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 machine-generated texts. This project focuses on the detection of AI-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**

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 project 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. Literature Review

In this field of Natural Language Processing, there is a notable lack of prior projects focusing on the Turkish language. Moreover, there are no existing datasets containing AI-generated Turkish text, necessitating manual data creation. 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 project.

Data cleaning and preprocessing are pivotal to the success of natural language processing (NLP) projects, 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 [4]. 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.

Given the limited size of Turkish AI-generated datasets in NLP, data augmentation plays a critical role in enhancing model performance by artificially increasing the volume and diversity of training data. According to a comprehensive survey of data augmentation techniques for NLP featured in a survey, methods such as synonym replacement, random insertion, and back translation are effective in creating robust models [5]. These techniques can be particularly useful in the context of AI-generated complaint reviews in Turkish, where the AI-generated dataset is initially small. By applying these augmentation strategies, the dataset can be expanded and diversified, providing a broader linguistic scope for the model to learn from, thus improving its ability to generalize across unseen data.

The significant disparity in dataset sizes between AI-generated and human-written texts poses a challenge, as it can lead to model bias towards the more extensively represented class. The article on best practices in data augmentation suggests that balancing these classes through sampling methods is essential for unbiased model training [6]. Techniques such as down-sampling the larger dataset or up-sampling the smaller dataset can help achieve a balance, ensuring that the model does not overfit to the characteristics of the more dominant class. This approach is critical in the detection of AI-generated texts, where equal representation of both classes in training data helps improve the model's accuracy and fairness.

By integrating data cleaning, preprocessing, augmentation, and balanced sampling techniques, the project can be effectively carried out. These methods collectively enhance the dataset's quality and diversity, which is crucial for training robust models capable of accurately classifying texts in real-world applications.

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. SVMs are widely used in text classification for their ability to handle high-dimensional data, such as text, and for their robustness in various linguistic contexts. Their effectiveness in binary classification tasks makes them particularly suitable for distinguishing between human-written and AI-generated texts. Research indicates that SVMs, when paired with appropriate kernel functions, can efficiently manage non-linear data separations, making them an excellent choice for initial baseline models in NLP tasks [7].

The Naive Bayes classifier is another fundamental model used in NLP due to its simplicity and speed in handling large datasets. It operates based on Bayes' Theorem, with the assumption of independence among predictors. Naive Bayes has been effectively utilized in text classification, particularly in spam detection and sentiment analysis, demonstrating substantial efficacy in environments where the independence assumption holds reasonably true despite real-world violations [8].

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.

TF-IDF is a statistical measure used to evaluate the importance of a word to a document in a collection or corpus. It increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. TF-IDF vectorization is beneficial for datasets where textual data vary significantly in terms of information redundancy and specific term relevance [9].

N-grams are contiguous sequences of n-items from a given sample of text. By using n-grams, one can capture not only the presence of individual words (unigrams) but also the contextual continuity within text data, such as bigrams (pairs of words) or trigrams (triplets of words). This method is particularly useful for understanding the context and semantic relationships in sentences, which is crucial for tasks like AI text detection, where subtle nuances can be the key to classification [10].

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 project progresses, these models and techniques can be iteratively refined and potentially augmented with more advanced neural approaches based on the initial insights gained from these traditional methods.

Evaluating models effectively in the context of significantly imbalanced datasets, such as those involving a smaller proportion of AI-generated text compared to human-written text, is crucial for assessing the true performance of NLP systems. The approach of using repeated random subsampling, also known as Monte Carlo Cross-Validation [11], provides a robust framework for this evaluation. This method is especially advantageous in scenarios where one class significantly outnumbers another because it allows for the comprehensive evaluation of the model across various subsets of the data, reducing the likelihood of biased training or overfitting to particular samples. Each subset is used to train the model, which is then tested on a separate set. This process is repeated multiple times, and the evaluation metrics are averaged to obtain a more reliable estimate of the model's performance. The use of this method in text classification tasks has been discussed in detail by researchers like Picard and Cook, who highlighted its effectiveness in providing stable and generalizable error estimates, particularly in situations with imbalanced classes. This method not only ensures that each sample is representative but also mitigates the risk of the model's performance being skewed by any particular idiosyncrasies in the data. 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.

Precision and recall are particularly crucial in the context of detecting AI-generated text, as it is essential not only to accurately identify such text (precision) but also to ensure that as many instances of AI-generated text as possible are detected (recall). The F1 Score, being the harmonic mean of precision and recall, provides a single metric that balances both concerns, especially useful when equal importance is placed on both precision and recall [12].

In summary, employing Monte Carlo Cross-Validation for model evaluation ensures a comprehensive and balanced assessment strategy, particularly effective for the imbalanced nature of the dataset. Coupled with the use of precision, recall, and the F1 Score, this methodology provides a robust framework for accurately gauging the performance of baseline models in distinguishing between human and AI-generated texts. This approach not only aligns with established statistical practices but also enhances the reliability of the outcomes in the field of NLP.

III. Related Work

A. Datasets

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, three of OpenAI's GPT Language models—GPT-3.5, GPT-4, and GPT-4o—were utilized, with each model producing 100 reviews. The prompts used to generate these texts were included within the dataset. Due to the small initial size of the AI-generated dataset, augmentation techniques are applied such as synonym replacement, random insertion/deletion/swapping. Finally, augmentation expands the dataset to approximately 6000 entries. After augmentation some preprocessing methods also are applied to such as tokenization, lowercasing, stopword removal, stemming.

For the human-written component, an existing dataset [3] 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.

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 a dataset specifically comprising user reviews, which could be more practically relevant for both companies and consumers.

B. 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 hyper-parameter optimization is necessary, though it has not yet been conducted. The outcomes from the preliminary tests and their comparative analysis are discussed in the Initial Results section.

C. Frameworks to be Utilized (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 project, 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. 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.

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 project 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, it will be needed to train or fine-tune the models 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 project 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 project 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 project, 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 might involve pre-training the model on a large corpus of Turkish texts or using a multilingual model that includes Turkish in its training data. So, there might be a need for a big Turkish dataset extra from the human-AI dataset that is generated for other models.

The adversarially fine-tuned model will be trained using both the original and adversarially modified Turkish complaint review texts. This training will 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 will be used. Additionally, considering the creation of a validation set that reflects the real-world distribution of human and AI-generated texts will be essential for assessing the model's practical effectiveness.

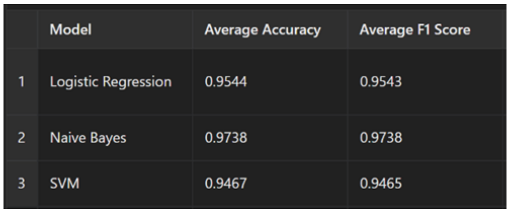
Based on initial results, to better capture the subtleties of AI-generated texts in the context of Turkish complaint reviews, the adversarial examples and fine-tuning processes will be iteratively refined. By utilizing the adversarial fine-tuning approach, this project can significantly advance the detection 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.

IV. Initial Results

We trained three base models: SVM, Logistic Regression, Naïve Bayes. While SVM and Naïve Bayes are trained with tf-idf measurements, Logistic Regression is trained with n-gram. Since we have a small AI dataset, we used an approach to improve generalization of scores. We created 10 different models for each and saved all scores, after that took average of those scores. For each iteration, augmentation and random sampling operations are repeated so that each model is unique.

Naïve Bayes outperformed SVM and Logistic Regression in the final results. The scores for each model are shown in Fig. 1. and Fig. 2.

The confusion matrices for Logistic Regression, Naive Bayes, and SVM models highlight interesting differences in how each model performs on the AI and Human text classification task.



	Model	Average Accuracy	Average F1 Score
1	Logistic Regression	0.9544	0.9543
2	Naive Bayes	0.9738	0.9738
3	SVM	0.9467	0.9465

Fig. 1. Accuracy and F1-score of the Models

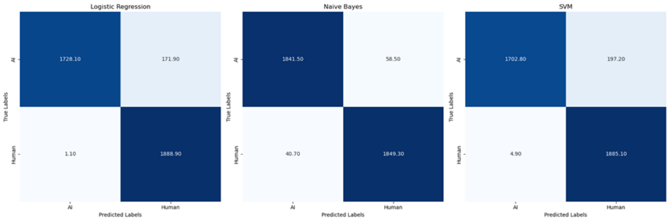


Fig. 2. Confusion Matrix for all Models

Naive Bayes demonstrates a balanced performance with strong detection capabilities for both AI and Human texts, making it a robust choice for this classification task.

Logistic Regression and SVM, on the other hand, excel in detecting Human texts but are less effective in identifying AI texts. This could be advantageous in scenarios where false positives (Human texts misclassified as AI) are more critical to avoid.

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

[4] 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)

[5] Steven Y. Feng and Varun Gangal and Jason Wei and Sarath Chandar and Soroush Vosoughi and Teruko Mitamura and Eduard Hovy, “A Survey of Data Augmentation Approaches for NLP”, 2105.03075, 2021

[6] Bayer, M., Kaufhold, MA., Buchhold, B. et al. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. Int. J. Mach. Learn. & Cyber. 14, 135–150 (2023)

[7] Schölkopf, Bernhard, and Alexander J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

[8] 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.

[9] 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.

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

[11] Picard, Richard R., and R. Dennis Cook. “Cross-Validation of Regression Models.” Journal of the American Statistical Association, vol. 79, no. 387, 1984, pp. 575–83. JSTOR

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