How MASX AI Resolves Geopolitical Forecasts Without a Human Editorial Team

https://forecast.masxai.com

The Resolution Problem

Every forecasting system faces the same question eventually.
Who checks whether the predictions were right?
Generating a forecast is half the job. The other half is resolution, determining whether the predicted event actually happened once the deadline passes. **When you produce 30+ forecasts per day across 195 countries, manual verification is not an option.** Without a resolution layer, every forecast just sits there forever marked open.

What Existing Forecasting Systems Do

Metaculus uses human editorial teams. Every question is written by a person, every resolution is judged by admins, and if the outcome is ambiguous they annul it. The Good Judgment Project, the IARPA funded program that put superforecasters on the map, uses predefined resolution committees with criteria baked in before any forecasts are made. Polymarket uses a decentralised oracle network called UMA where token holders vote on outcomes. Kalshi, which is CFTC regulated, has internal teams resolving against contract terms using named source agencies. Manifold Markets lets the question creator resolve their own markets, with community override if they disappear.
All of these require humans somewhere in the loop. That works when you have editorial staff, a research budget, or a token economy funding the verification layer. MASX AI has none of those. It is a solo built autonomous system. The resolution problem had to be solved with engineering, not headcount.

Research That Shaped the Design

Two pieces of recent research shaped the approach directly.
Bosse et al. [2026] built a system that generated 1,499 real world forecasting questions using LLM agents, then resolved them automatically using ReAct web search agents. Their paper reports resolution accuracy of around 95% with no human in the loop. The key insight was what they call the generation verification asymmetry. **It is easier to verify whether an event happened than to generate a good question about whether it will happen.** Their resolver uses multi turn search where the agent reformulates queries when initial results come back thin, rather than running a single fixed query and hoping for the best.

ForecastBench took a different angle. They auto generate questions from structured data sources like FRED, ACLED, and Yahoo Finance using templates, then assign fixed system horizons at 7 days, 30 days, 90 days, 1 year, and 10 years. A nightly pipeline pulls ground truth directly from the source APIs. The critical insight from ForecastBench is that **dates should be a system concern, not an LLM concern.** The model does not pick when things happen. The system tells it when to evaluate.

MASX AI sits between these two. The questions are about emerging geopolitical hotspots, not templated data series, so the approach is closer to Bosse et al. But the temporal grounding comes straight from ForecastBench.

MASX AI Design Decisions

The first and probably most important decision was removing date hallucination entirely. The original MASX AI design let the LLM freely choose resolution dates. That is the single most unreliable output you can ask a language model for. LLMs have no temporal grounding. They do not have access to summit calendars, reporting cycles, or legislative schedules. I watched the model confidently tie a resolution date to an EU Foreign Affairs Council meeting that did not exist. Following the Kalshi and Polymarket model where the platform sets dates rather than the forecasters, I moved dates into code. **Every hotspot gets three deterministic deadlines, 14 days, 30 days, and 90 days.** The LLM generates one question per window with the deadline passed in as fixed context. The date is not part of the LLM output schema at all.

Not every horizon fits every hotspot though. A blind 14 days from today might not coincide with any resolvable event. Following the IARPA principle that horizons should be useful rather than just short, the LLM can signal that no valid question exists for a given band. About 40 to 60 percent of hotspots get skipped at the 14 day mark, under 5 percent at 90 days. Fewer questions enter the pipeline, but every question that proceeds is genuinely resolvable.

The second decision was enforcing resolution criteria at question creation time. This comes from Metaculus, where admins apply what they call the Clairvoyance Test. If a clairvoyant who could see the future could not determine the answer based solely on the question text and criteria, the question is badly written. MASX AI enforces this through a structured template. **The criteria must name a specific observable event, a named authoritative source for verification, and an explicit edge case statement covering partial or ambiguous outcomes.** Vague terms like significant escalation or major developments are banned from the criteria outright.

The third decision was how to validate question quality. Bosse et al. use separate LLM verifier agents that check resolvability and ambiguity after generation. That is LLM checking LLM, which introduces a circular problem. The verifier has no ground truth about whether the criteria reference real sources or hallucinated ones. **I replaced it with a programmatic validator. Pure Python, zero API calls.** It checks criteria length, scans for vague terms from a growing rejection list, verifies that a concrete source of truth is named, and deduplicates against recent questions. Based on early testing, the catch rate sits somewhere around 60 percent. Not as high as an LLM verifier might achieve, but it is deterministic, costs nothing, and improves as real failure patterns feed back into the rejection rules from resolved forecasts.

Resolution Logic

When a forecast deadline passes, the resolution engine searches the web to determine what actually happened. The resolver uses multi-turn agentic search following Bosse et al. It builds multiple queries from the forecast event text, resolution criteria, and key drivers, then reformulates when initial results are thin. A forecast written 3 months ago rarely matches the vocabulary journalists use when the event lands. The question says EU Council energy sanctions April 2026. The headline says Brussels imposes punitive tariffs on Gazprom. The multi turn approach handles that vocabulary gap.

The critical design choice in resolution logic was rejecting LLM confidence floats. The obvious approach is having the resolver output a confidence score and auto resolving above some threshold, say 0.85. LLMs are poorly calibrated on self assessed confidence. When asked how confident are you, the model outputs 0.9 as a task completion signal, not as a statement about the world. Following the structured evidence approach from Bosse et al., the resolver instead reports countable metrics. How many distinct authoritative sources corroborate the outcome. Whether any source explicitly contradicts it. **Auto resolution fires only when corroboration reaches 2 or more and no contradictory evidence is found.** Below that, the forecast goes ambiguous.

The post mortem is structured data, not a freestyle essay. An LLM writing a narrative assessment of its own predictions will be generous, a tendency that shows up often in models trained on RLHF where agreeable outputs get rewarded. So the resolver reports which key drivers from the original forecast appeared in search evidence, which did not, whether any disconfirming evidence the forecast identified actually materialised, and what real world signals emerged that the forecast missed entirely. The Brier score, computed as the squared difference between predicted probability and binary outcome, is the ground truth. The structured comparison provides context. It does not replace math.

Ambiguity Handling

Not every forecast resolves cleanly to yes or no. A question about whether BMKG would revise an earthquake magnitude to 8.0 or higher came back with no evidence either way. A forecast about the Brent crude oil price reaching $150 per barrel found plenty of analysis but no actual price confirmation. A question about whether Viktor Orban would mention an assassination plot on state television returned zero relevant search results.

In rigorous forecasting practice, Metaculus annuls ambiguous questions and excludes them from scoring entirely. The Good Judgment Project voids them. **MASX AI does the same. Ambiguous forecasts get a null Brier score and are excluded from all aggregate metrics.** They are not scored at 0.5. This matters because scoring ambiguous outcomes at 0.5 artificially compresses calibration variance and makes aggregate performance numbers look better than they are. If the evidence was insufficient to call the outcome, the intellectually honest response is to void the forecast, not to assign a coin flip and pretend that it is a measurement.

Of the 33 forecasts resolved in the first production run, 24 came back ambiguous. That is a 73% ambiguity rate, which sounds high but reflects the reality that many of these forecasts were

backfilled with retroactive criteria and cover events where English language web coverage is sparse or the question was too specific for the search layer to find a definitive answer.

First Results

33 forecasts resolved. 9 auto-resolved with real Brier scores. 24 correctly voided as ambiguous. Zero failures. Search health at 100%.

The 9 auto resolved forecasts covered real geopolitical events. The PKK claimed responsibility for the Diyarbakir bombing, corroborated by 8 sources including Reuters, Al Jazeera, and TRT World. Iran used cluster munitions in strikes on Israel, confirmed by Human Rights Watch and 4 corroborating sources. Hezbollah publicly condemned the Lebanese government expulsion of the Iranian ambassador, backed by 12 sources. The IMD issued a red alert for Delhi dust storms, confirmed by 3 sources. The Italian Justice Minister did not resign after the referendum, confirmed by 6 sources. PHIVOLCS issued no tsunami threat statements for the Philippines, backed by 12 sources. The port of Bitung did not close for more than 48 hours after the earthquake, confirmed by 2 sources.

The Brier scores ranged from 0.0064 [nearly perfect on the Philippines tsunami forecast, where the system assigned 92% probability and the event happened] to 0.7569 [a poor call on Iran cluster munitions, where the system assigned only 18% probability but the event occurred]. **Average Brier across all 9 was 0.3013.** The distribution tells you more than the average. Four forecasts scored below 0.18, meaning the system was well calibrated on those. Two scored above 0.65, meaning the system got those materially wrong. **The honest read is that the system is capable of excellent calls but also capable of significant misses, which at least shows the system is not avoiding hard questions.** A system that only scores 0.05 across the board is either cherry picking easy calls or not forecasting anything meaningful.

Limitations

The resolution engine is only as good as the criteria it judges against. Of the 901 forecasts in the system, 890 were created before resolution criteria existed. Without criteria, the resolver cannot evaluate them. A backfill process is generating criteria retroactively, but backfilled criteria are inherently weaker than criteria written at forecast creation time when the full context is fresh.

There is an English language bias in the search layer. Some geopolitical events in the pipeline involve regions where English language coverage is sparse or delayed. The resolver may lack evidence not because the event did not happen, but because no English language source reported on it within the search window. The 73% ambiguity rate on the first batch partially reflects this.

The structured post mortem reduces but does not eliminate LLM self assessment bias. The resolver still decides which drivers were confirmed by evidence and which signals were missed. It is making judgment calls within the structure. A truly independent verification would require a different model family or an external data source as ground truth, which is the direction ForecastBench takes with their direct API pulls from FRED and ACLED.

33 resolved forecasts is not yet a meaningful sample. The numbers show promise but 9 scored data points cannot tell you whether the system is well calibrated. Anyone claiming calibration from this volume would be overfitting to noise.

What Comes Next

MASX AI now has an accountable resolution layer. Every resolved forecast preserves the original probability, the search evidence, the binary outcome, and the Brier error score. That record is permanent and public. There is no way to quietly revise a bad call after the fact.
890 legacy forecasts are queued for criteria backfill and will run through the sweep as soon as criteria generation completes. That batch will be the first real stress test of resolution accuracy at scale.
Based on forecasting literature, calibration curves generally need a few hundred resolved outcomes before they start to mean anything. At current volume that is probably 4 to 6 months out. Until then, per forecast Brier scores and simple hit rates are the honest metrics.
The first batch already shows a pattern worth watching. Natural disaster and military forecasts resolved cleanly with strong corroboration. Political forecasts about specific statements or endorsements mostly went ambiguous because web search cannot prove a negative with certainty. As the dataset grows, domain level Brier breakdowns will make that distinction precise, showing exactly where the system earns its confidence and where it does not.
Most AI forecasting products stop at the prediction.
MASX AI now scores its own homework.

Author: Ateet Bahamani
ab@masxai.com
https://ateet.masxai.com
https://www.linkedin.com/in/ateet-vatan-bahmani

References

- Bosse et al. [2026]. System for generating and resolving 1,499 real world forecasting questions using LLM agents and ReAct web search agents.
- ForecastBench. Project detailing auto-generation of questions from structured data sources like FRED, ACLED, and Yahoo Finance.