KorvexChecker: A BERT-Based Verification Framework for LLM Outputs in Turkish

Mehmet Göl[1], Savaş Öztürk[1], Uğur Dursun[2]
[1]Marmara University, Electrical and Electronics Eng. Department, Turkey
[2]Istanbul Law Office, Turkey
E-mail: mgol@marmara.edu.tr, savas.ozturk@marmara.edu.tr, ugurdursun@gmail.com

Keywords: large language models (LLMs), Turkish NLP, LLM validation, BERT fine-tuning, low-resource languages, hallucination detection

Received: February 23, 2025

Large Language Models (LLMs) have demonstrated exceptional capabilities in generating human-like text, yet their outputs often suffer from critical issues such as hallucination, ethical violations, and lack of meaningfulness. These challenges can result in misinformation, offensive language, and unreliable outputs, especially in niche languages like Turkish. To address these concerns, we introduce KorvexChecker, a modular and multi-functional tool for validating LLM outputs in terms of ethical compliance, meaningfulness, and hallucination detection. KorvexChecker employs a combination of fine-tuned BERT models for sentiment analysis, ethical violation detection, and text classification to identify whether a given text requires further verification. The tool integrates various external verification resources, such as Web Search API for factual accuracy and journal database for academical document validation. Furthermore, Zemberek, a Turkish NLP library, is utilized for preprocessing, normalization, and segmentation of Turkish text. The system was evaluated on a custom-labeled dataset of 7552 Turkish samples, divided into training, validation, and test sets. Experimental results demonstrate that the BERT-based classifiers achieved an average F1-score of 93.3% across multiple tasks, significantly outperforming both rule-based methods and traditional machine learning baselines. A small-scale user test showed the framework to be responsive and effective in real-time scenarios, with an average inference latency of 800 milliseconds per query. By leveraging both advanced machine learning techniques and external validation mechanisms, KorvexChecker offers a robust framework for ensuring the trustworthiness of AI-generated text, with potential applications in academia, media, and content moderation.

Povzetek: Študija predstavlja orodje KorvexChecker za preverjanje zanesljivosti besedil, ki jih ustvarijo veliki jezikovni modeli, zlasti v turščini, pri čemer z uporabo modelov BERT in zunanjih virov učinkovito zaznava halucinacije, etične kršitve in nesmiselne vsebine.

1 Introduction

Large Language Models (LLMs) have significantly advanced natural language processing (NLP) by enabling human-like text generation. However, their outputs frequently suffer from hallucinations, bias, misinformation, and ethical concerns, raising substantial reliability and trustworthiness issues. These limitations become even more pronounced in low-resource languages, where pre-trained models often lack sufficient coverage and linguistic adaptability. Addressing these challenges requires a multi-layered validation approach that combines ethical evaluation, hallucination detection, and sentiment analysis, particularly for languages like Turkish.

One of the most critical concerns in LLM research is hallucination detection, as models frequently generate fabricated, unverified, or logically inconsistent statements. Studies have demonstrated that statistical uncertainty estimators, such as entropy-based methods, can effectively detect confabulations in LLM-generated outputs by analyzing semantic inconsistencies rather than simple token-level discrepancies [1]. Another promising approach involves augmenting LLMs with external knowledge sources, such as knowledge graphs, to enhance factual consistency. Beyond hallucinations, bias in LLMs remains a major ethical concern. Research has shown that pre-trained models inadvertently propagate social and cultural biases, particularly in educational contexts [2]. These biases not only impact the trustworthiness of AI-generated content but also raise critical concerns regarding the ethical deployment of LLMs in academia, journalism, and public discourse [3]. To address these issues, KorvexChecker incorporates sentiment and ethical compliance evaluation methods to identify biases, offensive content, and ethically problematic statements.

A growing body of research highlights the limitations of current open-access LLMs in handling Turkish, emphasizing the need for language-specific adaptation

strategies [4]. Recent efforts have focused on creating large-scale Turkish datasets, fine-tuning existing models, and improving training methodologies to enhance LLM performance in low-resource linguistic environments [5]. Additionally, synthetic dataset generation through machine translation and adaptation of English corpora into Turkish has shown promise in improving few-shot and zero-shot learning performance.

Another critical task in Turkish Natural Language Processing (NLP) is sentiment analysis, which presents unique challenges due to Turkish's agglutinative structure and rich morphology. Agglutinative languages, such as Turkish, use extensive suffixation to convey grammatical meaning, leading to highly complex word forms that traditional machine learning (ML) approaches struggle to handle effectively. Conventional sentiment classification methods, such as Term Frequency-Inverse Document Frequency (TF-IDF) combined with Support Vector Machines (SVM), have shown limited effectiveness because they rely on word frequency and statistical patterns rather than deeper linguistic understanding [6]. However, recent advancements in deep learning (DL) have significantly improved sentiment analysis performance. Models such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) can capture contextual dependencies within sentences, making them more suitable for sentiment classification tasks. Additionally, pre-trained transformer-based models, including Bidirectional Encoder Representations from Transformers (BERT), Distilled BERT (DistilBERT), and Efficiently Learning an Encoder that Classifies Token Replacements Accurately (Electra), have outperformed traditional methods by leveraging context-aware word embeddings and self-attention mechanisms. Studies have shown that fine-tuned BERT models achieve the highest classification accuracy in Turkish sentiment analysis, effectively addressing the challenges posed by morphological complexity and sparse datasets [7]. TurkishBERTweet, a transformer-based model specifically trained on over 894 million Turkish tweets, has demonstrated state-of-the-art results in sentiment analysis and hate speech detection, surpassing existing alternatives such as BERTurk while offering significantly lower inference time [8].

Recent advances in NLP have emphasized the challenges of low-resource languages and AI-generated content validation. In [9] addressed sentiment analysis in Algerian dialectal texts, combining traditional machine learning (SVM, Naive Bayes variants) with deep learning (CNN, RNN) on a dataset of 11,760 social media comments. Their work introduced Word2Vec-based semantic enrichment (Skip-Gram/CBOW) and achieved 84.21% accuracy with Multinomial Naive Bayes on Latin-transcribed texts. However, their deep learning models underperformed (~65%), revealing limitations in handling morphologically complex dialects. While their preprocessing pipeline for hybrid scripts (Arabic/Latin) offers insights into low-resource language challenges, their focus remained limited to sentiment classification rather than holistic AI output validation.

In parallel, [10] investigated AI-generated fake news detection through their ERAF-News dataset and DuSTraMo, a dual-stream transformer model. Their study demonstrated RoBERTa and DistilBERT's efficacy (>98% accuracy) in distinguishing human/AI-generated content. Notably, their survey of 83 participants highlighted users' inability to differentiate AI-generated news, underscoring the urgency of automated verification tools. However, their work focused narrowly on fake news detection, omitting broader ethical and hallucination-related risks inherent to LLMs.

Ensuring that LLMs do not generate harmful or offensive content is crucial, particularly in social media environments, where hate speech and misinformation can have widespread consequences. Prior research has demonstrated that hate speech detection models trained on English datasets often fail to generalize to low-resource languages, necessitating language-specific dataset creation and model adaptation [11]. Studies have proposed automatic labeling methods to generate large-scale Turkish hate speech datasets, with BERT-based classifiers outperforming traditional ML approaches.

Additionally, researchers have investigated whether AI-generated text detection tools can reliably distinguish between human-written and LLM-generated content [12]. Their findings indicate that most existing AI-detection tools exhibit a strong bias towards misclassifying AI-generated text as human-written, raising concerns about their effectiveness in academic integrity enforcement. Beyond text classification and sentiment analysis, LLMs have demonstrated potential in software testing, particularly in automated test case generation and program debugging [13]. However, the reliability of LLM-generated test cases remains questionable, reinforcing the need for robust verification frameworks to validate LLM-generated outputs in diverse domains.

Moreover, LLMs have been applied in author profiling tasks, where they classify author characteristics such as gender and personality traits based on textual features [6]. These techniques have been widely used in forensic linguistics, marketing, and online fraud detection, showcasing the versatility of NLP models in different domains.

Studies have further emphasized the importance of reliable validation frameworks in NLP. In [14] conducted a comprehensive review of deep learning-based sentiment analysis techniques, highlighting the superiority of transformer models in handling linguistic complexity. In [15] demonstrated the effectiveness of fine-tuned BERT models for author profiling in Arabic, a morphologically rich low-resource language similar to Turkish. Sabir et al. explored sentiment classification on ChatGPT outputs using SVM, underlining the growing need to evaluate LLM-generated content [16]. Additionally, Tsani et al. (2023) showed that combining BERT and RoBERTa can yield strong performance in personality detection from social media data [17]. These studies support the design choices made in this work, which adopts a BERT-based, modular validation framework focused on Turkish LLM outputs.

Table 1: Comparative summary of related work

Study	Language	Dataset	Task	Method	Accuracy
Bsir et al. (2024)	Arabic	PAN 2018 (2,400 authors)	Author Profiling (Gender)	Fine-tuned AraBERTv2-large	79.7% Accuracy
Sabir et al. (2024)	English (ChatGPT Tweets)	Custom Tweet Dataset	Sentiment Analysis	TF-IDF + SVM	96.4% Accuracy
Etaiwi et al. (2021)	Multilingual	Various (Survey)	Sentiment Analysis	CNN, RNN, BERT, LSTM	-
Tsani et al. (2023)	Multilingual (Social Media)	Personality Corpus	Personality Profiling	Ensemble BERT + RoBERTa	0.741 F1
A. C. Mazari and A. Djeffal (2022)	Algerian Arabic	11,760 posts	Sentiment Analysis	Naive Bayes, CNN, RNN	84.2% (NB), ~65% (DL)
H. Moalla et al. (2024)	English	ERAF-News	AI-generated text detection	RoBERTa / DuSTraMo	>98% Accuracy
Kurt, M. S., & Yücel Demirel, E., (2023)	Turkish (translated)	Auto-labeled dataset	Hate Speech Detection	BERT	~86%
KorvexChecker	Turkish	7552 annotated samples	Hallucination, Ethical Compliance, Repetition, Context	Fine-tuned BERT + Rule-based + Web Search + Zemberek	Average 93.3% F1

A comparative summary of related studies—highlighting their datasets, methods, and limitations—is presented in Table 1, showing the relative lack of multi-dimensional verification systems in the context of Turkish LLM output analysis.

To address these interrelated challenges, there is a need for a unified validation system that can operate in real time, adapt to the morphological complexity of Turkish, and assess outputs from both linguistic and ethical perspectives. The methodology proposed in this study directly targets these requirements by combining transformer-based classifiers with rule-based and external verification mechanisms. This approach not only enables fine-grained classification of ethical violations and hallucinations in Turkish texts but also introduces domain-specific adaptation through preprocessing tools and curated datasets.

The innovative contribution of this study lies in the design of KorvexChecker, a modular and language-specific framework for validating large language model (LLM) outputs, with a particular focus on Turkish. KorvexChecker addresses key validation challenges such as hallucination detection, ethical compliance, sentiment consistency, and output meaningfulness through a verification-oriented pipeline rather than generative modeling. The framework integrates external knowledge sources for factual and academic validation, including web-based verification mechanisms and scholarly databases, to assess the reliability of factual claims. Turkish-specific preprocessing and sentence segmentation are handled using Zemberek, enabling robust processing of the language's rich morphological structure. Furthermore, transformer-based classification models are employed for tasks such as sentiment analysis and ethical content evaluation. By combining these components into a transparent and extensible architecture, KorvexChecker provides a systematic solution for improving the trustworthiness and testability of LLM outputs in Turkish-language applications.

2 Methodology

2.1 System architecture

Recent advancements in Large Language Models (LLMs) have significantly improved natural language understanding and text generation capabilities. However, the outputs generated by these models often lack consistency in terms of factual accuracy, ethical compliance, and linguistic coherence. This study introduces a comprehensive evaluation framework for assessing the performance of Turkish LLMs, focusing on three critical aspects: meaningfulness, hallucination, and ethical compliance. The system architecture is designed to evaluate outputs generated by large language models (LLMs) across three key dimensions: meaningfulness, hallucination, and ethical compliance. The framework comprises a modular pipeline that integrates multiple APIs, each tailored to a specific evaluation task.

Before fine-tuning BERT models, we experimented logistic regression classifiers. Although this solution offered faster training and inference, average F1-score remained below 75% for both hallucination and ethical classification tasks.

For all classification tasks in this study, we fine-tuned the dbmdz/bert-base-turkish-cased model, also known as BERTurk, which follows the original BERT-base architecture but is pre-trained entirely on Turkish data [18]. The model was developed by the MDZ Digital Library team at the Bavarian State Library and trained using a 35GB Turkish corpus including the OSCAR corpus, Turkish Wikipedia, multiple OPUS corpora, and a

special dataset curated by Kemal Oflazer. This pretraining corpus contains over 44 billion tokens and was trained for 2 million steps on a TPU v3-8 provided by Google's TensorFlow Research Cloud (TFRC). Due to its linguistic alignment with Turkish morphology and syntax—particularly casing and agglutination—BERTurk offers significant advantages over multilingual BERT variants or models trained on non-specific data. These properties made it a suitable and linguistically sound foundation for KorvexChecker's classification tasks. Compared to classical NLP approaches and even some other transformer-based models, BERT provides superior performance in text classification and semantic analysis. Traditional word embedding models such as Word2Vec and GloVe create static word representations, meaning the same word will always have the same embedding regardless of context. However, BERT is context-aware, thanks to its bidirectional attention mechanism. This is particularly useful in Turkish, where the meaning of a word often depends on the surrounding words due to rich morphology and flexible syntax. While other Transformer-based models, such as GPT, T5, or XLNet, also provide strong text-processing capabilities, they are primarily optimized for generative tasks rather than classification. In contrast, BERT was specifically designed for text classification and semantic similarity tasks, aligning perfectly with our evaluation goals.

Meaningful API: Determines the linguistic and contextual coherence of the text. Hallucination API: Assesses factual correctness by cross-referencing external resources and academic databases. Ethical API: Detects content that violates ethical standards, such as hate speech, misinformation, or harmful language. Each API is powered by fine-tuned BERT models and utilizes pre-trained tokenizers to ensure efficient processing. The architecture allows seamless integration with external data sources for validation, enabling a holistic assessment of LLM outputs.

The pipeline follows a sequential process: Text input is normalized using natural language processing tools and corrected for any spelling errors. The normalized text is passed through each API for evaluation. Outputs from the APIs are aggregated into a final report, detailing the quality and compliance of the text.

To enhance user experience and provide seamless interaction with large language models (LLMs), a graphical user interface (GUI) has been integrated into the system. Instead of relying solely IDE or bash, the framework now allows users to run models locally through LM Studio, an open-source platform for executing LLMs on personal hardware. This addition provides flexibility and ensures a smoother workflow for evaluating AI-generated content.

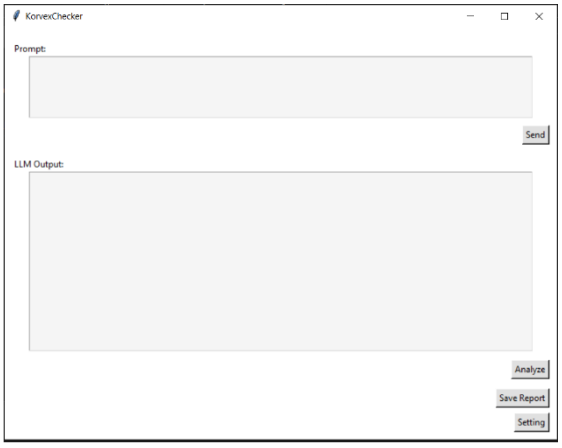

Figure 1: System GUI

Figure 1 illustrates the graphical user interface (GUI) of KorvexChecker designed for interactive evaluation of large language model (LLM) outputs. The interface allows users to input a prompt and view the generated response produced by a locally running LLM. Model selection is performed through the Settings panel, where users can choose among available models hosted in LM Studio, enabling seamless switching between different local LLMs without restarting the application.

After the LLM generates an output, users can initiate analysis directly from the interface. KorvexChecker evaluates the generated text across multiple dimensions, including meaningfulness, hallucination-oriented verification, and ethical compliance. The analysis results are presented within the same interface, providing an immediate and interpretable assessment of the model's behavior.

The GUI also supports saving analysis results for reporting purposes, facilitating systematic comparison of different models and prompts. By integrating local model execution with evidence-grounded verification and configurable model selection, the interface serves as a practical environment for testing, benchmarking, and validating LLM outputs.

Figure 2 illustrates the end-to-end workflow of the system, including GUI interaction and model validation.

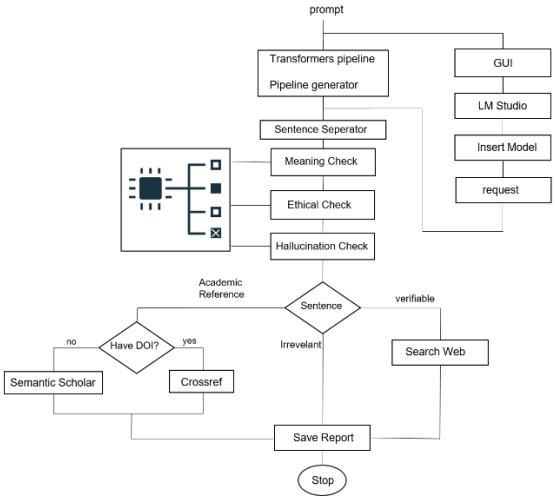

Figure 2: System architecture

The system is implemented in Python and leverages libraries such as PyTorch, Transformers, and Zemberek [12] for Turkish language processing.

2.2. Text preprocessing and sentence segmentation

To ensure robust sentence segmentation in academic and formal texts, a customized preprocessing and sentence extraction pipeline was implemented. The primary objective of this approach is to prevent erroneous sentence boundaries caused by abbreviations, enumerations, academic references, and numerical expressions, which are common sources of segmentation errors in Turkish-language texts. The preprocessing pipeline consists of the following steps:

1. Protection of Enumerations and Reference Patterns
Enumerated items and reference-like patterns (e.g., lettered lists, academic citations, and DOI expressions) are temporarily masked using placeholder tokens to prevent premature sentence splitting triggered by punctuation marks. This masking strategy is particularly effective for preserving reference integrity during segmentation.

2. Preservation of Abbreviations, Dates, and Numerical Expressions
Common abbreviations, parenthesized years, and numerical patterns frequently appearing in scholarly writing are replaced with custom tokens prior to segmentation. This step ensures that punctuation within these structures does not lead to incorrect fragmentation of semantically coherent units.

3. Sentence Segmentation with Turkish-Specific Linguistic Support
Following preprocessing, the modified text is processed using a Turkish-specific sentence segmentation module, which incorporates rule-based linguistic constraints to accurately detect sentence boundaries while accounting for the morphological characteristics of Turkish.

4. Restoration and Post-processing
After sentence extraction, all placeholder tokens are restored to their original forms. Additional post-processing steps are applied to correct formatting artifacts introduced during masking, such as duplicated punctuation or extraneous whitespace, thereby preserving grammatical and structural integrity.

By integrating domain-aware preprocessing with Turkish-specific linguistic rules, the proposed segmentation approach significantly reduces incorrect sentence splits in academic texts. This enhancement directly improves the reliability of downstream validation tasks, including factual verification, ethical assessment, and semantic consistency analysis.

2.3 BERT Models

The framework employs three BERT-based models, each fine-tuned for a specific task. Below, the details for each API and its corresponding BERT model are presented:
Meaningful API: This API evaluates whether the input text is linguistically meaningful and contextually coherent. The BERT model was fine-tuned on a custom dataset containing examples of meaningful and nonsensical sentences. The fine-tuning process involved binary classification, with labels for "meaningful" and "nonsensical." One of the common challenges in AI-generated text, especially with models trained on limited or imbalanced datasets, is the generation of nonsensical or structurally incoherent sentences. These outputs, while grammatically correct at times, often lack semantic consistency or logical flow, rendering them meaningless to human readers.
The Meaningful API addresses this issue by evaluating the semantic integrity and contextual relevance of generated text. It ensures that outputs are not only grammatically sound but also coherent, contextually appropriate, and aligned with the intended purpose of the text.
Hallucination API: The Hallucination API identifies factual inaccuracies by classifying content into three categories: Academic Reference, Historical Claim, and Needs Verification. The BERT model used for this task was fine-tuned on a manually labeled dataset consisting of representative samples for each class. In addition to semantic classification, the API incorporates external validation mechanisms to cross-check factual claims. Specifically, a keyword-based web search is performed, where queries are automatically generated by extracting named entities and temporal expressions (e.g., dates, years) from the input text. The top five search results retrieved from Web search are parsed, and the sentence is considered factually supported if matching evidence (i.e., names, dates, numerical values) appears in any of the retrieved snippets.
Ethical API: The Ethical API detects content violations, including hate speech, misinformation, and other harmful language. The BERT model was trained on a dataset annotated for ethical violations, encompassing multiple categories such as discrimination, abuse, and fraud. Outputs include confidence scores for each violation type, ensuring transparency in the evaluation process. All datasets were split into training (80%), validation (10%), and test (10%) sets using stratified sampling. Preprocessing included sentence normalization, punctuation correction, and tokenization using the BERTurk tokenizer.

The training process for all three BERT models was closely monitored using loss graphs to evaluate model convergence and performance. Below are the training loss graphs for model.

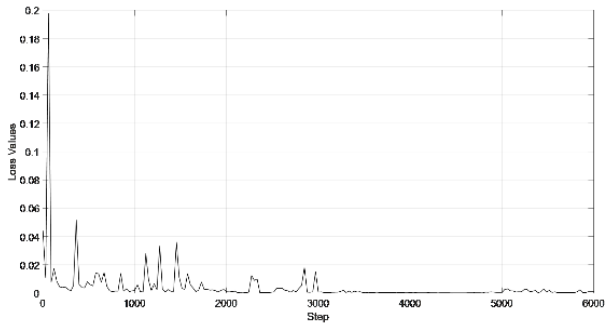

Figure 3: Training and validation loss across epoch 1 to 3

Fig 3 illustrate the training and validation loss curves over multiple epochs for the BERT models used in the KorvexChecker framework. The loss curves demonstrate a consistent decrease in training loss, indicating that the models effectively learned patterns from the training dataset while avoiding significant overfitting.

These results validate the robustness and reliability of the training process, ensuring that the models can generalize effectively to unseen data while maintaining a stable learning trajectory.

Meaningful API: A smooth decrease in loss, indicating effective learning of linguistic patterns.

Hallucination API: A gradual loss reduction with occasional fluctuations, attributed to the complexity of multi-class classification.

Ethical API: A steep decline in initial epochs followed by stabilization, demonstrating the model's ability to learn ethical distinctions.

These graphs illustrate the effectiveness of the fine-tuning BERT process and highlight the models' readiness for real-world applications. All fine-tuning experiments were conducted on a local machine using CPU-based training with the Hugging Face Transformers and PyTorch libraries. Due to computational constraints, the models were trained for 3 epochs with early stopping based on validation loss. The same hyper parameters were used across all three classification tasks, including a batch size of 16, a learning rate of 2e-5, and the AdamW optimizer.

3 Results and discussion

The proposed framework was evaluated for its ability to assess outputs from large language models (LLMs) across three dimensions: meaningfulness, hallucination, and ethical compliance. The results demonstrate that the modular architecture, powered by fine-tuned BERT models and external validation mechanisms, provides reliable and actionable evaluations.

3.1 Meaningfulness analysis

The Meaningful API achieved an accuracy of 96% on the test dataset, effectively distinguishing meaningful sentences from nonsensical ones. Table 2 illustrates the confusion matrix for this classification task, showcasing minimal false positives and false negatives. Qualitative examples highlight the API's robustness:

Table 2: Confusion matrix for meaningfulness

	Predicted: Meaningful	Predicted: Nonsensical
Meaningful	98 (TP)	2 (FN)
Nonsensical	6 (FP)	94 (TN)

The meaningfulness classifier was trained and evaluated on a dataset of 4,713 Turkish text samples manually written and labeled by the author. Each sentence was annotated as either “Meaningful” or “Not Meaningful”, based on criteria such as grammatical coherence, contextual relevance, and informativeness. While meaningful examples were composed as logically sound and syntactically correct sentences, non-meaningful examples were generated using a Python script that randomly combined Turkish sentences to produce grammatically or semantically incoherent outputs. For example, nonsensical sentences such as;

“*Bardak yürürken kitap uyur*” (The glass walks while the book sleeps) or

“*Sandalye hızlıca üzgün fikirleri koştu*” (The chair quickly ran sad ideas) simulate the type of low-quality content sometimes produced by under-trained LLMs. This ensured a clear distinction between coherent and nonsensical inputs, simulating low-quality LLM outputs..

The meaningfulness test results demonstrated strong performance across key evaluation metrics. The system achieved an Accuracy of 96%, indicating that 96% of all sentences were classified correctly. The Precision was 94.2%, showing that 94.2% of the sentences predicted as meaningful were indeed meaningful. Additionally, the Recall reached 98%, highlighting the system's ability to correctly identify 98% of all genuinely meaningful sentences. The F1-Score, representing the harmonic mean of Precision and Recall, was calculated as 96.1%, reflecting a balanced performance between precision and recall. These results suggest that the proposed model effectively distinguishes meaningful sentences from meaningless ones with high reliability and consistency.

3.2 Hallucination detection

The hallucination verification component of KorvexChecker is designed to perform sentence-level, evidence-grounded assessment of factual claims in large language model (LLM) outputs. Rather than producing binary hallucination labels, the system generates explanatory verification results by comparing extracted claims against information retrieved from external web sources and academic databases.

The verification process operates in two complementary stages: general factual verification and academic reference verification.

1. Factual claim verification

Sentences identified as requiring verification are analyzed individually. For each sentence, the system retrieves relevant information from external web sources and compares the generated claim with widely accepted or frequently cited facts. When inconsistencies are observed, the system produces an interpretable explanation highlighting the discrepancy rather than asserting absolute correctness.

LLM Output:
"Einstein won the Nobel Prize in 1925."

System Output: Web verification result: Likely incorrect

Explanation: The claimed year (1925) conflicts with widely cited sources indicating that Albert Einstein received the Nobel Prize in Physics in 1921.

Evidence summary: Multiple web sources report that Einstein was awarded the Nobel Prize in 1921 for his contributions to theoretical physics, particularly the discovery of the photoelectric effect. Sources also indicate that in 1925 he received the Copley Medal, which may contribute to confusion.

This example illustrates how the system provides context-aware explanations grounded in retrieved evidence, enabling users to understand why a claim is flagged rather than receiving a simple warning.

2. Academic reference verification

In addition to general factual claims, KorvexChecker performs structured verification of academic references. When a sentence is classified as an academic citation, the system analyzes the reference components (e.g., author names, publication year, title, venue, and DOI when available).

If a DOI is detected, the reference is automatically directed to authoritative scholarly metadata services such as Crossref for direct DOI-based validation. When a DOI is absent or incomplete, the system queries alternative academic search services, such as Semantic Scholar, using title and author information to identify the closest matching records.

LLM Output: D. Li, D. Zhao, Q. Zhang and Y. Chen, "Reinforcement Learning and Deep Learning Based Lateral Control for Autonomous Driving," 2019.

System Output: Academic verification: FOUND (*semanticscholar*).
Title: Reinforcement Learning and Deep Learning based Lateral Control for Autonomous Driving
Year: 2018
Source: IEEE Computational Intelligence Magazine
DOI: 10.1109/MCI.2019.2901089
URL:https://www.semanticscholar.org/paper/96f2f4792e19de7d24d9fe49b514700b668ff1db

LLM Output: Input Reference: Jun Wang, Li Lu, 'CNN-Based Iterative Decoding for Polar Codes in MIMO Systems: Performance Analysis and Computational Complexity Evaluation,' Informatica 49 (2025) 27–42, https://doi.org/10.31449/inf.v49i33.8431

System Output: Academic validation: FOUND (*crossref*).
Title: CNN-Based Iterative Decoding for Polar Codes in MIMO Systems: Performance Analysis and Computational Complexity Evaluation
Year: 2025
Source: Informatica
DOI: 10.31449/inf.v49i33.8431
URL: https://doi.org/10.31449/inf.v49i33.8431
Description: DOI verified on Crossref.

LLM Output: Smith, J. "Quantum Dreams and Multiverse Realities" Journal of Advanced Physics, 50(2), 101–120 2023.

System Output: Academic verification: NOT FOUND (*semanticscholar*).
Explanation: No matching publication found.

Table 3: Confusion matrix for hallucination

	Predict:Need Verification	Predict: Academic Reference	Predict:Non Relevant
Need Verification	89	0	11
Academic Reference	0	100	0
Non Relevant	14	0	86

Table 3 shows that the model demonstrates high accuracy across all classes, with most misclassifications occurring between "Need Verification" and "Non-Relevant" samples. The Hallucination Detection module achieved high performance across all classes, with a macro-averaged F1-score of 91.7%. The model classified all academic references correctly (F1: 100%), while minor confusion was observed between "Need Verification" and "Non-Relevant" categories. The precision and recall averaged 91.7%, indicating consistent and balanced performance in identifying potentially hallucinated or unsupported content in Turkish LLM outputs.

This distinction is essential for ensuring the accuracy of factual knowledge and maintaining the integrity of references in AI-generated content across both general knowledge domains and academic research contexts.

3.3 Ethical compliance

This evaluation is based on a manually created dataset of 1,228 Turkish text samples written by the authors and categorized into five predefined ethical classes. The annotation was guided by structured labeling rules, covering categories such as illegal or biased responses, profanity, violence, fraud, and ethically safe content. To ensure that the annotation scheme aligns with Turkish legal and cultural norms, a licensed attorney with expertise in digital media law reviewed a representative subset of 100 samples. The feedback received helped refine class boundaries and confirm the legal validity of the labeling framework.

The Ethical Compliance Evaluation section examines the performance of the Ethical API in identifying and categorizing unethical content across five predefined classes: illegal or biased responses, hate speech, violence and harassment, fraud, and safe content.

This evaluation is supported by quantitative metrics, including precision, recall, and a confusion matrix, which collectively illustrate the system's strengths and limitations. The analysis highlights the API's effectiveness in handling explicit violations while underscoring areas requiring further refinement, particularly in distinguishing subtle contextual nuances.

The system evaluates content across five distinct classes, each addressing a specific aspect of ethical compliance:

1. Illegal or Biased Response Detection: Identifies content containing illegal activities or presenting biased perspectives.
2. Profanity and Offensive Language Detection: Flags content containing explicit language or offensive remarks.
3. Violence and Harassment Detection: Detects content promoting violence, harassment, or abusive behavior.
4. Fraudulent Content/ Fraud Detection: Identifies content specifically designed to deceive users with the intent of financial or personal gain, such as scams, phishing attempts, or fraudulent schemes.
5. Safe Content: Represents content that adheres to ethical guidelines without any violations.

Confusion Matrix (Table 4) provides insights into the model's performance across these classes, offering a clear view of how well the system distinguishes between compliant and non-compliant content. The Ethical API achieved an average F1-score of 92.3% across five classes. Class-wise performance was highest for Profanity (F1: 96.1%) and Fraudulent Content (F1: 96.0%), while Illegal or Biased Responses (F1: 86.7%) showed relatively more misclassifications due to semantic overlap with Safe content. Precision and recall averaged 92.1% and 92.8%, respectively, reflecting consistent and balanced classification performance across ethical categories.

Table 4: Confusion matrix for ethical

	1	2	3	4	5
1	82	2	10	0	6
2	1	99	0	0	0
3	3	5	90	0	2
4	0	0	0	96	4
5	3	0	0	0	97

An analysis of Table 2 reveals key insights into the model's strengths and areas for improvement. The model demonstrates exceptional accuracy in detecting Profanity and Offensive Language (Class 2) and Fraudulent Content (Class 4), with minimal misclassifications. Fraudulent content detection rarely overlaps with other classes, except occasionally with Safe Content (Class 5), indicating a strong ability to identify fraudulent patterns distinctly. Similarly, profanity detection achieves near-perfect results, highlighting the model's robustness in recognizing explicit language with clarity.

However, Illegal or Biased Response Detection (Class 1) poses more significant challenges. Overlaps with Violence and Harassment Detection (Class 3) and Safe Content (Class 5) suggest the inherent difficulty of distinguishing nuanced biases in textual content. Class 3, while performing reasonably well in identifying violent or harassing content, occasionally overlaps with profanity detection, indicating semantic similarities between offensive and violent language.

To ensure the reliability and contextual appropriateness of the ethical annotations, the dataset was manually labeled by the authors based on a predefined set of five ethical categories. To strengthen the validity of the annotation framework, a licensed attorney with expertise in digital media law reviewed a representative subset of the labeled data. This legal review confirmed that the categorization and annotation schema were consistent with applicable ethical standards and legal norms in Turkish digital content environments. While formal inter-annotator agreement metrics were not computed, this expert validation provides an added layer of credibility and domain relevance to the ethical evaluation methodology.

In summary, the Ethical API delivers outstanding performance in detecting clear ethical violations, such as profanity and fraudulent content, while encountering challenges in addressing context-heavy biases and subtle overlaps between certain classes. These findings emphasize both the system's strengths and areas for improvement, particularly in refining Illegal or Biased

Content Detection to better handle nuanced textual contexts.

3.4 Evaluation

Table 5 summarizes the core features and performance metrics of KorvexChecker. The table highlights the system's high accuracy, comprehensive coverage of validation checks, and its ability to perform ethical analysis, meaningfulness evaluation, and hallucination detection. Additionally, the system's flexibility for integration into various workflows and compatibility with Hugging Face further underscore its practical utility and accessibility. The open-source nature of KorvexChecker makes it a versatile tool for developers and researchers aiming to enhance LLM output validation processes.

To evaluate the responsiveness of the system, inference latency was measured on 50 representative Turkish text samples using the fine-tuned BERT model. Tests were performed on an Intel i5-1240P CPU with 16 GB RAM. The average inference latency was 800 milliseconds per query, excluding loading time. This indicates that the system is suitable for real-time or near-real-time use in practical content moderation workflows.

Table 5: Performance metrics of KorvexChecker

Metric/Feature	Evaluation
F1 Scores respectively	96.1 – 91.7 - 92.3
Ethical Analysis	✓
Meaningfulness Analysis	✓
Hallucination Detection	✓
Coverage (checks)	3
Integration Flexibility	High
Open Source	✓
Platform	Any python IDE and Bash or GUI
Language Support	Turkish
Hugging Face Compatible	✓

3.5 Limitations

Despite its strengths, the framework has some limitations:

- Dependency on External Resources: The reliance on external tools and the database may limit performance if these resources are unavailable or contain outdated data.
- Language-Specific Features: While Zemberek ensures robust Turkish language processing, the framework may require additional adaptations for other languages.
- Complex Ethical Analysis: Certain nuanced ethical violations may require advanced contextual understanding, which is a challenge for current models.
- The project size is large because it includes BERT models.
-

3.6 Threats to validity

While the proposed framework demonstrates promising results, several factors may affect its validity. These threats are categorized as follows:

3.6.1 Internal validity

The framework depends on external web search results for factual verification, which introduces internal validity risks related to data availability and consistency. Web-based evidence retrieval is performed using SerpAPI, and limitations such as query rate restrictions may constrain large-scale or continuous verification. In addition, variations in search results over time may affect reproducibility. These risks are mitigated by focusing on inconsistency flagging rather than absolute factual correctness.

3.6.2 External validity

Although the framework is optimized for Turkish text analysis using the Zemberek NLP library, its applicability to other languages remains uncertain. Language-specific morphological structures and syntactic variations may require substantial modifications. Consequently, the generalizability of the results to different linguistic settings is limited. Future research should investigate the adaptability of the system across diverse languages by incorporating multilingual LLM models and expanding the scope of analysis.

3.6.3 Construct validity

Ethical assessment remains a challenging aspect of LLM evaluation, as nuanced ethical violations often require contextual interpretation that exceeds the capabilities of current automated models. Transformer-based models may exhibit inherent sensitivity and bias arising from their pre-training data and architectural characteristics, which constitutes a fundamental limitation rather than an issue that can be fully resolved through model refinement alone. Moreover, the identification of misinformation, hate speech, or biased content is inherently subjective and may vary across cultural, social, and ideological contexts.

In this framework, ethical evaluation is therefore treated as a risk-oriented screening task rather than a definitive ethical judgment. Future iterations may incorporate expert-driven annotation and human judge voting as calibration and interpretability mechanisms to improve annotation consistency and contextual understanding, rather than as solutions intended to eliminate model sensitivity or bias.

3.6.4 Model validity

BERT-based models increase computational requirements but do not affect the logical structure of the framework. This limitation may restrict real-time deployment, particularly on resource-constrained systems. Additionally, transformer-based architectures are known to be sensitive to training data distribution and may exhibit biases inherent to pre-trained models. To address this issue, future research should explore lightweight model

alternatives such as DistilBERT or TinyBERT while ensuring robustness through diverse training datasets.

By addressing these validity threats, the framework can be further improved to enhance reliability, generalizability, and computational efficiency.

4 Conclusion

Large Language Models (LLMs) have demonstrated impressive capabilities in generating coherent and human-like text. However, they still face critical challenges such as hallucinations, ethical violations, and semantic inconsistencies—particularly in morphologically rich, low-resource languages like Turkish. These issues pose risks of misinformation, offensive language, and unreliable outputs. In this study, we introduced KorvexChecker, a modular verification framework that integrates BERT-based classifiers, rule-based filtering, and external verification mechanisms to evaluate LLM outputs across ethical, factual, and linguistic dimensions.

Unlike previous work that typically focuses on isolated tasks—such as sentiment analysis, hate speech detection, or fake news identification—KorvexChecker provides a unified framework capable of addressing multiple verification tasks simultaneously (as summarized in Table 1). Experimental results demonstrate its effectiveness, with an average F1-score of 93.3% on ethically annotated Turkish datasets. This project is designed for use in various domains requiring trustworthy AI-generated content, including academia, digital media, and online content moderation platforms.

Our error analysis reveals that most misclassifications occur in ambiguous or culturally sensitive content, particularly when distinguishing implicit biases masked by polite or formal phrasing. For example, statements such as "He speaks very well for someone from that background." may appear neutral or complimentary but contain underlying social bias that often goes undetected by surface-level classifiers. Such cases expose the inherent limitations of BERT-based models in recognizing pragmatic and cultural nuances. While KorvexChecker mitigates some of these issues through rule-based filters and web-based fact-checking, deeper contextual reasoning remains a persistent challenge for future research.

Among Turkish LLMs, YTU Cosmos stands out as the first publicly available, locally trained large-scale model specifically optimized for Turkish. In contrast to multilingual models with partial Turkish support, Cosmos is fully trained on Turkish corpora, making it an essential benchmark for evaluating native-language-specific challenges such as ethical risks and misinformation. Although Cosmos was not directly integrated into this study, we recognize its significance within the Turkish NLP community and identify it as a promising candidate for future comparative evaluations alongside KorvexChecker [19].

We also acknowledge several limitations in the current framework:

- Challenges in interpreting cultural context, sarcasm, or implicit meaning,
- Dependence on external APIs for factual verification,
- Limited support for multi-modal content beyond text.

To address these limitations, our future work will focus on:

- Expanding the framework to additional application domains (e.g., legal, medical),
- Enhancing ethical classification by incorporating multi-modal models (e.g., combining image, video, and text analysis),
- Developing offline and privacy-preserving validation mechanisms to reduce reliance on external services.

KorvexChecker is publicly available on Hugging Face and remains open for community contributions. We invite researchers and practitioners to experiment with the tool, provide feedback, and collaborate in its continued development to advance the trustworthiness and ethical reliability of AI-generated content.

Appendix

Appendix A: How to use

KorvexChecker can be integrated with various AI platforms and frameworks to validate the outputs of language models. Below are two examples showcasing its integration:

1 Using OpenAI API

The following code demonstrates how to use KorvexChecker with OpenAI's GPT models:

```
import openai
from analysis import analysis

openai.api_key = "YOUR_API_KEY"

response = openai.Completion.create(
    engine="text-davinci-003",
    prompt="Einstein ne zaman Nobel Ödülü
kazandı?"
    max_tokens=100,
    temperature=0.7
)
formatted_output =
response.choices[0].text.strip()

analysis(formatted_output)
```

2 Using hugging face transformers

KorvexChecker is compatible with Hugging Face's open source LLM libraries for text generation.

```
from transformers import pipeline
from analysis import analysis

generator = pipeline("text-generation",
model="gpt2")
```

```
response = generator("Einstein ne zaman
Nobel Ödülü kazandı?", max_length=50,
num_return_sequences=1)
formatted_output =
response[0]["generated_text"]

analysis(formatted_output)
```

3 Accessing through LM Studio

KorvexChecker can analyze outputs from local models running in LM Studio. To integrate it:

```
import requests
from analysis import analysis

API_URL =
"http://localhost:1234/v1/chat/completions"
model_name = "llama2-7b-chat"

payload = {
    "model": model_name,
    "messages": [{"role": "user",
"content": "Einstein ne zaman Nobel Ödülü
kazandı?"}]
}

response = requests.post(API_URL,
json=payload).json()
formatted_output =
response["choices"][0]["message"]["content"
]

analysis(formatted_output)
```

Ensure that LM Studio's API Server is enabled in Settings, with open-sources models loaded locally.

4 Accessing KorvexChecker

The tool is available on Hugging Face. Developers can also clone the project and use it locally. For Linux users, we recommend using Environmental Interpreter to ensure smooth installation and execution of the framework. This helps mitigate potential dependency issues and ensures compatibility across different distributions.

```
setx SERPAPI_KEY "enter your serpapi key"

git clone
https://huggingface.co/mehmetgol/KorvexChec
ker

cd KorvexChecker

setup.bat

run.bat
```

References

[1] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal, "Detecting hallucinations in large language models using semantic entropy," *Nature*, vol. 630, pp. 625–630, Jun. 2024, doi: https://doi.org/10.1038/s41586-024-07421-0.

[2] Elham Tajik, "The Ethical Concerns Surrounding GPT in Education," *arXiv*, Jan. 2024, doi: http://dx.doi.org/10.2139/ssrn.4887922.

[3] Zhiheng Xi, Rui Zheng, Tao Gui, "Safety and Ethical Concerns of Large Language Models," *China Natl. Conf. Comput. Linguist.*, pp. 9–16, 2023.

[4] A. Taha Arslan, "Büyük dil modellerinin Türkçe verisetleri ile eğitilmesi ve ince ayarlanması," *arXiv*, vol. 2306.03978, Jun. 2023.

[5] T. Kesgin *et al.*, "Optimizing Large Language Models for Turkish: New Methodologies in Corpus Selection and Training," in *Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/ASYU62119.2024.10757019.

[6] İlhami Sel , Davut Hanbay, "Ön Eğitimli Dil Modelleri Kullanarak Türkçe Tweetlerden Cinsiyet Tespiti," *Fırat Üniversitesi Mühendis. Bilim. Derg.*, vol. 33, no. 2, pp. 675–684, Sep. 2021, doi: 10.35234.

[7] E. Ezin, R. S. Kiziltepe, and M. Karakus, "Using LLMs for Annotation and ML Methods for Comparative Analysis of Large-Scale Turkish Sentiment Datasets," presented at the 2024 9th International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, Oct. 2024, pp. 204–209. doi: 10.1109/UBMK63289.2024.10773485.

[8] A. Najafi and O. Varol, "TurkishBERTweet: Fast and reliable large language model for social media analysis," *Expert Syst. Appl.*, vol. Volume 255, 2024, doi: https://doi.org/10.1016/j.eswa.2024.124737.

[9] A. C. Mazari and A. Djeffal, "Sentiment Analysis of Algerian Dialect Using Machine Learning and Deep Learning with Word2vec," *Informatica*, vol. 46, no. 46, pp. 67–78, Mar. 2022, doi: https://doi.org/10.31449/inf.v46i6.3340.

[10] H. Moalla, H. Abid, D. Sallami, E. Aïmeur, and B. B. Hamed, "Exploring the Power of Dual Deep Learning for Fake News Detection," *Informatica*, vol. 48, pp. 567–594, Apr. 2024, doi: https://doi.org/10.31449/inf.v48i4.5977.

[11] Kurt, M. S., & Yücel Demirel, E., "Türkçe Hakaret ve Nefret Söylemi Otomatik Tespit Modeli," *Veri Bilimi*, vol. 6, no. 1, pp. 61–73, 2023.

[12] Weber-Wulff, D., Anohina-Naumeca, A., Bjelobaba, S. et al., "Testing of detection tools for AI-generated text," *Int. J. Educ. Integr.*, vol. 19, no. 26, Dec. 2023, doi: 10.1007.

[13] Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, Qing Wang, "Software Testing with Large Language Models: Survey, Landscape, and Vision," *arXiv*, vol. 2307.07221v3, Mar. 2024.

[14] W. Etaiwi, A. Awajan, and Dima Suleiman, "Deep Learning Based Techniques for Sentiment Analysis: A Survey," *Informatica*, no. 45, pp. 89–95, Aug. 2021, doi: https://doi.org/10.31449/inf.v45i7.3674.

[15] B. Bsir, N. Khoufi, and M. Zrigui, "Prediction of Author's Profile Basing on Fine-Tuning BERT Model," *Informatica*, no. 48, pp. 69–78, Feb. 2023.

[16] A. Sabir, H. A. Ali, and M. A. Aljabery, "ChatGPT Tweets Sentiment Analysis Using Machine Learning and Data Classification," *Informatica*, no. 48, Dec. 2023, doi: https://doi.org/10.31449/inf.v48i7.5535.

[17] E. F. Tsani and D. Suhartono, "Personality Identification from Social Media Using Ensemble BERT and RoBERTa," *Informatica*, no. 47, pp. 537–544, Mar. 2023.

[18] K. Oflazer, *bert-base-turkish-cased*. [Online]. Available: https://huggingface.co/dbmdz/bert-base-turkish-cased

[19] A. Zeer *et al.*, "Cosmos-LLaVA: Görselle Sohbet Etmek Cosmos-LLaVA: Chatting with the Visual," presented at the 8th International Artificial Intelligence and Data Processing Symposium (IDAP'24), Malatya, Türkiye, Sep. 2024.