Election-Based Evaluation Methods for NLG-Systems

Michał Tomasz Godziszewski[1,3,*], Mateusz Galiński[1], Rzyszard Tuora[1], Mateusz Czyżnikiewicz[1], Tomasz Ziętkiewicz[1,4], Adam Kozakiewicz[1] and Cristina Cornelio[2]

[1]*Samsung AI Center Warsaw*
[2]*Samsung AI Center Cambridge*
[3]*University of Łódź*
[4]*Adam Mickiewicz University in Poznań*

Abstract

Evaluating Natural Language Generation (NLG) systems is difficult: classical metrics are either reference-based or fail to capture important dimensions of quality such as factual accuracy or reasoning validity, while human evaluation is costly, slow, and sometimes inconsistent. Recent approaches that use Large Language Models (LLMs) as judges offer a scalable and reproducible alternative, but the standard methods such as single-answer grading and pairwise comparison suffer from significant limitations, including prompt design sensitivity, positional bias, global inconsistency, etc. To address these issues, we propose a voting-based evaluation method that prompts an LLM to rank all system outputs simultaneously, producing a complete preference ordering treated as a vote. By repeating this process under different random seeds and permutations and aggregating the resulting rankings with robust voting rules, our method mitigates positional bias, reduces noise, and enforces consistency and transitivity. Our experiments show that this approach is more reliable and efficient than existing LLM-based evaluation methods.

Keywords

evaluation, voting, computational social choice, RAG

1. Introduction

Evaluating the performance of Natural Language Processing (NLP) systems in a robust and meaningful way is a critical scientific challenge. This becomes even more difficult with generative models, where a unique "ground truth" answer often does not exist and output quality is difficult to define precisely. In recent years, NLP research has increasingly focused on generative tasks, where models produce open-ended text for applications such as translation, summarization, open-domain question answering, and dialogue systems. These tasks inherently involve balancing multiple quality dimensions (fluency, coherence, relevance, factual correctness, and sometimes creativity or style), which is challenging because there are often many plausible ways to fulfill a given request. As a result, there is no single gold-standard output – different valid answers can vary widely in wording, focus, or detail. This multiplicity of acceptable outputs, along with linguistic nuances like idiomatic expressions and pragmatic context, makes designing evaluation metrics particularly difficult. Existing approaches for evaluating NLG (Natural Language Generation) fall into three categories: classical (automated) metrics, human evaluation, and LLM (Large Language Model) based evaluation. Classical metrics cover a wide range of task-specific measures, such as exact match or F1 scores for QA, or BLEU/ROUGE for translation and summarization. While these metrics are useful in constrained settings, they also typically are reference-based and lack elasticity. Overlap-based scores like BLEU and ROUGE are somewhat more permissive, but they often fail to capture key aspects of quality such as factual accuracy and coherence. More

NL4AI 2026: Ninth Workshop on Natural Language for Artificial Intelligence, October 6-9th, 2026, Perugia, Italy
*Corresponding author.
mtgodziszewski@gmail.com (M. T. Godziszewski); m.galinski@samsung.com (M. Galiński); r.tuora@samsung.com (R. Tuora); m.czyznikiew@samsung.com (M. Czyżnikiewicz); tomasz.zietkiewicz@amu.edu.pl (T. Ziętkiewicz); a.kozakiewic@samsung.com (A. Kozakiewicz); c.cornelio@samsung.com (C. Cornelio)
0000-0002-3907-2242 (M. T. Godziszewski); 0009-0006-5375-9055 (M. Galiński); 0000-0001-9610-8048 (R. Tuora); 0009-0007-5248-208X (M. Czyżnikiewicz); 0000-0002-2594-4660 (T. Ziętkiewicz); 0000-0002-1377-5046 (A. Kozakiewicz); 0000-0001-5284-6487 (C. Cornelio)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-05

CEUR-WS.org/Vol-4315/paper26.pdf

broadly, classical metrics tend to focus on a single dimension (e.g., lexical similarity or precision/recall) while overlooking others like factual correctness or reasoning validity. As a result, they have low granularity for NLG evaluation and can be gamed by superficial strategies. Moreover, traditional metrics have shown only moderate correlation with human judgment on high-quality outputs [1].

Human judgment is often regarded as the ultimate standard for evaluating NLG systems. Annotators (domain expert or crowd workers) can assess outputs along multiple axes (e.g., fluency, correctness, relevance), or directly rank/score the outputs from different systems. For example, benchmarks like WMT use professional translators to rank translation outputs and annotate errors [2]. Obtaining reliable results, however, requires clear guidelines and sufficient training to avoid annotator bias and fatigue [3]. Even then, human evaluations can suffer from inconsistency: different annotators may disagree, and individuals may not remain consistent over time. Recent studies have also pointed out that even expert ratings are subject to variability and can be influenced by context and presentation [4]. While human evaluation provides rich and nuanced assessments, its high cost and variability usually restricts it to small sample sizes, which may not capture the full performance of a system.

More recently, with the advent of powerful LLMs, a new paradigm has emerged: using LLMs to evaluate the outputs of other models ("LLM-as-a-Judge"). This approach promises scalable, cheap, and reproducible evaluation, without relying on strict ground-truth references. In the literature "LLM-as-a-Judge" methods are usually implemented in two main forms, *Single-Answer Grading* and *Pairwise Comparison*, which differ in whether the LLM assigns an absolute score to a single output or instead makes a relative judgment between two candidate outputs. In more details, *Single-Answer Grading* leverages an LLM to score an individual output against specific criteria, either *directly*, or through a *multi-step* process that produces a single numeric score per answer [5]. In the direct setting, the LLM simply returns a number representing, for example, the comprehensiveness, or relevance of the answer, sometimes guided by chain-of-thought reasoning to better align with human values [6]. RAGAS (Retrieval Augmented Generation Assessment) [7], which is a collection of evaluation metrics, includes "Factual Correctness", which is a notable example of the *multi-step* setting. In this approach both the system answer and the reference are decomposed into individual semantic claims (claim decomposition), and then compared for NL-based entailment. These are then aggregated via different arithmetic functions to output the final scores (e.g., precision, recall or F1-score). The second approach, *Pairwise Comparison*, instead ask the LLM to compare two systems' outputs for the same query and determine which one is better (or whether they are tied). This method has been popularized in settings like chat model battles [8], Elo rating arenas [9], and MT (Machine Translation) benchmarks [10], where state-of-the-art LLMs serve as judges. Compared to single-output scoring, pairwise evaluation is often more discriminative, as the direct comparison between two outputs makes their differences more evident.

Although decomposition-based metrics may appear more sophisticated and explainable while remaining convenient, they have several critical weaknesses. Different runs of claim extraction can generate different sets of claims, leading to inconsistent scoring. [11] Moreover, single-output evaluation methods are highly sensitive to prompt engineering: the wording, choice of few-shot examples, and domain proximity can significantly bias both the decomposition, the entailment and thus the final scores. When outputs are lengthy and rich in content, claim decomposition becomes especially costly, as many claims must be extracted and verified, requiring many separate LLM calls (impacting the token usage). Another limitation is due to the fact that the extracted claims can vary drastically in information content (from trivial to substantive), yet they are treated as equally important to compute the final score. As a result, scoring can be dominated by uninformative checks, while crucial errors or omissions in the output remain overlooked or underrepresented. Finally, single-answer grading often leads to very coarse scores; in RAGAS Factual Correctness metric, for instance, only a couple of claims are typically extracted from an answer, so the final score becomes almost binary. This low resolution makes it difficult to distinguish fine gradations in quality.

Pairwise comparison methods also face their own set of problems. One common problem is the LLM's bias on the order of the items in the prompt [12][13], as an LLM preference depends on the order in which candidate outputs are presented. Recent studies [12] found that even with careful

prompt engineering, the agreement when swapping candidate order is only around 89% in the best case. Another issue is the inconsistency across comparisons: since each comparison is made in isolation, the outcomes may not form a consistent global ranking [14]. It is possible to get cycles (loop in preferences, e.g., $a > b$, $b > c$, $c > a$) where each pairwise decision seems rational locally but together they are mutually inconsistent. In that case a separate algorithm, such as Elo [9] or TrueSkill [15], ranking systems designed to turn pairwise comparisons into consistent global rankings, must be applied which are themselves sensitive to noise in the presence of cycles and contradictory votes. Pairwise evaluation also scales poorly, as comparing N systems via pairwise comparisons requires $\mathcal{O}(N^2)$ LLM evaluations (to compare all ordered pairs) which quickly becomes impractical as the number of systems increases (common case in NLG). A further limitation is the lack of global context, since standard of what an LLM judge considers "better" might shift depending on the particular opponent in the pair. For example, system A might barely outperform B and lose by a large margin to C, yet a pairwise-only view cannot capture the relative magnitude of these differences.

2. LLM-as-a-Voter

Our novel evaluation method builds on ideas from computational social choice, particularly the use of *voting rules*. Before introducing our method, referred to as **LLM-as-a-Voter**, we provide a brief overview of the voting framework it relies on.

Voting framework. Suppose we have a set $C = \{c_1, c_2, \ldots, c_N\}$ of N candidates (for example, the outputs of N different NLG systems under evaluation). When presented with these outputs, an LLM judge produces a ranking, that is, a total order over C from the best to the worst candidate. We denote this ranking using a preference relation $\succ$, where $c_i \succ c_j$ means the judge prefers output c_i over c_j. We allow the possibility of ties in the ranking (the judge might consider two outputs equally good), but for simplicity we will assume a strict order unless stated otherwise. A profile P is a collection of M such preference rankings, $P = \{\succ^{(1)}, \succ^{(2)}, \ldots, \succ^{(M)}\}$, obtained from either multiple independent judges or multiple runs of the same judge. Intuitively, each ranking in the profile can be seen as a vote on the relative quality of the outputs. The goal of a voting rule is to aggregate the profile P into a single collective ranking or a winner. More formally, a voting rule is a function that, given a set of candidates and the preference profile, outputs an aggregated ranking of the set of candidates (or the winner of the election). In our context, for each query $i = 1, 2, \ldots, Q$ (a question to an NLG system) we obtain a profile consisting of M rankings, resulting in a total of $Q \cdot M$ virtual votes whose ordinal "preferences" we aggregate.

LLM-as-a-Voter procedure. Our novel evaluation, LLM-as-a-Voter extends comparative evaluation by enforcing transitive preferences. A judge LLM is asked to rank outputs from N systems with or without reference gold-standard output. Its response consists of a string representing the ordering of candidates from best to worst. For instance, given 5 candidates labeled a, b, c, d, e, the ranking $c\ db\ e\ a$ indicate that c is the best candidate, a is the worst, b and d are considered indistinguishable (tied) and occupy the second place, while e is the second-to-worst. This method scales linearly with the number of systems N under evaluation, and provides a total ordering of the candidates. Suppose we have N NLG systems evaluated on Q tasks (e.g., natural language questions). To achieve position invariance we consider M random permutations of the N outputs for each task. Each permutation corresponds to a different ordering of the outputs in the prompts (i.e., a different way of presenting them to the vote LLM). The final system ranking is obtained through a two-step aggregation: First, for each task, the LLM judge produces ranking for all M permutations of the outputs from N systems. These M rankings form the first election, on which we apply a voting rule to generate a single winner ranking for each task. Second, the winner rankings across the Q tasks, form the second election on which we apply the (same) voting rule again, producing the final system ranking across tasks. In principle, it is possible that we have multiple winners. In such case, our method output them all as ex aequo winners.

This procedure guarantees a transitive ordering of systems performance across the tasks, while requiring exactly $M \cdot Q$ calls to the judge LLM. Figure 1 illustrates this process for an individual

question.

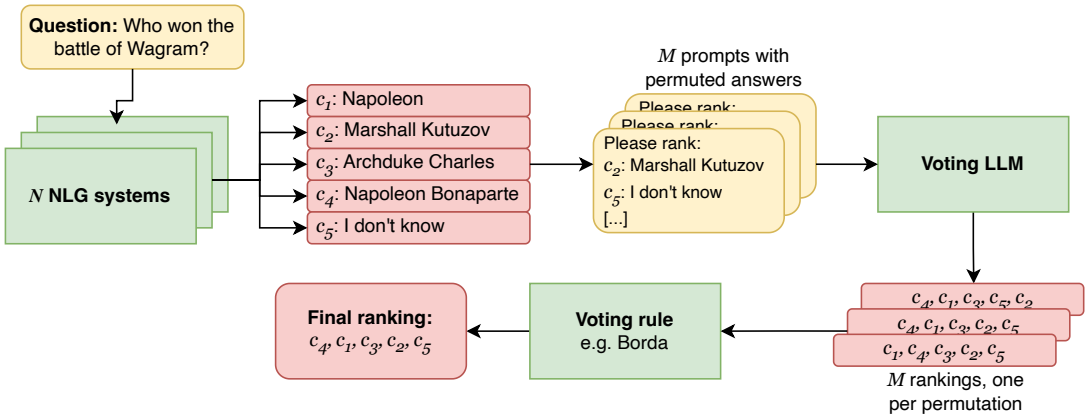


Figure 1: The design of the LLM-as-a-Voter system on an individual question. For a given task (e.g., a natural language question) an LLM ranks the outputs $(c_1, c_2, \ldots, c_N)$ of N candidate NLG systems under M different answer permutations. Each ranking is treated as a vote, and the set of votes is aggregated using a voting rule such as Borda count to produce a final system ranking.

3. Experimental setup

3.1. Voting Rules

In our experiments we focus on two voting rules, Borda and Copeland, for aggregating LLM-generated rankings, though our framework is more general and can accommodate also other voting rules.

Borda Count. The Borda count is a scoring rule that assigns a score to each candidate based on its position in each vote's ranking. In a ranking of N candidates, a candidate gets $N-1$ points from a vote if it is ranked first, $N-2$ points if ranked second, and so on, down to 0 points if ranked last. Formally, if $\text{pos}_v(c)$ gives the position of candidate c in vote v's ordering (where 1 is top-ranked and N is last), the Borda score of c is: $S_{\text{Borda}}(c) = \sum_{v=1}^{M} \big(N - \text{pos}_v(c)\big)$. This is equivalent to summing, over all votes, the number of other candidates that each vote ranked below c. For example, if there are $N = 5$ outputs and a particular vote ranks c in second place, c earns $5 - 2 = 3$ points from that vote. All the Borda points from all M rankings in the profile are totaled to get $S_{\text{Borda}}(c)$. The candidates can then be ordered by their total score, with a higher Borda score indicating a better overall ranking. Borda count produces a complete transitive ranking of the candidates (ties in score are possible but uncommon with enough votes). It captures the overall support a candidate receive across the profile; in our setting, this means favoring outputs that are consistently near the top of the LLM's rankings across many prompt permutations.

Copeland's Method. Given a profile, we can derive pairwise comparison counts for any two candidates. Let N_{ij} denote the number of rankings in P where $c_i \succ c_j$. Similarly, N_{ji} is the count preferring c_j over c_i. The majority margin between c_i and c_j is $m_{ij} = N_{ij} - N_{ji}$. If $m_{ij} > 0$, a majority of the votes prefer c_i to c_j; if $m_{ij} < 0$, they prefer c_j to c_i; and $m_{ij} = 0$ indicates a tie in the profile's preferences. Copeland's method is a pairwise majority rule where each candidate is evaluated by its win-loss record in pairwise comparisons against all other candidates. For each pair of candidates (c_i, c_j), we determine who wins the majority vote: if $N_{ij} > N_{ji}$ then c_i wins over c_j, and if $N_{ij} < N_{ji}$ then c_i loses to c_j (in case of a tie $N_{ij} = N_{ji}$, one can assign a half-win to each, but we omit tie-handling here for simplicity). The Copeland score of candidate c_i is then defined as: $S_{\text{Copeland}}(c_i) = \sum_{j \neq i} \Big(\chi(\{N_{ij} > N_{ji}\}) - \chi(\{N_{ij} < N_{ji}\})\Big)$, where $\chi(\{\cdot\})$ is an indicator function. In words, $S_{\text{Copeland}}(c_i) = W_i - L_i$ where W_i is the number of other candidates that c_i beats in a majority vote, and L_i is the number that beat c_i. A win contributes $+1$ to the score and a loss -1; (with an equal split for ties, $+0$ each). The candidate with the highest Copeland score is the overall winner under this

rule, and one can rank all candidates by their scores. Copeland's method thus identifies candidates that win many direct comparisons. In our setting it means that we focus on pairwise victories as the currency of success. This method yields a transitive ranking as well.

Borda and Copeland rules provide a principled way to combine multiple preference rankings into a single outcome. By using voting rules, we ensure the final evaluation is transitive (unlike arbitrary pairwise LLM judgments) and gains stability from multiple independent votes. We chose Borda and Copeland because they balance different criteria (overall consensus vs. majority wins) and have desirable properties suitable for our use case. Borda count captures a global consensus order and is less sensitive to outlier judgments, while Copeland's method aligns with the intuitive notion of *winning more comparisons* and will pick out a model that consistently outperforms others. Both methods are easy to compute and scale to many candidates, and they make no assumptions about the votes beyond having preference rankings, which fits our scenario of treating LLM outputs as preference votes.

3.2. Datasets

To evaluate our method, we test it on three datasets, each offering a different type of data:

MT-Bench is a dataset derived from LLM Arena [16] and consists of questions paired with a number of head-to-head human judgments over pairs of model outputs. We restrict the evaluation to three models (`vicuna-13b-v1.2`, `gpt-3.5-turbo`, `llama-13b`) chosen to maximize the pairwise coverage, resulting in a subset of 47 questions with a total of 287 individual judge preferences.

RewardBench 2 consists of (question, N model answers) pairs, each paired with human judgment identifying the best model's answer. A limitation of this dataset is that it provides only a single human-preferred answer per question rather than a full ranking. To ensure comparability, we select the largest subset of questions answered by the same set of models. This results in 59 questions, with answers from `Llama-3.1-70B-Instruct`, `claude-3.5-sonnet-20241022` and `gpt-4o-2024-08-06`.

WMT is an annual shared-task that focuses on evaluating general MT capabilities. For our experiments we use the MQM-based human scores released for WMT20 [17] and WMT23 [18] translation tasks[1], covering English-German and Chinese-English. Each translation (i.e., each *task* in our terminology) receives a numerical MQM severity-weighted score; by averaging these scores over all sentences in a subset we obtain a single quality indicator for every system, which we then use to construct a global ranking of the participating models. The four subsets contains 10–16 systems and between 400 and 2,000 translations, providing a balanced mix of different languages and text characteristics (e.g., shorter news snippets in 2020 and longer paragraphs in 2023). In particular, the 2020 data, collected before the widespread deployment of LLMs, supplies a diverse collection of traditional NLG systems for comparison.

3.3. Baselines

In our experiments we compare LLM-as-a-Voter against two LLM-as-a-judge baselines. The first one is a simple head-to-head LLM for **pairwise comparisons**, where we aim to reduce the impact of position bias by considering multiple calls per each permutation. The second is **Rubrics Score**, a task-agnostic single-answer-grading procedure implemented in RAGAS evaluation framework [7]. It involves listing a hierarchy of grades, with each grade being associated with a description in natural language, and asking the LLM to assign the answer to one of these grades. For all methods We use GPT-OSS-120B [2] as the judge LLM.

[1] https://github.com/google/wmt-mqm-human-evaluation
[2] https://github.com/openai/gpt-oss

3.4. Metrics

To assess how well our system-generated rankings match human judgments we employ three measures: *consistency*, *accuracy*, and *Kemeny distance*. **Consistency** measures the extent to which preferences deduced from our system agree with human judgments of preference over pairs of model outputs. For this metric we simply calculate the proportion of individual judgments for which the LLM-ranking-derived decision is identical. Since human judgments may themselves be inconsistent, achieving 100% agreement is often not possible, and we therefore also report a theoretical maximum. **Accuracy** is used to evaluate whether the human-preferred candidate also appears at the top of the ranking of a given method (ignoring the full ranking). In cases of ties on the first place, we assign $1/k$ points where k is the number of such tied-systems in the system outputs (which equals 1 when there is a unique winner). The final accuracy is the average of points over the set of questions. **Kemeny distance** (sometimes called Kendall's tau distance) is used to quantify ranking differences based on the number of pairwise disagreements between them. It is defined as follows: given two (possibly weak) rankings ρ and ζ over the set of candidates, the Kemeny distance between them is the minimum number of swaps needed to transform one ranking into the other. Equivalently, it is calculated as the number of pairs of items that are ranked in the opposite order in σ and ζ. More formally: $\bar{d}_K(\rho, \zeta) = \sum_{x \succ_\rho y} \chi(\{y \succ_\zeta x\})$. In our application, we allow ties (weak orderings) in system rankings. To handle ties, we use the following generalized Kemeny distance:

$$\begin{aligned} d_K(\rho, \zeta) \; = \; & \sum_{x \succ_\rho y} \big(\chi(\{y \succ_\zeta x\}) + \\ & + \frac{\chi(\{y \equiv_\zeta x\})}{2}\big) + \sum_{x \equiv_\rho y} \big(\frac{\chi(\{y \not\equiv_\zeta x\})}{2}\big), \end{aligned}$$

where $x \equiv_\rho y$ means that x and y are tied in the ranking ρ. The function above is equivalent to counting the cardinality of the symmetric difference of the (weak) orders ρ and ζ (treated as non-strict orderings, i.e., as a *less or equal* relation) and dividing it by 2. In our experiments we additionally normalize the Kemeny distance by dividing its original version by its maximal value $\frac{n(n-1)}{2}$.

Moreover, to situate our ranking approach with respect to WMT, we outline the human-centered quality metrics we compare against [17], namely the *Scalar Quality Metric (SQM)* and the *Multidimensional Quality Metric (MQM)*. **SQM** evaluates translation quality on a 0-6 scale, assessing both meaning preservation and grammatical correctness within document context. A score of 0 corresponds to nonsense with no meaning preserved, while 6 denotes perfect meaning and grammar. **MQM** evaluates translation quality by explicitly identifying and categorizing errors within document segments, with attention to context. Annotators highlight errors, label them with categories (e.g., Accuracy, Fluency, Terminology) and rate their severity (Major, Minor, Neutral). Errors are then weighted with numerical scores: Minor errors count as 1, Major errors as 5, and Non-translation as 25, with Minor Fluency/Punctuation errors weighted as 0.1. Segment-level scores range from 0 (perfect) to 25 (worst), and are averaged across annotators.

4. Results

The main goal of our experiments is to demonstrate that our new evaluation framework is both valid and consistent. By validity, we mean that the evaluation outcomes should align closely with human judgments, and by consistency, we mean that the framework should remain robust under noise and invariant to arbitrary transformations (e.g., permutations). In what follows, we present detailed experimental results with respect to these dimensions.

MT-Bench In the question/answering setting we define Q as the number of questions evaluated, and as a reminder, N as the number of system outputs per question, and M as the maximum number of permutations of these N outputs within the prompt.

To measure how well each method's results align with human judgments in MT-Bench, we use the following evaluation procedures, adapted to the system under consideration.

For LLM-as-a-Voter we check on each of the individual pairwise human preferences defined over one of the Q questions if the ranking of N systems obtained is consistent with them. This requires strict superiority of the chosen model in the ranking, or occupying the same place in the ranking if the human decided on a tie. The dataset also includes 25.1% judgments where human judge is indifferent and as such gives a tie judgment between the systems.

For Head-to-Head LLM-as-a-Judge in order to make it resistant to variant ordering we employ the following procedure. On each question we find all 2 element permutations of N systems. This in turn yields twice each combination since once its present as $(1, 2)$ and second time as $(2, 1)$. Then we show each combination $\lfloor \frac{M}{2} \rfloor$ times to the LLM, in the end each combination of system answers is being shown M times to the LLM judge. Once we have that, to calculate the consistency of this method's judgments with human judgments, we aggregate across each question Q and the 2 element output combinations to find the most common occurring system as the preferred one by the LLM judge (mode). Since M is even, it is possible that the judge LLM chose exactly $\frac{M}{2}$ times one system output and exactly $\frac{M}{2}$ the other system output. In such cases we assign tie judgment for this question, output combination pair. There are 10 questions, output combination pairs that meet this criteria out of the $47 \cdot 3$ considered (we have 47 questions and each model output is presented in 3 combinations). Then to calculate the final model consistency with the benchmark we check for each human judgment, if the verdict of that judge, on that output pair, was the same as this method's judgment on that output pair.

For Rubrics Score, we use RAGAS to obtain a 1–5 grade per output. These grades are then used to sort the outputs into a per-question ranking. Because of low granularity, the rankings contain much more ties than in other methods. To calculate the final consistency, we verify for each question how many of the human judgments are in agreement with the ranking constructed this way, and average this over all the questions.

We report the final comparison across methods as well as the theoretically achievable maximum and expectation in Table 1. Our system shows slightly higher performance than the competition, while achieving higher stability across runs and lower computational cost.

Table 1
Consistency of our method compared to the baselines on MT-Bench, showing mean (μ), variance (σ^2) and computational cost (as number of LLM calls), averaged over 5 independent runs.

Method	Consistency μ	Consistency σ^2	Computational Cost
LLM-as-a-Voter (our)	66.06 %	$1.09e^{-2}$ %	$Q \cdot M$
Head-to-Head LLM-as-a-Judge	65.51 %	$1.94e^{-2}$ %	$Q \cdot (N^2 - N) \cdot \lfloor \frac{M}{2} \rfloor$
Rubrics Score	65.37 %	$3.80e^{-2}$ %	$Q \cdot N$
Theoretical Max	83.62 %	—	—
Expected Value	34.60 %	—	—

RewardBench 2 For RewardBench 2 we are interested in the accuracy of selecting the best model. For LLM-as-a-Voter and Rubrics Score, we employ the same methods as above to derive the ranking of the outputs. For Head-to-Head LLM-as-a-Judge, all pairs are compared in both permutations, and a ranking is derived from pairwise comparisons using Copeland's method. Having the rankings ready, we then compare the first place in the ranking with the human-chosen best model. It is important to note, that human-annotated data does not allow for ties, but they do occur in all of the systems under evaluation. We report the final comparison across methods in Table 2. Similarly to the previous experiment, we are able to achieve the highest accuracy, while maintaining stability across runs at a low cost.

Table 2
Accuracy of our method compared to the baselines on RewardBench 2 (setup as in Table 1).

Method	Accuracy μ	Accuracy σ^2	Computational Cost
LLM-as-a-Voter (our)	81.52 %	$1.26e^{-2}$ %	$Q \cdot M$
Head-to-Head LLM-as-a-Judge	82.03 %	$3.62e^{-2}$ %	$Q \cdot (N^2 - N) \cdot \lfloor \frac{M}{2} \rfloor$
Rubrics Score	72.43 %	$2.21e^{-2}$ %	$Q \cdot N$

WMT To evaluate the validity of the proposed framework we benchmark it against the MQM-based human preference rankings released for WMT20 and WMT23. For every subset we generate system-level rankings with our two-step preference-aggregation pipeline (using either Borda or Copeland) and compare them with the rankings produced by the baseline Rubrics Score method. Fidelity to the gold-standard MQM ranking is quantified with the normalized Kemeny distance; for the WMT20 experiments we additionally compute Kemeny distances for rankings derived from the SQM human-preference metric, to provide an extra point of reference. Moreover, we consider two experimental conditions: (i) *reference-inclusive* where the human reference translations were added to the candidate pool as additional systems; and (ii) *reference-exclusive* where the references were omitted and only automatically generated outputs were ranked.

Table 3
Normalized Kemeny distance for system rankings for WMT datasets in comparison to MQM ranking. *r-incl* includes human reference translations, while *r-excl* excludes them and rank only system outputs.

	SQM		Borda		Copeland		Rubrics	
	r-incl	r-excl	r-incl	r-excl	r-incl	r-excl	r-incl	r-excl
WMT20_ende	0.022	0.048	0.444	0.095	0.489	0.143	0.567	0.262
WMT20_zhen	0.011	0.018	0.467	0.214	0.489	0.250	0.700	0.554
WMT23_ende	-	-	0.090	0.030	0.083	0.023	0.083	0.023
WMT23_zhen	-	-	0.083	0.095	0.083	0.095	0.058	0.067

The results are shown in Table 3. They reveal several important patterns. In the reference-inclusive setting both LLM-based approaches (our method and Rubrics Score) assign the human reference a considerably lower rank than human annotators do, indicating that the current LLM judge tends to undervalue human-written translations when they are mixed with model outputs (see Section 4.2 for more details). Despite this systematic bias, our method consistently yields smaller normalized Kemeny distances than Rubrics Score on the WMT20 data, demonstrating a closer alignment with the MQM gold ranking. For the newer WMT23 data the two methods achieve comparable Kemeny distances, and, importantly, the absolute distances are substantially lower for WMT23 overall, which makes the head-to-head comparison less discriminative.

4.1. Ranking stability

We first evaluate the stability of our election-based ranking method under repeated trials and varying prompt orders. Specifically, we consider two questions: (1) How consistent is the final aggregated ranking of systems when we repeat the entire evaluation process (with independent random seeds) multiple times? (2) For a given query, how similar are the per-query rankings (obtained from different LLM prompt permutations) to the ranking obtained on that query via election function. For this experiment we use a subset of the *RewardBench 2* dataset (from LLM Arena user preferences) where a

fixed set of three models have outputs for 59 questions. For the first question, we ran the full election procedure 5 times and recorded each model's aggregate score under the Borda and Copeland voting rules. Table 4 shows the mean (μ) and variance (σ^2) of each model's score across runs. The results indicate that randomness in LLM outputs has negligible effect on the final winner ranking. For the second question, we examine how each task's ranking, obtained from multiple LLM queries with different orderings of system answers in the prompt, deviates from the ranking obtained by applying election function on the aforementioned rankings. To do so we measure the average Kemeny distance between the rankings obtained from permutations of the LLM query and the resultant ranking on that task obtained via election. We then report those average Kemeny distances across all the tasks in 5 runs (with different random seeds) in order to account for randomness. This can be observed on Figure 2, which shows that in case of both election functions (Borda and Copeland) the rankings we obtain from permutations are close to the rankings obtained via elections.

Table 4
Stability of final rankings on subset of RewardBench 2. For each model we show the mean μ and (variance σ^2)

Voting rule	Claude μ (σ^2)	LLaMa μ (σ^2)	GPT μ (σ^2)
Borda	76.9 (1.34)	65.2 (2.16)	34.9 (0.74)
Copeland	2 (0)	0 (0)	-2 (0)

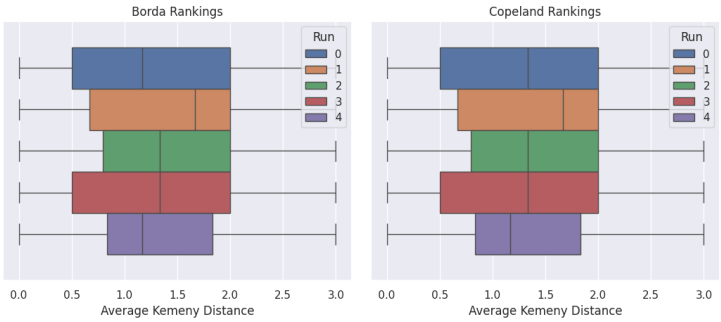


Figure 2: Stability and spread of permutation rankings to the rankings on each task on subset of RewardBench 2

4.2. Discrepancies Between LLM-Based and Human Evaluation

Our experiments show that while the election-based method produces internally consistent and transitive rankings, it does not always align with human evaluations. It seems that often LLM judges favor machine-generated outputs over human references. For example, in the WMT 2023 English–German benchmark, our method ranked a GPT-4 translations above the human reference, whereas MQM human annotation placed the reference first. We observe a similar pattern in WMT 2020, where aggregated LLM rankings pushed human translations below several machine systems. To highlight this effect, in Table 3 we report results both with and without including human references, showing how their presence can shift the evaluation. Since these differences appear systematic, they likely stem from underlying dynamics that distinguish LLM evaluation from human judgment. LLM evaluators often weigh errors differently than humans, overlooking nuances or penalizing minor issues too harshly, and they tend to reward consistency and formal style that resemble model outputs more than human translations. Their judgments can also reflect training-induced self-preference and a lack of domain knowledge, leading them to overvalue fluent but inaccurate content. As a result, our method sometimes ranks AI outputs above nuanced human references. However, this is not unique to our method but a broader limitation of LLM-based evaluation approaches.

5. Conclusions

To summarize, our election-based evaluation aligns with human judgment on many aspects and demonstrate a high degree of stability. T he results are encouraging in spite of the scarcity of data. We argue that the strengths of the election-based approach such as high validity, transitivity, scalability, objectivity, and computational tractability, make it a valuable and promising tool for evaluating NLG systems.

6. Limitations

Our study has several limitations. First, although the proposed election-based framework is general, our empirical evaluation is still limited in scale and scope. We evaluate the method on three datasets, covering question answering, preference modelling, and machine translation, but each experimental setting has its own restrictions. In MT-Bench, we use a subset of questions and models selected to maximize pairwise human-judgment coverage. In RewardBench 2, the available annotations identify only a single human-preferred answer rather than a full human ranking. In the WMT experiments, system-level rankings are derived from aggregated MQM scores, which provide a strong reference point but do not capture all possible dimensions of translation quality.

Secondly, we consider only a limited set of voting rules, namely Borda and Copeland. These rules were chosen because they are simple, scalable, and represent two natural aggregation principles: global positional consensus and pairwise majority success. However, social choice theory offers many other rules, each with different axiomatic and practical properties. Some rules may be more robust to outlier rankings, strategic or adversarial outputs, near-ties, or polarized preference profiles. We therefore do not claim that Borda or Copeland are optimal for all NLG evaluation scenarios. A more systematic comparison of voting rules, including their failure modes under noisy LLM preferences, remains an important direction for future work.

Finally, our evaluation focuses primarily on agreement with available human judgments and on ranking stability. These are important desiderata, but they do not exhaust all aspects of evaluation quality. We do not directly measure, for example, explanation quality, calibration of confidence, sensitivity to adversarially constructed answers, or the ability to detect subtle factual and reasoning errors. The proposed method should therefore be viewed as a complementary evaluation framework: it improves the aggregation of LLM preferences into stable and transitive rankings, but it does not eliminate the need for task-specific validation, expert analysis, and careful benchmark design.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] B. Peng, X. Li, L. Li, J. Gao, A. Celikyilmaz, S. Lee, K.-F. Wong, Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2017. URL: http://dx.doi.org/10.18653/v1/D17-1237. doi:10.18653/v1/d17-1237.

[2] T. Kocmi, E. Avramidis, R. Bawden, O. Bojar, A. Dvorkovich, C. Federmann, M. Fishel, M. Freitag, T. Gowda, R. Grundkiewicz, B. Haddow, M. Karpinska, P. Koehn, B. Marie, C. Monz, K. Murray, M. Nagata, M. Popel, M. PopoviÄ‡, M. Shmatova, S. SteingrÃmsson, V. Zouhar, Findings of the wmt24 general machine translation shared task: The llm era is here but mt is not solved yet, in: Proceedings of the Ninth Conference on Machine Translation, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 1–46. URL: https://aclanthology.org/2024.wmt-1.1.

[3] G. H. Chen, S. Chen, Z. Liu, F. Jiang, B. Wang, Humans or LLMs as the judge? a study on judgement bias, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 8301–8327. URL: https://aclanthology.org/2024.emnlp-main.474/. doi:10.18653/v1/2024.emnlp-main.474.

[4] T. Hosking, P. Blunsom, M. Bartolo, Human feedback is not gold standard, 2024. URL: https://arxiv.org/abs/2309.16349. arXiv:2309.16349.

[5] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, C. Zhu, G-eval: NLG evaluation using gpt-4 with better human alignment, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023. URL: https://openreview.net/forum?id=puMfaHb1hY.

[6] S. Saha, X. Li, M. Ghazvininejad, J. Weston, T. Wang, Learning to plan & reason for evaluation with thinking-llm-as-a-judge, 2025. URL: https://arxiv.org/abs/2501.18099. arXiv:2501.18099.

[7] S. Es, J. James, L. Espinosa Anke, S. Schockaert, RAGAs: Automated evaluation of retrieval augmented generation, in: N. Aletras, O. De Clercq (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, St. Julians, Malta, 2024, pp. 150–158. URL: https://aclanthology.org/2024.eacl-demo.16/. doi:10.18653/v1/2024.eacl-demo.16.

[8] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, B. Zhu, H. Zhang, M. I. Jordan, J. E. Gonzalez, I. Stoica, Chatbot arena: an open platform for evaluating llms by human preference, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024.

[9] M. Boubdir, E. Kim, B. Ermis, S. Hooker, M. Fadaee, Elo uncovered: Robustness and best practices in language model evaluation, in: S. Gehrmann, A. Wang, J. Sedoc, E. Clark, K. Dhole, K. R. Chandu, E. Santus, H. Sedghamiz (Eds.), Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), Association for Computational Linguistics, Singapore, 2023, pp. 339–352. URL: https://aclanthology.org/2023.gem-1.28/.

[10] M. Freitag, N. Mathur, D. Deutsch, C.-K. Lo, E. Avramidis, R. Rei, B. Thompson, F. Blain, T. Kocmi, J. Wang, D. I. Adelani, M. Buchicchio, C. Zerva, A. Lavie, Are LLMs breaking MT metrics? results of the WMT24 metrics shared task, in: B. Haddow, T. Kocmi, P. Koehn, C. Monz (Eds.), Proceedings of the Ninth Conference on Machine Translation, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 47–81. URL: https://aclanthology.org/2024.wmt-1.2/. doi:10.18653/v1/2024.wmt-1.2.

[11] H. Qisheng, L. Quanyu, W. Wenya, Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance?, 2025. URL: https://arxiv.org/abs/2411.02400. arXiv:2411.02400.

[12] L. Shi, C. Ma, W. Liang, X. Diao, W. Ma, S. Vosoughi, Judging the judges: A systematic study of position bias in llm-as-a-judge, 2025. URL: https://arxiv.org/abs/2406.07791. arXiv:2406.07791.

[13] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, B. Lin, Y. Cao, Q. Liu, T. Liu, Z. Sui, Large language models are not fair evaluators, 2023. URL: https://arxiv.org/abs/2305.17926. arXiv:2305.17926.

[14] Y. Xu, L. Ruis, T. Rocktäschel, R. Kirk, Investigating non-transitivity in llm-as-a-judge, 2025. URL: https://arxiv.org/abs/2502.14074. arXiv:2502.14074.

[15] R. Herbrich, T. Minka, T. Graepel, Trueskill™: A bayesian skill rating system, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, volume 19, MIT Press, 2006. URL: https://proceedings.neurips.cc/paper_files/paper/2006/file/f44ee263952e65b3610b8ba51229d1f9-Paper.pdf.

[16] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL: https://arxiv.org/abs/2306.05685. arXiv:2306.05685.

[17] M. Freitag, G. Foster, D. Grangier, V. Ratnakar, Q. Tan, W. Macherey, Experts, errors, and context: A large-scale study of human evaluation for machine translation, Transactions of the Association for Computational Linguistics 9 (2021) 1460–1474. URL: https://aclanthology.org/2021.tacl-1.87/. doi:10.1162/tacl_a_00437.

[18] M. Freitag, N. Mathur, C.-k. Lo, E. Avramidis, R. Rei, B. Thompson, T. Kocmi, F. Blain, D. Deutsch, C. Stewart, C. Zerva, S. Castilho, A. Lavie, G. Foster, Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent, in: P. Koehn, B. Haddow, T. Kocmi, C. Monz (Eds.), Proceedings of the Eighth Conference on Machine Translation, Association for Computational Linguistics, Singapore, 2023, pp. 578–628. URL: https://aclanthology.org/2023.wmt-1.51/. doi:10.18653/v1/2023.wmt-1.51.