Transforming User-Defined Criteria into Explainable Indicators with an Integrated LLM–AHP System

Anonymous ACL submission

Abstract

Evaluating complex texts across domains requires converting user-defined criteria into quantitative, explainable indicators, which is a persistent challenge in search and recommendation systems. Single-prompt LLM evaluations suffer from complexity and latency issues, while criterion-specific decomposition approaches rely on naive averaging or opaque black-box aggregation. We present an interpretable aggregation framework combining LLM scoring with the Analytic Hierarchy Process (AHP). Our method generates criterion-specific scores via LLM-as-judge, measures discriminative power using Hellinger distance, and derives statistically grounded weights through AHP pairwise comparison matrices. Experiments on Amazon review helpfulness prediction, summarization quality assessment, and depression-related text scoring demonstrate that our approach achieves high explainability and operational efficiency while maintaining predictive power comparable to black-box alternatives, making it suitable for latency-sensitive web services.

1 Introduction

Text quality evaluation is a core component of many web and data mining applications, including review recommendation (Moghaddam et al., 2011; Zheng et al., 2017; Qu et al., 2021; Chen et al., 2022), content moderation (Qiao et al., 2024; Huang, 2025), and survey analysis (McGillivray et al., 2020; Mellon et al., 2024). Traditional approaches rely on human annotations or heuristic metrics such as readability scores, which are costly, domain-specific, and limited in capturing semantic qualities like expertise, coherence, or persuasiveness (Louis and Nenkova, 2013).

Recent advances in large language models (LLMs) have enabled automated text evaluation frameworks such as G-Eval (Liu et al., 2023a), where LLMs score texts quality using Likert scales (Likert, 1932). However, practical deployment remains challenging. Small models lack capacity for complex judgments (Wang et al., 2025; Martinez et al., 2024), while large models incur prohibitive computational costs (Fernandez et al., 2025; Stojkovic et al., 2024). These constraints often require limiting model size or evaluation runs, which exacerbates score instability and bias.

A natural remedy is to decompose evaluation into per-criterion scores. When each LLM call focuses on a single explicit criterion rather than holistic quality, the scoring task becomes simpler and outputs more consistent. The question then shifts to how these criterion-level scores should be aggregated. Linear regression is a natural candidate. It directly optimizes coefficients to predict observed signals. Yet LLM-generated Likert scores frequently exhibit central tendency bias, with outputs concentrated near the scale midpoint (Rupprecht et al., 2025). This distributional compression distorts regression weights, undervaluing highly discriminative criteria (Lee and Chen, 2025), and normalization fails to recover missing variance in discrete scales (Bojić et al., 2025; Li et al., 2024). While nonlinear alternatives (random forests, neural networks) can better tolerate such distributional artifacts, they are inherently opaque. Either way, practitioners must choose between predictive power and interpretability.

To address these issues, we propose **UniScore**, an interpretable aggregation framework that integrates lightweight LLM-based criterion scoring with statistically grounded weighting. UniScore measures each criterion's discriminative power using Hellinger distance (Hellinger, 1909) and derives relative importance through the Analytic Hierarchy Process (AHP) (Saaty, 1980). By relying on pairwise, distribution-level comparisons rather than absolute score magnitudes, UniScore mitigates scale bias and produces signed, interpretable weights suitable for low-cost evaluation pipelines.

Our main contributions are:

- A principled integration of Hellinger distance and AHP for robust, interpretable aggregation of LLM-generated criterion scores.
- Empirical validation across multiple datasets, demonstrating stronger alignment with external quality signals than common baselines.
- Evidence of efficiency and interpretability, supporting real-time web deployment.

To our knowledge, UniScore is the first framework to transform distributional divergences between quality-conditioned score distributions into pairwise criterion importance, eliminating the need for absolute score calibration or subjective expert judgments in LLM-based evaluation settings.

2 Related Work

Predictive text quality evaluation plays a critical role in web applications such as review recommendation (Moghaddam et al., 2011; Zheng et al., 2017; Qu et al., 2021; Chen et al., 2022), content moderation (Qiao et al., 2024; Huang, 2025), and survey analysis (McGillivray et al., 2020; Mellon et al., 2024). Traditional heuristic metrics, including readability and lexical complexity (Kincaid et al., 1975), enable real-time evaluation but fail to capture deeper semantic qualities such as coherence, expertise, and persuasiveness (Louis and Nenkova, 2013). User feedback signals (e.g., helpful votes) provide reliable quality indicators (Zhang and Zhang, 2014; Singh et al., 2017), but are unavailable for newly generated content.

Despite recent progress in LLM-based evaluation frameworks (e.g., G-Eval (Liu et al., 2023a)), practical deployment remains challenging. LLM-based evaluators face a performance-cost trade-off: large models achieve high accuracy at prohibitive computational cost (OpenAI, 2025; Anthropic, 2025; Zhou et al., 2024), while smaller models are more efficient but less reliable (Zheng et al., 2023). Moreover, LLM scores are sensitive to prompt variations (Sclar et al., 2024) and are often impractical under strict latency constraints in real-time systems (Barros, 2025).

To improve reliability, recent work decomposes evaluation into multiple fine-grained criteria, such as FLASK (Ye et al., 2023). While effective for diagnosis, these methods typically do not yield a unified operational score. Aggregating multi-criteria scores remains challenging: naive averaging ignores criterion importance, while regression-based approaches suffer from skewed and compressed Likert-score distributions (Stureborg et al., 2024; Wang et al., 2023), violating statistical assumptions (Liddell and Kruschke, 2018) and reducing interpretability.

Some approaches incorporate human preference data to train aggregation models, such as HD-Eval (Liu et al., 2024), but these methods are primarily designed for offline benchmarking and lack transparency and efficiency for real-time deployment.

In contrast, our work adopts a Multi-Criteria Decision Analysis (MCDA) perspective, leveraging the Analytic Hierarchy Process (AHP) (Saaty, 1980) to derive relative criterion importance from a small amount of guiding signals. Unlike prior work that relies on LLMs to generate subjective pairwise judgments (Lu et al., 2024), UniScore derives weights from distributional differences measured via Hellinger distance, yielding a robust, interpretable, and data-driven aggregation framework suitable for low-latency applications.

3 Preliminaries

UniScore relies on two components: measuring discriminative power via information-theoretic distance, and deriving interpretable weights through structured decision-making.

3.1 Hellinger Distance

The Hellinger distance is a symmetric metric for quantifying differences between probability distributions (Hellinger, 1909). For discrete distributions P and Q:

$$H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_i (\sqrt{p_i} - \sqrt{q_i})^2}. \quad (1)$$

Unlike Kullback–Leibler divergence (Kullback and Leibler, 1951), Hellinger distance is a true metric with values bounded in $[0, 1]$, ensuring consistent scaling for AHP matrices. It offers key advantages: (1) numerical stability when probabilities approach zero, (2) geometric interpretability as Euclidean distance between square-root transformed distributions, and (3) robust empirical performance across diverse data (Section 5.4).

3.2 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) (Saaty, 1980) quantifies relative criterion importance through pairwise comparisons. The comparison matrix $A = [a_{ij}]$ is positive and reciprocal ($a_{ij} =$

$1/a_{ji}$, $a_{ii} = 1$). Weights are obtained as the normalized principal eigenvector:

$$A\,\mathbf{w} = \lambda_{\max}\,\mathbf{w}, \qquad \hat{\mathbf{w}} = \frac{\mathbf{w}}{\sum_{k=1}^{n} w_k} \tag{2}$$

where $\lambda_{\max}$ is the largest eigenvalue. Consistency is measured by the consistency ratio (CR), which should not exceed 0.1 (Saaty, 1990).

In our framework, pairwise comparisons are generated from Hellinger distances rather than expert judgments. While AHP traditionally aggregates multiple expert judgments via geometric means (Aczél and Saaty, 1983), our data-driven approach yields uniquely determined, reproducible ratio-scale weights compatible with standard eigenvector methods (Saaty, 1977).

4 Methods

In this section, we introduce our **UniScore** (Unified Scoring Framework). First, we describe the process of obtaining individual scores for user-defined criteria using LLM-based evaluation. Then, we explain how these criterion-level scores are integrated into a single interpretable scoring formula through AHP-based weight estimation. The overall architecture of this framework is illustrated in Figure 1.

4.1 Group Partitioning

UniScore requires two reference groups representing high and low quality examples, constructed via manual selection or automatic partitioning.

Manual Selection. A domain expert selects representative examples for each group:

$$G_{\text{low}} = \{x_i \mid \text{selected as low quality}\}, \quad G_{\text{high}} = \{x_j \mid \text{selected as high quality}\}. \tag{3}$$

Typically 50-200 samples per group suffice for stable weight estimation. This is useful when ground-truth labels are unavailable or quality is task-specific.

Automatic Partitioning. When a quality signal s_i is available, groups are constructed automatically. For categorical signals:

$$G_{\text{low}} = \{x_i \mid s_i = 0\}, \quad G_{\text{high}} = \{x_i \mid s_i = 1\}. \tag{4}$$

For continuous signals, a percentile threshold p extracts extreme groups:

$$G_{\text{low}} = \{x_i \mid s_i \le Q_p(S)\}, \quad G_{\text{high}} = \{x_i \mid s_i \ge Q_{1-p}(S)\} \tag{5}$$

where $Q_p(S)$ denotes the p-th percentile of signal set $S = \{s_i\}$.

Sampling Strategies. For efficiency and balance, we optionally apply: (1) *size balancing* to match $|G_{\text{low}}| = |G_{\text{high}}|$, or (2) *subsampling* to a target size n_{target} per group.

4.2 Criteria Scoring

UniScore employs LLM-based scoring for user-defined criteria

$$\mathcal{C} = \{c_1, c_2, \ldots, c_m\}, \tag{6}$$

where each c_k denotes a textual quality dimension. For each criterion, a prompt P_k instructs the LLM to return a 1–5 Likert score. For each sample x_i and criterion c_k:

$$s_{ik} = \text{LLM}(P_k, x_i) \in \{1, 2, 3, 4, 5\}. \tag{7}$$

For real-time deployment, we use lightweight LLMs with temperature 0 for deterministic outputs. Invalid responses are retried up to three times or assigned a neutral score of 3. For cost efficiency, scoring is applied only to the partitioned groups from Section 4.1:

$$\mathbf{s}_k^{\text{low}} = \left[s_{ik}\right]_{i \in \mathcal{I}_{\text{low}}}, \quad \mathbf{s}_k^{\text{high}} = \left[s_{ik}\right]_{i \in \mathcal{I}_{\text{high}}} \tag{8}$$

where $\mathcal{I}_{\text{low}}$ and $\mathcal{I}_{\text{high}}$ denote the index sets of each group.

For quantitative criteria (e.g., word count), we bypass LLM scoring and scale raw measurements to $[1, 5]$ using standard techniques (see Appendix A).

4.3 Distribution Estimation

In our framework, the importance of a criterion is defined by its ability to discriminate between texts associated with high and low quality signals. Accordingly, criteria whose score distributions differ more strongly between these groups are treated as more informative for quality assessment, and receive greater weight in the final aggregation. Based on this definition, we quantify each criterion's discriminative power by comparing score distributions across the two groups using Hellinger distance.

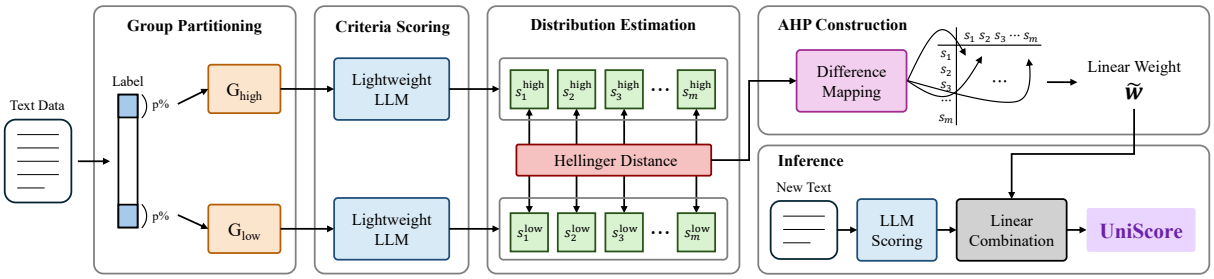

Figure 1: The overall architecture of the UniScore framework on continuous signals.

Given the group-wise score matrices from Section 4.2, $\mathbf{S}^{\text{low}}$ and $\mathbf{S}^{\text{high}}$, we estimate, for each criterion c_k, the empirical probability mass functions (PMFs) over the Likert levels $\mathcal{L} = \{1, 2, 3, 4, 5\}$ for the low and high groups. For each criterion k and Likert-scale value $v \in \mathcal{L}$, we define

$$\begin{aligned} n_k^{\text{low}}(v) &= \sum_{i \in G_{\text{low}}} \mathbf{1}[s_{ik} = v], \\ n_k^{\text{high}}(v) &= \sum_{i \in G_{\text{high}}} \mathbf{1}[s_{ik} = v]. \end{aligned} \tag{9}$$

These counts represent the score distributions for the two groups under criterion k.

With Laplace smoothing $\varepsilon > 0$ (we use $\varepsilon = 10^{-6}$) to avoid zero probabilities, the smoothed PMFs are

$$\begin{aligned} P_k^{\text{low}}(v) &= \frac{n_k^{\text{low}}(v) + \varepsilon}{\sum_{u \in \mathcal{L}} \left(n_k^{\text{low}}(u) + \varepsilon\right)}, \\ Q_k^{\text{high}}(v) &= \frac{n_k^{\text{high}}(v) + \varepsilon}{\sum_{u \in \mathcal{L}} \left(n_k^{\text{high}}(u) + \varepsilon\right)}. \end{aligned} \tag{10}$$

We quantify the discriminativeness of criterion c_k via the Hellinger distance:

$$d_k = \frac{1}{\sqrt{2}} \sqrt{\sum_{v \in \mathcal{L}} \left(\sqrt{P_k^{\text{low}}(v)} - \sqrt{Q_k^{\text{high}}(v)} \right)^2}. \tag{11}$$

For continuous (non-Likert) scores, we histogram both groups using common bin edges determined by Sturges' formula $\lceil \log_2(n) + 1 \rceil$ bins to ensure data-driven discretization. Users can adjust the binning method if desired. We then apply Laplace smoothing and normalize to obtain PMFs P_k^{low} and Q_k^{high}, and compute the Hellinger distance d_k as in Eq. (11).

To enable proper directional weighting in the final scoring formula (Section 4.4), we determine the direction from sample means.

$$\begin{aligned} \bar{s}_k^{\text{low}} &= \frac{1}{|G_{\text{low}}|} \sum_{i \in G_{\text{low}}} s_{ik}, \\ \bar{s}_k^{\text{high}} &= \frac{1}{|G_{\text{high}}|} \sum_{i \in G_{\text{high}}} s_{ik}. \end{aligned} \tag{12}$$

The direction indicator is:

$$sign_k = \text{sign}\left(\bar{s}_k^{\text{high}} - \bar{s}_k^{\text{low}}\right). \tag{13}$$

4.4 AHP Construction via Difference Mapping

To transform the criterion discriminativeness values into meaningful weights for the final indicator, we employ AHP, leveraging pairwise comparisons and the principal eigenvector method.

Using discriminativeness values $d_k \in [0, 1]$, we construct pairwise comparisons based on their differences, mapping each to a positive ratio scale for compatibility with the eigenvector method. This approach is grounded in the additive pairwise comparison framework (Barzilai and Golany, 1990; Lootsma, 1999), which models preference as differences ($a_{ij} = w_i - w_j$) rather than ratios ($a_{ij} = w_i / w_j$). Since our discriminative power values d_k are interval-scaled, difference-based comparison is theoretically appropriate (Kou et al., 2016; Cavallo and D'Apuzzo, 2009). The mapping $1 + 8\Delta_{ij}$ rescales these additive judgments to Saaty's $[1, 9]$ range while preserving compatibility with standard eigenvector methods. Empirical comparison with nonlinear alternatives (Section 5.4.1) confirms this choice.

Beyond theoretical grounding, difference-based comparison offers practical advantages: (1) it avoids numerical instability and ratio inflation when some d_j values approach zero, unlike ratio-based mappings (e.g., d_i / d_j); (2) it exhibits lower sensitivity to outliers than nonlinear mappings

enforcing strict multiplicative transitivity; and (3) it maintains an intuitive linear interpretation where larger differences correspond to stronger relative preferences, facilitating explanation to non-technical stakeholders.

For two criteria c_i and c_j, we define

$$\Delta_{ij} = d_i - d_j \in [-1,\, 1]. \tag{14}$$

The pairwise comparison entry is then

$$a_{ij} = \begin{cases} 1 + 8\,\Delta_{ij}, & \Delta_{ij} \geq 0, \\ \dfrac{1}{1 + 8|\Delta_{ij}|}, & \Delta_{ij} < 0, \end{cases} \qquad a_{ii} = 1. \tag{15}$$

This mapping ensures several desirable properties. Reciprocity is preserved exactly: $a_{ij} \cdot a_{ji} = 1$ for all $i \neq j$. The result adheres to Saaty's recommended range (Saaty, 1980) $a_{ij} \in [1/9, 9]$. The coefficient 8 maps maximum difference $\Delta_{ij} = 1$ to the upper bound 9, corresponding to slope $(9-1)/1 = 8$. When $\Delta_{ij} < 0$, the reciprocal form maps to $[1/9, 1)$. The linear form ensures interpretational stability: equal differences in $|\Delta_{ij}|$ translate to proportional preference intensities, enabling consistent weight interpretation across datasets.

Let $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times m}$ denote the comparison matrix. Following standard AHP, weights are obtained via the principal eigenvector:

$$A\mathbf{v} = \lambda_{\max}\mathbf{v}, \quad \mathbf{w} = \frac{\mathbf{v}}{\mathbf{1}^\top \mathbf{v}}, \quad \sum_{k=1}^{m} w_k = 1. \tag{16}$$

While linear mapping does not enforce strict multiplicative transitivity ($a_{ij} \cdot a_{jk} = a_{ik}$), the bounded range $[0, 1]$ and smooth distribution of Hellinger distances naturally produce near-transitive matrices. This is empirically confirmed in Appendix D.1, where all CR values remain well below 0.1 across 100 random samplings.

The final signed weights incorporate directional information:

$$\tilde{w}_k = sign_k \cdot w_k \tag{17}$$

ensuring criteria with higher scores in the high-quality group receive positive weights, while those higher in the low-quality group receive negative weights.

4.5 Inference

Once the signed weights $\tilde{\mathbf{w}} = (\tilde{w}_1, \ldots, \tilde{w}_m)$ are estimated, UniScore can be applied to new texts without re-partitioning or recomputing weights, enabling real-time deployment.

For a new text x^*, per-criterion scores $\hat{s}_k(x^*) \in [1, 5]$ are obtained using the same prompts P_k from Section 4.2. The final score is computed as:

$$\text{UniScore}(x^*) = \sum_{k=1}^{m} \tilde{w}_k\, \hat{s}_k(x^*) \tag{18}$$

where $\hat{\mathbf{s}}(x^*) = (\hat{s}_1(x^*), \ldots, \hat{s}_m(x^*))$.

Inference complexity is $O(m)$ per text, dominated by m LLM calls. The resulting score reflects each criterion's discriminative power (via $|w_k|$) and directional relationship with quality (via $sign_k$).

5 Experiments

To validate UniScore's effectiveness and generalizability, we design experiments addressing three research questions: (1) Does UniScore produce scores that correlate more strongly with ground-truth signals than baseline aggregation methods? (2) Is UniScore efficient enough for real-world web services under limited computational resources? (3) Are the AHP-derived weights reliable and consistent with domain knowledge? We conduct experiments with diverse baseline methods to ensure a rigorous and multifaceted evaluation.

5.1 Datasets

To evaluate the generalizability of UniScore, we use publicly available datasets with diverse characteristics, covering both continuous and discrete ground-truth signals. All datasets consist of naturally occurring human-generated text, reflecting realistic web deployment scenarios. Each dataset is split into training and test sets with an 80/20 ratio. Dataset statistics are provided in Appendix B.

Amazon Reviews. User-written product reviews where *helpful votes* serve as a continuous quality signal. We focus on the *Software* category for its technical depth and domain-specific language, sourced from the UCSD Amazon Review Dataset (Ni et al., 2019).

RoSE SAMSum. System-generated summaries with human-annotated ACU (Atomic Content Unit) scores as continuous signals, representing the proportion of reference information preserved (Liu et al., 2023b).

Depression Tweet. Tweets annotated as depressive or non-depressive, with binary labels as discrete signals. Selected based on its use as a benchmark in MentalHelp (Raihan et al., 2024).

5.2 Experimental Setup

For all experiments, we use Qwen3-1.7B (Yang et al., 2025) as our lightweight LLM scorer with a temperature of 0 for deterministic outputs.

5.2.1 Group Partitioning

For the continuous signal dataset (Amazon Software, ROSE SAMSum), we partition the texts into a G_{low} and G_{high} based on the top and bottom percentile of helpful votes. For discrete signal datasets (depression), the groups are naturally defined by their binary labels (G_{low} for label 0, G_{high} for label 1), and we sample up to n texts for each group to maintain computational feasibility.

5.2.2 Evaluation Criteria

To ensure a meaningful quality assessment, we design dataset-specific semantic criteria aligned with each dataset's ground-truth signal. All criteria reflect well-established quality dimensions in their respective domains. All criterion-specific prompts were systematically designed with the assistance of GPT-5.2 as a prompt engineering tool, following the definitions in Section 4.2. The full prompt templates are provided in Appendix E.

Amazon Reviews. We evaluate five dimensions commonly associated with review helpfulness: polarity, expertise, specificity, consistency, and word count(Appendix A) with $\sigma_{\text{scale}} = 2$, following prior work (Danescu-Niculescu-Mizil et al., 2009).

RoSE SAMSum. We adopt standard summarization quality dimensions aligned with ACU-based evaluation (Fabbri et al., 2021): coherence, fluency, relevance, and word count.

Depression Tweets. We assess linguistic markers of depression grounded in computational psychology, including negative affect, self-focus, absolutist thinking, and social isolation (Resnik et al., 2020; Chandra Guntuku et al., 2019).

5.3 Baselines and Metrics

To rigorously evaluate UniScore, we compare it against several baselines designed to isolate the benefits of our proposed weighting and aggregation method.

5.3.1 Baselines

To assess the effectiveness of UniScore, we compare it against a diverse set of baseline aggregation methods, including random weighting, linear regression, tree-based ensembles, and neural models. These baselines represent commonly used alternatives for aggregating multi-criteria scores and allow us to isolate the impact of our AHP-based weighting scheme. Implementation details for each baseline are provided in Appendix C.

5.3.2 Evaluation Metrics

We evaluate each method from three complementary perspectives: predictive power, discriminative ability, and score stability. Predictive power is measured using correlation metrics (Pearson r, Spearman ρ, Kendall τ). Discriminative ability is assessed via group separation tests and classification metrics (F1-score, accuracy). Score stability is evaluated using distributional statistics such as the coefficient of variation (CV) and skewness. Formal definitions and implementation details are provided in Appendix D.

5.4 Main Results

Our experimental results, summarized in Table 1, demonstrate that UniScore achieves the best or comparable performance across most settings without requiring additional training.

As summarized in Table 1, UniScore validates the aggregation-centric approach: AHP-based weighting yields the strongest or tied-strongest results without additional learning. Statistical significance was assessed via paired t-tests across 5 random seeds. On Amazon Reviews, UniScore significantly outperforms Regression and RandomForest (p<0.001) while matching NN (p=0.41) with lower variance (std=0.001 vs 0.013). On SAMSum, UniScore significantly outperforms all baselines (p<0.001). On Depression Tweet, UniScore achieves the highest F1-score (p<0.05). These results demonstrate that UniScore matches or exceeds black-box approaches while providing full interpretability.

Ablation Study on Distance Metrics. To validate our choice of Hellinger distance for measuring discriminative power, we compared seven distance metrics across all datasets (Table 2). Each metric was used to construct the AHP pairwise comparison matrix, with all other components held constant.

Hellinger distance achieves the best performance on Amazon Software Reviews and SAMSum, and competitive results on Depression Tweet, demonstrating consistent effectiveness across tasks. In contrast, KL divergence exhibits unstable behavior,

Table 1: Performance comparison across datasets (mean ± std over 5 seeds). Best results are in **bold**, second-best are underlined.

Dataset	Method	Spearman ρ	Kendall τ	Pearson r	F1	Acc	CV	Skew.
Software	Random Weight	.090 ± .194	.069 ± .150	.048 ± .082	–	–	0.95 ± 1.10	0.37 ± 0.40
	Regression	.376 ± .000	.291 ± .000	.215 ± .000	–	–	1.59 ± 0.00	1.61 ± 0.00
	Random Forest	.376 ± .002	.294 ± .001	.044 ± .005	–	–	5.19 ± 0.41	19.98 ± 1.09
	Neural Network	.428 ± .013	.332 ± .011	**.217 ± .003**	–	–	1.34 ± 0.09	1.61 ± 0.13
	UniScore (Ours)							
	– p=0.1%	**.434 ± .001**	**.336 ± .001**	.201 ± .002	–	–	0.38 ± 0.02	0.44 ± 0.05
	– p=0.3%	.424 ± .005	.328 ± .004	.188 ± .004	–	–	**0.32 ± 0.01**	**0.20 ± 0.06**
	– p=0.5%	.428 ± .003	.331 ± .003	.192 ± .003	–	–	0.35 ± 0.02	0.24 ± 0.05
SAMSum	Random Weight	-.067 ± .159	-.047 ± .111	-.081 ± .172	–	–	-1.02 ± 1.37	**-0.10 ± 0.50**
	Regression	.270 ± .000	.195 ± .000	.293 ± .000	–	–	0.25 ± 0.00	-1.29 ± 0.00
	Random Forest	.178 ± .018	.135 ± .013	.241 ± .011	–	–	0.56 ± 0.01	0.21 ± 0.04
	Neural Network	.112 ± .183	.084 ± .138	.113 ± .179	–	–	0.33 ± 0.09	0.74 ± 0.79
	UniScore (Ours)							
	– p=25%	**.321 ± .002**	**.226 ± .002**	**.306 ± .004**	–	–	**0.14 ± 0.00**	-1.31 ± 0.06
	– p=20%	.279 ± .082	.198 ± .057	.272 ± .099	–	–	0.16 ± 0.02	-0.83 ± 0.36
	– p=15%	.301 ± .020	.210 ± .015	.295 ± .020	–	–	0.39 ± 0.15	-0.73 ± 0.04
Depression	Random Weight	–	–	–	.243 ± .223	.451 ± .111	1.47 ± 2.05	**0.15 ± 0.78**
	Regression	–	–	–	.723 ± .000	.783 ± .000	0.69 ± 0.00	0.36 ± 0.00
	Random Forest	–	–	–	.726 ± .000	.786 ± .000	0.72 ± 0.00	0.39 ± 0.00
	Neural Network	–	–	–	.725 ± .001	**.786 ± .000**	0.71 ± 0.01	0.40 ± 0.01
	UniScore (Ours)							
	– n=100	–	–	–	.726 ± .004	.778 ± .008	0.22 ± 0.02	0.66 ± 0.10
	– n=300	–	–	–	**.729 ± .003**	.780 ± .005	**0.22 ± 0.01**	0.68 ± 0.06
	– n=500	–	–	–	.727 ± .001	.782 ± .001	0.22 ± 0.01	0.69 ± 0.03

Table 2: Ablation study on distance metrics. Values indicate Spearman ρ for Amazon/SAMSum and F1 for Depression Tweet (mean ± std over 5 seeds). Best results are in **bold**, second-best are underlined.

Metric	Software	SAMSum	DepTweet
Hellinger	**.434 ± .001**	**.322 ± .000**	.729 ± .003
Total Var.	.434 ± .002	.307 ± .000	.728 ± .003
JS Dist.	.432 ± .001	.286 ± .000	.725 ± .006
Cohen's d	.433 ± .001	.282 ± .000	.727 ± .002
Wasserstein	.430 ± .003	.304 ± .000	.712 ± .008
KL Div.	.429 ± .003	.179 ± .000	**.730 ± .002**
Mean Diff.	.427 ± .007	.302 ± .000	.712 ± .008

Table 3: Ablation study on AHP mapping functions. Spearman ρ for Amazon/SAMSum, F1 for Depression Tweet (mean ± std over 5 seeds).

Mapping	Amazon	SAMSum	DepTweet
Linear ($1+8\Delta$)	**.434 ± .001**	**.321 ± .002**	.729 ± .001
Sigmoid ($1 + 8\sigma$)	.433 ± .001	.312 ± .001	**.734 ± .000**
Exponential (9^{Δ})	.425 ± .002	.317 ± .002	.733 ± .000
Quadratic ($1+8\Delta^2$)	.425 ± .002	.307 ± .003	.733 ± .000
Cubic ($1+8\Delta^3$)	.414 ± .002	.309 ± .000	.734 ± .000

achieving high performance on one dataset but degrading severely on sparse distributions. Based on these results, we adopt Hellinger distance as our default metric.

From a theoretical perspective, Hellinger distance is well suited to low-dimensional, discrete Likert-score distributions under limited sample sizes. Its symmetric and bounded range $[0, 1]$ yields stable and comparable discriminativeness values, while the square-root transformation mitigates noise in low-probability bins.

5.4.1 Ablation Study on Mapping Functions

A key design choice in UniScore is the mapping function that transforms Hellinger distance differences into AHP pairwise comparison entries. To validate our linear mapping $a_{ij} = 1 + 8\Delta_{ij}$, we compare it against four nonlinear alternatives.

As shown in Table 3, linear mapping achieves the best performance on continuous-signal tasks (Amazon, SAMSum), while nonlinear mappings show marginal gains only on the binary classification task. The quadratic and cubic mappings, which compress small differences, consistently underperform on regression tasks where fine-grained discrimination matters. These results support our design choice: linear mapping preserves proportional relationships between discriminative power differences, yielding more stable and interpretable weight estimation across diverse task types.

5.5 Comparison with Flagship Models

Real-world web datasets such as *Amazon Software* and *Depression tweet* have been relatively underexplored in evaluation research compared to canonical LLM benchmarks, raising questions about robustness on noisy, user-generated content.

To assess performance under realistic web con-

ditions, we compare UniScore against state-of-the-art evaluators, including 2-shot G-Eval with GPT-5.2 (OpenAI, 2025), Gemini-3-flash (Google, 2025), and Claude Sonnet 4.5 (Anthropic, 2025). The comparison focuses on the trade-off between effectiveness and efficiency on real-world, user-generated datasets (Figure 2). All evaluator prompts (Appendix E.4) were designed with assistance from the respective models, following the criteria definitions in Section 4.2.

As shown in Figure 2, UniScore achieves the highest Spearman correlation (0.4303) while running locally on a single RTX 3090 GPU. It evaluates 100 samples in 2.25 minutes, substantially outperforming ensemble-based evaluators in efficiency. Notably, the entire pipeline operates on-premise with a 1.7B-parameter model, requiring no external API calls.

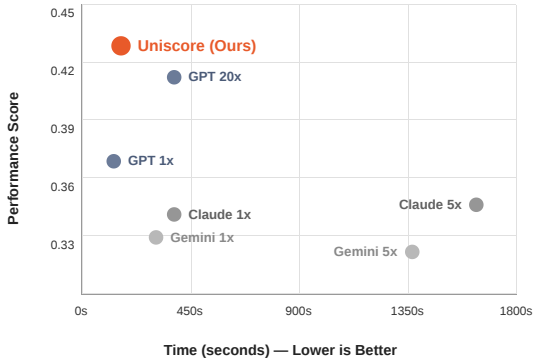

Figure 2: Performance and efficiency comparison of UniScore vs. Flagship model evaluators on *Amazon Reviews (Software)*, showing higher correlation and faster processing across all baselines.

5.6 Interpretability and Weight Analysis

By construction, AHP yields interpretable aggregation weights. To illustrate this advantage, we compare UniScore with a linear regression baseline on the Amazon Reviews (Software) dataset. As shown in Figure 3, the regression baseline assigns counter-intuitive weights, heavily relying on review length while assigning negative importance to expertise and specificity, contradicting established findings in prior work (Danescu-Niculescu-Mizil et al., 2009).

In contrast, UniScore distributes importance more plausibly across criteria, aligning with prior literature and indicating stronger, domain-consistent interpretability. While the weight for *Consistency* was slightly negative, its magnitude was small (7%), suggesting it had minimal influence on the overall prediction.

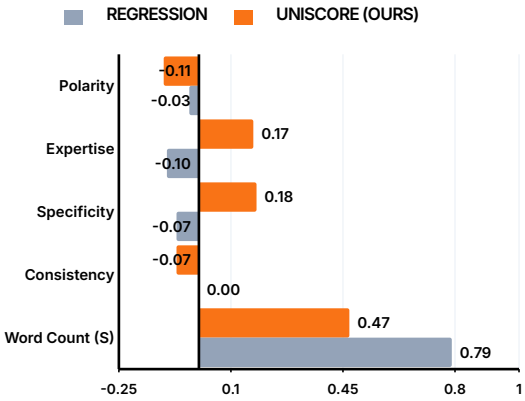

Figure 3: Weight distributions for UniScore and regression on *Amazon Reviews (Software)*.

6 Conclusion

This paper introduced UniScore, a principled framework for generating interpretable, efficient, and high-performance text quality scores. By integrating multi-criteria LLM-based evaluation with Hellinger distance-based discriminative power measurement and AHP-derived weighting, UniScore achieves competitive or superior performance compared to both traditional baselines and black-box neural approaches, while maintaining full interpretability.

UniScore provides a favorable balance of interpretability, efficiency, and predictive performance. AHP-derived weights are numerically explicit and domain-consistent, while the lightweight linear formulation enables near real-time inference. Moreover, relying on distributional differences rather than absolute score magnitudes makes UniScore robust to scale compression and central tendency bias in LLM-generated scores.

The practical implications are substantial for real-time web and industrial applications, including e-commerce review ranking, digital mental health analysis, and automated essay scoring. Because UniScore operates with compact local models (as small as 1.7B parameters), it enables fully on-premise deployment without external API calls. This is critical for privacy-sensitive domains such as psychiatric assessment and education. Beyond text evaluation, the Hellinger–AHP framework offers a generalizable approach to constructing interpretable linear models from arbitrary feature sets.

Limitations

While UniScore demonstrates strong performance across our experimental settings, several limitations warrant discussion. All experiments used Qwen3-1.7B as the sole LLM scorer; performance may vary with different model architectures or sizes, and cross-model validation remains necessary to establish broader generalizability. Additionally, despite providing systematic guidelines, criterion definition requires human judgment, meaning different criteria choices may yield different weight distributions. Automated criterion discovery remains an open research direction. Finally, as our experiments prioritized lightweight models for real-time web deployment, the effectiveness of UniScore when combined with large-scale models remains unexplored.

References

J. Aczél and T.L. Saaty. 1983. Procedures for synthesizing ratio judgements. *Journal of Mathematical Psychology*, 27(1):93–102.

Anthropic. 2025. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5. Accessed: 2025-12-15.

Sebastian Barros. 2025. Solving ai foundational model latency with telco infrastructure. *Preprint*, arXiv:2504.03708.

Jonathan Barzilai and Boaz Golany. 1990. Deriving weights from pairwise comparison matrices: The additive case. *Operations Research Letters*, 9(6):407–410.

Ljubiša Bojić, Olga Zagovora, Asta Zelenkauskaite, Vuk Vuković, Milan Čabarkapa, Selma Veseljević Jerković, and Ana Jovančević. 2025. Comparing large language models and human annotators in latent content analysis of sentiment, political leaning, emotional intensity and sarcasm. *Scientific reports*, 15(1):11477.

Bice Cavallo and Livia D'Apuzzo. 2009. A general unified framework for pairwise comparison matrices in multicriterial methods. *International Journal of Intelligent Systems*, 24(4):377–398.

Sharath Chandra Guntuku, Anneke Buffone, Kokil Jaidka, Johannes C. Eichstaedt, and Lyle H. Ungar. 2019. Understanding and measuring psychological stress using social media. *Proceedings of the International AAAI Conference on Web and Social Media*, 13(01):214–225.

Zaiqian Chen, Daniel Verdi do Amarante, Jenna Donaldson, Yohan Jo, and Joonsuk Park. 2022. Argument mining for review helpfulness prediction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8914–8922, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Cristian Danescu-Niculescu-Mizil, Vlad Danescu-Niculescu-Mizil, and Lillian Lee. 2009. Finding applause and boos in reviews. *arXiv preprint arXiv:0906.3741*.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Jared Fernandez, Clara Na, Vashisth Tiwari, Yonatan Bisk, Sasha Luccioni, and Emma Strubell. 2025. Energy considerations of large language model inference and efficiency optimizations. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32556–32569, Vienna, Austria. Association for Computational Linguistics.

Google. 2025. Gemini 3 flash: frontier intelligence built for speed. https://blog.google/products/gemini/gemini-3-flash/. Accessed: 2025-12-17.

Ernst Hellinger. 1909. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271.

Tao Huang. 2025. Content moderation by llm: From accuracy to legitimacy. *Artificial Intelligence Review*, 58(320). Published: July 19 2025.

J. Peter Kincaid, Jr. Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel. Technical Report TAEG-TR-75-4, Chief of Naval Technical Training, Naval Air Station Memphis, Millington, TN.

Gang Kou, Daji Ergu, Changsheng Lin, and Yi Chen. 2016. Pairwise comparison matrix in multiple criteria decision making. *Technological and Economic Development of Economy*, 22(5):738–765.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Hwiyoung Lee and Shuo Chen. 2025. Systematic bias of machine learning regression models and correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):4974–4983.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *Preprint*, arXiv:2412.05579.

Terrence M. Liddell and John K. Kruschke. 2018. Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology*, 79:328–348.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, EMNLP '23, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yixin Liu, Alexander Fabbri, Yilun Zhao, Pengfei Liu, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Towards interpretable and efficient automatic reference-based summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16360–16368, Singapore. Association for Computational Linguistics.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024. HD-Eval: Aligning large language model evaluators through hierarchical criteria decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '24, pages 7641–7660, Bangkok, Thailand. Association for Computational Linguistics.

Freerk A Lootsma. 1999. *Multi-Criteria Decision Analysis via Ratio and Difference Judgement*, volume 29 of *Applied Optimization*. Springer.

Annie Louis and Ani Nenkova. 2013. Automatically assessing review helpfulness. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 30–40, Seattle, Washington, USA. Association for Computational Linguistics.

Xiaotian Lu, Jiyi Li, Koh Takeuchi, and Hisashi Kashima. 2024. AHP-Powered LLM reasoning for multi-criteria evaluation of open-ended responses. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, EMNLP '24, pages 1847–1856, Miami, Florida. Association for Computational Linguistics.

Richard Diehl Martinez, Pietro Lesci, and Paula Buttery. 2024. Tending towards stability: Convergence challenges in small language models. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3275–3286, Miami, Florida, USA. Association for Computing Machinery.

Barbara McGillivray, Gard Jenset, and Dominik Heil. 2020. Extracting keywords from open-ended business survey questions. *Journal of Data Mining & Digital Humanities*, 2020(Project).

Jonathan Mellon, Jack Bailey, Ralph Scott, James Breckwoldt, Marta Miori, and Phillip Schmedeman. 2024. Do AIs know what the most important issue is? using language models to code open-text social survey responses at scale. *Research & Politics*, 11(1):20531680241231468.

Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. 2011. Review recommendation: Personalized prediction of the quality of online reviews. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM '11)*, pages 2249–2252, New York, NY, USA. Association for Computing Machinery.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

OpenAI. 2025. Introducing GPT-5.2. `https://openai.com/ko-KR/index/introducing-gpt-5-2/`. Accessed: 2025-12-11.

Wei Qiao, Tushar Dogra, Otilia Stretcu, Yu-Han Lyu, Tiantian Fang, Dongjin Kwon, Chun-Ta Lu, Enming Luo, Yuan Wang, Chih-Chun Chia, Ariel Fuxman, Fangzhou Wang, Ranjay Krishna, and Mehmet Tek. 2024. Scaling up LLM reviews for Google ads content moderation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, pages 1174–1175, Merida, Mexico. Association for Computing Machinery.

Xianshan Qu, Xiaopeng Li, Csilla Farkas, and John Rose. 2021. Review helpfulness evaluation and recommendation based on an attention model of customer expectation. *Information Retrieval Journal*, 24(1):55–83.

Nishat Raihan, Sadiya Sayara Chowdhury Puspo, Shafkat Farabi, Ana-Maria Bucur, Tharindu Ranasinghe, and Marcos Zampieri. 2024. Mentalhelp: A multi-task dataset for mental health in social media. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11196–11203.

Philip Resnik, William Armstrong, Leonardo Claudino, and Tri Nguyen. 2020. Discovering the experience of depression and anxiety on social media. *Proceedings of the International AAAI Conference on Web and Social Media*, 14:578–589.

Jens Rupprecht, Georg Ahnert, and Markus Strohmaier. 2025. Prompt perturbations reveal human-like biases in llm survey responses. *Preprint*, arXiv:2507.07188.

Thomas L Saaty. 1977. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3):234–281.

Thomas L. Saaty. 1980. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill, New York, NY.

Thomas L. Saaty. 1990. How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1):9–26. Decision making by the analytic hierarchy process: Theory and applications.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *Preprint*, arXiv:2310.11324.

Jyoti Prakash Singh, Seda Irani, Nripendra P. Rana, Yogesh K. Dwivedi, Sunil Saumya, and Pradeep Kumar Roy. 2017. Predicting the "helpfulness" of online consumer reviews. *Journal of Business Research*, 70:346–355.

Jovan Stojkovic, Esha Choukse, Chaojie Zhang, Inigo Goiri, and Josep Torrellas. 2024. Towards greener llms: Bringing energy-efficiency to the forefront of llm inference. *Preprint*, arXiv:2403.20306.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *Preprint*, arXiv:2405.01724.

Fali Wang, Minhua Lin, Yao Ma, Hui Liu, Qi He, Xianfeng Tang, Jiliang Tang, Jian Pei, and Suhang Wang. 2025. A survey on small language models in the era of large language models: Architecture, capabilities, and trustworthiness. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Volume 2*, KDD '25, pages 6173–6183, Toronto, ON, Canada. Association for Computing Machinery.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '23, pages 7397–7408, Toronto, Canada. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2023. Flask: Fine-grained language model evaluation based on alignment skill sets. *Preprint*, arXiv:2307.10928.

Yadong Zhang and Du Zhang. 2014. Automatically predicting the helpfulness of online reviews. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IRI)*, pages 662–668. IEEE.

Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*, pages 425–434, New York, NY, USA. Association for Computing Machinery.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhan Dong, and Yu Wang. 2024. A survey on efficient inference for large language models. *Preprint*, arXiv:2404.14294.

A Word Count Scaling

$$z_i = \frac{r_i - \mu}{\sigma},$$
$$\tilde{z}_i = \max\{1, \min\{5, z_i \cdot \sigma_{\text{scale}} + 3\}\}, \quad (19)$$
$$s_{ik} = \tilde{z}_i$$

where r_i is the raw measurement; μ and σ are the mean and standard deviation computed on the scored subset; σ_{scale} is a parameter to map the standardized scores to the 1–5 range. This approach handles outliers by clipping extreme values while preserving the common 1–5 scale.

B Dataset Statistics

Table 4: Dataset statistics. Software and RoSE SAMSum are regression tasks, while Depression Tweet is a binary classification task.

Dataset	Split	Count	Labels
Software	Train	10,224	-
	Test	2,561	-
RoSE SAMSum	Train	3,200	-
	Test	800	-
Depression Tweet	Train	22,090	0: 12,523 / 1: 9,567
	Test	5,523	0: 3,132 / 1: 2,391

C Baseline Implementations

We provide implementation details for all baseline aggregation methods used in our experiments.

Random Weights. Criterion weights are independently sampled from a uniform distribution $U[-1, 1]$ and normalized to sum to one in magnitude.

Linear Regression. A linear regression model is trained to predict the ground-truth signal from criterion scores using ordinary least squares. No regularization is applied to avoid biasing coefficients.

Random Forest. We use a random forest regressor with 200 trees and default hyperparameters, following standard practice for tabular feature aggregation.

Neural Network. A two-layer MLP with hidden sizes (64, 32), ReLU activation, and dropout rate 0.1 is trained using AdamW with early stopping based on validation loss.

D Evaluation Metrics

We briefly summarize the evaluation metrics used in the main experiments.

Correlation Metrics. Predictive alignment with continuous ground-truth signals is measured using Pearson r, Spearman ρ, and Kendall τ.

Classification Metrics. For binary signal datasets, discriminative performance is measured using accuracy and F1-score.

Stability Metrics. Score stability is evaluated using the coefficient of variation (CV), defined as the ratio of standard deviation to mean, and skewness to assess distributional asymmetry.

D.1 Consistency Ratio Check

To verify the consistency of the method discussed in Section 4.4, we performed a Consistency Ratio (CR) check by conducting 100 random samplings for each p (0.1%, 0.3%, and 0.5%). As shown in Table 5, not a single sampled CR value exceeded the conventional consistency error threshold of 0.1(Saaty, 1980). This result experimentally suggests that our difference-based approach does not have significant consistency issues.

Table 5: Consistency Ratio (CR) statistics across different p%. All CR values remain well below the 0.1 threshold, indicating reliable pairwise comparisons.

p-value	Mean	Variance	Max
0.5%	0.0123	7.88e-05	0.0277
0.3%	0.0169	7.83e-05	0.0389
0.1%	0.0123	3.34e-05	0.0378

E UniScore Prompt Templates

This appendix lists the exact prompts used for LLM-based evaluation in UniScore. All prompts require **JSON-only** output in the form `{"score": N}`.

E.1 Amazon Software Reviews (Multi-Dimensional)

Polarity

```
Rate the sentiment polarity of the review on a 1–5 Likert scale.
1=very negative, 2=negative, 3=neutral/mixed, 4=positive, 5=very positive.
Return JSON only.
Review: txt
"score": N
```

Expertise

```
Rate the reviewer expertise shown in the text (domain terms, procedures, precise specs,
comparisons).
1=none, 2=low, 3=moderate, 4=high, 5=very high expertise.
Return JSON only.
Review: txt
"score": N
```

Specificity

```
Rate how specific and concrete the review is (numbers, model names, scenarios, measurable
details).
1=very vague, 2=vague, 3=some specifics, 4=specific, 5=highly specific with concrete details.
Return JSON only.
Review: txt
"score": N
```

Consistency

```
Given the star rating and the review text, rate rating–content consistency (1–5).
1=strongly inconsistent, 3=partly consistent/mixed, 5=fully consistent in tone and claims.
Star rating (0–5): star
Review: txt
Return JSON only.
"score": N
```

E.2 Depression Tweet Analysis (DepTweet)

Negative Affect

```
Rate the negative affect in the tweet (e.g., sadness, hopelessness expressions, as identified in
depression-related social media analysis).
1=very positive, 2=positive, 3=neutral, 4=negative, 5=highly negative.
Return JSON only.
Tweet: txt
"score": N
```

Self-Focus

```
Rate the self-focus in the tweet (e.g., use of first-person pronouns like I, me, my, indicating
self-centered language in depressed texts).
1=none, 2=low, 3=moderate, 4=high, 5=very high.
Return JSON only.
Tweet: txt
"score": N
```

Absolutist Language

```
Rate the use of absolutist language in the tweet (e.g., words like always, never, everything,
which are markers of black-and-white thinking in depression).
1=none, 2=low, 3=moderate, 4=high, 5=very high.
```

```
Return JSON only.
Tweet: txt
"score": N
```

Social Isolation

```
Rate the level of social isolation in the tweet (e.g., lack of social words like friend, we, talk,
indicating withdrawal in depressed individuals).
1=highly social, 2=social, 3=neutral, 4=isolated, 5=highly isolated.
Return JSON only.
Tweet: txt
"score": N
```

E.3 SAMSum Summaries

We use the same prompt structure as Amazon, with criteria: `coherence`, `relevance`, `fluency`, and `word_count_scaled`. For brevity, we omit the duplicated templates.

E.4 G-Eval Full Prompts (Flagship Models)

System Prompt

```
You are an evaluator for product reviews. Judge whether a review is a good,
helpful review overall for making purchase or usage decisions.

You must respond with a single valid JSON object only. No prose, no extra keys.

Objective:
- Produce a single integer score in [1,5] that reflects overall
  helpfulness/quality of the review text.

Output format:
- Return only a single JSON object: "score": N
- N must be an integer in [1,5]. No explanations, no extra keys, no trailing text.

Decision checklist (evaluate internally; do not reveal reasoning):
1) Specificity: concrete details (numbers, model names, scenarios, steps,
   measurements, comparisons).
2) Expertise: correct domain terms, procedures, constraints, or trade-offs
   indicating real familiarity.
3) Evidence/Verification: measurable facts, side-by-side comparisons,
   reproducible steps, or cited conditions.
4) Coverage/Balance: addresses key pros/cons or "works if/doesn't if"
   conditions relevant to typical users.
5) Coherence/Clarity: clear, non-redundant, logically consistent; avoids
   contradictions.
6) Consistency with star (if provided): tone/claims align with the star;
   do not infer a star when missing.
7) Actionability: contains guidance that changes a user's decision
   (who should/shouldn't buy, settings, fixes).

Scoring rule (apply privately, then map to N):
- For each checklist item:
  present = +1, partial = +0.5, absent = +0.
  Let S be the total score.
- Map S to N before penalties:
  * S < 1.0     -> 1
  * 1.0--2.0    -> 2
  * 2.5--3.5    -> 3
  * 4.0--5.0    -> 4
  * S >= 5.5    -> 5
- Then apply penalties (subtract 1 per item, floor at 1):
  (a) Off-topic or product-irrelevant content.
  (b) Pure cheerleading/complaint with no specifics
      (e.g., <12 tokens and no concrete fact).
  (c) Strong inconsistency (e.g., 5* with negative content,
```

```
      or 1* with very positive content).
  (d) Advertising/spam/referral codes; copy-pasted spec sheet
      without user insight.
  (e) Safety-critical or potentially misleading claims stated
      without any supporting detail.
  (f) Toxic/abusive content unrelated to product performance
      or user safety.

Important constraints:
- Length alone must not inflate the score.
  Long but generic => low; short but highly specific can be high.
- Do not use outside knowledge or web search; judge strictly by
  provided text and star.
- Emojis or styling do not affect the score unless they add factual content.
- If star is unknown, ignore the consistency item (6).
- If the text is empty or meaningless, return "score": 1.
- Clamp the final N to [1,5] and output JSON only.
```

User Prompt Template

```
Examples for calibration:

[Very helpful]
Star: ex1_star
Review: ex1_text
Expected: "score": 5

[Unhelpful]
Star: ex2_star
Review: ex2_text
Expected: "score": 1

Now evaluate the user review below. Think privately and return only the final JSON.

Star: star
Review: text
```