Uluslararası Fen Araştırmalarında Yenilikçi Yaklaşımlar Dergisi
International Journal of Innovative Approaches in Science Research, Vol. 10 (2026), e9786
https://doi.org/10.29329/ijiasr.2026.1442.1

Original article

Evaluating Large Language Models for Emotion Classification and Contextual Word Prediction in NLP

Murat Eser *

Department of Information Technology, Çanakkale Onsekiz Mart University, Çanakkale, Türkiye

Abstract

Large Language Models (LLMs) are powerful deep learning models capable of understanding, interpreting, and generating natural language with high accuracy. These models significantly facilitate the processing of large-scale and complex data, providing substantial ease and efficiency in various natural language processing (NLP) tasks, including text classification, sentiment analysis, contextual understanding, and automatic content generation. This research was conducted to evaluate the sentiment analysis and contextual understanding capabilities of LLMs in the NLP domain. The study contributes to the literature by providing an integrated evaluation that assesses LLMs not only for their generative and classification capabilities but also for their ability to maintain and predict semantic integrity. A balanced dataset consisting of five emotion categories was used to test classification and fill-in-the-blank tasks with ChatGPT 5.3, Claude Sonnet 4.6, and Gemini 3.1 Pro models using the zero-shot method. In the classification task, model performance was evaluated using accuracy, precision, recall, and F1-score metrics. The results revealed that the Claude Sonnet 4.6 model demonstrated superior performance by achieving a 99.52% accuracy score. In the fill-in-the-blank task, the semantic similarity between the predicted words, the original words, and the completed sentences were measured using SBERT and cosine similarity. In this task, Gemini 3.1 Pro achieved the highest similarity performance with scores of 0.85 for word similarity and 0.94 for sentence similarity. The findings indicate that the examined LLMs generally exhibited high success in emotion classification and contextual word prediction tasks. Particularly, Sonnet 4.6 performed stronger in classification, while Gemini 3.1 Pro showed greater strength in semantic fill-in-the-blank tasks. These results highlight the potential of LLMs in understanding and completing emotion-bearing texts in everyday language, thereby underscoring their importance in NLP research.

Keywords: Large Language Models, Natural Language Processing, Classification, Semantic Similarity Analysis, Sentiment Classification

Received: 26 March 2026 * **Accepted:** 22 May 2026 * **Published:** 07 June 2026

* **Corresponding author:**

Murat Eser is an lecturer in the Department of Information Technology at Çanakkale Onsekiz Mart University in Çanakkale, Türkiye. His research interests include the Artificial Intelligence, Natural Language Processing, Machine Learning. He has lived, worked, and studied in Çanakkale, Türkiye.
Email: meser@comu.edu.tr

INTRODUCTION

Natural Language Processing (NLP) is a field that integrates linguistic, computational, and machine learning techniques to enable computer systems to comprehensively analyze textual and spoken data for understanding and generating human language. A major challenge in this domain is handling the wide variety of everyday language use, including dialects, slang, and grammatical irregularities. (Annepaka & Pakray, 2025). To overcome these challenges, various architectures such as rule-based approaches, machine learning methods, and deep learning techniques have been developed over time and employed in various language tasks. As models have evolved, the increasing hardware requirements and the growing volume of data they can process have led to a substantially higher demand for training data. For a long period, studies in the field of NLP were conducted using statistical models that relied on predefined grammatical rules and hand-crafted features (Raiaan et al., 2024). These models have generally been used in NLP for tasks such as binary classification. The methods employed in classification tasks perform classification using a predefined set of rules, which typically requires a high level of domain-specific knowledge (Tan & Liu, 2022). Due to their reliance on predefined rule sets and statistical information, the generalization ability of these models remained limited, rendering them insufficient for solving complex language problems (Raiaan et al., 2024). To overcome the limitations of rule-based approaches, machine learning methods emerged that learn to classify input text by recognizing patterns present in the data (Tan & Liu, 2022). These methods require the input to be transformed into feature vectors that models can understand in order to learn from data. Feature vectors enable models to make inferences from the data by manually converting textual inputs into numerical representations. This transformation of text into vectors was performed statistically, primarily based on word frequencies, without incorporating semantic approaches. The performance of machine learning methods is directly related to the ability of the feature vectors to effectively represent the input. To address the complex nature of natural language and provide semantically rich and meaningful representations of textual expressions, word embedding methods such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) were developed. Through these methods, semantic relationships between texts have been modeled and represented with significantly higher efficacy. In addition to word representation techniques, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks—which mitigate limitations such as polysemy, synonymy, sequential dependencies, and the comprehension of contextual information—have been extensively employed in natural language processing tasks. However, these methods remain constrained in capturing the nuanced meanings of words within a text due to challenges such as the inability to sufficiently learn relational meanings in long-range sequences, vanishing gradients, protracted training durations, and limited short-term memory capacity (Raiaan et al., 2024).

To overcome challenges such as semantic proximity between related words, long-range dependencies, and prohibitive training durations, the Transformer architecture—which focuses on the

interrelationships between words and possesses parallel processing capabilities—was proposed for NLP tasks by (Vaswani et al., 2017). At the core of this architecture, the self-attention mechanism calculates the relationships and relative importance of a word within an input sequence with respect to all other words in the same input. Consequently, long-range contextual dependencies between words are captured significantly more effectively (Leon, 2024; Min et al., 2023). The parallelization capability of Transformer models, which represents a significant superiority over other architectures, offers a substantial advantage in terms of training duration by enabling the capture of long-range dependencies in language and facilitating simultaneous training across multiple Graphics Processing Units (GPUs) with a high number of parameters (Raiaan et al., 2024). Consequently, this architecture provides significant advantages in terms of speed when training larger models compared to other architectures. By eliminating the limitations of traditional statistical models as well as RNN and LSTM methods, it has served as the foundational architecture for the development of Large Language Models (LLMs), which possess a far superior understanding of context and produce human-like text in tasks such as machine translation, text summarization, and classification.

LLMs are built upon the Transformer architecture and acquire fundamental language representation capabilities through pre-training on massive text corpora. During this process, the self-attention mechanism enables the effective modeling of long-range dependencies, while in-context learning allows these models to be utilized for novel tasks (Chang et al., 2024). Models constructed with the Transformer architecture undergo pre-training on large-scale datasets to learn general linguistic patterns and representations; subsequently, the parameters acquired during this stage are re-trained on task-specific datasets to enhance model performance and ensure applicability to real-world tasks (Leon, 2024).

Among the core capabilities of LLMs are the ability to generate text with a level of coherence closely approximating human language, perform translations, and execute question-answering tasks (Leon, 2024). Another characteristic feature of LLMs, in-context learning, involves training the models to generate text based on a provided context or prompt. This capability enables LLMs to produce more coherent and contextually relevant responses, making them suitable for interactive human-centric applications (Chang et al., 2024). The capabilities of LLMs include generating human-like text, synthesizing information, performing textual inference, executing question-answering tasks, conducting logical reasoning, writing code, processing diverse data types, handling data across various domains without prior specific training, strategic planning, and establishing causal relationships (Chang et al., 2024; Leon, 2024; Min et al., 2023; Naveed et al., 2025; Shao et al., 2024). Through these capabilities, LLMs offer advantages such as performing content analysis—which typically requires extensive human effort—in short durations, the versatility of a single model across numerous distinct tasks, and the ability to maintain human-like interaction (Annepaka & Pakray, 2025). In addition to these advantages, LLMs possess certain drawbacks, such as the generation of factually incorrect yet plausible-sounding

information (hallucination), inherent bias and ethical concerns, and the requirement for massive datasets and extensive hardware resources during the training phase (Annepaka & Pakray, 2025; Leon, 2024; Naveed et al., 2025).

LLMs offer time-efficiency advantages in numerous fields such as medicine, education, finance, software, and scientific research, achieving levels of success in natural language understanding and generation tasks previously unattainable with classical models (Sindhu et al., 2024). However, alongside these capabilities and opportunities, they harbor significant disadvantages and risks that cannot be overlooked, including societal biases and ethical issues inherent in large datasets, the generation of factually incorrect information presented as meaningful (hallucinations), hardware and energy costs associated with processing trillions of parameters, and data privacy vulnerabilities (Chang et al., 2024; Min et al., 2023; Naveed et al., 2025; Raiaan et al., 2024). Consequently, to safely leverage the potential of LLMs, it is essential to understand their limitations and drawbacks as thoroughly as the advantages they provide.

Song et al. (Song et al., 2025) systematically examined the GPT-3.5, GPT-4.0, and Llama 3.1 70B models to evaluate the statistical programming capabilities of large language models. They presented the models with 207 statistical analysis tasks—developed by the authors from 65 datasets obtained from various open-source repositories—alongside corresponding problem definitions and data descriptions for each task. The output codes generated by the models were evaluated by nine expert reviewers across 10 criteria using a 50-point scale, accumulating a total of 18,630 points. The evaluation results revealed that the three models achieved an average total score reflecting a 74.774% success rate; however, while they attained a 94.052% success rate in code quality assessment, a significant decline was observed in executability (61.202%) and output quality (51.690%). It was observed that GPT-3.5 performed better in code brevity and readability metrics, whereas the Llama model demonstrated superior performance in output accuracy; nevertheless, redundant repetitions and erroneous variable usage were detected in the Llama model during tasks requiring deep domain expertise.

Zhu et al. (Zhu et al., 2024) developed the Chinese Financial Language Understanding Evaluation (CFLUE) benchmark to comprehensively assess the Chinese financial language understanding capabilities of large language models and tested various LLMs in a zero-shot setting. In the knowledge evaluation section, 38K+ multiple-choice questions obtained from 15 different financial proficiency exams were provided with their answers and utilized for answer prediction and chain-of-thought tasks. Additionally, the models were evaluated on five classical NLP tasks—text classification, machine translation, relation extraction, reading comprehension, and text generation—using a total of 16K+ real-world financial examples. The GPT-4, GPT-4-turbo, and Qwen-72B models achieved over 60% accuracy in knowledge evaluation. In the application section consisting of NLP tasks, GPT-4 demonstrated the highest average performance, while Qwen-72B achieved the best results in two out of

the five tasks. Finance-specific models, namely FinGPT, DISC-FinLLM, and Tongyi-Finance, exhibited significantly lower performance in both knowledge and application tasks.

Leon (Leon, 2024) proposes a novel and unified ranking metric called the Comprehensive Language Model Performance Index (CLMPI) to address the lack of standardized evaluation that has emerged with the rise of LLMs such as GPT-4, LLaMA, and PaLM. The proposed metric combines five core dimensions—accuracy, contextual understanding, coherence, fluency, and resource efficiency—into a weighted formula. To demonstrate the mathematical and practical application of this flexible metric, the study performs sample calculations using three hypothetical models designated as LLM-A, LLM-B, and LLM-C, rather than real-world models. Unlike existing fragmented evaluation methods, the holistic and adjustable approach offered by CLMPI targets model performance based on specific task requirements.

Li et al. (Li et al., 2023) propose the Chinese Financial Generative Pre-Trained Transformer (CFGPT) model with the objective of achieving a deep understanding of Chinese financial texts and performing tasks such as stock movement prediction with high accuracy. This framework consists of a dataset containing 584 million documents for fine-tuning, 1.5 million instruction pairs for supervised fine-tuning, and a financial large language model named CFLLM, which was trained using a two-stage process of InternLM-7B-based continued pre-training and supervised fine-tuning. Additionally, the framework includes a deployment module (CFAPP) designed for real-world applications, featuring capabilities such as causal reasoning, price prediction, and risk management to adeptly manage financial texts. The CFLLM-ins-7B model significantly improved zero-shot and few-shot performance in Chinese financial tasks, while CFAPP yielded results that support financial decision-making processes through user-friendly interactions.

Feng et al. (Feng et al., 2024) systematically compared general-purpose models—including GPT-4, GPT-3.5-Turbo, Flan-T5-XXL, Llama-3-8B-Instruct, Yi-1.5-34B-Chat, and Zephyr-7B-Beta—against domain-specific biomedical models such as Medicine-Llama3-8B, Meditron-7B, and MedLLaMA-13B across 13 datasets within the Biomedical Language Understanding and Reasoning Benchmark (BLURB). The evaluation encompassed tasks such as named entity recognition, relation extraction, PICO (population, interventions, comparators, and outcomes), sentence similarity, document classification, and question answering. The models were assessed without any additional training or fine-tuning, utilizing prompting strategies that included short/long instructions, zero-shot/few-shot examples, and random/semantically similar example selection. The study results indicated that GPT-4 achieved the highest overall BLURB score of 64.6, while general-purpose models like Flan-T5-XXL and Llama-3-8B-Instruct demonstrated significant superiority over biomedical models in most tasks. Furthermore, it was observed that prompts constructed by incorporating semantically similar examples to the input text consistently enhanced performance across nearly all tasks and models.

Jiang et al. (Jiang et al., 2024) proposed TriSum, a framework designed to distill the text summarization capabilities of large language models into local models for use in resource-constrained and privacy-oriented environments. The model consists of three stages: in the first stage, document-based aspect-triple rationales and summaries are generated using GPT-3.5; in the second stage, "golden rationales" are selected through a dual-scoring method; and in the third stage, a BART-Large-based local model is trained using a curriculum learning strategy. Evaluations conducted on the CNN/DailyMail, XSum, and ClinicalTrial datasets demonstrated improvements in ROUGE scores by 4.5%, 8.5%, and 7.4%, respectively, compared to baseline models. Furthermore, the use of aspect-triple rationales significantly enhanced the interpretability of the summarization process.

Ma et al. (Ma et al., 2024) propose the domain-specific ImpressionGPT framework, which utilizes ChatGPT as a foundation model for the automated summarization of the "Impression" section in radiology reports. This method constructs context by incorporating reports similar to the test case into a dynamic prompt and subsequently employs an iterative optimization algorithm to automatically evaluate and continuously refine ChatGPT's output. The most significant contribution of the proposed approach, in terms of data and training, is its applicability without requiring additional training or fine-tuning. Evaluated on the Medical Information Mart for Intensive Care - Chest X-ray (MIMIC-CXR) and Open Access Biomedical Image Search Engine (OpenI) datasets, the framework demonstrated a clear superiority over existing methods, producing high-quality and interpretable summaries with a minimal sample size of 5–20 examples.

Xu and Ashley (Xu & Ashley, 2023) developed an approach supported by argumentative segmentation to overcome the input length constraints faced by large language models during the automated summarization of long legal texts and to enhance the quality of the generated summaries. In the employed method, legal decision texts were first grouped according to their semantic coherence using the C99 algorithm and Sentence-BERT and subsequently classified as argumentative or non-argumentative sections based on the presence of "Issue, Reason, and Conclusion" sentences using the LegalBERT model. These identified argumentative sections were then fed into GPT-3.5 and GPT-4 models without any fine-tuning to obtain legal summaries. The findings indicate that the proposed method, reinforced by argumentative segmentation, offers a solution for long texts exceeding context limits. Furthermore, it demonstrated higher performance in ROUGE, BLEU, and BERTScore metrics compared to standard GPT-3.5 and GPT-4 models.

Su and McMillan (Su & McMillan, 2024) propose an approach that automates source code summarization used in software development processes by transferring 2.15 million Java method summaries generated by the GPT-3.5 model to smaller models, such as Jam, StarCoder, and AttenGRU, via knowledge distillation. The proposed method is primarily expected to eliminate dependency on third-party services in code development workflows. By employing fine-tuned models with the dataset across various parameter sizes, it was observed that the values obtained in METEOR and USE scores improved

as the model and data scale increased. Notably, the 350M-parameter Jam model, trained on 2.15M data points, demonstrated performance close to GPT-3.5 while remaining operational on a 16GB GPU, thereby offering a cost-effective alternative that largely reproduces the capabilities of GPT-3.5 while preserving data privacy.

MATERIALS and METHODS

Datasets

In the study, the same dataset was utilized for both classification and fill-in-the-blank tasks, and evaluations were conducted by considering the class distribution balance. The dataset used for the classification and fill-in-the-blank tasks consists of five classes, comprising examples that include the categories of entertainment, fear, trust, surprise, and sadness (Tintin & Yücebaş, 2026). The dataset was reduced for use, with 500 samples selected for each class.

Large Language Models

Large Language Models (LLMs) are deep learning models capable of understanding natural language spoken by humans and generating meaningful responses to inputs or queries originating from diverse sources (Shao et al., 2024). LLMs consist of multi-layered neural networks and possess billions of parameters, resulting in a much more complex structure than traditional neural network architectures. Following the emergence of the Transformer architecture as a turning point, they began to be utilized in natural language processing tasks; today, they are employed in diverse tasks such as text generation, language translation, video synthesis, audio generation, and image creation. During their training, they require massive datasets and hardware resources, necessitating extensive training durations. The process for LLMs begins with pre-training, during which they learn linguistic patterns, structures, and contextual features. LLMs can be trained using various learning strategies during pre-training; these include methods such as Masked Language Modeling (MLM) to predict missing words or unsupervised learning strategies to generate the next word in a sequence. Subsequently, they undergo fine-tuning processes supported by human feedback for specific tasks (Shao et al., 2024). In the study, analyses and evaluations were conducted using advanced LLMs, including ChatGPT 5.3, Sonnet 4.6, and Gemini 3.1 Pro, which are among the most widely utilized language models today.

Performance Metrics

In the study, an evaluation of the models' predictions is required to analyze their outputs. For the performance analysis of the classification phase, accuracy, precision, recall, and F1-score metrics—which are frequently employed in machine learning problems—were used, while similarity analysis was utilized for the fill-in-the-blank phase. The equations for these metrics are presented in Equations 1–4.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall} \tag{4}$$

$$Cosine\ Similarity\ (A, B) = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_1^2}\ \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5}$$

Accuracy, precision, recall, and F1-score are among the fundamental metrics used in the performance evaluation of machine learning models. While accuracy represents the overall success of the model, precision indicates the reliability of positive predictions, and recall measures the rate at which actual positive instances are correctly identified. The F1-score, defined as the harmonic mean of precision and recall, provides a balanced evaluation of model performance (Veziroğlu & Bucak, 2025). The confusion matrix presented in Figure 1 was utilized in the calculation of these metrics.

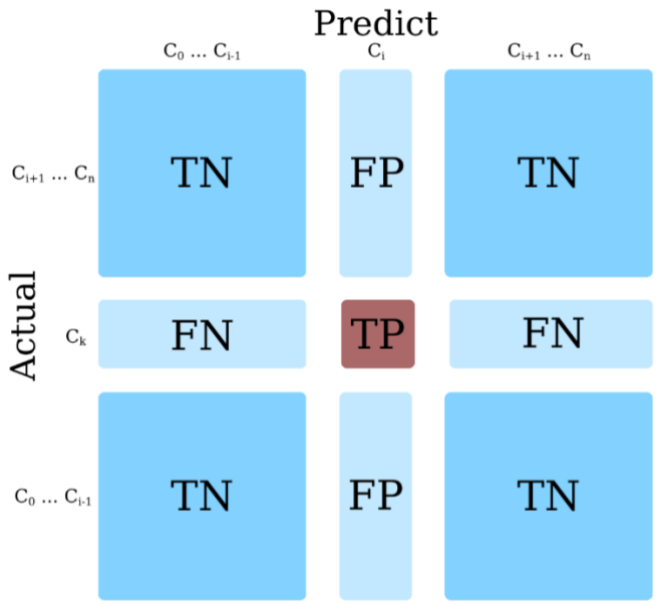

Figure 2. The confusion matrix in classification problems.

A confusion matrix evaluates the performance of a model in a classification problem by comparing predictions with actual classes. During this comparison, it is expressed as True Positive (TP) when a positive instance is correctly classified as positive, False Positive (FP) when a negative instance is incorrectly classified as positive, False Negative (FN) when a positive instance is incorrectly classified as negative, and True Negative (TN) when a negative instance is correctly classified as negative. In the similarity analysis, after tokenizing the relevant examples using the Sentence Bidirectional Encoder Representations from Transformers (SBERT) (Reimers & Gurevych, 2019) method, similarities were

calculated using the cosine distance measurement method. The distance measurement method is presented in Equation 5.

Cosine similarity is a metric used to measure the degree of similarity between two vectors and is frequently preferred for text similarity tasks in the field of natural language processing. Mathematically, cosine similarity is defined as the dot product of two vectors divided by the product of their magnitudes (Venkatesh Sharma et al., 2024). In this approach, the semantic similarity of texts is evaluated based on the geometric opening between the corresponding vectors. A narrower angle between the vectors symbolizes a higher degree of overlap and semantic proximity between the texts; therefore, the minimum angle represents the highest level of similarity (Lahitani et al., 2016). In this study, analyses were conducted by employing the SBERT method to calculate cosine similarity.

Proposed Method

In this study, Large Language Models (LLMs) were employed in a zero-shot setting to classify given instances into predefined categories. The classification performance was evaluated and compared by calculating standard metrics against ground truth labels. As a second task, a cloze test (fill-in-the-blank) was conducted where a random word was removed from each instance, and the models were tasked with predicting the missing token. The predictive performance was quantified using cosine similarity; specifically, the evaluation measured both the semantic proximity between the predicted and original words, and the contextual similarity between the reconstructed sentence and the original sentence. Figure 2 presents the pseudocode for the proposed experimental framework.

Input: Dataset D, Set of LLMs M, Predefined Classes C, Embedding Model E

Output: Classification Metrics, Semantic Similarity Scores

Step 1: Zero-Shot Classification Task
for each model m in M do
for each sample s in D do
Predict class $\bar{y}$ using m in zero-shot setting: $y \leftarrow m(s, C)$
Store $\bar{y}$ alongside ground truth y
end
Compute Accuracy, Precision, Recall, and F1-score for model m
end

Step 2: Semantic Fill-in-the-Blank Task
for each model m in M do
for each sample s in D do
Randomly select and remove a word w_{orig} from s to create s_{masked}
Predict missing word: $w_{pred} \leftarrow m(s_{masked})$
Generate reconstructed sentence: s_{rec} insert w_{pred} into s_{masked}
Word Level: Calculate $cos_{sim}(E(w_{orig}), E(w_{pred}))$
Sentence Level: Calculate $cos_{sim}(E(s), E(s_{rec}))$
end
Calculate average Word and Sentence similarity for model m
end

Figure 2. The proposed experimental framework for zero-shot classification and semantic word prediction tasks.

RESULTS and DISCUSSION

In this section of the study, the results obtained by LLMs in classification and fill-in-the-blank tasks are analyzed. Advanced language models, specifically ChatGPT 5.3, Sonnet 4.6, and Gemini 3.1 Pro, were utilized in the study. In the first task, the models were required to classify previously unseen texts into specific categories. The categories within the dataset were predefined for the models, and they were tasked with assigning one of these categories to the relevant sample. Based on the classification responses provided by the models, comparisons were made with the ground truth values to obtain accuracy, precision, recall, and F1-score metrics. In the second task, the models were asked to select the most appropriate word to fill a blank within a given sentence. While preparing the dataset, the blank spaces within the texts were selected randomly, depending on the sentence length. During the task, the same blank-space queries were presented to all models. The semantic proximity between the models' responses and the target words was measured using cosine similarity to obtain evaluation scores. In the similarity calculation, both the word-to-word similarity and the overall semantic similarity of the sentences were scored. The classification performance analysis is presented in Table 1, and the similarity analysis evaluations are provided in Table 2.

Table 1. Classification performance of llms on texts not seen during training.

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
Gemini 3.1 Pro	92.20	93.78	92.20	92.98
Sonnet 4.6	99.52	99.53	99.52	99.53
ChatGPT 5.3	84.84	89.61	84.84	87.16

When the classification performances obtained by the LLMs are analyzed, the Sonnet 4.6 model demonstrates superior performance with an accuracy of 99.52%, followed closely by the Gemini 3.1 Pro model with an accuracy rate of 92.20%. Although Gemini 3.1 Pro ranks second, its results across all metrics are considered highly effective within the context of natural language processing problems. ChatGPT 5.3, which ranks last in terms of overall success, achieved an acceptable performance with an accuracy metric of 84.84%. In a study conducted on the same dataset, values of 96.00% for accuracy, 90.00% for precision, 89.00% for recall, and 90.00% for F1-score were achieved; while Gemini 3.1 Pro and ChatGPT 5.3 fell behind these results, the Sonnet 4.6 model proved to be more successful. The confusion matrices generated based on the models' predictions are presented in Figure 3.

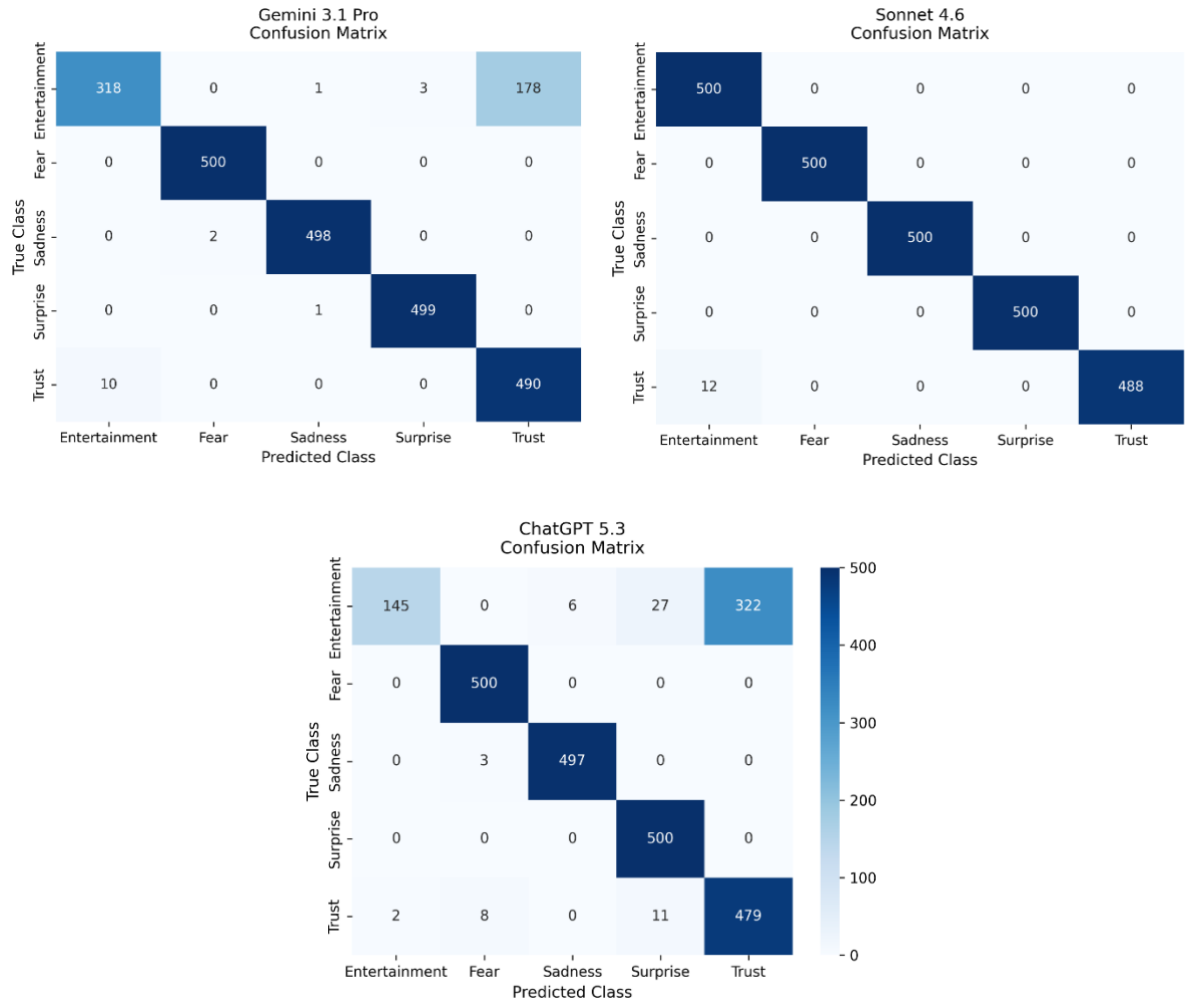

Figure 3. The confusion matrix in classification problems.

When the confusion matrices are analyzed, it is observed that the distinctions between classes are generally clear. However, a significant portion of the predictions made as Trust by the Gemini 3.1 Pro and ChatGPT 5.3 models are Entertainment in reality, indicating that these two models struggle to differentiate between these specific classes. In contrast, the Sonnet 4.6 model's predictions for Entertainment include only a 0.02% error rate involving the Trust class.

Table 2. Similarity analysis of words and sentences by category.

Model	Class	Words Similarity	Sentence Similarity
ChatGPT 5.3	Entertainment	0.81	0.93
	Trust	0.80	0.91
	Fear	0.80	0.91
	Surprise	0.85	0.96
	Sadness	0.79	0.91
	Average	**0.81**	**0.92**
Sonnet 4.6	Entertainment	0.84	0.94
	Trust	0.81	0.91
	Fear	0.82	0.94
	Surprise	0.91	0.97
	Sadness	0.81	0.91
	Average	**0.83**	**0.93**
Gemini 3.1 Pro	Entertainment	0.86	0.95
	Trust	0.82	0.92
	Fear	0.83	0.95
	Surprise	0.92	0.98
	Sadness	0.83	0.93
	Average	**0.85**	**0.94**

The results obtained from the evaluation of LLMs for the fill-in-the-blank task are presented in Table 2, both by class and as an overall average. Cosine similarity values range between 0 and 1, where values closer to 1 indicate higher semantic similarity. The Gemini 3.1 Pro model achieved the best results across all classes and averages for both word-level and sentence-level similarity. Although it produced results close to the top performer, Sonnet 4.6 ranked second, while the ChatGPT 5.3 model ranked last. The lowest score, 0.79, was recorded for the "Sadness" class by ChatGPT 5.3; however, this is still considered a good similarity rate.

According to the word similarity metric, the highest score among the classes was achieved by Gemini 3.1 Pro in the Surprise class with a value of 0.92. Similarly, Gemini 3.1 Pro obtained the highest value in the sentence similarity metric for the Surprise class with a score of 0.98, predicting the blank word with an exceptionally high degree of semantic significance. In the sentence similarity evaluation, all models across all classes achieved results above 0.91, generating sentences that were nearly identical in meaning to the original sentences.

In the word similarity task, it was observed that while selecting words based on the sentence's meaning, the models occasionally chose synonyms or terms that did not alter the overall semantic structure. This situation is reflected in the results as a discrepancy between word similarity and sentence similarity values. There is an approximate 10% increase in the sentence similarity scores compared to the word similarity results. This increase indicates that even when there are variations in specific word choices, there is an exceptionally high degree of semantic proximity between the original sentence and the new sentence generated through the fill-in-the-blank task. The distribution of scores and the data points obtained by the models in sentence similarity are presented in Figures 4 and 5, respectively.

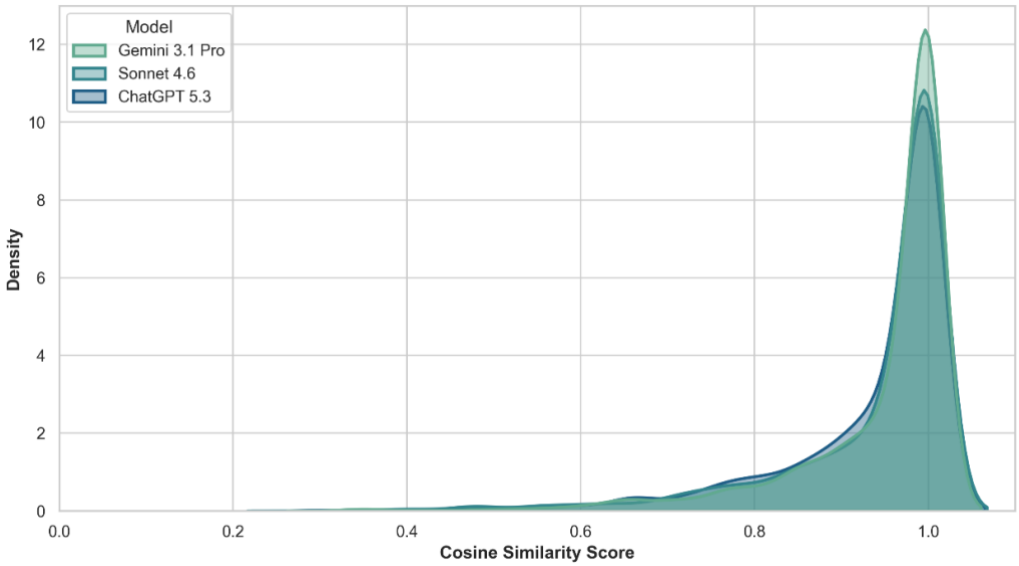

Figure 4. The distribution of similarity scores obtained by the models in sentence similarity.

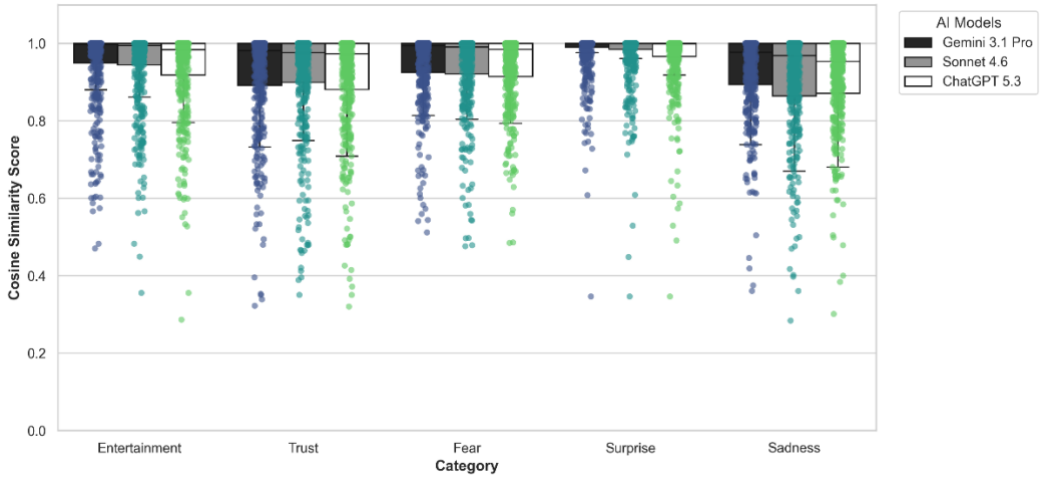

Figure 5. The point distribution of the similarity scores obtained by the models in sentence similarity.

Upon examining the figures, it is observed that both the density and the distribution of individual scores become sparse below the 0.80 similarity threshold. These distributions indicate that only a very small fraction of the words predicted by the models yield low similarity scores. Conversely, for scores

above 0.80, the concentration increases in both figures, reaching a maximum around approximately 0.95.

CONCLUSION

In this study, the performance of LLMs was evaluated in classification and fill-in-the-blank tasks using an emotion-based dataset. The experimental analysis measured classification accuracy, the semantic proximity between predicted and original words, and the semantic alignment between original and reconstructed sentences.

The findings demonstrate that the LLMs performed these tasks with high accuracy and similarity across all evaluations. In the classification task, the Sonnet 4.6 model achieved a peak accuracy of 99.52%, whereas the ChatGPT 5.3 model yielded a comparatively lower accuracy of 84.84%. Similarly, regarding the F1-score metric, the Sonnet 4.6 model attained the highest result at 99.53%, while the ChatGPT 5.3 model recorded the minimum score of 87.16%. In the task of predicting missing words, the Gemini 3.1 Pro model reached the highest semantic scores with 0.85 for word similarity and 0.94 for sentence similarity, demonstrating superior semantic capture. Although the ChatGPT 5.3 model also exhibited high semantic similarity with scores of 0.81 (word) and 0.93 (sentence), it ranked below the other evaluated models.

The results indicate that while classification tasks achieve near-perfect prediction success, word discovery tasks maintain high semantic integrity. These performance variances between models are attributed to differences in architecture and training datasets. Future work is planned to incorporate diverse datasets, open-source language models, and various similarity measurement methodologies.

Additional Declaration

Author Contributions

Funding

This study was not funded by any institution or organization.

Responsible Artificial Intelligence Statement

No artificial intelligence support was received in any part of this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest related to the publication of this study.

Ethics Approval

In all processes of this study, the principles of Pen Academic Publishing Research Ethics Policy were followed. This study does not require ethics committee approval as it does not involve any direct application on human or animal subjects.

REFERENCES

Annepaka, Y., & Pakray, P. (2025). Large language models: a survey of their development, capabilities, and applications. *Knowledge and Information Systems*, *67*(3), 2967–3022.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., & Wang, Y. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, *15*(3), 1–45.

Feng, H., Ronzano, F., LaFleur, J., Garber, M., De Oliveira, R., Rough, K., Roth, K., Nanavati, J., El Abidine, K. Z., & Mack, C. (2024). Evaluation of large language model performance on the biomedical language understanding and reasoning benchmark: Comparative study. *MedRxiv*, 2024–2025.

Jiang, P., Xiao, C., Wang, Z., Bhatia, P., Sun, J., & Han, J. (2024). Trisum: Learning summarization ability from large language models with structured rationale. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2805–2819.

Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016). Cosine similarity to determine similarity measure: Study case in online essay assessment. *2016 4th International Conference on Cyber and IT Service Management*, 1–6.

Leon, M. (2024). Benchmarking large language models with a unified performance ranking metric. *International Journal in Foundations of Computer Science & Technology*, *4*.

Li, J., Bian, Y., Wang, G., Lei, Y., Cheng, D., Ding, Z., & Jiang, C. (2023). Cfgpt: Chinese financial assistant with large language model. *ArXiv Preprint ArXiv:2309.10654*.

Ma, C., Wu, Z., Wang, J., Xu, S., Wei, Y., Liu, Z., Zeng, F., Jiang, X., Guo, L., & Cai, X. (2024). An iterative optimizing framework for radiology report summarization with ChatGPT. *IEEE Transactions on Artificial Intelligence*, *5*(8), 4163–4175.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781*.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. Ben, Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., & Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, *56*(2), 1–40.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2025). A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, *16*(5), 1–72.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., Ahmad, J., Ali, M. E., & Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, *12*, 26839–26874.

Shao, M., Basit, A., Karri, R., & Shafique, M. (2024). Survey of different large language model architectures: Trends, benchmarks, and challenges. *IEEE Access*, *12*, 188664–188706.

Sindhu, B., Prathamesh, R. P., Sameera, M. B., & KumaraSwamy, S. (2024). The evolution of large language model: Models, applications and challenges. *2024 International Conference on Current Trends in Advanced Computing (ICCTAC)*, 1–8.

Song, X., Xie, K., Lee, L., Chen, R., Clark, J. M., He, H., He, H., Min, J., Zhang, X., & Zheng, S. (2025). Performance evaluation of large language models in statistical programming. *ArXiv Preprint ArXiv:2502.13117*.

Su, C.-Y., & McMillan, C. (2024). Distilled GPT for source code summarization. *Automated Software Engineering*, *31*(1), 22.

Tan, E., & Liu, H. (2022). Performance Comparison of Seven Pretrained Models on a text classification task. *Proceedings of the 2022 5th International Conference on Signal Processing and Machine Learning*, 8–12.

Tintin, R., & Yücebaş, S. C. (2026). Duygu-Turk: A Context-Aware Sentiment Analysis Framework for Turkish, Based on Plutchik's Emotion Model. *Journal of Universal Computer Science*, *32*(4).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Venkatesh Sharma, K., Ayiluri, P. R., Betala, R., Jagdish Kumar, P., & Shirisha Reddy, K. (2024). Enhancing query relevance: leveraging SBERT and cosine similarity for optimal information retrieval. *International Journal of Speech Technology*, *27*(3), 753–763.

Veziroğlu, M., & Bucak, İ. (2025). Haber Sınıflandırma Sistemlerinde Naive Bayes ve Makine Öğrenmesi Algoritmaları Arasında Performans Karşılaştırması. *Journal of the Institute of Science and Technology*, *15*(1), 57–70.

Xu, H., & Ashley, K. (2023). Argumentative segmentation enhancement for legal summarization. *ArXiv Preprint ArXiv:2307.05081*.

Zhu, J., Li, J., Wen, Y., & Guo, L. (2024). Benchmarking large language models on CFLUE-a Chinese financial language understanding evaluation dataset. *Findings of the Association for Computational Linguistics: ACL 2024*, 5673–5693.