

Beyond Facts: Benchmarking Distributional Reading Comprehension in Large Language Models

Pei-Fu Guo¹, Ya-An Tsai¹, Chun-Chia Hsu¹, Kai-Xin Chen¹, Yun-Da Tsai¹,
Kai-Wei Chang², Nanyun Peng², Mi-Yen Yeh³, Shou-De Lin^{1,4}

¹National Taiwan University ²University of California, Los Angeles

³Academia Sinica, Taiwan ⁴NTU AI-CoRE

Correspondence: r12922217@csie.ntu.edu.tw

Abstract

While most reading comprehension benchmarks for LLMs focus on factual information that can be answered by localizing specific textual evidence, many real-world tasks require understanding distributional information, such as population-level trends and preferences expressed across collections of text. We introduce TEXT2DISTBENCH, a reading comprehension benchmark for evaluating LLMs' ability to infer distributional knowledge from natural language. Built from real-world YouTube comments about movie and music entities, the benchmark provides models with entity metadata and associated comments, and requires them to answer distributional questions, such as estimating the proportions of positive and negative comments, or identifying the most and second most frequent topics discussed among viewers. To support reliable and long-term evaluation, the construction pipeline of TEXT2DISTBENCH is fully automated and continuously updated to incorporate newly emerging entities over time. Experiments across multiple LLMs show that while models substantially outperform random baselines, performance varies widely across different distribution types and characteristics. These findings highlight both the capabilities and limitations of current LLMs in distributional reading comprehension and demonstrate the value of TEXT2DISTBENCH as a practical and scalable testbed for future research.¹

1 Introduction

Large language models (LLMs) are increasingly used as general-purpose language understanding systems, where they must comprehend diverse information from text and produce appropriate responses. Most existing reading comprehension benchmarks focus on *factual knowledge*, where answers can be obtained by retrieving or reasoning

¹Code and data are available at [link](#).

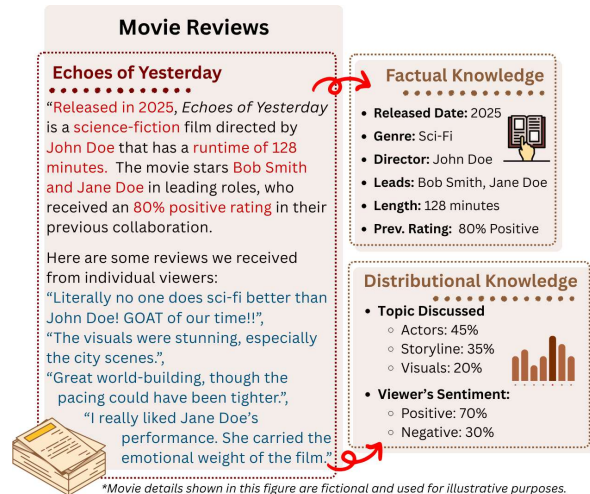


Figure 1: Factual vs. Distributional Knowledge.

over specific pieces of textual evidence, such as sentence spans (Rajpurkar et al., 2016; Kwiatkowski et al., 2019; Dua et al., 2019) or a set of localized facts (Tafjord et al., 2019; Welbl et al., 2018).

However, many real-world information needs go beyond factual knowledge and require models to understand *distributional knowledge* expressed across text. As illustrated in Figure 1, factual information such as movie genre or director is localized and can be answered by identifying specific textual evidence. In contrast, distributional information, such as proportion of viewer sentiment or topic prevalence, must be derived by aggregating patterns across many individual comments. This form of understanding is important as LLMs are increasingly used to support human decision making and analyze public opinion in real-world settings such as market research and product review analysis, where users summarize collective opinions, identify dominant trends, and compare preferences across aspects from large volumes of text.

Recent work has explored models' *distributional knowledge* from different perspectives. Some benchmarks study models' distributional knowl-

edge of human values, using survey-style questions to probe models’ beliefs about cultural values or political attitudes associated with specific demographic groups (Durmus et al., 2023; Meister et al., 2025; Röttger et al., 2024; Zhao et al., 2024). Other benchmarks focus on probabilistic reasoning, where models are provided with explicitly specified distributions (e.g., Poisson distributions) or numeric data and are asked to calculate statistics or sample outcomes (Paruchuri et al., 2024; Pourneemat et al., 2025; Ozturkler et al., 2023). While valuable, these benchmarks primarily probe distributional knowledge encoded in models during pretraining. They do not evaluate whether models can derive population-level distributions, such as the *frequencies* or *prevalences* of different opinions, by reading and aggregating information from natural language text.

Based on this gap, we introduce TEXT2DISTBENCH, a reading comprehension benchmark designed to systematically evaluate LLMs’ ability to infer *distributional knowledge* from text. Constructed from real-world YouTube comments, the benchmark provides models with metadata and user comments for movie and music entities released after the model knowledge cutoff date. Specifically, each benchmark instance includes: (1) **entity metadata**, which provides background information needed to interpret the comments; (2) a set of **human comments** associated with the entity; and (3) **distributional QA pairs**, which require models to aggregate information across comments to infer population-level statistics (e.g., estimating the percentage of positive and negative comments or identifying the most frequently discussed topics).

To define the underlying distributions, each comment is automatically annotated using multiple LLMs across two attributes: *sentiment* (e.g., positive or negative) and *topic* (e.g., acting, storyline, visuals, or audio). These annotations induce a discrete distribution over attributes for each entity. Based on this formulation, we design questions that probe different aspects of distributional understanding. Specifically, models are asked to estimate category proportions (*estimation*) and to identify the most and second most frequent categories (*mode queries*). These questions are instantiated over marginal distributions (e.g., overall sentiment or topic prevalence), conditional distributions (e.g., viewer sentiment given a topic), and joint distributions that capture attribute co-occurrence patterns (e.g. percentage of users that are positive on the

lyrics of a song).

In addition, TEXT2DISTBENCH is constructed through a fully automated pipeline that supports continuous updates. By continuously introducing newly emerging entities, the benchmark reduces the risk of data leakage, where models may have already been exposed to the entity-related information during training, and instead encourages them to derive distributions from the provided text. This design makes TEXT2DISTBENCH more sustainable and allows it to remain reliable as LLMs and their training data evolve over time.

Using TEXT2DISTBENCH, we evaluate a range of state-of-the-art LLMs. Our experiments show that while current models substantially outperform random baseline, their performance varies widely across distributional settings. In particular, models generally perform better on questions related to marginal distributions than on conditional or joint distributions, and are sensitive to intrinsic properties of the target distribution (e.g., uniformity, probability mass concentration). Moreover, we find that models can form informative prior beliefs from factual information alone, which often closely approximate the target distributions even for unseen entities. These findings highlight both the strengths and limitations of current LLMs in understanding population-level information from text and demonstrate the utility of TEXT2DISTBENCH for distributional reading comprehension.

2 Related Work

Reading Comprehension Benchmarks Most reading comprehension benchmarks focus on evaluating *factual knowledge* that can be derived from text, where answers can be obtained by retrieving or reasoning over specific pieces of textual evidence, such as sentence spans or localized facts. Representative benchmarks include SQuAD, WikiReading, Natural Questions, and DROP, which evaluate factual understanding over short or long documents (Rajpurkar et al., 2016; Hewlett et al., 2016; Kwiatkowski et al., 2019; Dua et al., 2019). Other benchmarks emphasize more advanced forms of comprehension, such as contextual commonsense reasoning (Huang et al., 2019), qualitative relations (Tafjord et al., 2019), and multi-hop reasoning (Welbl et al., 2018). In contrast, TEXT2DISTBENCH focuses on distributional knowledge, where the goal is to infer population-level patterns and statistics by aggregating informa-

tion across a collection of textual comments.

Probabilistic Reasoning Benchmarks A separate line of work evaluates models’ understanding of probabilistic concepts (Paruchuri et al., 2024; Pournemat et al., 2025; Freedman and Toni, 2025). In these benchmarks, models are typically provided with explicitly specified distributions (e.g., Normal or Poisson distributions) or tabular numerical data, and are asked to estimate percentiles, sample outcomes, or compute specific probabilities. In contrast to these settings, where the underlying distributions are directly given or numerically represented, TEXT2DISTBENCH requires models to infer distributional properties from unstructured natural language.

Human Values Distribution Benchmarks Recent work has evaluated LLMs by comparing their response distributions to human distributions, often using survey-style questions and divergence-based metrics (Durmus et al., 2023; Meister et al., 2025). These benchmarks typically probe models’ beliefs about cultural values (Naous et al., 2024; Wang et al., 2024), political attitudes (Röttger et al., 2024; Stambach et al., 2024), and public opinions (Zhao et al., 2024) associated with specific demographic groups. In contrast, TEXT2DISTBENCH evaluates distributional understanding grounded in text, requiring models to derive population-level statistics by aggregating information from the provided context rather than relying on pretrained knowledge.

3 Methodology: Benchmark Overview

TEXT2DISTBENCH is a reading comprehension benchmark designed to evaluate a model’s ability to understand *distributional knowledge* from text. We use movie and music comments from YouTube as data sources and construct reading comprehension questions that require models to infer different properties of the distributions expressed across the comments. Each TEXT2DISTBENCH instance consists of three components: (1) *entity metadata*, which provides contextual background for interpreting the comments; (2) a set of *human comments* associated with the entity; and (3) *distributional QA pairs*, which ask information about the underlying distribution expressed in the comments. To enable systematic evaluation, questions are organized along two axes: *distribution type*, which specifies the distribution being queried, and *task type*, which specifies how the model is asked about

that distribution.

3.1 Distribution Type

In real-world scenarios, distributional information can come from different types of distributions. For example, in product reviews that include ratings and topic tags, determining whether overall feedback is mostly positive or negative corresponds to a marginal distribution, while analyzing sentiment toward a specific product feature corresponds to a conditional distribution.

In TEXT2DISTBENCH, each viewer comment is annotated along two dimensions: *sentiment* S (e.g., positive or negative) and *topic* T (e.g., Actor, Storyline, Visual, Audio). Together, these annotations define a discrete joint distribution $P(S, T)$ over the comment set. Based on this formulation, we define three *distribution types*, each corresponding to a distribution derived from $P(S, T)$: **Marginal distributions**, such as $P(S)$ and $P(T)$, capture the overall distribution of sentiment or topic across all comments; **Conditional distributions**, such as $P(S | T)$ and $P(T | S)$, describe how one variable is distributed given a specific value of the other; and **Joint distributions**, $P(S, T)$, represent the full co-occurrence between sentiment and topic. These distribution types specify which distribution a question targets. Question examples associated with each distribution type are shown in Table 1.

3.2 Task Types

We design three tasks to probe model understanding for each distribution type. Question examples of each task are shown in Table 2.

(1) Estimation The model is asked to estimate the full numerical distribution by predicting the percentage associated with each support category. This task evaluates whether the model can recover the overall shape of the distribution and correctly allocate probability mass across categories.

(2) Most Frequent The model is asked to identify the support category with the largest probability mass. This task tests whether the model can reliably recognize the dominant trend in the distribution, corresponding to use cases where users primarily care about the majority outcome.

(3) Second Frequent The model is asked to identify the support category with the second-largest probability mass. This task probes a more subtle form of distributional understanding, requiring

Distribution	Question Example (Task Type: Most Frequent)
Marginals	$P(S)$: What overall attitude do most viewers express? Positive or Negative. $P(T)$: Which aspect of the movie is discussed most often? Actor, Storyline, Visual, or Audio.
Conditionals	$P(S T)$: Among the comments that talk about the actor, what attitude do viewers express most commonly? Positive or Negative. $P(T S)$: Among the comments that express a positive attitude, which aspect of the movie is mentioned most often? Actor, Storyline, Visual, or Audio.
Joint	$P(S, T)$: Considering both (1) which aspect (Actor, Storyline, Visual, Audio) is being talked about and (2) whether the attitude is positive or negative. Which (aspect, sentiment) combination appears most often in the comments?

Table 1: Question Examples of different distribution type.

Question	Question Example (Distribution Type: Marginals)
Estimation	Movie Information: {meta_data} YouTube Viewer Comments: {comments} What aspects of the movie do viewers talk about? Summarize how frequently each aspect appears in the comments using percentages. Output your answer in the following format: { "Actor": "<int>%", "Storyline": "<int>%", "Visual": "<int>%", "Audio": "<int>%" }
Most Frequent	Movie Information: {meta_data} YouTube Viewer Comments: {comments} Which aspect of the movie is discussed most often? Actor, Storyline, Visual, or Audio.
Second Frequent	Movie Information: {meta_data} YouTube Viewer Comments: {comments} Which aspect of the movie is discussed the second most often? Actor, Storyline, Visual, or Audio.

Table 2: Question Examples of different task type

the model to correctly rank alternatives beyond the dominant mode.

3.3 Evaluation Metrics

Evaluation metrics are defined by tasks, independent of the distribution type.

(1) Estimation In this task, the model outputs a distribution. To quantify the discrepancy between the predicted distribution \hat{p}_i and the ground-truth distribution p_i , we follow prior work on measuring distributional alignment (Pournemat et al., 2025; Meister et al., 2025; Gupta et al., 2025) and adopt the *Total Variation Distance (TVD)* as the evaluation metric. TVD measures the minimum amount of probability mass that must be redistributed to transform \hat{p}_i into p_i and is invariant to the size of the support:

$$\text{TVD}(\hat{p}_i, p_i) = \frac{1}{2} \sum_{k=1}^K |\hat{p}_{i,k} - p_{i,k}| \quad (1)$$

where K denotes the number of support categories. Since TVD is bounded in $[0, 1]$, with lower values

indicating better alignment, we report $1 - \text{TVD}$ for consistency with other metrics where higher values represent better performance.

(2) Most / Second Frequent In this task, the model identifies the category with the largest or second-largest probability mass, and performance is evaluated using classification accuracy :

$$\text{ACC} = \mathbf{1}[\hat{y}_i = y_i] \quad (2)$$

where \hat{y}_i is the model’s prediction and y_i is the ground-truth category label.

4 Methodology: Benchmark Construction

TEXT2DISTBENCH is constructed through a fully automated data generation pipeline designed to support continuous updates. As illustrated in Figure 2, the pipeline consists of four stages: (1) entity selection, (2) comment annotation, (3) empirical distribution estimation, and (4) QA generation.

Stage 1: Opinion Entity Selection To reliably evaluate whether models can infer distributional knowledge from the provided text, it is crucial that

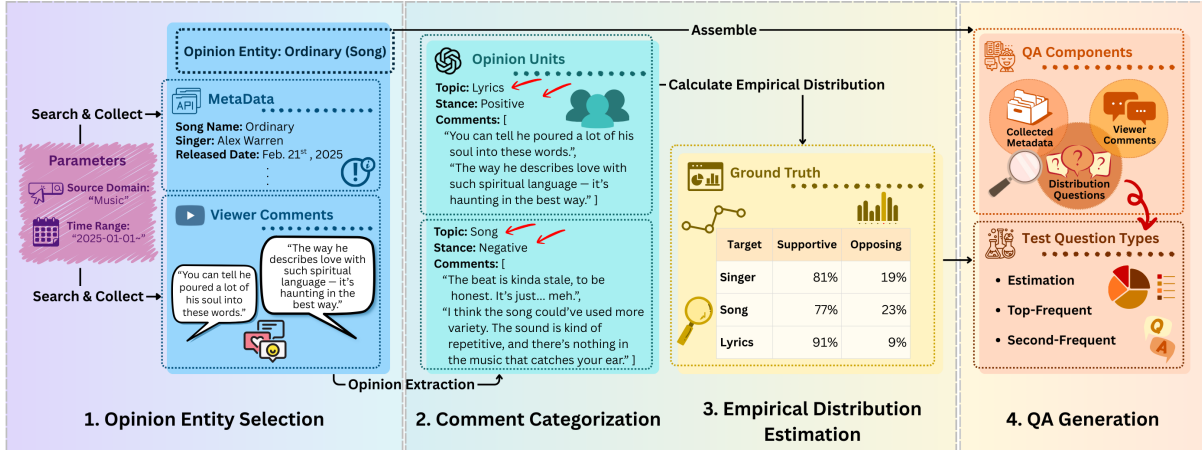


Figure 2: **TEXT2DISTBENCH Generation Pipeline.** The construction process consists of four stages: (1) selecting valid opinion entities; (2) annotating comment topic and sentiment; (3) estimating empirical opinion distributions; and (4) generating distributional reading-comprehension questions.

target entities are not already familiar to the model. If an entity has been widely discussed during pre-training, the model may have already encountered entity-related comments or human opinions, rather than actually comprehending the provided input, leading to a contaminated evaluation.

To mitigate this issue, we select entities according to two criteria. First, each entity must appear at least six months after the model’s pretraining cutoff date, reducing the likelihood that large-scale public discussions are included in the training data. Second, entities must be largely self-contained, such that audience reactions are primarily driven by the entity itself rather than by other similar or previously existing entities. Based on these criteria, newly released movies and musics are well-suited domains for our benchmark.

In practice, we retrieve candidate entities using domain-specific APIs, including IMDB and TMDb for movies and YouTube Music for songs. These APIs allow us to filter entities by release time and obtain relevant metadata. For each entity, we then use the YouTube Data API to retrieve associated videos (e.g., movie trailers or music videos). Entities with comment volumes below a minimum threshold are discarded. For the retained entities, viewer comments and metadata are collected. Additional details of the retrieval and filtering process are provided in Appendix B.1.

Stage 2: Comment Annotation For each selected entity, viewer comments are automatically annotated with two attributes: *topic* and *sentiment*. Motivated by prior work showing that LLMs can match or even outperform human annotators on

common text classification tasks (Gilardi et al., 2023), we employ multiple state-of-the-art LLMs to independently assign a (*topic, sentiment*) label to each comment. We then apply majority voting across model outputs to mitigate model-specific biases and improve annotation robustness (Farr et al., 2024; Qiu et al., 2025). Comments for which no majority is reached, or that are identified as expressing multiple topics, are discarded.

To assess annotation quality, we conduct human verification on all comments in the dataset (Section 5). The automated annotations achieve 95% precision against human judgments, indicating high annotation reliability. Details of the annotation process, prompt design, and human evaluation are provided in Appendix B.2 and C.

Stage 3: Empirical Distribution Estimation

Following annotation, each entity is associated with a collection of comments, each assigned a (*topic, sentiment*) label. Treating each comment as one observation, we compute an empirical joint distribution by counting label frequencies and normalizing over all comments. From this joint distribution, we derive the corresponding marginal and conditional distributions, which serve as ground-truth statistics for distributional question answering.

Stage 4: QA Generation

Finally, we generate distributional reading-comprehension questions using the templates described in Section 3. Each benchmark instance consists of entity metadata, the associated viewer comments, and a set of question-answer pairs, where each answer is computed directly from the empirical distributions. Examples

are provided in Appendix D.

Benchmark Maintenance TEXT2DISTBENCH supports easy maintenance. By periodically retrieving newly released entities and re-running the automated generation pipeline, the benchmark can be continuously updated with new content. The entire process requires no human annotation, enabling up-to-date evaluation for newly released LLMs.

5 Experimental Setup

Models We evaluate both closed-source and open-source LLMs, including GPT-5.1², GEMINI-2.5-PRO³, GROK-4-FAST⁴, CLAUDE-SONNET-4.5⁵, QWEN3-32B⁶, and LLAMA3.3-70B⁷. These models vary in pretraining data, parameter scale, and alignment strategies, providing a diverse testbed for evaluation.

Benchmark Configuration Since TEXT2DISTBENCH is continuously updated, we construct a dataset using the pipeline in Section 4. The dataset consists of 20 movie and music entities released between July and October 2025,⁸ Under this configuration, each task type yields 40 marginal, 120 conditional, and 20 joint questions, resulting in a total of 540 evaluation questions across the three tasks. For each entity, we randomly sample 50 comments from the associated videos to balance question token length. Dataset statistics and construction details are reported in Appendix A.

Inference Configuration We evaluate all models using fixed zero-shot prompt templates. Each prompt includes definitions of sentiment and topic, the entity metadata, a set of associated comments, and a distributional question. Full prompt templates are provided in Appendix D. All models are evaluated in text-only mode, without tool calling or retrieval. Each question is sampled once using the model’s default decoding parameters, including temperature and maximum token limits.

²<https://platform.openai.com/docs/models/gpt-5.1>

³https://aistudio.google.com/app/prompts/new_chat?model=gemini-2.5-pro

⁴<https://x.ai/news/grok-4-fast>

⁵<https://platform.claude.com/docs/en/about-claude/models/overview>

⁶<https://huggingface.co/Qwen/Qwen3-32B>

⁷<https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>

⁸The most recent model evaluated, GEMINI-2.5-PRO, has a knowledge cutoff date of 2025-01.

6 Results & Analysis

Building on the setup described in Section 5, we first present the main results, followed by additional analyses on model behavior.

6.1 Overall Performance and Ranking

Table 3 summarizes model performance on TEXT2DISTBENCH across different distribution and task types. Across all settings, every model substantially outperforms the random baseline, indicating that current LLMs can understand non-trivial distributional information from text.

Model rankings differ across task types. For estimation task, GPT-5.1 achieves the highest average performance, with GEMINI-2.5-PRO and GROK-4-FAST closely following, and a similar ordering is observed for most-frequent task. In contrast, second-frequent task favors GEMINI-2.5-PRO, with GPT-5.1 and GROK-4-FAST ranking slightly lower. Mid-tier models such as CLAUDE-SONNET-4.5 and LLAMA-3.3-70B remain competitive on estimation tasks but exhibit larger drops on most/second frequent questions, while QWEN3-32B consistently ranks lowest despite outperforming the random baseline.

Across all tasks, a consistent difficulty ordering emerges across distribution types. Marginal questions are the easiest for models, followed by conditional questions, while joint distribution questions are the most challenging. This trend holds for estimation tasks and is especially pronounced for most/second frequent questions, where performance drops sharply on joint distributions.

Overall, while several models consistently form a top-performing group, their performance varies across distribution and task types. This implies that, in practice, model selection should depend on the specific information the user is interested in.

6.2 Target Distribution Characteristics

Figure 3 analyzes how model performance correlates with intrinsic properties of the target distributions. We characterize each ground-truth distribution using four statistics: (1) *JS divergence from the uniform distribution*, which measures the degree of non-uniformity; (2) *Top-1 probability mass*, which captures the dominance of the most frequent category; (3) *Top-2 margin*, defined as the probability difference between the most and second most frequent categories, reflecting how sharply peaked the distribution is; and (4) *Support size*, the number

Model	Estimation (1-TVD) \uparrow				Most Frequent (ACC) \uparrow				Second Frequent (ACC) \uparrow			
	M	C	J	Avg	M	C	J	Avg	M	C	J	Avg
GPT-5.1	0.930	0.865	0.847	0.881	0.861	0.877	0.778	0.839	0.834	0.805	0.556	0.732
Gemini-2.5-Pro	0.903	0.869	0.805	0.859	0.805	0.886	0.722	0.804	0.778	0.835	0.667	0.760
Grok-4-Fast	0.911	0.856	0.838	0.868	0.889	0.858	0.667	0.805	0.833	0.728	0.611	0.724
Claude-Sonnet-4.5	0.907	0.818	0.828	0.851	0.833	0.809	0.611	0.751	0.805	0.728	0.500	0.678
Llama-3.3-70B	0.855	0.836	0.744	0.812	0.778	0.819	0.667	0.755	0.805	0.728	0.278	0.604
Qwen3-32B	0.701	0.747	0.619	0.689	0.777	0.743	0.500	0.673	0.555	0.495	0.167	0.406
Random Baseline	0.651	0.594	0.485	0.577	0.375	0.435	0.132	0.314	0.375	0.444	0.125	0.315

Table 3: **TEXT2DISTBENCH Results** We report performance across distribution (Marginals, Conditionals, Joint) and task types. Random baselines are computed from 50 runs of random guessing for Most/Second Frequent questions and from a uniform distribution for Estimation questions. Higher values indicate better performance.

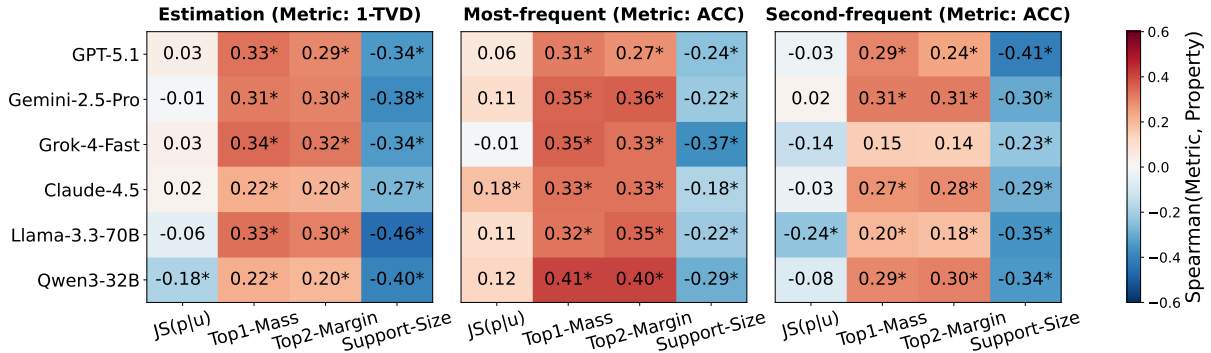


Figure 3: **Sensitivity to target distribution characteristics.** Each cell reports the Spearman correlation between model performance and distribution statistics. (*) indicates statistical significance ($pvalue < 0.05$).

of categories with non-zero probability mass.

Across models and tasks, we observe consistent trends. Performance exhibits positive rank correlations with both Top-1 probability mass and Top-2 margin, indicating that distributions with stronger dominance and clearer separation between leading categories are easier for models to handle. In contrast, support size shows a negative correlation with performance, suggesting that distributions spread across more categories pose greater difficulty. Finally, performance shows little systematic correlation with JS divergence from the uniform distribution, indicating that overall non-uniformity alone is not a strong predictor of model success.

6.3 Scaling Reasoning Effort

As an additional analysis, we examine whether increasing reasoning effort improves model performance. We use GPT-5.1 as the test model and vary the reasoning effort from low to high. Figure 4 shows the performance of GPT-5.1 under different reasoning-effort settings, with GPT-4.1, a strong non-reasoning model from the same family, included as a comparison baseline.

Overall, increasing reasoning effort leads to

consistent performance improvements, with the largest gains observed for joint distribution questions. Across task types, estimation and most-frequent questions benefit more from additional reasoning effort than second-frequent questions. Performance improves as reasoning effort increases up to a medium level, after which gains diminish and may slightly degrade, indicating diminishing returns from further increases in reasoning effort.

6.4 Prior Belief from Factual Information

Beyond the main experiments, we consider an extreme setting in which viewer comments are removed from the input and only entity metadata is provided. This metadata-only setting mirrors a common human behavior of forming initial population-level beliefs from factual information alone, without observing individual responses.

Figure 5 compares metadata-only (prior) and full-context (posterior) performance across the three task types. Across all models, prior predictions consistently outperform random baselines, indicating that models can leverage factual metadata to form non-trivial distributional hypotheses even for previously unseen entities. Importantly,

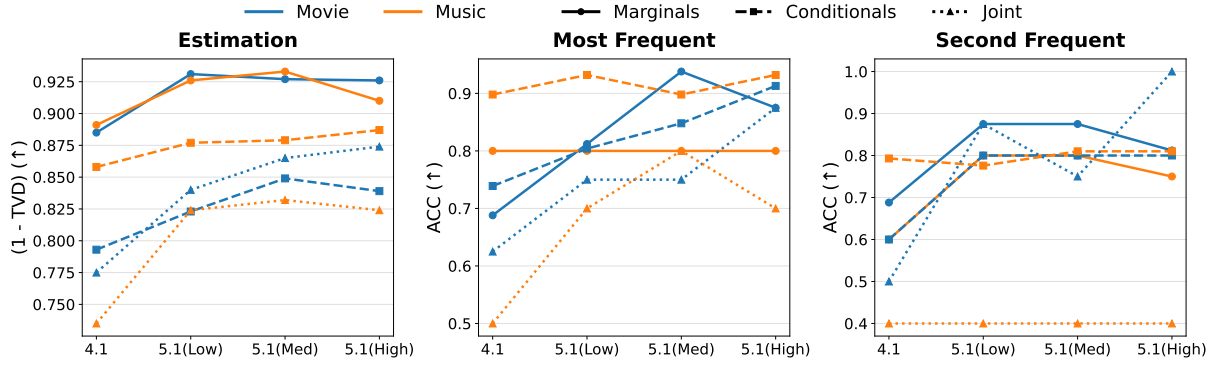


Figure 4: **Effect of scaling reasoning effort.** Performance of GPT-5.1 under different reasoning-effort settings, with GPT-4.1 shown as a non-reasoning baseline for comparison.

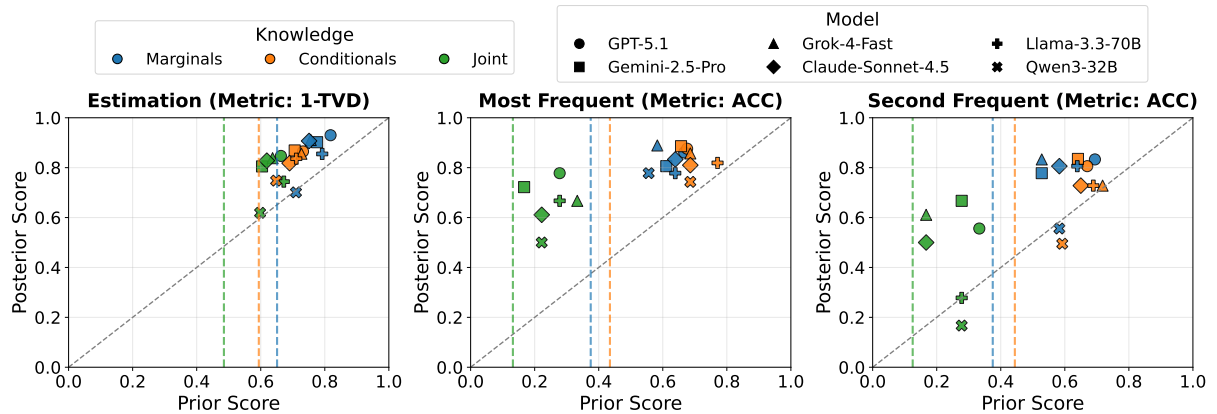


Figure 5: **Prior belief from factual information.** Each subplot compares prior (metadata-only) and posterior (with comments) performance for one task. The diagonal denotes equal prior and posterior performance, with points above it indicating improvements after observing viewer comments. Colored vertical dashed lines indicate random baselines for each distribution type.

Model	Esti- mation	Most Frequent	Second Frequent
GPT-5.1	0.142	0.220*	0.246*
Gemini-2.5-Pro	0.215*	0.221*	0.316*
Grok-4-Fast	0.186*	0.341*	0.181*
Claude-Sonnet-4.5	0.069	0.274*	0.158*
Llama-3.3-70B	0.219*	0.154*	0.122
Qwen3-32B	0.255*	0.304*	0.077

Table 4: Spearman correlation between model prior and posterior performance. (*) indicates statistical significance $pvalue < 0.05$.

incorporating viewer comments leads to consistent performance improvements, with most points lying above the diagonal, including cases where prior predictions (e.g., joint distribution) are weak.

Table 4 further shows that prior and posterior performance are positively correlated for most models and tasks. This suggests that models with stronger prior distributional beliefs tend to benefit more after observing comments. Together, these results

demonstrate that models actively integrate textual evidence to refine their initial beliefs, highlighting the value of `TEXT2DISTBENCH` for evaluating text-conditioned distributional understanding.

7 Conclusion

In this work, we introduced `TEXT2DISTBENCH`, an automated and continuously updatable benchmark for evaluating distributional reading comprehension in LLMs. Our benchmark probes models’ ability to infer distribution knowledge from text, spanning marginal, conditional, and joint distributions under multiple question types. Our empirical results show that while LLMs substantially outperform random baselines, their performance varies widely across different distribution settings. We also find that LLMs can form informative prior beliefs for unseen entities from factual information alone, and that models with stronger priors tend to benefit more from incorporating viewer comments.

Together, these findings highlight both the capabilities and limitations of current LLMs in distributional reading comprehension, and demonstrate the value of TEXT2DISTBENCH as a practical and scalable testbed for future research.

Limitations

While TEXT2DISTBENCH provides an automated and continuously updatable benchmark for systematically evaluating distributional reading comprehension, several limitations remain that could be addressed in future work. Currently, the benchmark defines three tasks. However, in real-world scenarios, humans may ask a wider range of questions about distributional information expressed in text, such as threshold-based or comparative queries. Extending the framework to support a richer set of query types would enable a more comprehensive evaluation of distributional understanding. Moreover, expanding the benchmark to more domains could further improve its coverage and provide a more diverse testbed for evaluation.

Ethical Considerations

TEXT2DISTBENCH is constructed using publicly available APIs, including IMDB/TMDB for movie metadata and YouTube Data API for music metadata and viewer comments. While these sources may occasionally contain inaccuracies due to human error or reporting delays, they are widely recognized and provide verifiable records of real-world content. All data are derived from post-release or officially published materials, and no private, sensitive, or personally identifiable information is included in the benchmark.

Use of Ai Assistants

In this work, we leveraged large language models (LLMs) to assist research in two ways. First, multiple LLMs were employed as part of the automated benchmark generation pipeline, including comment annotation and relevant video verification. Second, an AI assistant (OpenAI GPT-5.2) was used for minor writing support, such as grammar correction and improving manuscript clarity. All AI-assisted contents were carefully reviewed by the authors to ensure factual accuracy and consistency with the authors' original intent.

Acknowledgment

This material is based upon work supported by National Science and Technology Council, ROC under grant number 114-2221-E-002-134-MY3 and 113-2628-E-001-003-MY4, NTU AI Center of Research Excellence within Taiwan Centers of Excellence in Artificial Intelligence, and by National Taiwan University and Academia Sinica Innovative Joint Program, under grant AS-NTU-114-06.

References

- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.
- Esin Durmus, Karina Nguyen, Thomas I. Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, Liane Lovitt, Sam McCandlish, Orowa Sikder, Alex Tamkin, Janel Thamkul, Jared Kaplan, Jack Clark, and Deep Ganguli. 2023. [Towards measuring the representation of subjective global opinions in language models](#). *arXiv preprint arXiv:2306.16388*.
- David Farr, Nico Manzonelli, Iain Cruickshank, Kate Starbird, and Jevin West. 2024. [Llm chain ensembles for scalable and accurate data annotation](#). *Preprint*, arXiv:2410.13006.
- Gabriel Freedman and Francesca Toni. 2025. [Exploring the potential for large language models to demonstrate rational probabilistic beliefs](#). *Preprint*, arXiv:2504.13644.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. [Chatgpt outperforms crowd workers for text-annotation tasks](#). *Proceedings of the National Academy of Sciences*, 120(30).
- Ritwik Gupta, Rodolfo Corona, Jiaxin Ge, Eric Wang, Dan Klein, Trevor Darrell, and David M. Chan. 2025. [Enough coin flips can make llms act bayesian](#). *Preprint*, arXiv:2503.04722.
- Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. [WikiReading: A novel large-scale language understanding task over Wikipedia](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1545, Berlin, Germany. Association for Computational Linguistics.
- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. [Cosmos QA: Machine reading](#)

- comprehension with contextual commonsense reasoning. 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 2391–2401, Hong Kong, China. Association for Computational Linguistics.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Ming-Wei Chang, Andrew Zhai, Jacob Davis, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Nicole Meister, Carlos Guestrin, and Tatsunori Hashimoto. 2025. [Benchmarking distributional alignment of large language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2025), Volume 1: Long Papers*, pages 24–49. Association for Computational Linguistics.
- Tarek Naous, Michael Ryan, Alan Ritter, and Wei Xu. 2024. [Having beer after prayer? measuring cultural bias in large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16366–16393, Bangkok, Thailand. Association for Computational Linguistics.
- Batu Ozturkler, Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. 2023. [Thinksum: Probabilistic reasoning over sets using large language models](#). *Preprint*, arXiv:2210.01293.
- Akshay Paruchuri, Jake Garrison, Shun Liao, John Hernandez, Jacob Sunshine, Tim Althoff, Xin Liu, and Daniel McDuff. 2024. [What are the odds? language models are capable of probabilistic reasoning](#). *Preprint*, arXiv:2406.12830.
- Mobina Pournemat, Keivan Rezaei, Gaurang Sriraman, Arman Zarei, Jiaxiang Fu, Yang Wang, Hamid Eghbalzadeh, and Soheil Feizi. 2025. [Reasoning under uncertainty: Exploring probabilistic reasoning capabilities of llms](#). *Preprint*, arXiv:2509.10739.
- Jiaying Qiu, Dongliang Guo, Natalie Papini, Noelle Peace, Hannah F. Fitterman-Harris, Cheri A. Levinson, Tom Hartvigsen, and Teague R. Henry. 2025. [Labeling free-text data using language model ensembles](#). *Preprint*, arXiv:2501.08413.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Paul Röttger, Valentin Hofmann, Valentina Pyatkin, Musashi Hinck, Hannah Rose Kirk, Hinrich Schütze, and Dirk Hovy. 2024. [Political compass or spinning arrow? towards more meaningful evaluations for values and opinions in large language models](#). *Preprint*, arXiv:2402.16786.
- Dominik Stambach, Philine Widmer, Eunjung Cho, Caglar Gulcehre, and Elliott Ash. 2024. [Aligning large language models with diverse political viewpoints](#). *Preprint*, arXiv:2406.14155.
- Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. [QuaRTz: An open-domain dataset of qualitative relationship questions](#). 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 5941–5946, Hong Kong, China. Association for Computational Linguistics.
- Wenxuan Wang, Wenxiang Jiao, Jingyuan Huang, Ruyi Dai, Jen tse Huang, Zhaopeng Tu, and Michael Lyu. 2024. [Not all countries celebrate thanksgiving: On the cultural dominance in large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6349–6384, Bangkok, Thailand. Association for Computational Linguistics.
- Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. [Constructing datasets for multi-hop reading comprehension across documents](#). *Transactions of the Association for Computational Linguistics*, 6:287–302.
- Wenlong Zhao, Debanjan Mondal, Niket Tandon, Danica Dillion, Kurt Gray, and Yuling Gu. 2024. [World-valuesbench: A large-scale benchmark dataset for multi-cultural value awareness of language models](#). *Preprint*, arXiv:2404.16308.

A Benchmark Statistics

Following the experimental setup described in Section 5, we construct TEXT2DISTBENCH using entities released between July and October 2025. The benchmark contains 20 entities in total, evenly split across two domains (10 movies and 10 music entities). Each entity is annotated with two attributes: *sentiment* ($|S| = 2$) and *topic* ($|T| = 4$), which together define the underlying distributions. For each entity, we generate reading comprehension questions over three distribution types: marginals ($P(S)$ and $P(T)$), conditionals ($P(S|T)$ and $P(T|S)$), and the joint distribution ($P(S, T)$). Each question is instantiated under three *task types*: Estimation, Most-Frequent, and Second-Frequent. For a single task, each entity is associated with a fixed set of questions: (1) 2 marginal questions ($P(S)$ and $P(T)$); (2) 6 conditional questions (4 for $P(S|T)$ and 2 for $P(T|S)$); and (3) 1 joint question ($P(S, T)$). This results in 9 distributional questions per entity per task type. Across the three tasks, each entity therefore yields 27 evaluation questions. Table 5 summarizes the number of questions per task and the total number of evaluation instances across all tasks.

Entities	#Marginals	#Conditionals	#Joints	#Total (per Task)	#Total (3 Tasks)
20 (10 Movie + 10 Music)	40	120	20	180	540

Table 5: **Benchmark Statistics Summary.**

B Data Generation

This section describes the *Opinion Entity Selection* and *Comment Annotation* stages of the automated generation pipeline used to construct TEXT2DISTBENCH.

B.1 Opinion Entity Selection

We retrieve candidate opinion entities using domain-specific APIs, including IMDB and TMDb for movies and YouTube Music for songs. These APIs allow us to filter entities by release time and obtain relevant metadata. For each entity, we then use the YouTube Data API to retrieve associated videos, such as movie trailers or music videos. Because the YouTube Data API relies on text-based matching, some retrieved videos may be unrelated to the target entity. To address this issue, we employ an LLM (GPT-5.1-MINI) to filter out irrelevant or noisy video results. The prompt template used for this verification step is shown in Appendix C. Entities with comment volumes below a minimum threshold are discarded. For the remaining entities, we collect viewer comments and associated metadata.

B.2 Comment Annotation

Method We use three API-based LLMs (GPT-5.1-MINI, GEMINI-2.5-FLASH, and GROK-4-FAST) to annotate viewer comments. For each comment, models are prompted to assign both a *topic* and a *sentiment*. Tables 6 and 7 list the topic definitions provided to the models for the movie and music domains, respectively. All models are prompted using the template shown in Appendix C. When model predictions disagree, we apply a majority-voting scheme to determine the final label. If no majority is reached, or if a model indicates that a comment expresses multiple topics, the comment is discarded.

Annotation Quality To assess annotation quality, we conduct human verification on the test set described in Section 5, which consists of 20 entities with 50 comments each, totaling 1,000 comments. Human annotators are provided with the same annotation instructions as the models, including label definitions, entity metadata, and viewer comments. We recruit three annotators (two undergraduate students and one graduate student), and determine the final human label for each comment by majority vote. Annotation precision is then computed by comparing the automated labels against the human annotations.

Based on the evaluation, with $n=1,000$ and an observed precision of 0.95, the sampling distribution follows a Binomial distribution, which, according to the Central Limit Theorem, yields a 95% confidence interval of [0.936, 0.964] with a margin error of $\pm 1.4\%$. This narrow interval indicates that the estimated

precision is statistically stable, suggesting that our annotation pipeline produces results that are largely consistent with human judgments.

Attribute	Description
Actor	Performance, emotion, chemistry, casting quality.
Storyline	Plot, pacing, themes, character arcs, dialogue.
Visual	Cinematography, lighting, color, design, VFX, costumes.
Audio	Soundtrack, score, songs, sound effects, audio mix.

Table 6: Movie Comment Topics

Attribute	Description
Song	Melody, harmony, rhythm, structure, arrangement.
Singer	Tone, technique, emotion, appearance.
Lyrics	Themes, message, storytelling, rhymes.
Visual	Cinematography, lighting, concept, choreography, VFX.

Table 7: Music Comment Topics

C Data Generation Prompt Examples (Movie)

Movie Metadata

- Movie Title: {title}
- Release Date: {date}
- Cast: {casts}
- Summary: {summary}
- Synopsis: {synopsis}

Relevant Video Verification Prompt

You searched YouTube for: {query}
 Following is the retrieved video information:
 Title: {title}
 Description: {description}

Determine if this video matches the intended search query and satisfies the information need.

If yes, return True. If not, return False.

Comment Topic Annotation Prompt

You are analyzing public reactions to a movie by assigning each viewer comment to one or more attributes (multi-label).

Attributes (use EXACTLY these keys; prefer the single most dominant attribute unless the comment clearly discusses multiple):

- Actor : Comments about the actors' performances: delivery, emotion, chemistry, casting.
- Storyline : Comments about the movie plot, narrative, themes, pacing, character arcs, or dialogue.

- Visual : Comments about cinematography, animation, lighting, color, production design, costumes, or visual effects.
- Audio : Comments about the soundtrack, score, songs, sound effects, or audio mix.
- Other : Use only if none of the above clearly fit (off-topic, spam, unclear).

Movie Information:

```
{meta_data}
```

YouTube Viewer Comments (0-based indexing, e.g., 0, 1, 2, ...):

```
{comments}
```

Instructions:

- 1) MULTI-LABEL is allowed, but if uncertain choose the single most dominant attribute.
- 2) Use 0-based indices exactly as shown; do not invent indices.
- 3) If a comment does not clearly fit any attribute, include it under "Other".
- 4) Return ONLY the JSON object.
- 5) The JSON must be a single object whose keys are EXACTLY the attributes below and whose values are lists of integer indices.
- 6) Do not add, rename, or remove keys.

Output JSON:

```
{"Actor": [], "Storyline": [], "Visual": [], "Audio": [], "Other": []}
```

Comment Sentiment Annotation Prompt

Classify the sentiment expressed in YouTube comments toward the movie.

Movie Information (for information reference only):

```
{meta_data}
```

YouTube Viewer Comments (0-based indexing, e.g., 0, 1, 2, ...):

```
{comments}
```

Labels (choose exactly one per comment):

- support : praise / approval / positive attitude
- oppose : criticism / disapproval / negative attitude

Rules:

- 1) Focus on the overall tone or attitude of the comment toward the movie.
- 2) Use movie information only to resolve references (e.g., who/what "he", "it", or "this scene" refers to), not to guess sentiment.
- 3) Consider emojis, slang, irony/sarcasm (e.g., quotes, "/s", exaggeration, laugh reactions).
- 4) Use 0-based indices exactly as shown; do not invent or skip indices.
- 5) Each index must appear in EXACTLY ONE list (support OR oppose).

Output JSON:

```
{"support": [], "oppose": []}
```

D QA Template Examples (Movie)

System Template

You will be given information about a movie, followed by a collection of viewer comments.

Each comment reflects what a viewer thinks about the movie and focuses on a particular aspect while expressing a certain attitude.

When reading the comments, keep in mind two dimensions:

- Sentiment: whether the comment expresses a positive or negative attitude.
- Topic: which aspect of the movie the comment mainly talks about.

Sentiment categories:

- positive: expressing approval, enjoyment, or praise.
- negative: expressing criticism, dissatisfaction, or disappointment.

Topic categories:

- Actor: acting performance, casting, chemistry, emotional expression.
- Storyline: plot, narrative, pacing, themes, dialogue, character development.
- Visual: cinematography, animation, lighting, color, visual effects, production design.
- Audio: soundtrack, music, sound effects, or audio quality.

Read the following movie information and the viewer comments carefully.
Then answer the question based on this information.

Movie Information: {meta_data}

Viewer Comments: {comments}

P(S) Estimation

Question:

Read the viewer comments.

How do the expressed opinions break down in terms of overall attitude toward the movie?

Summarize how common each sentiment is among the comments.

Ensure the percentages sum to 100 and use integers only.

Output your answer in the following JSON format:

```
{ "positive": "<int>", "negative": "<int>" }
```

Answer:

P(T) Estimation

Question:

Read the viewer comments.

What aspects of the movie do viewers talk about?

Summarize how frequently each aspect appears in the comments.

Ensure the percentages sum to 100 and use integers only.

Output your answer in the following JSON format:

```
{"Actor": "<int>", "Storyline": "<int>", "Visual": "<int>", "Audio": "<int>"}
```

Answer:

$P(S,T)$ Estimation

Question:

Read the viewer comments.

Consider both which aspect is being discussed and whether the expressed opinion is positive or negative.

Summarize how these combinations appear in the comments.

Ensure the percentages sum to 100 and use integers only.

Output your answer in the following JSON format:

```
{
  "percentages": {
    "(Actor,positive)": "<int>",
    "(Actor,negative)": "<int>",
    "(Storyline,positive)": "<int>",
    "(Storyline,negative)": "<int>",
    "(Visual,positive)": "<int>",
    "(Visual,negative)": "<int>",
    "(Audio,positive)": "<int>",
    "(Audio,negative)": "<int>"
  }
}
```

Answer:

$P(S|T)$ Estimation

Question:

Read the viewer comments.

Focus only on the comments that talk about the {topic} aspect of the movie.

How are viewers' attitudes divided?

Ensure the percentages sum to 100 and use integers only.

Output your answer in the following JSON format:

```
{"positive": "<int>", "negative": "<int>" }
```

Answer:

$P(T|S)$ Estimation

Question:

Read the viewer comments.

Focus only on the comments that express a {sentiment_label} attitude toward the movie.

Which aspects of the movie do these comments discuss?

Ensure the percentages sum to 100 and use integers only.

Output your answer in the following JSON format:

```
{"Actor": "<int>", "Storyline": "<int>", "Visual": "<int>", "Audio": "<int>"}
```

Answer:

$P(S)$ Most Frequent

Question:

Read the viewer comments.

What overall attitude do most viewers express?

Positive or Negative.

Answer:

$P(T)$ Most Frequent

Question:

Read the viewer comments.

Which aspect of the movie is discussed most often?

Choose from: Actor, Storyline, Visual, or Audio.

Answer:

$P(S, T)$ Most Frequent

Question:

Read the viewer comments.

Considering both (1) which aspect is being talked about and (2) whether the attitude is positive or negative.

Which combination appears most often in the comments?

Choose one pair from:

(Actor,positive), (Actor,negative),

(Storyline,positive), (Storyline,negative),

(Visual,positive), (Visual,negative),

(Audio,positive), (Audio,negative).

Answer:

$P(S|T)$ Most Frequent

Question:

Read the viewer comments.

Focus only on the comments that talk about the {topic} aspect of the movie.

What attitude do viewers most commonly express?

Positive or negative.

Answer:

$P(T|S)$ Most Frequent

Question:

Read the viewer comments.

Focus only on the comments that express a {sentiment_label} attitude toward the movie.

Which aspect of the movie is mentioned most often?

Choose from: Actor, Storyline, Visual, or Audio.

Answer:

$P(S)$ Second Most Frequent

Question:

Read the viewer comments.

What overall attitude is the second most commonly expressed by viewers?

Positive or Negative

Answer:

$P(T)$ Second Most Frequent

Question:

Read the viewer comments.

Which aspect of the movie is discussed the second most often?

Choose from: Actor, Storyline, Visual, or Audio.

Answer:

$P(S, T)$ Second Most Frequent

Question:

Read the viewer comments.

Considering both (1) which aspect is being talked about and (2) whether the attitude is positive or negative.

Which combination appears the second most often in the comments?

Choose one pair from:

(Actor,positive), (Actor,negative),

(Storyline,positive), (Storyline,negative),

(Visual,positive), (Visual,negative),

(Audio,positive), (Audio,negative).

Answer:

$P(S|T)$ Second Most Frequent

Question:

Read the viewer comments.

Focus only on the comments that talk about the {topic} aspect of the movie.

What attitude is the second most commonly expressed?

Positive or Negative

Answer:

$P(T|S)$ Second Most Frequent

Question:

Read the viewer comments.

Focus only on the comments that express a {sentiment_label} attitude toward the movie.

Which aspect of the movie is mentioned the second most often?

Choose from: Actor, Storyline, Visual, or Audio.

Answer: