Approaching Human-Level Forecasting with Language Models

Danny Halawi[*] dhalawi@berkeley.edu
UC Berkeley

Fred Zhang[*] z0@eecs.berkeley.edu
UC Berkeley

Chen Yueh-Han[*] john0922ucb@berkeley.edu
UC Berkeley

Jacob Steinhardt jsteinhardt@berkeley.edu
UC Berkeley

Abstract

Forecasting future events is important for policy and decision making. In this work, we study whether language models (LMs) can forecast at the level of competitive human forecasters. Towards this goal, we develop a retrieval-augmented LM system designed to automatically search for relevant information, generate forecasts, and aggregate predictions. To facilitate our study, we collect a large dataset of questions from competitive forecasting platforms. Under a test set published after the knowledge cut-offs of our LMs, we evaluate the end-to-end performance of our system against the aggregates of human forecasts. On average, the system nears the crowd aggregate of competitive forecasters, and in some settings surpasses it. Our work suggests that using LMs to forecast the future could provide accurate predictions at scale and help to inform institutional decision making.

1 Introduction

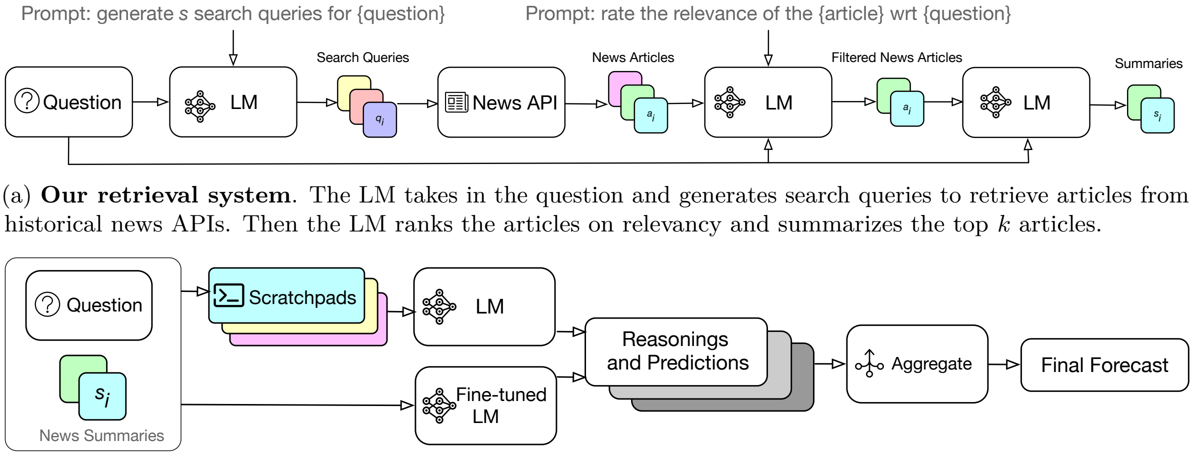
Forecasting events is important in the modern world. Governments rely on economic and geopolitical forecasts for decision-making. Companies hire and invest based on forecasts of market conditions (Armstrong, 2001). In 2020, epidemiological forecasts for COVID-19 prompted national lockdowns across the globe (Adam, 2020).

There are two main approaches to forecasting. *Statistical forecasting* primarily uses tools from time-series modeling. This methodology typically excels when data are abundant and under minimal distributional shift. By contrast, in *judgmental forecasting*, human forecasters assign probabilities to future events based on their own judgments, making use of historical data, domain knowledge, Fermi estimates, and intuition. They draw information from diverse sources and reason based on detailed contexts of the task. This enables accurate forecasts even with scarce past observations or under significant distributional shift (Tetlock and Gardner, 2015). We will refer to judgmental forecasting simply as "forecasting".

Since forecasting relies on human effort and expertise, it can be expensive, delayed, or applicable only in specific domains. Moreover, most human forecasts contain little or no explanatory reasoning. These limitations motivate using language models (LMs) to automate forecasting (Hendrycks et al., 2021). Because they can parse and produce texts rapidly, LMs can provide cheap and timely forecasts. Because they are pre-trained on web-scale data, they are endowed with massive, cross-domain knowledge. And because we can elicit their reasonings through prompts, we can examine them to (partially) understand the final forecast.

In this work, we build a LM pipeline for automated forecasting, with a focus on predicting binary outcomes. Our system implements and automates three key components in the traditional forecasting process: (1) retrieval, which gathers relevant information from news sources; (2) reasoning, which weighs available data

[*] Joint authorship.



(a) **Our retrieval system**. The LM takes in the question and generates search queries to retrieve articles from historical news APIs. Then the LM ranks the articles on relevancy and summarizes the top k articles.

(b) **Our reasoning system**. The system takes in the question and summarized articles and prompts LMs to generate forecasts. The forecasts are then aggregated into a final forecast using the trimmed mean.

Figure 1: **Overview of our retrieval and reasoning systems**. Our retrieval system retrieves summarized new articles and feeds them into the reasoning system, which prompts LMs for reasonings and predictions that are aggregated into a final forecast.

and makes a forecast; and (3) aggregation, which ensembles individual forecasts into an aggregated prediction. Each step makes use of an LM or a collection of LMs (either prompted or fine-tuned) (Figure 1).

To optimize and evaluate our system, we collect a large dataset of forecasting questions from 5 competitive forecasting platforms. The test set consists only of (binary) questions published after June 1st, 2023. Since this is after the knowledge cut-off date of our models, this prevents leakage from pre-training. The train set contains questions before June 1st, 2023, which we use for hyperparameter search and fine-tuning our system.

We use a self-supervised approach to fine-tune a LM to make accurate predictions and explanatory reasonings. We first prompt a base LM with various scratchpads to elicit forecasts to questions in our training set. We then fine-tune a new LM on the outputs that outperformed the crowd, which teaches the model what reasoning method to apply in a given context and improves forecasting performance. For hyperparameter search, we identify system configurations, including retrieval and LM prompting strategies, that lead to the best end-to-end performance.

Our optimized system approaches the performance of aggregated human forecasts over the test set, as measured by Brier score, a standard metric in forecasting. To our knowledge, this is the first automated system with forecasting capability that nears the human crowd level, which is generally stronger than individual human forecasters (Section 3.1). We also consider a selective setting where our system uses heuristics, based on the LM's strengths, to decide whether to submit a forecast for a given question and date. In this setting, our system outperforms the human crowd.

To summarize our main contributions:

1. We curate the largest, most recent dataset of real-world forecasting questions to date, for evaluating and optimizing automated forecasting systems.
2. We build a retrieval-augmented LM system that significantly improves upon the baseline and approaches the human crowd performance on competitive forecasting platforms.
3. We propose and apply a self-supervised fine-tuning method to improve LM's capability in reasoning about forecasting tasks.

2 Related Work

Event forecasting. Machine learning systems that make accurate, automated forecasts can help inform human decision-making (Hendrycks et al., 2021). Jin et al. (2021) provided ForecastQA, the first dataset

Field	**Content**
Question	Will Starship achieve liftoff before Monday, May 1st, 2023?
Background	On April 14th, SpaceX received a launch license for its Starship spacecraft. A launch scheduled for April 17th was scrubbed due to a frozen valve. SpaceX CEO Elon Musk tweeted: "Learned a lot today, now offloading propellant, retrying in a few days . . . "
Resolution Criteria	This question resolves Yes if Starship leaves the launchpad intact and under its own power before 11:59pm ET on Sunday, April 30th.
Key Dates	Begin Date: 2023-04-17 | Close Date: 2023-04-30 | Resolve Date: 2023-04-20

Table 1: **A sample question** with its background, resolution criteria, and key dates. The question resolved early (with a final resolution of Yes). See Table 12 for the complete sample point.

for this task, which contains questions created by crowdworkers based on events from news articles. Zou et al. (2022) introduced Autocast, a benchmark dataset compiled from forecasting competition questions up to 2022. In a competition with a large prize pool, no machine learning system was able to approach the performance of human forecasters on Autocast (Zou et al., 2022). The knowledge cut-offs of LMs have moved past 2022, necessitating more recent data. In this work, we source questions in 2023–2024, enabling us to apply recent LMs.

Yan et al. (2024) built a retrieval system that led to improved accuracy on Autocast. They trained a Fusion-in-Decoder model to directly predict the final (binary) resolution (Izacard and Grave, 2021) and reported accuracy, whereas we elicit both explanatory reasonings and probability forecasts from LMs and measure performance with the standard Brier score metric.

Schoenegger and Park (2023); Abolghasemi et al. (2023) evaluated GPT-4 and other LLMs on forecasting tournaments and found that they underperform the human crowd. This observation is in line with ours in Section 3.4. Unlike us, they make little or no efforts to improve these LMs on forecasting.

Finally, there has been recent work on using transformer models or LMs for statistical time-series forecasting (Nie et al., 2023; Gruver et al., 2023; Dooley et al., 2023; Rasul et al., 2023; Jin et al., 2024; Das et al., 2024; Woo et al., 2024), but this is distinct from our focus on judgmental forecasting.

Information retrieval (IR). IR can improve question-answering capabilities of LMs (Lewis et al., 2020; Shuster et al., 2021; Nakano et al., 2021). In event forecasting, access to diverse, up-to-date information is crucial (Tetlock and Gardner, 2015). Thus, a key component of our system is an IR architecture that furnishes the reasoning model with news articles, using LMs for query expansion, relevance ranking and summarization. Beyond our setting, using LMs for IR is an active research topic (Zhu et al., 2024).

Calibration. Calibration is important for accurate forecasting (Tetlock and Gardner, 2015). Hence, on competitive forecasting tournaments, forecasters are evaluated by proper scoring rules, such as Brier score (Brier, 1950), which incentivize calibration (Gneiting and Raftery, 2007). There is a vast literature on calibration in deep learning; see Gawlikowski et al. (2021); Wang (2023) for surveys.

3 Preliminaries: Data, Models and Baseline

3.1 Dataset

Data format. Forecasting platforms such as Metaculus, Good Judgment Open, INFER, Polymarket, and Manifold invite participants to predict future events by assigning probabilities to outcomes of a question. Each question consists of a *background description*, *resolution criterion*, and 3 timestamps: a *begin date* when the question was published, a *close date* when no further forecasts can be submitted, and (eventually) a *resolve date* when the outcome is determined. A forecast can be submitted between the begin date and min(resolve date, close date). See Table 1 for an example question with these main fields.

Crowd prediction. On any given question, as individual forecasts are submitted, forecasting platforms continuously aggregate them into a crowd prediction; see Section A.3 for details about the aggregation