Chatbot Arena and P2L: Ranking LLMs with Human Preferences

Overview

- **Chatbot Arena:** An open platform for evaluating LLMs using crowdsourced human preference data.
- **Prompt-to-Leaderboard (P2L):** A method for generating prompt-specific model rankings.
- We summarize Arena's ranking methodology and P2L's training and deployment framework.

Chatbot Arena Platform

- Users compare anonymized model responses and vote.
- Voting options: Model A, Model B, or Tie.
- **2.8M+ votes**, **219+ models** (as of March 24, 2025).
- Supports multilingual prompts and diverse domains.

Top Models on Arena (as of March 2025)

Rank	Model	Score	95% CI	Votes	
1	Gemini-2.5-Pro-Exp-03-25	1443	+8/-13	2540	
2	Grok-3-Preview-02-24	1404	+5/-6	10398	
2	GPT-4.5-Preview	1398	+7/-6	10615	
4	Gemini-2.0-Flash-Thinking	1381	+4/-3	22659	Latest
4	ChatGPT-4o (2025-01-29)	1374	+5/-4	22517	
7	DeepSeek-R1	1360	+5/-5	12772	
10	Qwen2.5-Max	1340	+5/-3	17124	
13	DeepSeek-V3	1318	+5/-4	22845	
15	Claude 3.7 Sonnet	1304	+8/-7	4917	

leaderboard from `lmarena.ai`, based on Bradley–Terry model and bootstrapped CIs.

Bradley–Terry (BT) Ranking Model (New Algorithm)

Arena collects pairwise comparisons (Model A vs. Model B). BT is designed for this.

Goal: Derive a global ranking of models from pairwise human preferences.

- Assumes each model i has a latent score $\xi_i \in \mathbb{R}$.
- Predicts preference probability as:

$$P(i \succ j) = \frac{1}{1 + e^{\xi_j - \xi_i}} = \sigma(\xi_i - \xi_j)$$

- Scores $\boldsymbol{\xi}$ are fitted via **batch maximum likelihood estimation (MLE)**.
- **Statistical advantages:**
  - Enables confidence intervals and regularization
  - Compatible with convex optimization and bootstrapping

Elo Rating System (Original Online Algorithm)

Goal: Dynamically update model ratings based on sequential user comparisons.

- Each model i has a score $\xi_i \in \mathbb{R}$.
- For each new comparison between model i and model j:
  - Predict win probability:

$$\hat{y} = \sigma(\xi_i - \xi_j) = \frac{1}{1 + e^{-(\xi_i - \xi_j)}}$$

  - Observe outcome $Y \in \{0, 1\}$:
    $Y = 1$ if model i wins; $Y = 0$ if model j wins.
  - **Online update**:

$$\xi_i \leftarrow \xi_i + \eta(Y - \hat{y}), \quad \xi_j \leftarrow \xi_j - \eta(Y - \hat{y})$$

- No need to store history — each update only uses the current match.
- **Learning rate η:** Controls adaptation speed (e.g., 0.01).

Elo Rating System: Properties and Comparison with BT

Key Properties

- **Online:** Updates occur after each new match.
- **Lightweight:** No need to store full comparison history.
- **Limitations:** No uncertainty quantification; score may drift over time.

Relation to BT Model

- Elo can be seen as a online approximation to batch MLE in the Bradley–Terry model.

MLE for Bradley–Terry Scores

Goal: Estimate scores $\boldsymbol{\xi} \in \mathbb{R}^M$ that best explain pairwise preferences.

- Each model i is assigned a latent score ξ_i.
- For each pair (i, j), let n_{ij} be the number of times i is preferred over j.
- BT model:

$$P(i \succ j) = \sigma(\xi_i - \xi_j) = \frac{1}{1 + e^{\xi_j - \xi_i}}$$

- Maximize the regularized log-likelihood:

$$\mathcal{L}(\xi) = \sum_{i \neq j} n_{ij} \log \sigma(\xi_i - \xi_j) - \lambda \|\xi\|^2$$

MLE for Bradley–Terry Scores

Optimization Procedure (Convex):

- Initialize all $\xi_i = 0$
- Compute gradient for each coordinate:

$$\frac{\partial \mathcal{L}}{\partial \xi_k} = \sum_{j \neq k} [n_{kj}(1 - \sigma(\xi_k - \xi_j)) - n_{jk}\sigma(\xi_j - \xi_k)] - 2\lambda\xi_k$$

- Use convex optimizers such as gradient descent or L-BFGS.
- Enforce constraint $\sum_i \xi_i = 0$ to resolve identifiability.
- Sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$

Extended vs Standard Bradley–Terry Models

Extended Bradley–Terry (BT) generalizes the standard BT model by modeling each player as a collection of subsystems.

Feature	**Standard BT**	**Extended BT**
Player Representation	Scalar score ξ_i per player	Multiple subsystems per player
Input Format	Pair (i, j)	Feature vector $x \in \mathbb{R}^d$ (encodes active subsystems)
Prediction Model	$\sigma(\xi_i - \xi_j)$	$\sigma(\theta^\top x)$
Expressiveness	Simple, interpretable	Captures richer structure and interactions
Optimization	Logistic regression over scalar scores	Logistic regression over structured features
Use Cases	LLM ranking in Arena	Evaluation of agents, tools, and frameworks (e.g. RedTeam Arena)

P2L: Prompt-to-Leaderboard

- Learns a function $\hat{\boldsymbol{\theta}} : z \mapsto \mathbb{R}^M$ mapping prompts z to BT scores.
- Trained on Arena's pairwise human preference data.
- **Input:** Natural language prompt z
- **Output:** $\hat{\theta}(z)$ — predicted BT scores for each of the M models
- **Applications:**
  - *Prompt-aware evaluation:* Generate leaderboards tailored to specific prompts or topics.
  - *Model capability profiling:* Identify strengths and weaknesses across prompt clusters.
  - *Intelligent routing:* Select models dynamically based on cost, quality, and prompt characteristics.

Training the P2L Model

- **Training triplets:** (X, Y, Z)
  - Z: Natural language prompt
  - $X \in \mathbb{R}^M$: Two-hot encoding of a model pair: $X_i = -1$, $X_j = +1$
  - $Y \in \{0, 1\}$: 1 if j wins, 0 if i wins
- **Model:** Pretrained LLM with BT head outputting $\hat{\theta}(Z) \in \mathbb{R}^M$
- **Loss:**

$$\ell(\hat{y}, Y) = -Y \log \hat{y} - (1 - Y) \log(1 - \hat{y}), \quad \hat{y} = \sigma(X^\top \hat{\theta}(Z))$$

- Trained on **2.8M comparisons** over **219+ models**.

P2L Evaluation: Simple & Category Metrics

1. **Simple Metrics**
   - **Goal:** Evaluate local prediction quality on individual comparisons.
   - **Measures:** Accuracy, loss, and output stability.
   - **Examples:** `Loss`, `Accuracy`, `BCELoss`, `Tie_Accuracy`, `Spread-BT`
2. **Category Metrics**
   - **Goal:** Evaluate ranking consistency within task types (e.g., math, code).
   - **Measures:** Agreement with category-specific ground truth.
   - **Examples:** `Kendall-1bs`, `Spearman-1bs`, `Leaderboard`, `Top-k-Fraction`, `Aggr_Loss`

P2L Evaluation: Subset & Scale Metrics

3. Random Subset Metrics

- **Goal:** Test ranking robustness across sampled prompt subsets.
- **Use case:** Sample N prompts (e.g., 250, 500, 1000) repeatedly.
- **Measures:** Ranking stability under prompt variation.
- **Metrics:** `Kendall-1bs`, `Spearman-1bs`, `L1-Dist-Prob`

4. Aggregation Scale Metrics

- **Goal:** Evaluate scalability with leaderboard size.
- **Use case:** Rank models over simulated leaderboards (e.g., 10 to 1000).
- **Measures:** Consistency as model pool grows.
- **Metrics:** `Kendall-1bs`, `Spearman-1bs`, `L1-Dist-Prob`

Loss-Based Metrics in P2L Evaluation

Purpose: Quantify how well the model fits human preference data at different evaluation levels.

Types of Loss Metrics:

- `Loss`, `BCELoss`, `MSELoss` (Simple metrics)
  - **Purpose:** Evaluate local pairwise prediction accuracy.
  - **Used in:** `simple_metrics`
- `Aggr_Loss`, `Aggr_BCELoss` (Aggregated metrics)
  - **Purpose:** Measure global consistency with aggregated rankings.
  - **Used in:** `category_metrics`
- `Aggr_Tie_Loss` (Tie-aware models)
  - **Purpose:** Support multi-label preference with tie predictions.
  - **Used in:** `category_metrics` (for tie models)

Interpretation:

- Lower loss = better alignment with human preferences.
- Aggregated loss = better reflection of leaderboard-level fit.

Kendall's τ: Rank Correlation

Purpose: Evaluate how well the predicted model rankings match the ground-truth BT rankings.

Setup: Let $\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_M)$ be the predicted scores from P2L, and $\theta = (\theta_1, \ldots, \theta_M)$ be the ground-truth BT scores.

For each pair (i, j):

- **Concordant:** $\hat{\theta}_i > \hat{\theta}_j$ and $\theta_i > \theta_j$, or vice versa.
- **Discordant:** Rankings are reversed.

Definition:

$$\tau = \frac{(\#\text{ concordant pairs}) - (\#\text{ discordant pairs})}{\binom{M}{2}}$$

Range: $-1 \leq \tau \leq 1$

- $\tau = 1$: Perfect rank agreement
- $\tau = 0$: No correlation
- $\tau = -1$: Perfect inverse rank

Spearman's Rank Correlation

Spearman's ρ measures the similarity between two rankings by computing the correlation between their ranks.

Mathematical Definition:
Let x_i and y_i be the ranks of the i-th item in two ranked lists of size n:

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} (x_i - y_i)^2}{n(n^2 - 1)}$$

Interpretation:

- $\rho = +1$: perfect agreement in ranks
- $\rho = 0$: no correlation
- $\rho = -1$: perfect inverse ranking

Notes:

- Equivalent to Pearson correlation on ranks.
- More tolerant to small misorderings than Kendall's τ.

L1 Distance Between Predicted and Ground Truth Probabilities

What is L1-Dist-Prob?

- Measures the L1 (Manhattan) distance between predicted preference probabilities and ground truth probabilities.
- Focuses on full distribution similarity, not just rank order.

Definition: Let **p** be predicted preference probabilities, and **q** the ground truth:

$$\text{L1-Dist-Prob} = \sum_i |p_i - q_i|$$

Interpretation:

- 0 indicates perfect match
- Higher values indicate larger divergence

Used in: `random_subset_metrics`, `aggr_scale_metrics`

Top-k-Fraction & Top-k-Displace Metrics

Top-k-Fraction:

- Measures the fraction of predicted top-k models that also appear in the ground truth top-k.
- Value range: $[0, 1]$

Top-k-Displace:

- Computes the average absolute position difference for predicted top-k models vs ground truth.
- Penalizes rank displacement within top performers.

Used in: `category_metrics`
Goal: Focus on head-of-leaderboard accuracy

Spread-BT: Score Separation in P2L Predictions

What is Spread-BT?

- Measures how far apart the model's output scores are across models.
- High spread $\rightarrow$ confident differentiation; low spread $\rightarrow$ uncertain ranking.

Definition: Often computed as:

$$\text{Spread-BT} = \max_i \beta_i - \min_i \beta_i$$

where β_i are model scores from the P2L head.

Used in: `simple_metrics`
Goal: Evaluate confidence / sharpness of predicted rankings.

Optimal Routing with P2L

Goal: Given a prompt z, select a distribution $\pi(z)$ over models to maximize performance within cost.

- $\hat{\theta}(z) \in \mathbb{R}^M$: P2L model output — predicted BT scores for M models.
- $\pi(z) \in \Delta^M$: Routing distribution — how likely to query each model.
- $c \in \mathbb{R}^M$: Inference costs of models (e.g., latency, price per token).
- C: Total cost budget for a single query.
- $W \in \mathbb{R}^{M \times M}$: Win-rate matrix, defined by:

$$W_{ij} = \sigma(\hat{\theta}_i(z) - \hat{\theta}_j(z))$$

- $q \in \Delta^M$: Baseline model distribution, e.g., uniform or recent Arena queries.

Routing objective:

$$\max_{\pi \in \Delta^M,\ \pi^\top c \leq C} \pi^\top W q$$

Optimal Routing with P2L

Interpretation:

- $\pi^\top Wq$: Expected win rate if a model is selected according to π, compared against models drawn from q.
- $\pi^\top c \leq C$: Cost constraint — limit average inference cost.

Solution strategy:

- Convex optimization over the simplex Δ^M with linear constraints.
- Use standard solvers (e.g., CVXPY, projected gradient descent).
- In practice: $\pi(z)$ is sparse — typically routes to top 1–2 models.

Deployment:

- Used to deploy P2L-based router in Chatbot Arena.
- Achieved top ranking in January 2025 leaderboard.
- Enables efficient, prompt-specific inference with controlled budget.

Applications and Impact

- Fine-grained evaluation of LLMs at the prompt level
- Personalized model selection and adaptive inference
- Efficient deployment under cost/latency constraints
- Public dataset and open-source code available

Conclusion

- Chatbot Arena provides scalable, human-aligned evaluation for LLMs.
- P2L introduces prompt-aware modeling and deployment.
- Together, they enable data-driven benchmarking and optimization for the next generation of AI systems.