AI-Enabled Semantic Matching for Cross-Platform Prediction Market Arbitrage

Allen Sirolly[1]

[1]Decision, Risk, and Operations Division, Columbia Business School

Abstract

Identifying arbitrage opportunities in prediction markets is challenging because markets are represented by questions in natural language (e.g., "What will the Fed decide in its December 2025 meeting?"). To detect arbitrage opportunities at scale, one must consider the *semantic* similarity between market descriptions, while taking into account unobserved contextual information. This paper uses an LLM agent to detect sets of markets on Kalshi and Polymarket, two popular prediction market platforms, which are *logically equivalent* and therefore form a basis for a potential cross-platform arbitrage trade.

Introduction

A *prediction market* allows participants (or *traders*) to bet on the outcomes of future events. Traders buy and sell contingent contracts, labeled YES and NO, which pay $1 if the referenced occurs does or does not occur, and $0 otherwise. Contracts on the two most popular exchanges at present, *Kalshi* and *Polymarket*, are traded using a continuous double auction which is implemented by a central limit order book (CLOB) for each market. In theory, this mechanism aggregates the beliefs of traders, such that the market price may be interpreted as the probability of the event in question.

The simplicity of the single-market order book makes trading easy for participants, but means that price consistency is not enforced between markets whose outcomes are logically dependent. This gives rise to opportunities for *arbitrage*, in which a trader purchases contracts at a total price less than a guaranteed payout amount, ensuring a risk-free profit.

In securities markets, securities traded on different exchanges share a common ticker symbol, making same-ticker arbitrage opportunities straightforward to identify. In contrast, prediction market exchanges define markets using YES/NO questions expressed in natural language, and there is no standardized method to compare them. This makes cross-platform arbitrages non-trivial to identify comprehensively.

Consider the following example. On Kalshi, there is an event titled "Best AI at end of 2025?"; on Polymarket, there is an event titled "Which company has best AI model end of 2025?" Figure 1 shows the sum of the most recent traded YES price for "Google/Gemini" on Kalshi and the most recent traded NO price for "Google" on Polymarket. On June 25, 2025, one could have purchased YES contracts for the "Google/Gemini" outcome on Kalshi at $0.45 per contract, and NO contracts for the "Gemini" outcome on Polymarket at $0.40 per contract. Therefore, buying "both sides" of the outcome would have cost $0.85 for a guaranteed payout of $1 per contract at the end of 2025, yielding a 17% return (35% annualized), ignoring fees.[1] In fact, this return could have been achieved as soon as July 8, 2025, when the sum of the prices returned to $1.

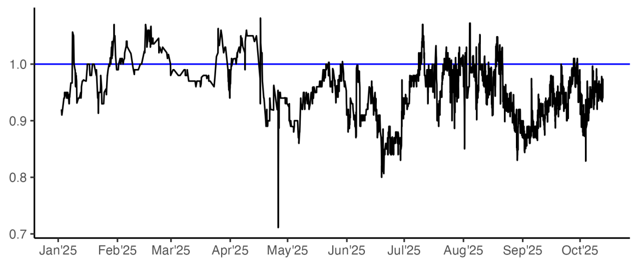

Figure 1: The sum of the most recent traded YES price for "Google/Gemini" in the Kalshi market "Best AI at year end?" and the most recent traded NO price in the Polymarket market "Will Google have the top AI model on December 31?" Deviations from $1 indicate an arbitrage opportunity (ignoring fees).

The identification of arbitrage opportunities such as this may be aided by large-language models, which excel at making inferences based on unstructured text.

Consider a more complex example: On both Kalshi and Polymarket, there are events which allow traders to bet on the score (0–100) that a new movie release will receive on

[1]For most markets, Kalshi charges a taker fee equal to $0.07 \times p \times (1 - p)$ per contract at price p, i.e., at most $0.028 per contract; as of this writing, Polymarket does not charge fees. Furthermore, both platforms pay competitive interest rates on open positions; therefore one does not need to explicitly consider the opportunity cost of arbitrage (relative to depositing money in an interest-bearing account).

the review aggregation websites Rotten Tomatoes and Metacritic. The scores are partitioned differently on the two platforms, as illustrated in Table 1. On examination, there are several sets of markets which are equivalent and may therefore be the basis for an arbitrage trade. For instance, betting YES on “Above 97” and NO on “Above 97” on Kalshi is equivalent to betting YES on “96-97” on Polymarket; thus, betting YES, NO, and NO on the three markets, repsectively (or NO, YES, and YES) yields a guaranteed payout of $1, but the total price may be less than $1 minus fees.

Kalshi	Polymarket
Above 70	
Above 75	
Above 80	
Above 85	
Above 87	<90
Above 90	90-91
Above 92	92-93
Above 95	94-95
Above 97	96-97
	98+

Table 1: Kalshi and Polymarket YES/NO markets for the event “‘The Long Walk” Rotten Tomatoes score?’

Related Literature

Several existing studies have examined the extent of arbitrage in prediction markets. Most similar to this study is Saguillo et al. (2025), which examines arbitrages within (i) *negative risk* events on Polymarket—events with multiple markets which are mutually exclusive and exhaustive, i.e., with exactly one market resolving to YES; as well as (ii) pairs of events which belong to the same category and resolve on the same day—suggesting that they may be related—but which are not explicitly grouped as negative risk markets are. For the latter exercise, the authors use an LLM prompt to detect *semantic dependency* between pairs of events.

Rothschild and Pennock (2014) document the existence of cross-platform arbitrage opportunities for a subset of markets on Betfair and Intrade related to the 2012 U.S. presidential election. For example, they find that the price of an “Obama to win” contract was higher on Betfair than on Intrade, and that this difference persisted throughout 2012, peaking at $0.23 on Election Day. They conjecture that the large price difference existed because the potential arbitrage profits were not large enough to compel sophisticated investors to close the gap. Furthermore, they use a trading bot to demonstrate that the potential arbitrage profit is “an order of magnitude larger” than what one would naively estimate from the liquidity (i.e., limit orders) observable in order book snapshots. This is due to the “shadow order book”, referring to additional liquidity that may be placed by (likely automated) traders after trades are executed, and which may be purchased by an arbitrageur.

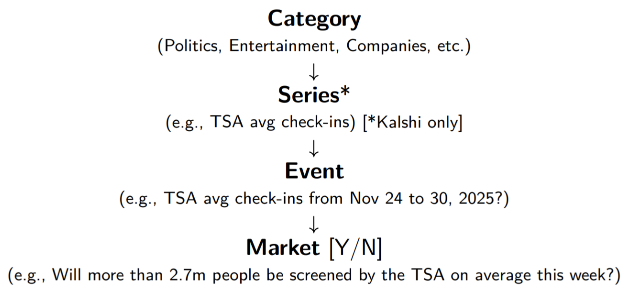
Category
(Politics, Entertainment, Companies, etc.)
↓
Series*
(e.g., TSA avg check-ins) [*Kalshi only]
↓
Event
(e.g., TSA avg check-ins from Nov 24 to 30, 2025?)
↓
Market [Y/N]
(e.g., Will more than 2.7m people be screened by the TSA on average this week?)

Figure 2: The taxonomy of markets on Kalshi and Polymarket. Note that categories differ between platforms, and there is no concept of a series on Polymarket.

Finally, Ng et al. (2025) document persistent price differences between 2024 U.S. presidential election contracts on Kalshi, Polymarket, and Intrade, and estimate the magnitude of cross-platform arbitrage profits to be on the order of tens of thousands of dollars.

Data

I obtain all market-level data using the Kalshi API and Polymarket API. The markets under consideration cover the entire set of markets traded on the respective exchanges until October 2025.

The markets on each platform are organized (loosely) according to a hierarchy. On Kalshi, a *series* defines a recurring *event*. Each *event* nests one or more markets; each *market* corresponds to a binary (YES/NO) question. For example, traders may bet on the volume of passengers screened by the TSA, denoted by series titled “TSA avg check-ins”; a representative event in this series is “TSA avg check-ins from Nov 24 to 30, 2025?”, and this event has multiple outcomes (markets) titled, e.g., “Will more than 2.7m people be screened by the TSA on average this week?” Series are additionally classified into *categories* such as Sports, Entertainment, and Companies. This taxonomy is illustrated in Figure 2.

On Polymarket, an *event* nests one or more *markets*, similar to Kalshi, but there is no series designation. I construct series for Polymarket by stripping away date/time information from event titles using a pre-defined set of regular expressions and grouping events. For example, “Highest temperature in NYC on Jan 22 [2025]?” and 265 other events are grouped under the series title “Highest temperature in NYC?” following this procedure. Moreover, Polymarket’s category labels are not the same as Kalshi’s category labels, so I classify the constructed series into one of Kalshi’s 16 categories using OpenAI’s `gpt-4o-mini`. (See the Appendix for prompt details.)

To make the subsequent analysis more manageable, I exclude series belonging to the Sports category. Table 2 shows the total count of categories, series, events, and markets for each platform following this exclusion.

	Categories	Series	Events	Markets
Kalshi	15	4,056	47,568	349,464
Polymarket	15	9,177	32,665	73,533

Table 2: The total count of Kalshi and Polymarket categories, series, events, and markets.

Methods

Below is a formal definition of arbitrage in the context of prediction markets:

Definition 1. *Consider two sets of events $\mathcal{A}$ and $\mathcal{B}$. An* arbitrage *exists if*

$$\bigcup_{A_i \in \mathcal{A}} A_i = \bigcup_{B_j \in \mathcal{B}} B_j$$

and

$$\mathit{cost}(\mathcal{A}) \neq \mathit{cost}(\mathcal{B}),$$

that is, blah. The cost of an event set is the sum of the contract prices for the outcomes of the constituent markets, i.e.,

$$\mathit{cost}(\mathcal{A}) = \sum_{A_i \in \mathcal{A}} \mathit{ContractPrice}(A_i)$$

In practice, given two equivalent sets of markets $\mathcal{A}$ and $\mathcal{B}$, an arbitrageur should take a *long* position in one set and a *short* position in the second set to guarantee a payout of \$1. Then, an arbitrage exists if either

$$\mathit{cost}(\mathcal{A}) + \mathit{cost}(\mathcal{B}^c) < 1 - \mathit{fees}$$

or

$$\mathit{cost}(\mathcal{A}^c) + \mathit{cost}(\mathcal{B}) < 1 - \mathit{fees}$$

where $\mathcal{A}^c$ represents the complementary set of outcomes for all $A_i \in \mathcal{A}$.

Strategy

Comparing all markets (or events) pairwise is computationally infeasible, as there are 349,464 $\times$ 73,533 $\approx 2.57 \times 10^{10}$ market pairs (and 47,568 $\times$ 32,665 $\approx$ 1.55×10^9 event pairs). To identify candidate events for an AI agent to evaluate, I first match series within categories using the similarities of vector embeddings for the series labels. There are approximately 6.89×10^6 within-category series pairs.

For each candidate series pair, I compute the cosine similarity of the vector embeddings for the series titles. I use OpenAI's `text-embedding-3-small` for the embeddings, which provides vectors with dimension 1,536.[2] To match Kalshi and Polymarket series referencing the same topic, I run a deferred-acceptance algorithm (using R package matchingR) where the cosine similarity θ_{ij} between Kalshi series i's embedding and Polymarket series j's embedding is i's "preference" for j (and vice versa).[3] I then filter for matches with $\theta_{im(i)} \geq 0.8$ which, based on inspection, seemed to be a cutoff which effectively separated correct from incorrect matches.

To simplify the subsequent analysis, I further restricted the series matching to series with exactly one underlying event (i.e., to non-recurring series). After this restriction, the series-level matching is effectively an event-level matching. To further verify that the matched events are true matches, I used an LLM agent to evaluate whether the two events are indeed equivalent. (See the Appendix for prompt details.)

Finally, I use an LLM agent to identify equivalent sets of markets within each (verified) matched event (subsequently referred to as the "Discovery" step). The system prompt is show in the Appendix. I use OpenAI's `gpt-4o` and `gpt-4.1` models, and specify structured output requirements using a Pydantic data validation model.[4]

Results

The procedure described in the preceding section yields 626 matched series. The count of matched series by category is shown in Table 3. (Surprisingly, there are few matches relative to the number of series per category, suggesting that there may room for improvement, either in the consistent assignment of series to categories, the embedding model, or in the procedure used to match series within categories.)

Category	Kalshi	Polymarket	Matches
Climate and Weather	102	137	2
Companies	160	473	19
Crypto	110	1,300	29
Economics	183	141	5
Education	2	4	0
Elections	327	873	127
Entertainment	1,069	2,119	161
Financials	87	354	3
Health	69	194	4
Mentions	116	282	16
Politics	1,455	2,762	253
Science and Technology	125	141	14
Social	62	0	0
Transportation	23	53	0
World	122	344	3

Table 3: Count of Kalshi and Polymarket series, and the number of matches (with cosine similarity $\theta_{im(i)} \geq 0.8$) by category.

Next, I evaluate the accuracy of the LLM agent in the Discovery step by comparing its output to a set of 300 randomly sampled and hand-labeled instances. Overall, the `gpt-4o` agent achieves 59.7% accuracy; the `gpt-4.1` agent achieves 70% accuracy. However, performance varies considerably by the type of event under consideration. Figure 3 shows the accuracy when split by event type: for events

[2]See https://platform.openai.com/docs/guides/embeddings for details.

[3]This produces a 1-1 matching m which is stable, in the sense that i cannot be matched to another $j' \neq m(i)$ with $\theta_{ij'} > \theta_{ij}$, since j' must already have a match $m(j')$ with $\theta_{m(j')j'} > \theta_{ij'}$. Since the number of Polymarket series exceeds the number of Kalshi series, there are Polymarket series which are not matched to any Kalshi series.

[4]See https://platform.openai.com/docs/guides/structured-outputs.

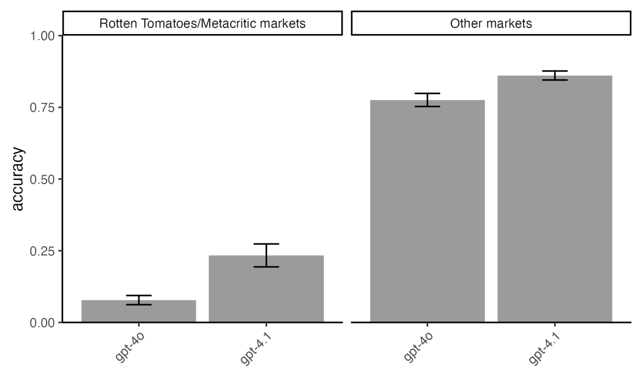

Figure 3: The fraction of 300 randomly-sampled Kalshi markets for which the AI agent correctly returned a subset of logically-equivalent Polymarket markets, split by (a) OpenAI ChatGPT model version and (b) markets which involved predicting a Rotten Tomatoes or Metacritic movie score, or not.

about the Rotten Tomatoes or Metacritic scores of movie releases (77 out of the 300 sampled events; example in Table 1), the agents achieve only 7.8% (`gpt-4o`) and 23.4% (`gpt-4.1`) accuracy. In all other markets, they achieve 77.6% and 86.1% accuracy, respectively. At least for these LLM models, performance seems to be adversely affected when the number of candidate markets is large, or when detecting logical equivalences involves numeric reasoning.

Conclusion

This study leverages agentic AI to identify potential cross-platform arbitrages between the prediction markets Kalshi and Polymarket. The approach explicated here may be extended to identify arbitrage opportunities across other platforms, or between events listed on the same platform.

The method described here merely provides a necessary input to a cross-platform arbitrage strategy; it does not quantify potential arbitrage profits or detail the implementation of the strategy itself. To do this, one would need to obtain the best bid and ask prices for each set of markets for which a potential arbitrage exists. One caveat is that the identified arbitrage-able markets may have low liquidity, limiting potential dollar profits.

Further improvements to the method described here are possible. For instance, identifying matches may require additional contextual information beyond that included in the relatively terse textual descriptions for each series or event. Thus, performance may improve if the AI agents are equipped with web search capabilities to discover this unobserved context. Furthermore, sets of markets which appear equivalent based on their textual descriptions may nevertheless differ subtly according to resolution criteria (the specific rules by which markets' outcomes are determined). A robust arbitrage strategy should also take these criteria into account to avoid unexpected market risk.

Appendix I. Implementation Details

Polymarket series classification system prompt:

> Please classify the following question as belonging to one of the following categories: Entertainment; Politics; Climate and Weather; Crypto; Financials; Economics; Elections; Science and Technology; Transportation; Companies; Health; World; Sports. Report only your chosen category and nothing else.

Event match verification prompt:

> Report TRUE if the two questions below are essentially logically equivalent, i.e., the same or opposite, and FALSE otherwise:
> Question 1: [q1]
> Question 2: [q2]

Equivalent markets identification system prompt:

> Your goal is to find a subset of Polymarket markets which is _logically equivalent_ to the given Kalshi market, in that the Kalshi market outcome implies the Polymarket market outcomes, and vice versa. If there are no equivalences, return an empty list.
>
> Here are some examples:
>
> Example 1. The Kalshi market “The Witcher: Season 4 Rotten Tomatoes score Above 60?” is logically equivalent to the markets “The Witcher: Season 4 Rotten Tomatoes score is less than 45?” “The Witcher: Season 4 Rotten Tomatoes score is between 45 and 60?” since being above 60 is the the opposite of being either less than 45 or between 45 and 60.
>
> Example 2. The Kalshi market “Tesla Robotaxi unveiling delayed?” is logically equivalent to the market “Tesla reveals robotaxi on time?” since being delayed is the opposite of being on time.

Structured output definition:

```
class Market(BaseModel):
    id: int
    title: str

class Result(BaseModel):
    ticker_kalshi: str
    title_kalshi: str
    polymarket_markets: List[Market] = []
```

References

Ng, H.; Peng, L.; Tao, Y.; and Zhou, D. 2025. Price Discovery and Trading in Prediction Markets. *SSRN Electronic Journal*.

Rothschild, D.; and Pennock, D. M. 2014. The extent of price misalignment in prediction markets. *Algorithmic Finance*, 3(1-2): 3–20.

Saguillo, O.; Ghafouri, V.; Kiffer, L.; and Suarez-Tangil, G. 2025. Unravelling the Probabilistic Forest: Arbitrage in Prediction Markets. arXiv:2508.03474.