RankLLM: A Multi-Criteria Decision-Making Method for LLM Performance Evaluation in Sentiment Analysis

Huzhi Xue[1,2] **Butian Zhao**[3] **Haihua Xie**[2,*] **Zeyu Sun**[4]

[1] School of Economics and Management, Beihang University (BUAA), Beijing 100191, China
[2] Beijing Institute of Mathematical Sciences and Applications (BIMSA), Beijing 101408, China
[3] School of Management, Beijing University of Chinese Medicine, Beijing 100029, China
[4] School of Economics and Management, Beijing Jiaotong University, Beijing 100044, China
xuehz@buaa.edu.cn; haihua.xie@bimsa.cn;
btzhao@bucm.edu.cn; 19113052@bjtu.edu.cn.

Abstract

Large Language Models (LLMs) have made significant advancements in sentiment analysis, yet their quality and reliability vary widely. Existing LLM evaluation studies are limited in scope, lack a comprehensive framework for integrating diverse capabilities, and fail to quantify the impact of prompt design on performance. To address these gaps, this paper introduces a set of LLM evaluation criteria with detailed explanations and mathematical formulations, aiding users in understanding LLM limitations and selecting the most suitable model for sentiment analysis. Using these criteria, we apply the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS), a classic decision-making method, to rank the performance of LLMs in sentiment analysis. We evaluated six popular LLMs on three Twitter datasets covering different topics and analyze the impact of prompt design by assessing model-prompt combinations. Additionally, a validation experiment on a publicly available annotated dataset further confirms our ranking results. Finally, our findings offer valuable insights into the evaluation and selection of LLMs for sentiment analysis.

CCL 2025

1 Introduction

Large Language Models (LLMs) have experienced considerable development in recent years and the number of LLMs (e.g., ChatGPT) has significantly increased in the past few years (Li et al., 2024a). Composed of billions of parameters and trained on billions of tokens, LLMs have achieved remarkable advancements in numerous real-world tasks (Chang et al., 2024). Sentiment analysis is one of the key application areas of LLMs (Zhang et al., 2023). With the exponential growth in the number of LLMs, the quality and reliability of LLMs vary widely (Li et al., 2024b; Chu et al., 2023). For instance, these models sometimes tend to generate text that is factually inaccurate (Liu et al., 2023). This leads to a deficiency in criterion and methodology for users to determine the suitability of an LLM (Pan et al., 2024). Hence, there is a growing need for evaluating the performance of LLMs to assist users in selecting the most appropriate LLM and understanding the limitations of different LLMs.

To quantify the performance of LLMs, evaluation criteria are a key but challenging issue and there exist some overlooked issues (Li et al., 2024a). First, as an important emerging technology, existing related research predominantly focuses on exploring "LLM as an evaluator" while ignoring to assess the quality of LLMs themselves (Huang et al., 2024). Second, the capabilities of LLMs are quite extensive, enabling them to produce different outputs from diverse tasks, which makes it challenging to identify the evaluation criteria. Third, although some studies have proposed criteria for evaluating and selecting LLMs, they have overlooked the more practical implications by quantifying them (Hu et al., 2024). **Hence, a set of evaluation criteria for assessing the sentiment analysis capabilities of LLMs is proposed in this paper**. The criteria should not only provide detailed explanations but also should offer mathematical formulations.

When assessing the performance of LLMs, one of the most important parts is the evaluation method and there exist some problems that need to be solved. Firstly, most LLM evaluation methods rely on

[*] Corresponding Author

©2025 China National Conference on Computational Linguistics
Published under Creative Commons Attribution 4.0 International License

human review (manual evaluation), leading to subjectivity and uncertainty in the evaluation process due to the vacillation of human cognition (Yang et al., 2024; Shankar et al., 2024). Secondly, current LLM evaluation methods are deficient in taking into account multiple LLM evaluation criteria (Zhang et al., 2024). Accordingly, there is an urgent need to consider a method that is capable of evaluating LLMs.

In practice, LLM is a complex system composed of various features and capabilities. It is challenging to directly conclude the performance of LLM from a single aspect because different LLMs may excel in different aspects (Pan et al., 2024). Hence, LLM evaluation can be treated as a Multi-Criteria Decision-Making (MCDM) problem that focuses on selecting the optimal one from a set of alternatives under considering several criteria from different dimensions (Aruldoss et al., 2013). The evaluation of LLMs requires aggregating these aspects to provide a comprehensive ranking result, which aligns with the core principle of MCDM methods.

In the past few decades, some representative MCDM methods have emerged and utilized (Hwang et al., 1981). Among them, Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) is one of the most highly regarded and widely used methods (Chakraborty, 2022). The core concept of the TOPSIS method is to rank the alternatives by calculating their closeness coefficient to the Positive Ideal Solution (PIS) and Negative Ideal Solution (NIS) (Chen, 2021). This approach is capable of making full utilization of the criterion information, offering cardinal ranking results without requiring the independency of criterion preferences (Corrente and Tasiou, 2023). TOPSIS has been widely used in areas such as service quality assessment (Du, 2023), performance evaluation (Sun and Yu, 2021), etc (Song et al., 2025). **Thus, based on the criteria, the TOPSIS is employed to evaluate the performance of LLMs.** We tasked LLMs with analyzing the sentiment attitudes of different users towards Artificial General Intelligence (AGI) and three other datasets on Twitter, and subsequently evaluated LLMs using the TOPSIS method based on the outputs from them.

In services developed based on LLMs, users can interact with LLMs through natural languages, called “prompts” (instructions that generate responses from LLMs). The impact of different prompts on the output results of LLMs is significant (Mizrahi et al., 2023). However, the performance of the combination of different LLMs with distinct questioning formats fails to be explored (Li et al., 2024b). **Therefore, we integrate different prompts with LLMs to explore the impact of different prompts on LLM performance.** In this way, the optimal combination between LLMs and prompts can be identified, as well as the impact of prompts on enhancing the performance of LLMs.

In this paper, firstly the datasets are collected through web scraping, which consists of user comments on three datasets with different topics from Twitter. Next, we invoked the official APIs of different LLMs to conduct sentiment analysis. Then, the results of sentiment analysis from various LLMs and prompts are transformed and calculated to serve as the input of the TOPSIS method, which is employed to evaluate the performance of LLMs. Finally, we obtain the ranking outcomes of LLMs. The results were further validated using a publicly available dataset.

The main contributions of this paper are summarized as follows:

- A set of LLM evaluation criteria as well as their mathematical formulations and statistical interpretations are introduced, which has provided a solid foundation for evaluating the performance of LLM for sentiment analysis.

- We integrate the TOPSIS method with LLM performance evaluation for sentiment analysis. By calculating the closeness coefficient to the ideal solution, an objective and quantitative evaluation of LLM can be achieved.

- By utilizing the proposed evaluation criteria and TOPSIS method, we examined the differences in performance when different prompts are paired with various LLMs, thereby assisting users in identifying the optimal combination of LLMs and prompts.

2 Related works

In this section, some related research on LLM evaluation criteria, LLM evaluation methods, and the integration of LLMs and prompts are introduced.

CCL 2025

LLM evaluation criteria. LLM evaluation criteria are crucial in assessing the performance and capabilities of these sophisticated models. Each criterion has its characteristics and limitations. Early research on generative models evaluation primarily employed BLUE (Papineni et al., 2002), ROUGE (Lin, 2004), and MoverScore (Zhao et al., 2019), etc. These ones can only capture one of a few aspects and are less effective (Pan et al., 2024). More recently, some manually designed criteria emerged. For example, Wang et al. used consistency and reliability to evaluate the use of LLM in clinical medicine (Wang et al., 2024). Hu et al. designed criteria such as fluency, perturbations, etc. (Hu et al., 2024). Gao et al. employed metrics such as accuracy and fairness to assess the effectiveness of LLMs (Gao et al., 2024). However, few of them considered multiple evaluation dimensions with computational formulations in sentiment analysis.

LLM evaluation methods. There have been some other evaluation methods emerged in these years. For example, approaches like star scoring utilizes the average scores (Zhang et al., 2024). BERTScore allocates scores to the outputs of LLMs by utilizing another LLM (Zhang et al., 2019), which may heavily rely on the pre-trained model and lacks flexibility in considering multiple criteria. ChatBot Arena enables public manual evaluators to cast votes between two LLMs in order to assess their performance (Zheng et al., 2024), which is less efficient. Human evaluation also remains a crucial component, complementing automated metrics with real-world judgments of quality and relevance, which lead to high costs and subjective judgment. However, none of the existing ones have compared and evaluated LLMs from the perspective of Multi-Criteria Decision-Making (MCDM). Among all MCDM methods, TOPSIS possesses strong generalization capabilities and is capable of addressing the aforementioned issues. First, TOPSIS allows for the assignment of weights to different criteria, reflecting their relative importance in the evaluation of LLMs (Irfan et al., 2022). Second, TOPSIS can be adapted to include any number of criteria and is not limited to a predefined set, making it flexible for evaluating LLMs across various tasks and domains. Third, TOPSIS provides an objective way to evaluate LLMs by reducing subjectivity and relying on mathematical computations.

Integration of LLM and prompts to sentiment analysis. The existing works have shown that the prompt formulation has a strong impact on the performance of LLMs (Sun et al., 2024). For example, Mizrahi et al. analyzed the sensitivity of several prompts during task execution on ChatGPT 3.5 (Mizrahi et al., 2024). Khurana et al. investigated the effectiveness of prompt-based interactions (Khurana et al., 2024). Tian et al. evaluated the capability of GPT-4 with four different prompting through human evaluators (Tian et al., 2024). However, the effectiveness of prompts varies significantly across different LLM architectures, or training paradigms (Mizrahi et al., 2024). The extent to which prompts affect the comprehensive performance of sentiment analysis across various LLMs remains to be explored.

3 Design of RankLLM

3.1 Selection of LLM evaluation criteria

In this part, the LLM evaluation criteria selected in RankLLM and their detailed explanation are introduced. We categorize the criteria into six dimensions. By quantifying them, we can comprehensively assess the performance of LLMs.

Deviation. Different LLMs may generate distinct outcomes when processing a same task. Deviation is introduced to determine the extent of discrepancy between the output of LLM and the standard answer. The standard answer is determined through manual annotation for each tweet.

Compliance. LLMs may encounter challenges in accurately capturing all the details of lengthy and complex prompts (Gani et al., 2023). Compliance is used to judge whether LLMs adhere to the prompt. It can be validated by examining whether the format of the results obtained from multiple analyses of the same prompt by an LLM is consistent with the format specified in the prompt.

Consistency. LLMs exhibit great confidence or certainty towards the source content when objective answers are lacked (Miao et al., 2021). Consistency is utilized to assess the level of confidence LLMs have in their own results. The consistency of an LLM can be determined by examining the identical results upon repeated analysis of the same prompt.

Miscalibration. LLMs may occasionally become uncertain when facing with multiple choices (Zhou

CCL 2025

et al., 2023). Miscalibration can be validated by examining the average deviation of the results from multiple analyses of the same tweet by an LLM compared to the accurate solution.

Robustness. The errors in a prompt may fail to answer the question correctly (Liu et al., 2023). Robustness is designed to investigate the impact of LLMs to attacks on prompts. To be specific, we use this criterion to detect whether grammatical or typo errors can cause LLMs to output wrong and low-quality content.

Stability. The time consumptions of different LLMs vary when performing the same task (Wilkins et al., 2024). Stability is a criterion used to assess the variation in time consumption, which can be quantified by the variance of time costs of an LLM when processing the same task multiple times under the conditions where other variables remain constant. The consumption of time is greatly influenced by the network conditions. Hence, we only focus on the standard deviation of time consumption.

3.2 Evaluation method of LLMs

In this part, we present the procedure of TOPSIS method employed for assessing the performance of LLMs, including the integration of criteria introduced in the previous subsection, the calculation of weights among criteria, and the determination of final ranking results. The algorithm of the TOPSIS method is shown in Algorithm 1. In the following, the detailed steps of TOPSIS method for assessing LLM performance in sentiment analysis are shown.

Stage 1: Normalize Evaluation ratings

Step 1. Acquisition of the original evaluation ratings. We consider that there are n LLM evaluation criteria $C = \{c_1, c_2, ..., c_n\}$ and a set of m candidate LLMs $X = \{x_1, x_2, ..., x_m\}$ that need to be evaluated. All of the LLM evaluation criteria are assumed to be beneficial. Then the original ratings of each LLM x_j $(j = 1, 2, ..., m)$ under each criterion c_k $(k = 1, 2, ..., n)$ can be represented as a_{jk}.

Step 2. Linear normalization of the original evaluation ratings. After obtaining the original evaluation ratings, it is necessary to normalize these values under the same criterion due to the different dimensions of criteria and the original ratings are transformed as,

$$\widetilde{a}_{jk} = \frac{a_{jk} - \min\limits_{1 \leq j \leq m} (a_{jk})}{\max\limits_{1 \leq j \leq m} (a_{jk}) - \min\limits_{1 \leq j \leq m} (a_{jk})}. \tag{1}$$

Thus, the original LLM evaluation matrix $\widetilde{a}_{jk}$ is established.

Stage 2: Determine the weights among criteria

Step 3. In this step, we calculate the weight of each LLM evaluation criterion w_k $(k = 1, 2, ..., n)$ through Shannon entropy in information theory. The core principle of entropy is to calculate the amount of information carried by a criterion (Chen, 2021). Entropy based weight calculation follows a rule that the lower the entropy E_k (the higher the value $1 - E_k$), the higher the weight of criterion c_k is (Li et al., 2022). Entropy is not only an objective weighting method which can fully exploit the information of data itself, but also conform to reality. The entropy E_k and the weights of each LLM evaluation criterion w_k can be determined by equation (2) and equation (3) respectively in the following:

$$E_k = 1 - \sum_{j=1}^{m} \widetilde{a}_{jk} \log(\widetilde{a}_{jk}), \tag{2}$$

$$w_k = \frac{1 - E_k}{m - \sum\limits_{k=1}^{n} E_k}. \tag{3}$$

Hence, the weight of each LLM evaluation criterion w_k is obtained.

Step 4. Construction of weighted LLM evaluation matrix. In this step, the weights of LLM evaluation criteria w_k are assigned to the normalized decision matrix $\widetilde{a}_{jk}$ denoted as,

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nm} \end{bmatrix}, \tag{4}$$

CCL 2025

Algorithm 1 LLM evaluation algorithm based on the TOPSIS method

Input: LLM evaluation information matrix a_{jk} across multiple criteria, $(j = 1, 2, ..., m)$, $(k = 1, 2, ..., n)$
Output: Closeness coefficient CC_j of the j-th LLM.

```
for each $j = 1, 2, ..., m$ do
    Normalize the original evaluation information $a_{jk}$ to $\widetilde{a}_{jk}$ based on eq.(1);
end for
for all $k = 1, 2, ..., n$ do
    Calculate weights of criteria $w_k$ through entropy;
    Assign weights of criteria $w_k$ to the normalized decision matrix $\widetilde{a}_{jk}$;
end for
for $k = 1, 2, ..., n$ do
    for $j = 1, 2, ..., m$ do
        Determine the NIS and PIS;
        Calculate the distance between the $j$-th LLM to PIS and NIS;
        Obtain the closeness coefficient $CC_j$ of the $j$-th LLM.
    end for
end for
return $CC_j$
```

$$z_{jk} = w_k \times \widetilde{a}_{jk}, \tag{5}$$

where Z is the weighted matrix.

Stage 3: Acquire the ranking results of LLMs

Step 5. Determination of the positive ideal solution (PIS) and negative ideal solution (NIS). To be specific, PIS is the LLM that has the highest evaluation value among all criteria and PIS is performed as,

$$Z^+ = \left\{z_1^+, z_2^+, ..., z_m^+\right\} = \left\{\max z_{jk} \left| k = 1, 2, ..., n\right.\right\}. \tag{6}$$

Similarly, NIS is the LLM that has the lowest evaluation value among all criteria. NIS is outlined as,

$$Z^- = \left\{z_1^-, z_2^-, ..., z_m^-\right\} = \left\{\min z_{jk} \left| k = 1, 2, ..., n\right.\right\}. \tag{7}$$

Step 6. Calculation of the distance to NIS and PIS. The positive and negative distance can be determined by equation (8) and equation (9), respectively.

$$D_j^+ = \sqrt{\sum\nolimits_{j=1}^{m} \left(z_{jk} - z_k^+\right)}, (j = 1, 2, ..., m) \tag{8}$$

$$D_j^- = \sqrt{\sum\nolimits_{j=1}^{m} \left(z_{jk} - z_k^-\right)}, (j = 1, 2, ..., m) \tag{9}$$

Step 7. Calculation of closeness coefficient CC_j of each LLM. The LLM with highest closeness coefficient represents the best performance. The equation is as follows:

$$CC_j = \frac{D_j^-}{D_j^- + D_j^+} \tag{10}$$

The workflow of RankLLM is shown in Figure 1.

4 Experiment

4.1 Dataset

In our experiment, three datasets with different topics, including artificial intelligence, sports, and politics are constructed.

Dateset 1: artificial general intelligence (AGI) The first 1000 Twitter user comments targeting AGI after October 1st, 2022.

Dateset 2: offside in football (sports) The first 1000 Twitter user comments regarding offside in football after January 1st, 2023.

Dateset 3: Israeli Palestinian Conflict (politics) The first 1000 Twitter user comments regarding the Israeli Palestinian Conflict after October 6th, 2023.

CCL 2025

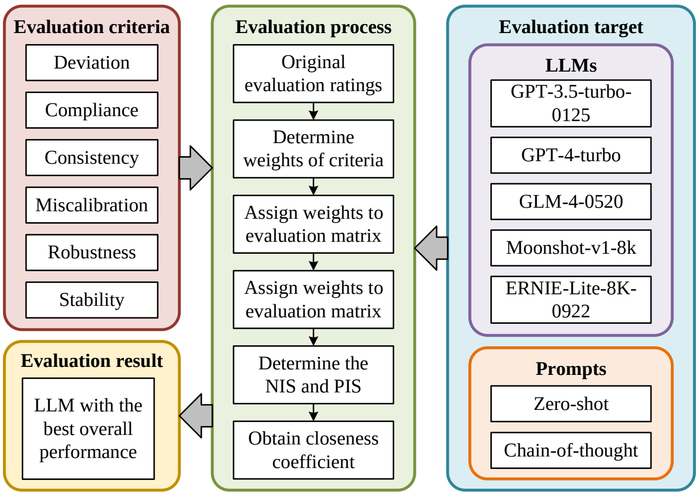

Figure 1: The workflow of RankLLM

We performed manual annotation on the three datasets. The annotators were two graduate students majored in computer science in China. They were compensated for their work to ensure a high-quality annotation process. The annotation process followed these steps: First, two annotators independently labeled the data. If their results fall within a ±10% agreement range, the annotations are considered consistent. In cases of disagreement, a second round of annotation will be conducted, with up to three rounds performed. For any remaining discrepancies after three rounds, the annotators discussed and resolved them to finalize the annotations.

CCL 2025

4.1.1 Selection of LLMs

In the experiment, six LLMs are selected including OpenAI's gpt-3.5-turbo-0125[0] and gpt-4-turbo[1], glm-4-0520[2] and moonshot-v1-8k[3] from Zhipu AI, ERNIE-Lite-8K-0922[4] from Baidu, as well as DeepSeek's deepseek-v3[5].

4.1.2 Selection of prompts

We selected two representative categories of in-context learning methods, including zero-shot (ZS) and chain-of-thought (CoT) settings. Zero-shot aims to process tasks without any examples and Chain-of-thought (CoT) means LLMs generate outputs through step-by-step reasoning. Table 1 illustrates the prompt design with the politics dataset as an example.

Each LLM answers p times based on the prompt in the evaluation process. For ease of calculation, we process the results of sentiment analysis for data i output by the j-th LLM into the form of probabilistic linguistic terms $L^{i}_{j(t)} = [l_1^{j(t),i}, l_2^{j(t),i}, ..., l_o^{j(t),i}]$, in which o represents the number of emotions that the user wishes to analyze and t represents the instance number of the LLM's response to the user's query ($t \leq p$).

4.2 Evaluation criteria

As introduced above, the criterion set are used to evaluate the performance of LLMs in sentiment analysis in this paper. The calculation process of these evaluation criteria is described in detail as follows.

Each data have a standard sentiment analysis answer. The standard answers are generated via human annotation to ensure accuracy. Suppose that the volume for a dataset is r, then the set of standard answers

[0] https://platform.openai.com/docs/models/gpt-3.5-turbo
[1] https://platform.openai.com/docs/models/gpt-4-turbo
[2] https://open.bigmodel.cn/dev/api/normal-model/glm-4
[3] https://platform.moonshot.ai/
[4] https://agicto.com/model/ERNIE-Lite-8K-0922
[5] https://deepseekv3.org/

(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Table 1: Prompt design of politics dataset

Prompt	Design
zero-shot (zs)	You are a sentiment analysis assistant. Analyze the sentiment of the following tweet about the Israeli Palestinian Conflict. Strictly provide the sentiment for Palestinian or Palestine in the format [percentage, positive], [percentage, neutral], [percentage, negative]:
Chain-of-thought (CoT)	You are a sentiment analysis assistant. Let's analyze the sentiment of the following tweet about the Israeli Palestinian Conflict. **First**, read the tweet carefully to understand the overall tone and context. **Next**, consider positive language that indicate approval, excitement, or support for Palestinian or Palestine. **Then**, identify any neutral statements or facts that do not express a strong sentiment. **Finally**, look for any negative language or phrases that express criticism, concern, or skepticism about Palestinian or Palestine. Based on your analysis, provide the sentiment strictly in the format [percentage, positive], [percentage, neutral], [percentage, negative].

for this dataset is $S = [S^1, S^2, ..., S^r]$. For data i, the standard answer is S^i.

(1) Deviation

For a single data i, let S^i be the standard sentiment analysis answer, $L^i_{j(t)}$ be the answer output by the j-th LLM in the t-th response, and both S^i and $L^i_{j(t)}$ are expressed in the form of probabilistic linguistic terms. The deviation of the j-th LLM on the dataset is

$$Dev_j = \frac{1}{r}\frac{1}{p}\sum_{i=1}^{r}\sum_{t=1}^{p} \text{CosineDistance}\left(S^i, L^i_{j(t)}\right), \tag{11}$$

in which

$$\text{CosineDistance}\left(S^i, L^i_j\right) = \frac{\left\|S^i\right\|\left\|L^i_{j(t)}\right\| - S^i \cdot L^i_{j(t)}}{\left\|S^i\right\|\left\|L^i_{j(t)}\right\|}. \tag{12}$$

(2) Compliance

For a dataset, let F_j be the set of all answers generated by the j-th LLM that conform to the format specified by the prompt, then the compliance can be described as

$$Comp_j = \frac{N\left(F_j\right)}{r}. \tag{13}$$

N is the counting operation on the set elements.

(3) Consistency

For a dataset, let $L_j = \left[L_{j(1)}, L_{j(2)}, ..., L_{j(p)}\right]$ be the answers output by the j-th LLM, then the consistency can be donoted as

$$Cons_j = \frac{\text{Mode}\left(L_j\right)}{p}, \tag{14}$$

in which Mode represents the mode of the answer generated by the j-th LLM.

(4) Miscalibration

For a dataset, let $L_{j(1)}$ be the answer generated by the first response of the j-th LLM, and $L_{j(t)}$ be the answers output by the j-th LLM in its t-th response ($t > 1$), and the first and other answers are expressed in the form of probabilistic linguistic terms. The miscalibration of the j-th LLM is

$$Mis_j = \frac{1}{p}\sum_{t=2}^{p} \text{MSE}\left(L_{j(1)}, L_{j(t)}\right). \tag{15}$$

MSE represents the mean square error between the standard answers and answers output by LLM, in which

$$\text{MSE}\left(L_{j(1)}, L_{j(t)}\right) = \frac{1}{r}\sum_{i=1}^{r}\left(L^i_{j(1)} - L^i_{j(t)}\right). \tag{16}$$

CCL 2025

(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

(5) Robustness

For a dataset, the j-th LLM is used to additionally generate answers for all data denoted as R_j. We alter some words in the prompt to mimic type errors that people might make. Let $Nan_{j(t)}$ be and $Nan_{R,j}$ be the missing rate of answer $L_{j(t)}$ and R_j, then the robustness of the j-th LLM can be described as

$$Rob_j = \frac{\frac{1}{p}\sum_{t=1}^{p} Nan_{j(t)}}{Nan_{R,j}}. \tag{17}$$

(6) Stability

For a dataset, let $T_{j(t)}$ be the total time consumption for generating answers for all data of the j-th LLM in its t-th response. Then, the stability can be described as

$$Stab_j = \sqrt{\frac{\sum_{t=1}^{p}\left(T_{j(t)} - \bar{T}_j\right)}{p}}. \tag{18}$$

$\bar{T}_j$ is the mean of total time consumption for generating answers for all data by the j-th LLM in p responses.

The missing rate Nan is the ratio of the amount of data that has not generated a complete answer to the total amount of data.

4.3 Implementation details

For the evaluation environment, in order to avoid potential influence from other variables on the results, all the experiments were conducted under the same network using Python 3.9 and the Python package OpenAI, zhipuai, Numpy, and pandas are utilized for the processing of outputs produced by LLMs and the computation of evaluation methods. The experiments were run on a laptop equipped with an Intel Core i7-9750H CPU, a NIVDIA GeForce RTX 2070 GPU, and 16 GB of RAM. In the experiment, we utilized the official API to interact with LLMs to generate sentiment analysis responses for a tweet 10 times ($p = 10$).

On the evaluation criteria, the evaluation process requires that all criteria ought to be beneficial (the larger, the better), while Deviation, Miscalibration, and Stability are non-beneficial criteria (the smaller, the better), so they need to be converted to be beneficial while maintaining their meaning as follows:

$$Dev'_j = 1 - Dev_j, \tag{19}$$

$$Mis'_j = \frac{1}{Mis_j}. \tag{20}$$

$$Sta'_j = \frac{1}{Sta_j}. \tag{21}$$

For criterion Robustness, we assess the robustness of LLMs by altering one verb (“analyze” to “analyzee”) and one noun (“percentage” to “percenatge”) respectively in the prompt we designed.

4.4 Results

Based on the set of evaluation criteria and the evaluation process, we get the final performance of LLMs on three datasets, which is shown in Table 2 and Figure 2.

In AGI dataset, gpt-4-turbo with CoT prompt had the best performance and ERNIE-Lite-8K-0922 with chain-of-thought prompt was the last. In sports dataset, gpt-4-turbo achieved the top two rankings using CoT and ZS prompt strategies, respectively. ERNIE-Lite-8K-0922 with two prompts was ranked the last two, respectively. In politics dataset, gpt-4-turbo with chain-of-thought and glm-4-0520 with zero-shot prompt ranked the top two models, while ERNIE-Lite-8K-0922 was still positioned the last.

Next, we remark the ranking results of LLMs from different dimensions. In the criterion Deviation, ERNIE-Lite-8K-0922 scored the lowest, which means it has the highest level of discrepancy between the outputs and the standard answer. In the criterion Compliance, the model deepseek-v3 performed

CCL 2025

(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Table 2: Closeness coefficients and ranking results of the experiment

Prompt	Model	AGI dataset		Sport dataset		Politics dataset		annotated dataset	
		CC_j	Rank	CC_j	Rank	CC_j	Rank	CC_j	Rank
zero-shot	gpt-3.5-turbo-0125	0.1136	10	0.1150	9	0.1223	9	0.0470	7
	gpt-4-turbo	0.4252	7	0.7466	4	**0.9739**	1	0.0963	4
	glm-4-0520	0.8133	2	0.8117	2	0.6990	2	0.1117	3
	moonshot-v1-8k	0.6412	4	0.3142	7	0.2919	6	0.0473	6
	ERNIE-Lite-8K-0922	0.0096	12	0.0156	12	0.0185	12	0	11
	deepseek-v3	0.4458	5	0.7662	3	0.3215	5	0.5734	2
chain-of-thought	gpt-3.5-turbo-0125	0.3892	8	0.3463	6	0.1247	8	**0.9975**	1
	gpt-4-turbo	**0.9705**	1	**0.8828**	1	0.4795	4	0.002	10
	glm-4-0520	0.7913	3	0.7344	5	0.5411	3	0.077	5
	moonshot-v1-8k	0.2236	9	0.0494	10	0.0820	10	0.038	8
	ERNIE-Lite-8K-0922	0.0307	11	0.0379	11	0.0278	11	0	11
	deepseek-v3	0.4263	6	0.2170	8	0.1900	7	0.0376	9

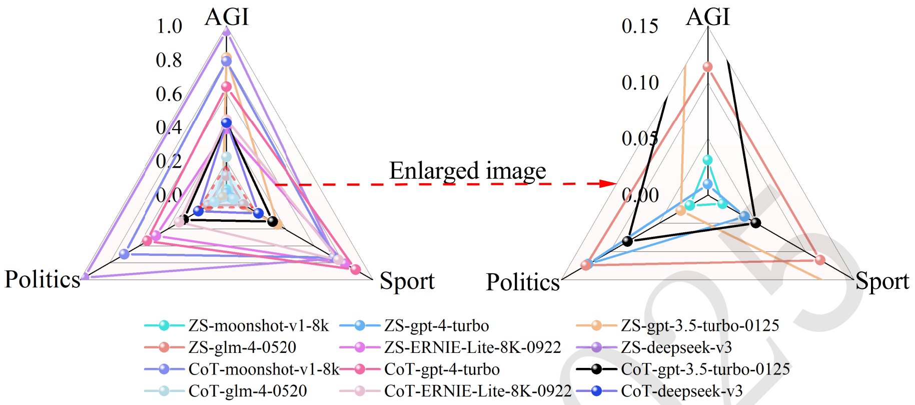

Figure 2: Closeness coefficients of LLMs across three datasets.

the worst which means it is incapable of adhering to the prompt, while gpt-3.5-turbo-0125 scored almost highest. In the criterion Consistency, the performance of the model ERNIE-Lite-8K-0922 was the most inferior, while gpt-3.5-turbo-0125 and gpt-4-turbo were much greater than others. In the criterion Miscalibration, the model ERNIE-Lite-8K-0922 still performed worst, while gpt-4-turbo was the most superior. In the criterion Robustness, the model moonshot-v1-8k performed the best. In the criterion Stability, the model deepseek-v3 had the greatest standard deviation in the time consumed across all datasets and prompts, which means that the model is less stable than others.

5 Discussions

5.1 Discussion on ranking results

It can be seen that the performance of an LLM is basically consistent across different datasets. This suggests that the capability of an LLM in sentiment analysis does not heavily rely on the specific type of data to which it is applied.

To begin with, gpt-4-turbo demonstrated excellent performance across all three datasets, indicating its strong comprehensive capability in handling sentiment analysis tasks. Secondly, glm-4-0520 also showed stable performance, particularly under the zero-shot prompt, highlighting its strong adaptability to tasks. The results revealed that deepseek-v3 with zero-shot prompt generally performed worse than the model with chain-of-thought prompt. This suggests that its training may be more focused on relatively simple tasks. Finally, ERNIE-Lite-8K-0922 had nearly the worst performance under all experiments. This indicates that the model struggles with both prompts compared with other models. This is due to its exceptionally low scores in criteria like consistency and miscalibration.

5.2 The impact of prompts on LLMs

In order to discuss the impact of prompts on LLMs, we subtract the metric scores obtained using the zero-shot prompt from those using the chain-of-thought prompt for the same LLM, and then calculate

CCL 2025

Table 3: Average performance change of LLMs with prompt variations across three datasets

LLM	Deviation	Compliance	Consistency	Miscalibration	Robustness	Stability
gpt-3.5-turbo-0125	0.0557	0.0270	0.0820	**-15.2061**	-0.5827	**0.0294**
gpt-4-turbo	0.0327	**0.3688**	**0.2593**	2.1893	-0.1350	0.0112
glm-4-0520	-0.0434	0.0051	0.0112	4.7608	-0.5776	0.0082
moonshot-v1-8k	-0.0438	-0.0273	0.0129	9.7164	0.1557	0.0009
ERNIE-Lite-8K-0922	**0.0704**	0.0939	-0.1088	-1.233	0.1222	0.0046
deepseek-v3	-0.0551	0.0162	0.0890	34.6464	**-1.5723**	0.0008

the average of these differences across the three datasets, which is shown in Table 3.

It is worth noting that there is a significant decrease in the overall performance of moonshot-v1-8k when using the chain-of-thought prompt compared to the zero-shot prompt. A similar problem is also observed with deepseek-v3 and glm-4-0520 to a lesser extent than moonshot-v1-8k. For gpt-4-turbo and gpt-3.5-turbo-0125, their rankings significantly improved when using the chain-of-thought prompt. This indicates that the chain-of-thought prompt effectively enhances the reasoning capabilities of these models in complex tasks, particularly in sentiment analysis tasks requiring multi-step logical reasoning. This prompting strategy better guides the models to generate more reliable results. For underperforming models like ERNIE-Lite-8K-0922, the chain-of-thought prompting offers little to no improvement, which indicates that the model's limitations can not be easily addressed by changes in prompting strategy.

Next, the impacts of prompts on the performance of various dimensions are presented. The chain-of-thought prompt may significantly reduce model stability by increasing the time required for LLMs to process tasks according to Table 3. Among the compared models, deepseek and moonshot-v1-8k exhibit more prominently. The consistency of LLMs shows decreases with the use of chain-of-thought prompt except ERNIE-Lite-8K-0922. The robustness of models are enhanced with the use of chain-of-thought prompt except moonshot-v1-8k and ERNIE-Lite-8K-0922. This is because the chain-of-thought prompt is better equipped to detect and correct errors at each stage of the reasoning process, which contributes to overall robustness. However, the deviations of models are decreased with the use of chain-of-thought prompt except for moonshot-v1-8k and ERNIE-Lite-8K-0922.

5.3 Verification of ranking results

To validate the reliability of our ranking results, we conducted an additional experiment using an annotated publicly dataset obtained from Kaggle[6]. The ranking results on this dataset are shown in the last two columns of Table 2 and Figure 3. The results indicate that the ranking results of the LLMs on the annotated publicly dataset have similar trends observed in the other three datasets, which demonstrate the reliability of our method and ranking results. The difference between the annotated dataset and the other three datasets may stem from the annotation rules: the publicly annotated dataset labels positive as 1, neutral as 0, and negative as -1. This labelling rule reduces the evaluation precision and does not fully align with the output format required for LLMs. Moreover, ERNIE-Lite-8K-0922 has failed to provide useful outputs during testing and acquired with no closeness coefficients.

6 Conclusions

In summary, we focus on the problem of evaluating the performance of LLMs in sentiment analysis tasks. First, we propose a set of comprehensive LLM evaluation criteria with detailed explanations and mathematical formulations. Based on the LLM evaluation criteria, the TOPSIS method is employed to evaluate the LLMs, which provides a quantitative and objective evaluation process from the perspective of MCDM. After that, we obtained the performance ranking of different LLMs in sentiment analysis. Finally, we explore the influence of prompts on the performance of LLMs, which is beneficial for users and developers to select the optimal combination of LLMs and prompts.

CCL 2025

[6] semeval-2013-dev.csv: https://www.kaggle.com/datasets/azzouza2018/semevaldatadets

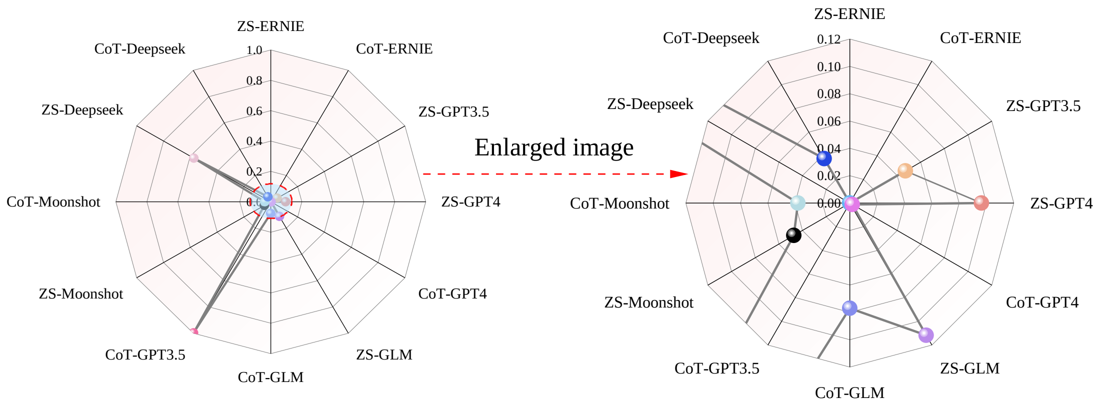

Figure 3: Closeness coefficients of LLMs on publicly annotated dataset.

7 Limitations

While this research provides valuable insights, some limitations still exist in this work. Firstly, the scale of LLMs and prompts are relatively limited, which potentially impacts the comprehensiveness of results. In the future, the number of LLMs and prompts can be increased to obtain more detailed findings. Secondly, the calculation of criterion stability may cause errors in the results due to fluctuations in network conditions, which may potentially affect the ranking results. Lastly, the criterion weight is determined by the amount of information entropy it contains and cannot be flexibly adjusted according to the confidence level of the criterion.

CCL 2025

References

Martin Aruldoss, T Miranda Lakshmi, and V Prasanna Venkatesan. 2013. A survey on multi criteria decision making methods and its applications. *American Journal of Information Systems*, 1(1):31–43.

Subrata Chakraborty. 2022. Topsis and modified topsis: A comparative analysis. *Decision Analytics Journal*, 2:100021.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Pengyu Chen. 2021. Effects of the entropy weight on topsis. *Expert Systems with Applications*, 168:114186.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2023. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. *arXiv preprint arXiv:2311.17667*.

Salvatore Corrente and Menelaos Tasiou. 2023. A robust topsis method for decision making problems with hierarchical and non-monotonic criteria. *Expert Systems with Applications*, 214:119045.

Shan Du. 2023. Hybrid kano-dematel-topsis model based benefit distribution of multiple logistics service providers considering consumer service evaluation of segmented task. *Expert Systems with Applications*, 213:119292.

Hanan Gani, Shariq Farooq Bhat, Muzammal Naseer, Salman Khan, and Peter Wonka. 2023. Llm blueprint: Enabling text-to-image generation with complex and detailed prompts. *arXiv preprint arXiv:2310.10640*.

Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024. Llm-based nlg evaluation: Current status and challenges.

Xinyu Hu, Mingqi Gao, Sen Hu, Yang Zhang, Yicheng Chen, Teng Xu, and Xiaojun Wan. 2024. Are llm-based evaluators confusing nlg quality criteria? *arXiv preprint arXiv:2402.12055*.

Hui Huang, Yingqi Qu, Jing Liu, Muyun Yang, and Tiejun Zhao. 2024. An empirical study of llm-as-a-judge for llm evaluation: Fine-tuned judge models are task-specific classifiers. *arXiv preprint arXiv:2403.02839*.

(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Ching-Lai Hwang, Kwangsun Yoon, Ching-Lai Hwang, and Kwangsun Yoon. 1981. Methods for multiple attribute decision making. *Multiple attribute decision making: methods and applications a state-of-the-art survey*, pages 58–191.

Muhammad Irfan, Rajvikram Madurai Elavarasan, Munir Ahmad, Muhammad Mohsin, Vishal Dagar, and Yu Hao. 2022. Prioritizing and overcoming biomass energy barriers: Application of ahp and g-topsis approaches. *Technological Forecasting and Social Change*, 177:121524.

Anjali Khurana, Hariharan Subramonyam, and Parmit K Chilana. 2024. Why and when llm-based assistants can go wrong: Investigating the effectiveness of prompt-based interactions for software help-seeking. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 288–303.

Zhao Li, Zujiang Luo, Yan Wang, Guanyu Fan, and Jianmang Zhang. 2022. Suitability evaluation system for the shallow geothermal energy implementation in region by entropy weight method and topsis method. *Renewable Energy*, 184:564–576.

Miao Li, Ming-Bin Chen, Bo Tang, ShengbinHou ShengbinHou, Pengyu Wang, Haiying Deng, Zhiyu Li, Feiyu Xiong, Keming Mao, Cheng Peng, et al. 2024a. Newsbench: a systematic evaluation framework for assessing editorial capabilities of large language models in chinese journalism. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9993–10014.

Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ninghao Liu, and Mengnan Du. 2024b. Quantifying multilingual performance of large language models across languages.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: A survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*.

Mengqi Miao, Fandong Meng, Yijin Liu, Xiao-Hua Zhou, and Jie Zhou. 2021. Prevent the language model from being overconfident in neural machine translation. *arXiv preprint arXiv:2105.11098*.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2023. State of what art? a call for multi-prompt llm evaluation. *arXiv preprint arXiv:2401.00595*.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Qian Pan, Zahra Ashktorab, Michael Desmond, Martin Santillan Cooper, James Johnson, Rahul Nair, Elizabeth Daly, and Werner Geyer. 2024. Human-centered design recommendations for llm-as-a-judge. *arXiv preprint arXiv:2407.03479*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya G Parameswaran, and Ian Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. *arXiv preprint arXiv:2404.12272*.

Wenyan Song, Huzhi Xue, and Wan Rong. 2025. An integrated method for resilient-sustainable supplier selection based on action-oriented practices. *Advanced Engineering Informatics*, 67:103570.

Fukang Sun and Junqi Yu. 2021. Improved energy performance evaluating and ranking approach for office buildings using simple-normalization, entropy-based topsis and k-means method. *Energy Reports*, 7:1560–1570.

Shuoqi Sun, Shengyao Zhuang, Shuai Wang, and Guido Zuccon. 2024. An investigation of prompt variations for zero-shot llm-based rankers. *arXiv preprint arXiv:2406.14117*.

Xiaoyi Tian, Amogh Mannekote, Carly E Solomon, Yukyeong Song, Christine Fry Wise, Tom Mcklin, Joanne Barrett, Kristy Elizabeth Boyer, and Maya Israel. 2024. Examining llm prompting strategies for automatic evaluation of learner-created computational artifacts.

CCL 2025

(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Li Wang, Xi Chen, XiangWen Deng, Hao Wen, MingKe You, WeiZhi Liu, Qi Li, and Jian Li. 2024. Prompt engineering in consistency and reliability with the evidence-based guideline for llms. *npj Digital Medicine*, 7(1):41.

Grant Wilkins, Srinivasan Keshav, and Richard Mortier. 2024. Offline energy-optimal llm serving: Workload-based energy models for llm inference on heterogeneous systems. *arXiv preprint arXiv:2407.04014*.

Dayu Yang, Fumian Chen, and Hui Fang. 2024. Behavior alignment: A new perspective of evaluating llm-based conversational recommendation systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2286–2290.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

Yue Zhang, Ming Zhang, Haipeng Yuan, Shichun Liu, Yongyao Shi, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Llmeval: A preliminary study on how to evaluate large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19615–19622.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. arxiv abs/2302.13439 (2023).

CCL 2025

Proceedings of the 24th China National Conference on Computational Linguistics, pages 818-830, Jinan, China, August 11-14, 2025.

(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China