Platform	Train	Validation	Test
Metaculus	1,576	230	275
GJOpen	806	161	38
INFER	52	50	4
Polymarket	70	229	300
Manifold	1,258	170	297
All Platforms	**3,762**	**840**	**914**

(a) Dataset distribution

Model	Zero-shot	Scratchpad
GPT-4-1106-Preview	**0.208** (0.006)	**0.209** (0.006)
Llama-2-13B	0.226 (0.004)	0.268 (0.004)
Mistral-8x7B-Instruct	0.238 (0.009)	0.238 (0.005)
Claude-2.1	0.220 (0.006)	0.215 (0.007)
Gemini-Pro	0.243 (0.009)	0.230 (0.003)
Trimmed mean	0.208 (0.006)	0.224 (0.003)

(b) Baseline performance of pre-trained models

Table 2: **(a) Distribution of our train, validation, and test sets across all 5 forecasting platforms.** Importantly, every question in the test set is from June 1, 2023 or later, after the training cut-off of our base LMs. Meanwhile, all questions in the train and validation sets were resolved before June 1, 2023, ensuring no leakage from the tuning process. **(b) Baseline performance** of pre-trained models on the test set, with 1 standard error (SE) (see full results in Table 7). Random baseline: 0.250; human crowd: 0.149. The results underscore that models are not naturally good at forecasting.

mechanisms. The crowd prediction is a strong benchmark to compete with. For example, Metaculus (2023) shows that an ensemble of all forecasters consistently outperforms using just the top 5, 10, ..., 30 best forecasters (based on past scores). In this work, we compare our system performance to the crowd aggregates.

Raw data. We source forecasting questions from the 5 above-mentioned platforms. This yields a total of 48,754 questions and 7,174,607 user forecasts spanning from 2015 to 2024. The dataset includes 33,664 binary questions, 9,725 multiple-choice questions, 4,019 numerical questions, and 1,346 questions of other types. The questions cover a wide range of topics across the globe (Figure 10).

The raw dataset contains questions that are ill-defined, overly personal, or of niche interests. Furthermore, recent questions are highly unbalanced, with over 80% of questions since June 1, 2023 coming from Manifold and Polymarket.

Data curation. To address the above issues, we curate a subset by filtering ill-defined questions and removing questions that received few forecasts or trading volume on Manifold and Polymarket. We focus on predicting binary questions and split multiple-choice questions into binary ones.

To guard potential leakage from LMs' pre-training, we only include questions in the test set that appear after the knowledge cut-off for the models we use (June 1, 2024). All test set questions were opened after the date, and all train and validation questions were resolved before. Questions that span across the date are discarded.

This yields a set of 5,516 binary questions, including 3,762 for training, 840 for validation, and 914 for testing (Table 2a). See Table 12 for a sample data point and Appendix C for details about the curation process.

3.2 Evaluation

Retrieval schedule. We can simulate forecasting the future by leveraging the fact that models are only trained up to a cut-off date (Zou et al., 2022). To simulate a forecast for a question that has been resolved, we query a historical news corpus to retrieve articles between the question begin date and a specified *retrieval date* (Zou et al., 2022; Yan et al., 2024). The retrieval date can be viewed as the "simulated date" of the forecast, as we are mimicking the information the model would have had access to on that date.

To create a set of retrieval dates for each question, we use geometrically increasing time points between the open and close dates. We choose this schedule for two reasons: (1) questions are often most active shortly after they open, and (2) some questions have overly conservative close dates that are long after the question resolves. We use $n = 5$ retrieval dates per question; the kth retrieval date is calculated as

$$\text{retrieval\_date}_k = \text{date}_{\text{begin}} + (\text{date}_{\text{close}} - \text{date}_{\text{begin}} - 1)^{k/n}. \tag{1}$$

For questions that resolve before they close, we exclude all dates occurring after the question has been resolved. Under this geometric retrieval schedule, we retain 86% of retrieval dates on average across all questions (Figure 11b). The average question window in our corpus is approximately 70 days, and the average time until resolution is 42 days.

In our dataset, questions can get resolved long before their official close date. This occurs for questions like "Will $\langle event \rangle$ happen by $\langle date \rangle$", where resolving early indicates that the event did occur (see Table 1 for an example). It is tempting to choose retrieval dates with respect to the resolve date so that each question can receive the same number of retrieval dates, e.g. by retrieving at geometric intervals between the open and resolve date. However, this would leak information, since the retrieval date would now depend on the resolve date, which, as we explained, correlates with the resolution.

Metric. Our work focuses on binary questions and uses the Brier score as the performance metric, defined as $(f - o)^2$, where $f \in [0, 1]$ is the probabilistic forecast and $o \in \{0, 1\}$ is the outcome. The Brier score is a strictly proper scoring rule: assuming the true probability that $o = 1$ is p, the optimal strategy is to report $f = p$. This is desirable, since improper scoring rules would incentivize reporting distorted probabilities. As a baseline, an (unskilled) forecast of .5 attains a Brier score of .25.

To compute the final Brier score, we first average the Brier scores across retrieval dates for each question, then average across questions. We also report standard errors; however, note that the computation of standard errors assumes the data are i.i.d., while our data are in fact time-series, so this likely underestimates the true error. Finally, we also measure calibration with root mean square (RMS) calibration error.

3.3 Models

We evaluate 14 instruction-tuned LMs: GPT-3.5-Turbo, GPT-3.5-Turbo-1106 (Brown et al., 2020); GPT-4, GPT-4-1106-Preview (OpenAI, 2023); Llama-2-7B, Llama-2-13B, Llama-2-70B (Touvron et al., 2023); Mistral-7B-Instruct, Mistral-8x7B-Instruct (Jiang et al., 2024), Nous Hermes 2 Mixtral-8x7B-DPO, Yi-34B-Chat, Claude-2, Claude-2.1 (Anthropic, 2023), and Gemini-Pro (Gemini Team, 2023); see Section A.1 for details.

3.4 Models are not naturally good at forecasting

As a baseline, we evaluate all 14 LMs with no additional information retrieval. We use zero-shot prompts and scratchpad prompts (Nye et al., 2021). For each prompting strategy, we craft candidate prompts, pick the best prompt on the validation set, and report its Brier scores on the test set. The results are given in Table 2b, where we report the best model in each series; see Table 7 for full statistics. The prompt choices appear in Figure 5 and Figure 6 and further details are in Appendix B.

None of the models are naturally good at forecasting. Most models' scores are around or worse than random guessing (.25). Only the GPT-4 and Claude-2 series beat the unskilled baseline by a large margin ($> .02$). Moreover, while GPT-4-1106-Preview achieves the lowest Brier score of .208, it trails significantly behind the human crowd performance of .149.

4 Our System

As observed in Table 2b, all models perform poorly in the baseline setting. We intuit that models require detailed contexts and up-to-date information to make accurate forecasts. Our system addresses this issue via news retrieval and elicits better reasoning via optimized prompting strategies and fine-tuning.

4.1 Retrieval

Our retrieval system consists of 4 steps: search query generation, news retrieval, relevance filtering and re-ranking, and text summarization (Figure 1a).

First, we generate search queries that are used to invoke news APIs to retrieve historical articles. We initially implement a straightforward query expansion prompt (Figure 12a), instructing the model to create queries

Criteria	Brier Score ↓			% Accuracy ↑			% Data Retained ↑	
	Ours	Crowd	Aggregate	Ours	Crowd	Aggregate	Forecasts	Questions
All Questions	$.179_{.003}$	$.149_{.003}$	$\underline{\mathbf{.146}_{\mathbf{.002}}}$	$71.5_{.7}$	$77.0_{.7}$	$\underline{\mathbf{77.8}_{\mathbf{.6}}}$	100%	100%
Crowd Uncertain	$\mathbf{.238}_{\mathbf{.004}}$	$.240_{.003}$	$\underline{\mathbf{.233}_{\mathbf{.002}}}$	$58.1_{1.3}$	$58.3_{1.3}$	$\underline{\mathbf{60.2}_{\mathbf{1.2}}}$	51%	56%
Early Retrieval	$.186_{.003}$	$.162_{.004}$	$\underline{\mathbf{.159}_{\mathbf{.003}}}$	$70.0_{.9}$	$74.4_{.9}$	$\underline{\mathbf{75.0}_{\mathbf{.8}}}$	84%	100%
5+ Articles	$.175_{.003}$	$.142_{.003}$	$\underline{\mathbf{.140}_{\mathbf{.002}}}$	$72.3_{.8}$	$77.7_{.7}$	$\underline{\mathbf{78.7}_{\mathbf{.7}}}$	84%	94%
All Criteria	$\mathbf{.240}_{\mathbf{.005}}$	$.247_{.004}$	$\underline{\mathbf{.237}_{\mathbf{.003}}}$	$\underline{\mathbf{58.0}_{\mathbf{1.7}}}$	$54.2_{1.7}$	$\mathbf{56.6}_{\mathbf{1.7}}$	22%	43%

Table 3: **System performance** on the test set. "All Questions" shows the Brier score on the full test set. Other rows show selective evaluation when specified criteria are met, averaging over qualifying questions and retrieval dates. "Crowd Uncertain" refers to questions with crowd predictions between 0.3-0.7. "Early Retrieval" refers to the first 3 retrieval dates. "5+ Articles" refers to forecasting when at least 5 relevant articles are retrieved. Finally, "All Criteria" refers to forecasting when the 3 criteria are jointly met. Notably, in every setting the aggregate (average) of our system and crowd prediction is the best. Subscript numbers indicate 1 standard error. We bold entries that outperform the crowd aggregate, and underline the best entry in each category.

based on the question and its background. However, we find that this overlooks sub-considerations that often contribute to accurate forecasting. To achieve broader coverage, we prompt the model to decompose the forecasting question into sub-questions and use each to generate a search query (Min et al., 2019); see Figure 12b for the prompt. For instance, when forecasting election outcomes, the first approach searches directly for polling data, while the latter creates sub-questions that cover campaign finances, economic indicators, and geopolitical events. We combine both approaches for comprehensive coverage.

Next, the system retrieves articles from news APIs using the LM-generated search queries. We evaluate 5 APIs on the relevance of the articles retrieved and select NewsCatcher[1] and Google News (Section E.2).

Our initial retrieval provides wide coverage at the cost of obtaining some irrelevant articles. To ensure that they do not mislead the model at the reasoning step, we prompt GPT-3.5-Turbo to rate the relevancy of all articles (Figure 14) and filter out low-scoring ones. Since the procedure is costly in run-time and budget, we only present the article's title and first 250 words to the model in context. We validate that this approach achieves high recall and precision while saving 70% cost (see Section E.3 for alternative methods and results).

Since LMs are limited by their context window, we summarize the articles. In particular, we prompt GPT-3.5-Turbo to distill the most relevant details from each article with respect to the forecasting question (Figure 13). Finally, we present the top k article summaries to the LM, ordered by their relevancy. We choose the ranking criterion, article count k, and summarization prompt based on end-to-end Brier scores over the validation set; see Section 5.2 for the hyperparameter sweep procedure.

4.2 Reasoning

Prior work in forecasting has focused on eliciting predictions from models without requiring rationales (Zou et al., 2022; Yan et al., 2024). However, accurately predicting the future is a difficult task that often requires computation beyond a single forward pass. Having the model externalize its reasoning also allows us to understand the explanation for the forecast and improve it accordingly.

We use open-ended scratchpad to structure model's reasoning paths. Our prompt begins with posing the question, providing a description, and specifying resolution criteria and key dates, followed by the top k relevant summaries (Figure 16). To guide the model to reason about the forecasting question, the optimal scratchpad prompt (Figure 15), as identified in Section 5.2, also incorporates four additional components:

- First, to ensure that the model comprehends the question, we prompt it to rephrase the question. It is also instructed to expand the question with its own knowledge to provide further information. Intuitively, a more detailed and precise phrasing of the question elicits better responses (Deng et al., 2023).

[1] https://www.newscatcherapi.com/