Empirical Evaluation of LLM Pipeline Architectures and Output Formats for Automated Candidate Screening

Nathan El Khoury
School of Computer Science
Carleton University
Ottawa, ON, Canada

Abstract

Large language models (LLMs) are increasingly adopted for automated candidate screening, yet practitioners lack empirical guidance on how pipeline architecture affects accuracy, cost, and consistency. We present a controlled comparison of four LLM pipeline architectures—single-pass, multistage keyword-semantic fusion, multi-agent debate, and hybrid retrieval—applied to ranking 100 production resumes for a Marketing Specialist position using GPT-4.1. Each condition is evaluated over three independent runs using top-20% precision, recall, and F1, Spearman and Kendall rank correlations, and quintile agreement against percentile-based ground truth derived from an institutional screening workflow. The multistage pipeline achieves the highest accuracy (F1 = 0.800, Spearman $\rho = 0.896$), significantly outperforming single-pass (0.700), multi-agent debate (0.600), and hybrid retrieval (0.696) as confirmed by a Kruskal–Wallis test ($H = 16.31$, $p = 0.001$). Contrary to expectations, the multi-agent advocate–critic–judge debate architecture underperforms even the single-pass baseline, producing compressed score distributions with reduced discriminative power. We additionally compare structured JSON output against Token-Oriented Object Notation (TOON) on the winning pipeline, finding that TOON reduces token usage by 73% while maintaining 100% parse reliability at a modest F1 cost of 0.067. These results provide actionable guidance for deploying LLM-based screening systems: multistage fusion maximizes accuracy, while TOON format offers substantial cost savings for high-volume applications.

Keywords: Large Language Models, Candidate Screening, Pipeline Architecture, Structured Output, TOON, Retrieval-Augmented Generation

1 Introduction

The adoption of large language models (LLMs) for automated candidate screening has accelerated rapidly, with organizations deploying these systems to evaluate hundreds or thousands of resumes per open position. The economic stakes are significant: a single screening pipeline may process tens of thousands of candidates annually, where even modest improvements in ranking accuracy can surface better hires while reducing recruiter workload. However, practitioners face a fundamental architectural decision: how should the LLM pipeline be structured? Options range from a single prompt that ingests the full job description and resume, to multi-stage decompositions that separate keyword extraction from semantic evaluation, to multi-agent systems where distinct LLM instances debate a candidate's merits, to retrieval-augmented pipelines that use embedding-based pre-filtering to reduce the number of full LLM evaluations. Each architecture implies different tradeoffs in accuracy, cost, latency, and consistency—yet no controlled empirical comparison exists to guide this choice.

A parallel design decision concerns the output format imposed on the LLM. Structured formats like JSON enable reliable downstream parsing but consume substantial token budgets. Recent work by Tam et al. (2024) demonstrates that format restrictions can significantly affect LLM task performance, suggesting that more compact output notations might reduce cost without proportionally degrading quality. Token-Oriented Object Notation (TOON) is one such compact alternative (Schopplich, 2025), but its impact on evaluation accuracy has not been measured empirically.

This paper addresses two research questions:

RQ1. How do four LLM pipeline architectures compare in candidate ranking accuracy, cost, and consistency?

RQ2. Does output format (structured JSON vs. compact TOON) affect evaluation quality or token efficiency?

Our contributions are:

1. An empirical comparison of four pipeline architectures on 100 production resumes evaluated with GPT-4.1 across three independent runs per condition.
2. The first empirical evaluation of TOON format for structured LLM output in a screening task, demonstrating 73% token savings with modest accuracy loss.
3. Evidence that multi-agent debate architectures underperform simpler single-pass approaches for candidate evaluation, contrary to their reported benefits in factuality tasks.
4. A percentile-based evaluation methodology combining top-20% classification metrics with rank correlation and quintile agreement measures.

Our key findings are as follows. The multistage pipeline achieves $F1 = 0.800$ with Spearman $\rho = 0.896$, outperforming single-pass ($F1 = 0.700$), A2A debate ($F1 = 0.600$), and hybrid retrieval ($F1 = 0.696$). TOON format saves 73% of tokens while reducing F1 by only 0.067 relative to JSON, with both formats achieving 100% parse success. Section 2 reviews related techniques, Section 3 describes our experimental setup and presents results, and Section 4 offers practical recommendations and directions for future work.

2 Techniques to Tackle the Problem

2.1 LLM-based Candidate Screening

The use of LLMs for resume evaluation represents a growing area of applied AI, yet empirical work comparing pipeline designs remains scarce. Lo et al. (2025) propose a context-aware multi-agent framework for resume screening that uses specialized agents for different aspects of candidate assessment, demonstrating that structured multi-agent collaboration can improve explainability. However, their evaluation focuses on qualitative explainability rather than ranking accuracy against ground truth, leaving open the question of whether multi-agent architectures actually improve screening decisions or merely produce more interpretable—but not more accurate—outputs. This gap motivates our controlled comparison across four distinct pipeline architectures.

2.2 Pipeline Architectures

The simplest LLM pipeline is a *single-pass* approach: the model receives the complete job description and resume in a single prompt and produces an evaluation. This baseline is fast and inexpensive but forces the model to simultaneously extract requirements, match qualifications, and assign scores in one inference step.

Multi-stage pipelines decompose the evaluation into sequential steps. In our design, a first stage extracts keywords from the job description (amortized across candidates), a second stage performs semantic evaluation of each candidate, and a third stage matches extracted keywords against the resume. A weighted fusion then combines these signals. This decomposition is motivated by the observation that lexical matching (specific tools, certifications) and semantic assessment (overall fit) capture complementary information.

Multi-agent debate systems assign different perspectives to distinct LLM instances. Du et al. (2024) show that multi-agent debate improves factuality and reasoning on mathematical and strategic tasks by having models critique and refine each other's outputs. Their results are compelling for tasks with verifiable answers, but candidate screening is inherently subjective—there is no objectively correct score, only a ranking that correlates with human judgment. We adapt their approach using an advocate–critic–judge architecture to test whether the benefits of debate transfer to this more ambiguous evaluation setting.

Retrieval-augmented pipelines use embedding-based pre-filtering to reduce computational cost. Following the retrieval-augmented generation (RAG) paradigm introduced by Lewis et al. (2020), our hybrid retrieval approach embeds all resumes and the job description, ranks candidates by cosine similarity, and applies full LLM evaluation only to the top-K candidates. While RAG has proven effective for knowledge-intensive QA tasks where queries have clear semantic matches in a corpus, job–candidate matching requires keyword-level precision (specific tools, certifications, years of experience) that general-purpose embeddings may not capture. This trades recall for efficiency—the question is how much accuracy is lost.

2.3 Structured Output Formats

The format in which an LLM produces structured output affects both token consumption and task performance. Tam et al. (2024) find that imposing format constraints such as JSON, XML, or YAML can degrade LLM reasoning, with the magnitude depending on the task and the restrictiveness of the format. Complementing this, Geng et al. (2025) introduce JSONSchemaBench, a benchmark for evaluating structured JSON generation, revealing that schema complexity significantly impacts generation quality. Together, these findings suggest that the choice of output format is not merely an engineering convenience but a design variable that can meaningfully affect evaluation accuracy.

Token-Oriented Object Notation (TOON) is a compact serialization format designed specifically for LLM output (Schopplich, 2025). It uses key:value pairs with inline arrays and indentation-based nesting, eliminating the syntactic overhead of braces, quotes, and commas present in JSON. Given the token-cost sensitivity of high-volume screening (where thousands of evaluations multiply any per-candidate overhead), TOON's compactness is particularly relevant. We evaluate whether this token reduction translates to cost savings without disproportionate accuracy loss.

2.4 Evaluation Methodology

Evaluating ranking quality requires metrics beyond simple classification accuracy. The Spearman rank correlation coefficient (Spearman, 1904) measures the monotonic relationship between predicted and ground-truth rankings, while Kendall's τ captures pairwise concordance. These rank-based metrics are essential because in screening, the relative ordering of candidates matters as much as the binary advance/reject decision. For LLM evaluation more broadly, Zheng et al. (2023) introduce the MT-Bench framework and the LLM-as-a-judge paradigm, demonstrating that strong LLMs can produce evaluations that correlate well with human preferences when properly calibrated. Our work differs in that we use LLMs as the *evaluated system* rather than the evaluator, measuring their outputs against human-derived ground truth. We combine top-k% classification metrics with rank correlation and quintile agreement to capture both decision accuracy and full-ranking quality.

3 Empirical Evaluation

3.1 Experimental Setup

Dataset. We use 100 anonymized production resumes (PDF and DOCX format) submitted for a Marketing Specialist position, along with the corresponding job description specifying requirements, responsibilities, and preferred qualifications. Ground truth is derived from human labeling by hiring managers, who classified candidates into broad tiers rather than providing a precise 1–100 ranking. We convert these tier-level labels into a percentile-based ranking: the top 20% (ranks 1–20) are labeled "advance" and the remainder "reject." This percentile-based threshold avoids arbitrary absolute score cutoffs, though within-tier ordering carries inherent uncertainty.

Pipeline architectures. We evaluate four architectures, all sharing an identical 3-component scoring rubric (requirements fit, responsibilities alignment, preferred qualifications match) on a 0–10 scale:

- **Single-pass** (baseline): Full job description and resume in one prompt. 1 LLM call per candidate.
- **Multistage**: (1) Keyword extraction from the job description (amortized), (2) semantic evaluation per candidate, (3) keyword matching per candidate, (4) weighted score fusion ($w_{\text{sem}} = 0.6$, $w_{\text{kw}} = 0.4$). 2 LLM calls per candidate.
- **A2A debate**: (1) Advocate evaluates focusing on strengths, (2) Critic evaluates focusing on gaps, (3) Judge reconciles both perspectives. 3 LLM calls per candidate.
- **Hybrid retrieval**: (1) Embed all resumes and the job description with `text-embedding-3-small`, (2) rank by cosine similarity, (3) LLM-evaluate top-K candidates only. Tested at $K \in \{5, 10, 20, 50\}$.

Format comparison. On the Phase 1 winning pipeline (multistage), we compare two output formats: JSON via LangChain's `JsonOutputParser`, and TOON using a custom parser for the compact key:value notation.

Table 1: Pipeline architecture comparison (mean ± std across 3 runs). Best values in **bold**.

Metric	Single-pass	Multistage	A2A Debate	Hybrid (k=50)
Top-20% F1	0.700 ± 0.000	**0.800 ± 0.000**	0.600 ± 0.041	0.696 ± 0.000
Spearman ρ	0.856 ± 0.009	**0.896 ± 0.004**	0.729 ± 0.003	0.764 ± 0.028
Kendall τ	0.694 ± 0.009	**0.742 ± 0.005**	0.555 ± 0.001	0.596 ± 0.028
Quintile exact	59.3%	**62.3%**	50.5%	51.3%
Quintile ±1	93.7%	**96.3%**	82.9%	86.0%
Cost/run (USD)	\$1.46	\$3.60	\$1.92	**\$0.73**
Latency/eval (s)	**6.7**	15.3	13.0	8.5
Error rate	0.0%	0.0%	0.3%	0.0%

Protocol. All experiments use GPT-4.1. Each condition is run 3 times; we report mean ± standard deviation. Metrics include top-20% precision, recall, and F1 (binary: advance vs. reject), Spearman ρ and Kendall τ (rank correlation with ground truth), quintile exact and ±1 agreement, token usage, estimated cost (USD), latency (ms), and parse success rate. Statistical significance across pipeline architectures is assessed via the Kruskal–Wallis H-test. All candidates per run are ranked by their final composite score; the top 20% by rank are classified as "advance."

Implementation. The system is implemented in Python 3.10 using LangChain 0.3.26 and Pydantic 2.11, with asynchronous evaluation at a concurrency level of 10 LLM calls.

3.2 Pipeline Architecture Comparison

Table 1 presents the main results across all four pipeline architectures. The multistage pipeline achieves the highest accuracy on every metric: F1 = 0.800, Spearman $\rho = 0.896$, and quintile ±1 agreement of 96.3%. A Kruskal–Wallis test confirms that the differences across architectures are statistically significant ($H = 16.31$, $p = 0.001$).

Figure 1 provides a visual overview. The multistage pipeline's advantage stems from its fusion of complementary signals: keyword extraction captures explicit requirement matches (specific tools, certifications, years of experience) that pure semantic evaluation overlooks, while the semantic stage captures holistic candidate–job fit. The 0.6/0.4 fusion weighting, while not optimized, proves effective.

The multi-agent debate architecture produces the lowest F1 (0.600), underperforming even the single-pass baseline despite using three times as many LLM calls. The advocate–critic framing introduces systematic bias: advocates inflate scores while critics deflate them, and the judge tends to average rather than reason, producing compressed score distributions with reduced discriminative power. Notably, debate has the lowest inter-run variance (std = 0.077)—it is consistent but consistently wrong.

All architectures achieve their highest quintile match rates at the extremes (Q1 and Q5). The multistage pipeline is the most balanced (80% Q1, 83% Q5), while middle quintiles are harder for all architectures, as borderline candidates are inherently ambiguous.

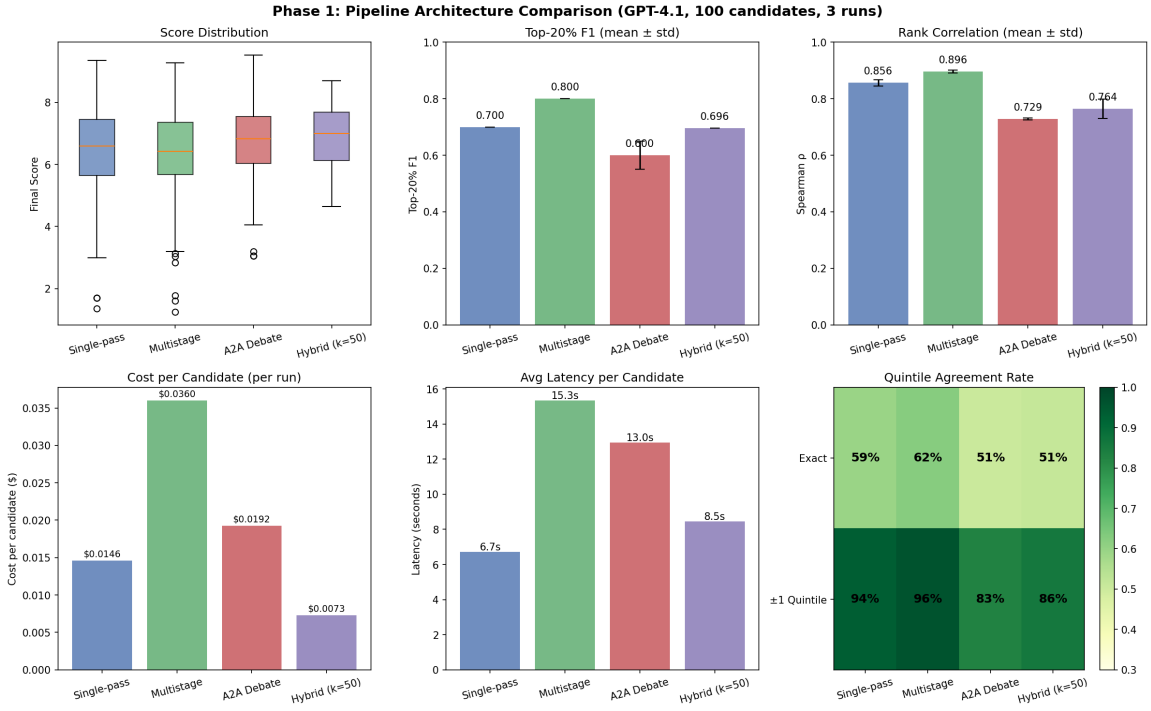

Figure 1: Phase 1 pipeline architecture comparison across 100 candidates and 3 runs. Top row: score distributions, top-20% F1, and Spearman rank correlation. Bottom row: cost per candidate, latency per candidate, and quintile agreement heatmap.

Table 2: Hybrid retrieval Recall@K at different shortlist sizes.

K	Recall@K	Cost Saving
5	0.00	95%
10	0.10	90%
20	0.30	80%
50	0.62	50%

3.3 Retrieval Pre-filtering Analysis

Table 2 and Figure 2 present the Recall@K analysis for the hybrid retrieval pipeline. Vector similarity pre-filtering shows a steep accuracy–cost tradeoff: at $K = 50$ (50% cost saving), only 62% of the top candidates identified by full LLM evaluation are recovered by the embedding-based shortlist.

The `text-embedding-3-small` model captures broad semantic similarity but lacks the granularity for keyword-level job matching. The low recall at small K values (0.00 at $K = 5$, 0.10 at $K = 10$) demonstrates that the candidates ranked highest by cosine similarity bear little resemblance to those ranked highest by full LLM evaluation. Specialized or fine-tuned embeddings trained on job–resume matching might improve recall substantially.

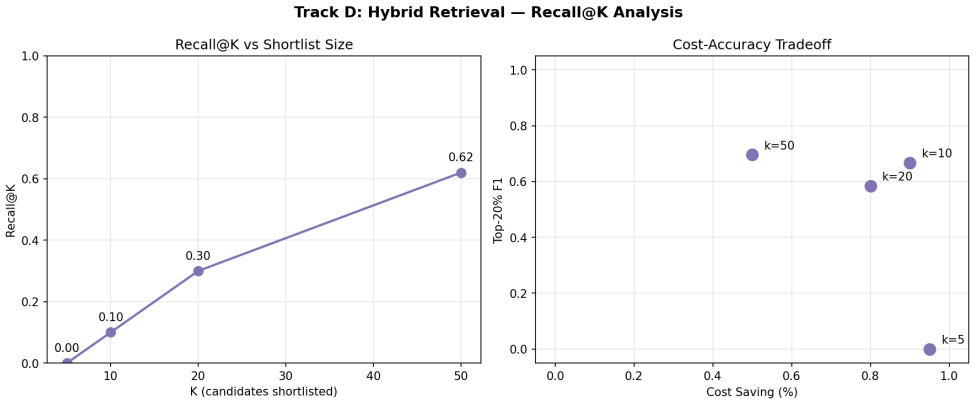

Figure 2: Left: Recall@K as a function of shortlist size. Right: cost–accuracy tradeoff showing each K value's position. Higher cost savings correspond to lower recall.

Table 3: Output format comparison on the multistage pipeline (mean ± std across 3 runs).

Format	Top-20% F1	Tokens/eval	Cost/run	Parse Success	Token Savings
JSON	**0.800 ± 0.000**	11,331	\$3.60	100%	—
TOON	0.733 ± 0.029	3,043	\$0.95	100%	73.1%

3.4 Output Format Comparison

Table 3 compares JSON and TOON output formats on the multistage pipeline. TOON achieves a 73.1% reduction in token usage (11,331 to 3,043 tokens per evaluation) at a modest F1 cost of 0.067, making each run 3.8× cheaper (\$0.95 vs. \$3.60).

Both formats achieve 100% parse success across 300 evaluations. The accuracy drop is concentrated in middle quintiles (Q2–Q4), where TOON's compact format may constrain the model's ability to reason through borderline cases; for Q1 and Q5 candidates, the difference is smaller. Figure 3 visualizes these differences.

For high-volume screening where slight accuracy loss is acceptable, TOON provides substantial cost savings. This finding extends the observations of Tam et al. (2024) to a specific applied task: format restrictions do affect performance, but the magnitude depends on the difficulty of the discrimination being made.

3.5 Limitations

Several limitations bound generalizability. All experiments use a single job description (Marketing Specialist); results may not transfer to technical or senior roles. Ground truth is human-labeled at the tier level, so within-tier rankings are interpolated rather than directly observed. Only GPT-4.1 was tested; other models may exhibit different architecture preferences. The dataset of 100 candidates is modest. The multistage fusion weights (0.6/0.4) were set by design rather than optimized via ablation. Finally, hybrid retrieval cost reflects only LLM calls on shortlisted candidates, excluding embedding computation.

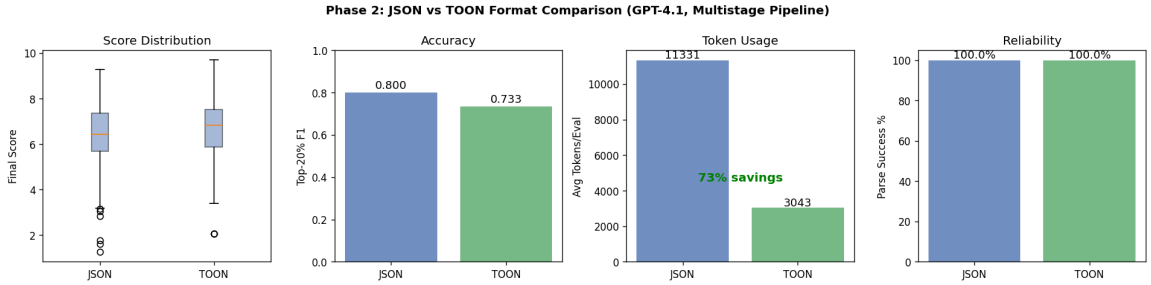

Figure 3: Phase 2 format comparison on the multistage pipeline. From left: score distributions, top-20% F1, average tokens per evaluation, and parse success rate.

4 Conclusion

We present a controlled empirical comparison of four LLM pipeline architectures and two output formats for automated candidate screening. Our results demonstrate that multistage pipelines with keyword–semantic fusion achieve the highest ranking accuracy ($F1 = 0.800$, Spearman $\rho = 0.896$), significantly outperforming single-pass, multi-agent debate, and hybrid retrieval approaches. The multi-agent debate architecture, despite its theoretical appeal and demonstrated benefits in factuality tasks (Du et al., 2024), underperforms even the single-pass baseline for candidate evaluation—the advocate–critic framing introduces conflicting signals that degrade ranking quality.

TOON format offers a compelling cost–accuracy tradeoff, reducing token usage by 73% with a modest F1 drop of 0.067. Both JSON and TOON achieve 100% parse reliability with GPT-4.1, making TOON a practical choice for cost-sensitive, high-volume screening deployments.

Is any technique good enough to declare the problem solved? At $F1 = 0.800$ and Spearman $\rho = 0.896$, the multistage pipeline achieves strong agreement with human-derived rankings, but the 20% miss rate on top candidates and the difficulty of ranking middle-quintile applicants indicate that fully automated screening without human review remains premature. These systems are best deployed as decision-support tools that surface a ranked shortlist for human judgment.

Based on these findings, we offer the following practical recommendations: (1) use multistage pipelines when accuracy is the primary concern; (2) use single-pass evaluation for budget-constrained deployments where the cost of multi-stage processing is not justified; (3) adopt TOON format for high-volume screening where cost reduction outweighs modest accuracy loss; and (4) avoid multi-agent debate architectures for candidate evaluation, as the added complexity reduces rather than improves accuracy.

Future work should investigate fusion weight optimization through ablation studies, fine-tuned embedding models for retrieval pre-filtering, generalization across diverse job types and candidate pools, and cross-model comparisons to determine whether these architectural preferences hold for other LLMs such as Claude and Gemini.

References

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, 2024. URL https://arxiv.org/abs/2305.14325.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. JSONSchemaBench: A rigorous benchmark of structured outputs for language models. *arXiv preprint arXiv:2501.10868*, 2025. URL https://arxiv.org/abs/2501.10868.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020. URL https://arxiv.org/abs/2005.11401.

Frank P.-W. Lo, Jianing Qiu, Zeyu Wang, Haibao Yu, Yeming Chen, Gao Zhang, and Benny Lo. AI hiring with LLMs: A context-aware and explainable multi-agent framework for resume screening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4223–4232, 2025. URL https://arxiv.org/abs/2504.02870.

Johann Schopplich. TOON: Token-oriented object notation specification. GitHub repository, 2025. URL https://github.com/toon-format/spec.

Charles Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. Let me speak freely? A study on the impact of format restrictions on performance of large language models. *arXiv preprint arXiv:2408.02442*, 2024. URL https://arxiv.org/abs/2408.02442.

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. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.05685.