompassing models like GPT versions, GROVER, CTRL, XLM, XLNET, and others.

There are two main benchmark tasks within TuringBench: Turing Test (TT) and Authorship Attribution (AA). Turing Test involves a binary classification problem where the goal is to classify texts as either human or AI-generated. It is modelled after the classical Turing Test, which assesses a machine’s ability to exhibit human-like intelligence. The test contains subtasks for each pair of human and machine model. Authorship Attribution task extends beyond binary classification to identify which specific neural model generated a given text if it is determined to be AI-generated. However, since the AI dataset utilized in this study only includes data from GPT-3.5 and GPT-4o, this part of the framework was not practiced in the experiments. The framework also features a website with leaderboards that track the performance of various models on the benchmark tasks, providing a competitive and open platform for researchers. Thus, it can also be used for evaluation and comparison with the other models. However, preliminary results from TuringBench experiments suggest that newer models like GPT-3 and FAIR_wmt20 generate text that is highly indistinguishable from human writing, posing challenges for current detection methods. So, it might be hard to surpass the Machine Learning models that have been on use forever as in accuracy or recall metrics.

For this study on detecting AI-generated complaint reviews in Turkish, TuringBench offers a robust framework for developing and evaluating specific models. Given that TuringBench primarily deals with English texts, the models are needed to be trained or fine-tuned on the Turkish dataset. This involves either adapting existing models within TuringBench to understand Turkish through transfer learning. TuringBench’s tasks are also designed for general text, but they can be customized to focus specifically on complaint reviews. This might involve adjusting the types of prompts used for generating machine text which is already available in the newly generated Turkish dataset or the features which used in model training to better capture the nuances of complaint language.

The evaluation metrics are going to stay as usual traditional ones such as precision, recall, F1 scores, and accuracy to evaluate the models. Given the class imbalance in real-world scenarios (possibly more human than machine-generated texts), and also in this study which the human dataset is many times greater than AI-generated, these metrics can help assess the effectiveness of the detection framework.

To leverage the benchmarking aspect of TuringBench, setting up similar leaderboard systems to compare different models’ performances on Turkish complaint review dataset would be appropriate. This could foster a collaborative environment and push for further improvements in the models. By integrating TuringBench’s methodologies and adapting its tasks and datasets for Turkish, the study can establish a pioneering framework for detecting AI-generated texts in underrepresented languages and specific domains like complaint reviews.

The Adversarial Fine-Tuning framework focuses on enhancing the detection of AI-generated text through adversarial fine-tuning of pre-trained language models (PLMs), such as BERT (Bidirectional Encoder Representations from Transformers).

The framework generates adversarial examples that mimic human modifications to texts. These examples are crafted using the T5 model, which modifies the input text by introducing subtle perturbations that are typically indistinguishable to humans but can mislead machine learning models. These adversarial examples are then used in training the PLMs. The process involves re-training the PLMs with a mix of original and adversarial texts, which helps the models learn to differentiate between human-like AI-generated text and genuine human text.

The PLMs are fine-tuned using a binary classification approach where the model learns to classify texts as either AI-generated or human-written. The fine-tuning process leverages adversarial examples to improve the robustness and accuracy of the models in detecting nuanced differences in text. The adversarially fine-tuned models show a significant improvement in detecting AI-generated texts, reducing misclassification rates and enhancing metrics like accuracy and F1 score compared to traditional fine-tuning methods.

To apply this framework into this study, the first step is to utilize a model like T5 to create adversarial examples from the Turkish dataset. Since the focus is on complaint reviews, the adversarial modifications should mimic common expressions and nuances specific to complaint narratives in Turkish. Since the original framework is demonstrated primarily on English data, adapting the language model to understand Turkish is crucial. This step involves pre-training the model on a large corpus of Turkish texts or using a multilingual model that includes Turkish in its training data. The adversarially fine-tuned model will be trained using adversarially modified Turkish complaint review texts. This training aims to help the model learn the specific characteristics of AI-generated versus human-generated complaint texts.

To evaluate the model's performance, standard metrics like precision, recall, and F1 score are used. Additionally, considering the creation of a validation set that reflects the real-world distribution of human and AI-generated texts is essential for assessing the model's practical effectiveness.

Based on initial results, to better capture the subtleties of AI-generated texts in the context of Turkish complaint reviews, the adversarial examples and fine-tuning processes are iteratively refined.

By utilizing the adversarial fine-tuning approach, this study can significantly advance the detection of AI-generated text in a less commonly studied language and application area, providing valuable insights into the capabilities and limitations of current NLP technologies in new domains.

C. Results

Before testing on existing human-written reviews, the generation of AI-produced Turkish reviews was necessary, as no dataset of such content existed. To create this dataset, two of OpenAI's GPT Language models—GPT-3.5 and GPT-4o—were utilized, with each model producing approximately 3000 reviews. The prompts used to generate these texts were included within the script that is used for data scraping. After scraping all, some preprocessing methods also are applied to such as tokenization, lowercasing, stop-word removal, stemming.

In order to make the human and AI data similar, the human dataset was processed. In the human dataset, the first sentence of each instance showcases like a title for the complaint. So, the example prompt to produce the AI-generated dataset is like below:

prompt = (first sentence of an instance from human dataset) + ” Sen bir tüketicisin. Bu konuda günlük dilde 60-100 kelimelik bir şikayet metini yaz. Talep ediyorum, Aksi takdirde, arz ederim gibi kalıplar kullanma, resmi dil kullanma. Örnek Girdi: A101 Reklamınızı Yaptığınız Ürününü Bulamamak. Örnek Çıktı: A101 Reklamınızı Yaptığınız Ürününü Bulamamak. 21.05.2020 tarihinde satışa sunduğunuz filtre kahve makinenizi Ankara'da esat semtinde 8.40'de reşit galip şubesi. 8.50'de esat caddesindeki daha sonra yine 9.02'de esat caddesindeki öbür şubenize sordum ikisi gelmediğini diğeri de 1 tane geldiğini onunda satıldığını söylediler.”

This prompt was constructed with trial and error. It was first experimented manually, giving different prompts and focusing on the unwanted parts. For example, the LLM kept writing the responses like a letter, but the structure should look like a human is complaining on a social media cite like sikayetvar.com. Then there were some restrictions added to the prompt. The final prompt was as given above.

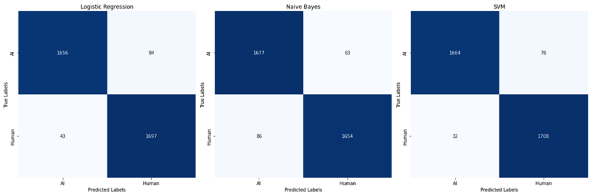

Fig. 1. Confusion Matrices of Baseline Models

For the human-written component, an existing dataset [?] was utilized for training purposes. It was down-sampled to achieve a balanced dataset when combined with the AI-generated texts (AI/Human). For pre-processing, the methods explained above were also applied to the human data and also the instances (the first sentences) from this dataset which were used for producing the AI dataset are dropped since it might cause data leakage.

Initially, the project aimed to predict whether any text was written by a human or an AI. However, it was later realized that to accurately perform such broad detection, a much larger dataset would be required than could feasibly be produced manually. At that point, a general form of data had already been produced and trained using Turkish Wikipedia data. Although the results exhibited higher than expected accuracy and recall, they were not reliably indicative of real-world performance. Consequently, the focus was shifted to producing and utilizing the datasets explained and later merged above specifically comprising user reviews, which could be more practically relevant for both companies and consumers.

Three base models were trained: SVM, Logistic Regression, Naïve Bayes. While SVM and Naïve Bayes are trained with tf-idf measurements, Logistic Regression is trained with n-gram vectorization. Since the dataset is still relatively small, a different approach was utilized to improve generalization of scores. 10 different models are created for each and saved all scores, after that took average of those scores. For each iteration, random sampling operation is repeated so that each model is unique.

TABLE I
PERFORMANCE METRICS OF BASELINE MODELS

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.9635	0.9517	0.9747	0.9631
Naive Bayes	0.9572	0.9638	0.9512	0.9575
SVM	0.9690	0.9563	0.9811	0.9686

Table 1 and Fig. 1 provide a comparative analysis of three baseline models—Logistic Regression, Naive Bayes, and SVM—used in the AI versus Human text classification task. The models are evaluated based on their accuracy, precision, recall, and F1 scores, which are crucial metrics for assessing the performance of classifiers, especially in imbalanced datasets.

The SVM model exhibits the highest accuracy at 0.9690 and also leads in terms of F1 score with 0.9686. This high performance suggests that SVM is highly effective in distinguishing between AI-generated and human-written texts. The confusion matrix for SVM further supports this, showing a lower number of misclassifications: 76 AI texts misclassified as Human and 32 Human texts misclassified as AI. Notably, SVM also has the highest recall at 0.9811, indicating its robustness in detecting Human texts. This could be advantageous in scenarios where accurately identifying human-authored content is critical, such as in plagiarism detection or content authenticity verification.

Logistic Regression performs slightly below SVM with an accuracy of 0.9635 and an F1 score of 0.9631. The confusion matrix shows that it misclassifies 84 AI texts as Human and 43 Human texts as AI. Logistic Regression demonstrates high recall (0.9747), meaning it is also effective at identifying Human texts, but its precision (0.9517) is slightly lower, which suggests it may produce more false positives compared to SVM. This might be less ideal in applications where false positives could lead to significant consequences.

Naive Bayes, while still performing well, shows the lowest accuracy at 0.9572 and an F1 score of 0.9575. The confusion matrix reveals that Naive Bayes misclassifies 63 AI texts as Human and 86 Human texts as AI. This model has the lowest recall (0.9512) among the three, indicating that it is less effective at detecting Human texts, which could lead to a higher false negative rate. This might be a disadvantage in applications where misclassifying Human text as AI could have significant implications, such as in legal or academic settings.

In summary, while all three models demonstrate competent performance, SVM stands out as the most reliable classifier for this specific task, particularly in scenarios where the precision and recall of detecting human-authored texts are crucial. Logistic Regression also performs well and might be preferred in contexts where a balance between precision and recall is desired. Naive Bayes, although effective, may require additional tuning or feature engineering to improve its detection capabilities, particularly in distinguishing Human text from AI-generated content.

TABLE II
ACCURACY SCORES FOR ADVANCED MODELS

Model	Normal Training	Adversarial Training
XLNet	99.18	95.25
GPT-2	99.00	95.13
RoBERTa	98.53	95.17
BERTurk	97.8	97.6
BERT	96.5	96.4
DistilBERT	96.16	96.16
Electra	94.74	94.5

In normal training conditions, XLNet emerges as the top performer with an accuracy of 99.18%, closely followed by GPT-2 at 99.00%. These results demonstrate the strong capabilities of these models in understanding and classifying text with high accuracy. RoBERTa also performs admirably with an accuracy of 98.53%, indicating its robustness in text classification tasks. BERTurk, a model tailored for the Turkish language, achieves a commendable accuracy of 97.8%, showcasing its effectiveness in handling Turkish text specifically.

When it comes to adversarial training, which introduces challenges designed to test the models' robustness, the performances generally drop. However, BERTurk proves to be the most resilient, with only a slight decrease in accuracy from 97.8% to 97.6%. This suggests that BERTurk's specialized training on Turkish data helps it maintain high performance even under adversarial conditions.

On the other hand, XLNet experiences the most significant drop, from 99.18% to 95.25%. While it remains effective, the drop indicates that XLNet might be more sensitive to adversarial perturbations compared to other models. Similarly, GPT-2 also shows a considerable decline in performance, dropping from 99.00% to 95.13%, which suggests that despite its strong capabilities in generative tasks, it might struggle more with adversarial examples.

RoBERTa also sees a noticeable reduction in accuracy under adversarial training (from 98.53% to 95.17%), but it remains relatively robust compared to XLNet and GPT-2. The consistency of DistilBERT is notable as it maintains the same accuracy (96.16%) across both training scenarios, which might be attributed to its simplified architecture and efficiency.

On the other hand, models like Electra and DistilBERT exhibit relatively minor drops in performance. This resilience might be due to their design philosophies— Electra is trained to distinguish between real and fake data, which might make it more robust to adversarial attacks, as mentioned in Pre-training Text Encoders as Discriminators Rather Than Generators [**?**]. And DistilBERT, being a distilled version of BERT, might have inherent resistance due to its simplified architecture.

When comparing advanced models like BERTurk, RoBERTa, GPT-2 and XLNet to baseline models such as Logistic Regression, Naive Bayes, and SVM, the difference in performance is evident. Advanced models, particularly GPT-2 and XLNet, achieve significantly higher accuracy, with XLNet reaching an accuracy of 99.18% under normal training conditions. This is a clear improvement over the baseline models, where the best-performing baseline, SVM, achieves an average accuracy of 96.90%.

In summary, the analysis clearly shows that advanced models outperform traditional baseline models by a significant margin, making them better suited for complex text classification tasks like distinguishing between AI and Human-generated content. However, the introduction of adversarial training highlights the vulnerability of even the most sophisticated models. While GPT-2 and RoBERTa excel under normal conditions, they are more susceptible to performance drops when faced with adversarial inputs. This suggests that while advanced models provide substantial accuracy gains, they also require robust adversarial defences to maintain their performance in real-world applications where adversarial attacks might be prevalent. Ultimately, the choice of model should consider both the baseline performance and the potential impact of adversarial scenarios, depending on the application's requirements.

IV. Conclusion

This study aimed to detect AI-generated complaint reviews using a comprehensive approach that includes both traditional machine learning models and advanced neural network models. The baseline models employed in this research were Naive Bayes, Logistic Regression, and Support Vector Machines (SVM), which provided foundational performance benchmarks. Advanced models such as BERTurk, BERT, RoBERTa, GPT-2, XLNET, Electra, and DistilBERT were also trained and evaluated, both with and without adversarial training. The experimental setup, inspired by the TuringBench framework [1], involved training a wide array of models to assess their utility in distinguishing between human-written and AI-generated user complaints. The inclusion of adversarial training, as detailed in the work by Hee Lee and Jang [2], aimed to enhance the robustness and accuracy of the advanced models against sophisticated AI-generated texts.

The advanced models generally outperformed the baseline models in detecting AI-generated text. Metrics such as accuracy and F1-score were significantly higher for the advanced models, indicating their superior capability in handling the complex nuances of AI-generated language.

Adversarial training did not make much improvement on top of the performances of the advanced models. Models trained with adversarial examples demonstrated that the dataset can be robust to adversarial attacks but also did not outperform the advanced models trained with normal data.

The comparative analysis revealed that models like BERT and RoBERTa, when adversarially trained, achieved the highest performance metrics. This aligns with the findings from the referenced articles, underscoring the effectiveness of adversarial fine-tuning in improving model resilience and accuracy.

The results validate the hypothesis that advanced neural network models, particularly when enhanced with adversarial training, significantly outperform traditional machine learning models in detecting Turkish AI-generated text. The frameworks and methodologies adopted from TuringBench and the work by Hee Lee and Jang provided a robust foundation for this research. Future work will focus on further refining these models, exploring additional adversarial training techniques, and expanding the dataset to include more diverse and sophisticated AI-generated user complaints to protect both the new customers and supplier.

By integrating advanced detection techniques and rigorous training methodologies, this study contributes to the ongoing efforts in developing reliable systems for AI-generated text detection, ensuring the integrity and authenticity of online content.

Future research will focus on expanding the dataset to include a wider variety of AI-generated user complaints, incorporating different styles, tones, and languages to improve the generalizability of the detection models. Exploring new adversarial training methods and fine-tuning strategies to further enhance model robustness against more sophisticated and varied adversarial attacks. Implementing the trained models in real-world scenarios to assess their performance in dynamic and unpredictable environments, and making necessary adjustments based on these assessments. Incorporating explainable AI techniques to provide transparency and interpretability of the detection models, helping users understand how decisions are made and increasing trust in the system.

The primary aim of this experiment was to adapt the detection models to Turkish user complaints, providing a safeguard for both sellers and customers against AI-generated comments. By ensuring the authenticity of reviews, this system can help maintain trust and reliability in online marketplaces, protecting stakeholders from the potential negative impacts of synthetic reviews. The link to the presentation: https://youtu.be/Siy7X9KL9fc

References

[1] Uchendu, Adaku & Ma, Zeyu & Le, Thai & Zhang, Rui & Lee, Dongwon. (2021). TURINGBENCH: A Benchmark Environment for Turing Test in the Age of Neural Text Generation. 2001-2016.

[2] D. Hee Lee and B. Jang, ”Enhancing Machine-Generated Text Detection: Adversarial Fine-Tuning of Pre-Trained Language Models” in IEEE Access, vol. 12, pp. 65333-65340, 2024

[3] Xiaobing Sun, Xiangyue Liu, Jiajun Hu, and Junwu Zhu. 2014. Empirical studies on the NLP techniques for source code data preprocessing. In Proceedings of the 2014 3rd International Workshop on Evidential Assessment of Software Technologies (EAST 2014)

[4] Thirumoorthy, K., Muneeswaran, K. Feature Selection for Text Classification Using Machine Learning Approaches. Natl. Acad. Sci. Lett. 45, 51–56 (2022).

[5] Sch¨olkopf, Bernhard, and Alexander J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

[6] Rennie, Jason D., et al. ”Tackling the poor assumptions of naive bayes text classifiers.”Proceedings of the 20th international conference on machine learning (ICML-03). 2003.

[7] Ramos, Juan. ”Using tf-idf to determine word relevance in document queries.” Proceedings of the first instructional conference on machine learning. Vol. 242. No. 1. 2003.

[8] Brown, Peter F., et al. ”Class-based n-gram models of natural language.” Computational linguistics 18.4 (1992): 467-480.

[9] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

[10] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

[11] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv preprint arXiv:1906.08237.

[12] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI GPT-2.

[13] Güneş, O., Can, T., & Can, O. A. (2020). BERTurk: A Pretrained Turkish BERT Model. arXiv preprint arXiv:2004.03323.

[14] Powers, David MW. ”Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.” arXiv preprint arXiv:2010.16061 (2020).

[15] Wu, Junchao, et al. ”A survey on llm-gernerated text detection: Necessity, methods, and future directions.” arXiv preprint arXiv:2310.14724 (2023).

[16] Guerrero, Jesus, and Izzat Alsmadi. ”Synthetic text detection: Systemic literature review.” arXiv preprint arXiv:2210.06336 (2022).

[17] https://huggingface.co/datasets/kmkarakaya/turkishReviews-ds

[18] CLARK, Kevin, et al. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555, 2020.