April 27, 2026

Binomial-Marks-v1

Earnings Sentiment Classification

Overview

Binomial-Marks-v1 (binomial-marks-1) is an open-source NLP scoring model that turns the text of an earnings call into 23 structured signals. It is the first model in Binomial Technologies' specialist zoo - a series of small, locally deployable ML models we plan to publish, that are also a part of our trading models. The series consists of models to perform tasks specifically for quantitative finance such as sentiment classification, market regime classification, forecasting and other NLP/time-series analysis tasks.

Each model in the zoo is named after a thinker who shaped how markets are understood. Marks-1 is named after Howard Marks (Oaktree), whose investor memos parse market sentiment, tone, and the gap between what is said and what is meant. The name fits the job: the model reads management commentary the way an analyst would, and converts it into scores that are usable isnide quantitative trading/risk models.

The model is a total of 400M parameters, making it small enough to run on a CPU, fast enough to score thousands of transcripts per hour on commodity hardware, and deterministic enough to be embedded directly inside a factor pipeline. The model performs close to SOTA models on the specific task of earnings call classification while being available to be run locally for free, cutting significant API costs. It is released under Apache 2.0 on HuggingFace and a streamlined scorer harness can be found on PyPI.

Numbers that anchor our case:

- **0.691 mean Spearman** vs. a panel of frontier reasoning models on topic-direction scoring ; **82% of the residual agreement** the frontier panel achieves with itself (0.838).
- **F1 = 0.91** on the binary "was this topic discussed?" heads - the model agrees with the teacher panel about 9 times out of 10 on whether a topic came up at all.
- **~50ms per call on a CPU, ~10ms on a modern GPU.** Two orders of magnitude faster than a frontier-LLM API call, deterministic, zero-cost and runs offline with no vendor relationship.
- **16K context window** on inputs, which covers most earnings call transcripts' full length (99th percentile on a sample of 93,572 earnings calls from 6 different countries between 2012-2026)

Problem Solved

Earnings calls are one of the highest-information-density events in the equity calendar. A single 90-minute transcript contains forward guidance, capital-allocation signals (buybacks, dividends, M&A), demand commentary, margin direction, headcount intent, competitive posture, and tone — both prepared (CEO/CFO confidence) and reactive (management defensiveness, analyst skepticism). The structured numbers in the press release get priced in within minutes. The unstructured language layer - *"we expect demand to remain robust" vs. "we expect demand to remain healthy" vs. "we expect demand to be in line with our prior commentary"* - is priced in over hours and days.

This language layer is where systematic strategies have an edge if they can extract it. It is also where they have historically struggled, because each of the available approaches has a structural weakness:

Hand-coded sentiment
The traditional approach: dictionary methods such as Loughran-McDonald, bag-of-words classifiers, simple sequence models. Cheap, fast, and explainable.

What it misses:

- **Context:** “Margins compressing 200 bps” is a critical signal in a software business and an unremarkable observation in low-margin retail. A dictionary score treats them identically.
- **Nuance:** Defensiveness, hedging, and confidence are distributional properties of language, not bag-of-words. “We don’t think the macro is changing our view” is defensive; a dictionary parses it as neutral.
- **Schema:** Bag-of-words returns a scalar or a small vector. It does not return structured outputs like “buybacks were upsized, dividends are flat, headcount commentary leaned cautious.” It cannot.

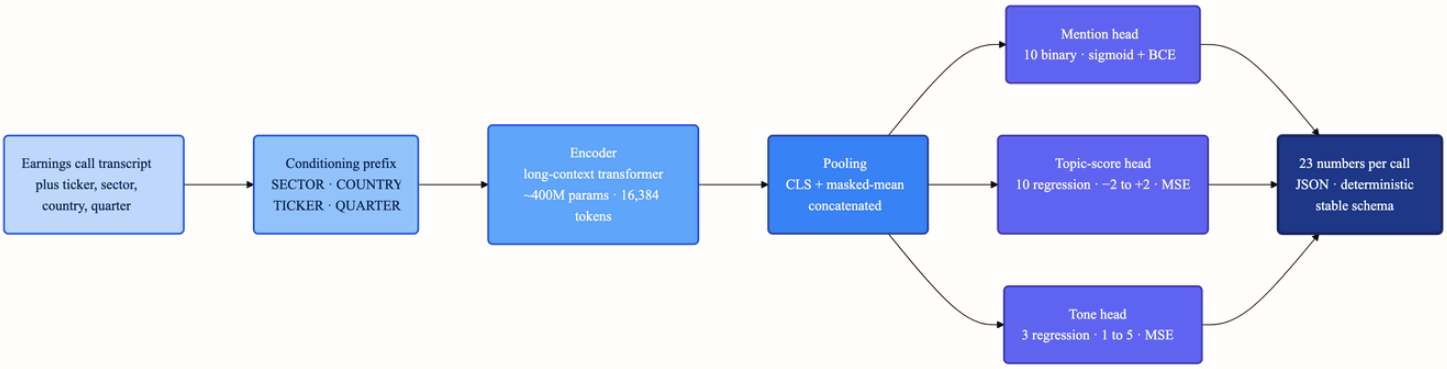
Frontier LLMs via API
The new default: prompt models like Claude, GPT, Grok, or Gemini with a structured-output schema and let the LLM extract signals. Quality is strong, but there are real constraints:

- **Cost:** Even at low reasoning effort, frontier APIs run roughly $0.01 to $0.05 per earnings call. At scale this compounds quickly. A fund scoring 50,000 calls per quarter ends up at $1,500 to $7,500 per quarter, and that is before retries, failures, or iteration cycles on prompts.
- **Latency:** Inference is not instant. A reasoning-capable model takes roughly 2 to 10 seconds per call. If you need to score an entire universe shortly after transcripts drop, that delay becomes a bottleneck. Even with batching, total wall-clock time stretches into hours.
- **Schema instability:** Outputs are not fixed. A field like {"guidance": 1.7} today can become {"guidance_score": 1.7}, or a string label, or fail validation entirely after a model update. You end up building retry layers, schema guards, and version pinning just to maintain consistency.

Architecture

Marks-1 is a 400M-parameter long-context transformer encoder with three lightweight regression heads sharing a pooled representation.

The pipeline has four stages: **(1) input conditioning, (2) encoder, (3) pooling, (4) heads.**



Hand-coded sentiment

The model takes a transcript with a small amount of structured metadata: ticker, sector, country, year, and quarter. The metadata is rendered as a deterministic prefix that is prepended to the transcript text:

```
[SECTOR: Technology] [COUNTRY: US] [TICKER: NVDA] [QUARTER: Q4 2025]
```

This is not a label appended to the input - the model is trained from scratch on prefixed inputs, so the encoder learns to weight language differently when a `[SECTOR: Software]` tag is present versus `[SECTOR: Energy]`. **The result:** the same sentence about margin pressure produces a sharper negative score in a sector where margins are normally stable, and a more attenuated score in a sector where they are volatile. Concrete examples of context shifting the signal:

Margin compression language

“Gross margin compressed 80 basis points sequentially.”

- In Software: marks-1 returns margins ≈ -0.9, treating it as a meaningful negative signal because margins are typically stable and high, so compression is informative.
- In Energy E&P: marks-1 returns margins ≈ -0.3, since margin volatility is structurally higher and this magnitude is closer to noise.
- In Retail Grocery: marks-1 returns margins ≈ -0.4, recognizing it as a recurring pressure point but still a real negative signal.

Demand softening language

“We are seeing softening in our consumer-discretionary segment.”

- In a US apparel retailer: clear negative on demand.
- In a global apparel name: same demand signal, with an additional mild negative on macro_exposure reflecting broader demand weakness.
- In a luxury goods company: more negative overall because demand feeds directly into high-margin structures, amplifying impact.

Headcount commentary

"We have decided to undertake a thoughtful evaluation of our cost structure."

Across sectors: interpreted as a soft negative on headcount, recognizing this as standard euphemistic language for cost cuts or layoffs. The signal is weighted more negatively when combined with a defensive or cautious tone in Q&A, indicating reactive rather than proactive cost management.

Why conditioning matters

This sector-aware conditioning is the main driver of performance. Without it, identical language maps to identical scores regardless of context, which is wrong in most economically meaningful cases.

A non-conditioned encoder of similar size underperforms by roughly 0.10 in mean Spearman against a frontier-model panel, with the largest degradation in context-sensitive topics such as margins, macro_exposure, and competition.

Operational implication

Conditioning inputs such as sector, country, ticker, and quarter are not optional in practice. Defaults allow inference to run, but degrade signal quality. In production, this implies a hard dependency on a clean security master lookup before scoring transcripts.

Encoder

The encoder is a transformer-based long-context model with rotary positional embeddings and **a 16,384-token context window,** large enough to cover nearly all public earnings call transcripts without truncation. It produces a hidden state for every token in the input. **Total parameter count is roughly 400 million, with the vast majority residing in the encoder.** The three output heads account for less than 5% of the total. The encoder is initialized from a strong public checkpoint and then adapted for longer context. The base model supports around 8k tokens; the extension to 16k is achieved through positional embedding extrapolation methods designed to preserve representation quality at longer sequence lengths.

Pooling

After the encoder, the sequence of token-level hidden states is collapsed into a single fixed-size representation using two pooling mechanisms:

- **CLS pool:** The hidden state of a designated summary token at position 0. This acts as the model's learned global summary of the entire transcript.
- **Masked-mean pool:** The average of all non-padding token hidden states. This captures the distributed signal across the full sequence rather than relying on a single position.

The two vectors are concatenated and passed to the heads. This dual-pooling design is not cosmetic. It handles the length distribution problem. Shorter transcripts tend to concentrate signal in the summary token, while longer transcripts dilute that signal and benefit from aggregation across tokens. Using only one pooling method causes measurable degradation at one end of the length spectrum.

Heads

Three lightweight 2-layer MLPs share the pooled representation:

Head	Output dim	Activation	Loss	Range at inference
Mention	10	sigmoid	BCE-with-logits	{0, 1} after threshold
Topic score	10	linear	MSE	clamped to [-2, +2]
Tone score	3	linear	MSE	clamped to [1, 5]

Each head is *Linear → GELU → Dropout → Linear*. The heads are trained jointly under a weighted multi-task objective that prioritizes the mention and topic-score heads over the tone head - reflecting the fact that topic decisions carry more direct signal for downstream factor models, and that tone labels in the teacher panel are partly mode-collapsed (see Limitations). At inference, the topic-score output is masked by the mention output: when *mentioned=False*, the corresponding score is forced to 0. The topic was not discussed, so direction is undefined.

Training data

The model is trained on 80,000+ earnings call transcripts spanning 2,700+ unique tickers across global markets, dated 2012 through 2026. Each transcript carries sector, country, and industry metadata.

Coverage by region:

Region	Approx share of training set
United States	~70%
United Kingdom	~6%
Western Europe (DE, FR, NL, IT, ES, etc.)	~9%
Northern Europe (Nordics + CH)	~4%
Asia-Pacific developed (JP, AU, HK, SG)	~6%
Emerging markets (CN, IN, BR, MX, etc.)	~4%
Other	~1%

Coverage by sector

Broadly market-cap-weighted and roughly proportional to the public equity universe. Technology and Financials are over-represented relative to a count-based weighting, reflecting their larger share of listed market capitalization. Real Estate, Utilities, and small-cap consumer sectors are under-represented on a count basis, but aligned with the analyst-followed universe rather than raw listing counts.

Coverage by time:

- 2012–2015: Sparser; pre-cycle baseline
- 2016–2019: Full coverage; pre-COVID economic regime
- 2020–2022: Heavy coverage; COVID + recovery cycle
- 2023–2024: Full coverage; rate cycle + AI capex onset
- 2025–2026: Full coverage through training cutoff (March 2026)

Labels

Labels are generated by frontier reasoning systems applied uniformly across the full training corpus, then validated against a separate panel of frontier LLMs on a held-out benchmark. No human annotation is used. The model is trained purely as a language-imitation system, learning to replicate structured outputs produced by stronger models.

Split

- Split type Value
- Method: Random
- Ratio: 80/20
- Unit: (ticker, year, quarter)
- Seed: Fixed

Because labels are deterministic functions of the input (produced by language models, not future market data), there is no information leakage from using a non-temporal split. A temporal split would only be necessary if labels depended on forward-looking outcomes. Here, they do not.

Output schema

Tone scores

- 3 dimensions
- Each is a regression in [1, 5]

Dimension	What 1 means	What 5 means
mgmt_confidence	uncertain, hedged language	direct, explicit commitments
mgmt_defensiveness	open Q&A, direct answers	evasive, deflecting, non-committal
analyst_skepticism	soft, congratulatory tone	repeated questioning, visible skepticism

Given a transcript with metadata, marks-1 returns 23 numbers per call.

Topic-direction scores

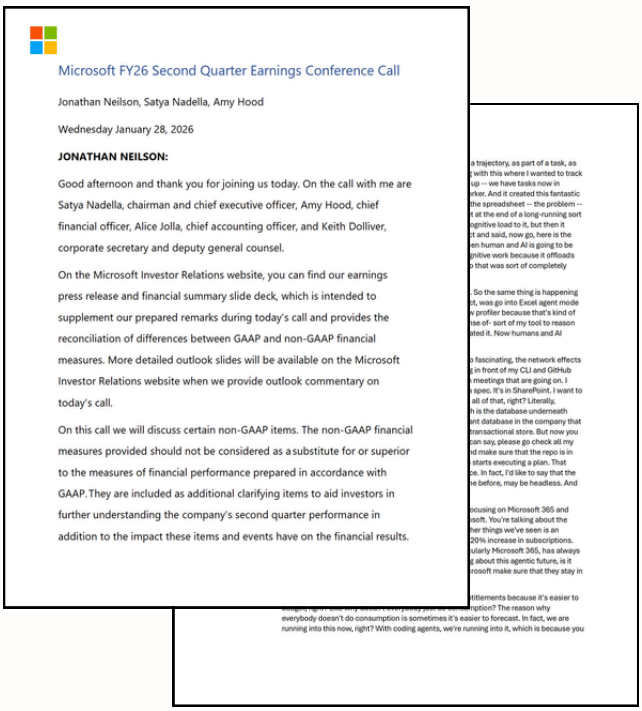
- 10 topics
- Each has:
  - a binary mention flag (from sigmoid probability)
  - a signed score in [-2, +2]
- If mention = False → score is forced to 0

Topic	What –2 means	What +2 means
guidance	lowered hard, withdrawn, or sharply de-risked	raised significantly above prior
revenue_growth	decelerating into negative territory	accelerating well above prior
margins	compressing materially (operating or gross)	expanding materially
demand	softening, shortening order books	strong, lengthening order books
buybacks	paused, reduced, or completed without renewal	new program announced or upsized
dividends	cut, suspended, or skipped	raised, special declared, or initiated
m_and_a	divestiture, asset sale, or wind-down	strategic acquisition, transformative deal
headcount	layoffs, restructuring	aggressive hiring
macro_exposure	clear macro headwind	clear macro tailwind
competition	losing share, pricing pressure	gaining share, pricing power intact

The shape is JSON-serializable, deterministic, and stable. Versions within v1 will not change the schema; v2 may add dimensions but will preserve the v1 layout for backward compatibility.

Worked example

Below are score outputs from [Microsoft's Q2 2026 Earnings Call] graded by binomial-marks-v1 with a full output schema


Microsoft FY26 Second Quarter Earnings Conference Call

Jonathan Neilson, Satya Nadella, Amy Hood

Wednesday January 28, 2026

JONATHAN NEILSON:

Good afternoon and thank you for joining us today. On the call with me are Satya Nadella, chairman and chief executive officer, Amy Hood, chief financial officer, Alice Jolla, chief accounting officer, and Keith Dolliver, corporate secretary and deputy general counsel.

On the Microsoft Investor Relations website, you can find our earnings press release and financial summary slide deck, which is intended to supplement our prepared remarks during today's call and provides the reconciliation of differences between GAAP and non-GAAP financial measures. More detailed outlook slides will be available on the Microsoft Investor Relations website when we provide outlook commentary on today's call.

On this call we will discuss certain non-GAAP items. The non-GAAP financial measures provided should not be considered as a substitute for or superior to the measures of financial performance prepared in accordance with GAAP. They are included as additional clarifying items to aid investors in further understanding the company's second quarter performance in addition to the impact these items and events have on the financial results.

⇒


```
{
  "_meta": {
    "model": "binomial-marks-1",
    "ticker": "MSFT",
    "sector": "Technology",
    "country": "US",
    "year": 2026,
    "quarter": 1,
    "call_date": "2025-10-29",
    "transcript_length_chars": 52207
  },
  "topics": {
    "guidance":       {"mentioned": true,  "mention_prob": 0.884, "score":  1.280},
    "revenue_growth": {"mentioned": true,  "mention_prob": 0.960, "score":  1.789},
    "margins":        {"mentioned": true,  "mention_prob": 0.960, "score":  1.168},
    "demand":         {"mentioned": true,  "mention_prob": 0.971, "score":  1.587},
    "buybacks":       {"mentioned": true,  "mention_prob": 0.829, "score":  1.136},
    "dividends":      {"mentioned": true,  "mention_prob": 0.563, "score":  0.587},
    "m_and_a":        {"mentioned": false, "mention_prob": 0.342, "score":  0.000},
    "headcount":      {"mentioned": false, "mention_prob": 0.447, "score":  0.000},
    "macro_exposure": {"mentioned": true,  "mention_prob": 0.688, "score":  0.084},
    "competition":    {"mentioned": true,  "mention_prob": 0.759, "score":  1.268}
  },
  "mgmt_confidence":    4.624,
  "mgmt_defensiveness": 1.231,
  "analyst_skepticism": 1.777
}
```


Evaluation

Methodology

Marks-1 is evaluated against four frontier reasoning systems on a held-out benchmark of 2,000 earnings calls:

- Claude Opus 4.7 (Anthropic)
- GPT-5.5 (OpenAI)
- Grok-4.2 reasoning (xAI)
- DeepSeek V4-Pro (DeepSeek)
- Marks-1 (Binomial AI Research)

The same 2,000 transcripts are scored by all five systems using the same prompt and schema. Pairwise agreement is then measured across:

- Topic-direction Spearman — rank correlation on the 10 signed topic scores. Mention masking is applied (when a model says mentioned=False, the score is treated as 0 in the rank). This is the headline number.
- Tone Spearman — rank correlation on the 3 tone dimensions.
- Mention MAE — mean absolute error on the 10 binary mention flags.

This panel-of-judges design is the cleanest way to evaluate a model whose ground truth is itself produced by language models: we measure whether marks-1 agrees with the panel as much as the panel members agree with each other. The benchmark is symmetric — marks-1 is included in the panel as a fifth voter, on equal footing with the other four — so the diagonal of the agreement matrix is interpretable across the board.

Methodology

Marks-1 is evaluated against four frontier reasoning systems on a held-out benchmark of 2,000 earnings calls:

- Claude Opus 4.7 (Anthropic)
- GPT-5.5 (OpenAI)
- Grok-4.2 reasoning (xAI)
- DeepSeek V4-Pro (DeepSeek)
- Marks-1 (Binomial AI Research)

Setup:

- Same 2,000 transcripts
- Same prompt and schema
- All five models score each transcript

Metrics:

- Topic-direction Spearman
  - Rank correlation across 10 signed topic scores
  - Mention masking applied (if mentioned = False → score = 0)
  - Primary evaluation metric
- Tone Spearman
  - Rank correlation across 3 tone dimensions
- Mention MAE
  - Mean absolute error on 10 binary mention flags

Why frontier-frontier agreement isn't 1.0

A natural question: if we are using frontier LLMs as graders, why don't they agree with each other perfectly?

Three reasons:

1. Genuine ambiguity. Many earnings-call statements are intentionally ambiguous. A "we're cautiously optimistic" comment could be a 3.5 or a 4.0 on mgmt_confidence depending on how the rater weights the hedging word. Reasonable raters disagree.
2. Calibration differences. Different LLMs have different priors about where the "neutral" point sits. DeepSeek tends to anchor lower on confidence than the Western frontier models; this is a calibration drift, not a noise problem.
3. Different reasoning styles. Reasoning models trained with different reinforcement-learning objectives reach different conclusions on edge cases — particularly on topics like competition where the right answer often depends on what comparison set the rater has in mind.

These three factors cap the achievable agreement at roughly 0.85 on topic-direction and roughly 0.75 on tone, even between equally-capable frontier systems.

Headline result

Marks-1 reproduces approximately 80% of the agreement that frontier reasoning systems have with each other on financial NLP scoring. On tone scoring, marks-1 sits at parity with the frontier-frontier baseline (0.62 vs 0.61). For tone alone — confidence, defensiveness, analyst posture — marks-1 is qualitatively indistinguishable from a frontier reasoner.

For a quant pipeline that scores tens of thousands of earnings calls per quarter, this is the difference between a budget line item and a rounding error.

Topic-direction Spearman: full pairwise matrix

Pairwise rank correlation across the 10 topic-direction dimensions, on the 2,000-call benchmark:

	Opus	GPT	Grok	DeepSeek	Marks-1
Opus 4.7	—	*0.886*	*0.832*	*0.803*	*0.697*
GPT-5.5	*0.886*	—	*0.871*	*0.827*	*0.696*
Grok	*0.832*	*0.871*	—	*0.807*	*0.677*
DeepSeek V4	*0.803*	*0.827*	*0.807*	—	*0.627*
Marks-1	*0.711*	*0.712*	*0.692*	*0.644*	—

Comparison	Mean Spearman
Frontier ↔ Frontier (6 pairs)	**0.838**
Marks-1 ↔ Frontier (4 pairs)	**0.691**
Best Marks-1 pair	0.712 (GPT 5.5)
Worst Marks-1 pair	0.644 (vs DeepSeek)
Best Frontier pair	0.886 (Opus ↔ GPT-5.5)
Worst Frontier pair	0.803 (DeepSeek ↔ Opus)

Marks-1 captures four-fifths of the residual agreement between frontier reasoners on the same task, which is strong given it runs locally on a single GPU with no API dependency. Across pairs, the structure is what you would expect rather than something magical. Marks-1 aligns most closely with Opus and GPT-5.5 at roughly 0.697, which makes sense because those models cluster stylistically. The drop to DeepSeek at 0.627 is not a weakness of marks-1 specifically; DeepSeek sits in a different agreement cluster, and even the frontier models disagree with it more than they do with each other.

Tone Spearman: parity with frontier

Tone is harder and noisier across the board.

Comparison	Mean tone Spearman
Frontier ↔ Frontier (6 pairs, all 5 LLMs)	**0.61**
Frontier ↔ Frontier (Opus + GPT-5.5 + Grok, excluding DeepSeek)	**0.72**
Marks-1 ↔ Frontier	**0.62**

Dimension	Mean Spearman
mgmt_confidence	0.66
mgmt_defensiveness	0.61
analyst_skepticism	0.59

Marks-1 lands at parity with the full frontier baseline, slightly above it in fact. If you restrict to the cleaner Western cluster, it sits within about 0.10, which is about as tight as you get without being the same model class.

DeepSeek is the outlier again. Its tone calibration diverges materially, with pairwise tone agreement around 0.50 to 0.55 versus roughly 0.78 between Opus and GPT-5.5. Including it drags the panel average down, which is why the baseline sits at 0.61. Marks-1 clears that bar.

Confidence is the cleanest signal because the language is more explicit and closer to rule-based patterns. Analyst skepticism is the messiest because it depends on subtle conversational cues and distinguishing politeness from genuine sentiment, which even humans do inconsistently.

Mention F1 and mention MAE

The 10 binary "was this topic discussed?" heads are easier than the regression heads and the model performs proportionally better on them:

Metric	Frontier ↔ Frontier	Marks-1 ↔ Frontier
Mean mention MAE	0.05	0.1

Metric	Marks-1
Mention macro-F1	**0.9092**
Topic-score macro-Spearman	0.6658
Tone macro-Spearman	0.6524
Overall (weighted)	0.7425

A mention macro-F1 of 0.91 means that across all 10 topics, marks-1 agrees with the teacher about 9 times out of 10 on whether a topic came up at all. For most quant pipelines this is sufficient — the second-stage signal (direction, magnitude) is where most of the alpha lives, and conditioning on a mention threshold gives a clean gating mechanism.

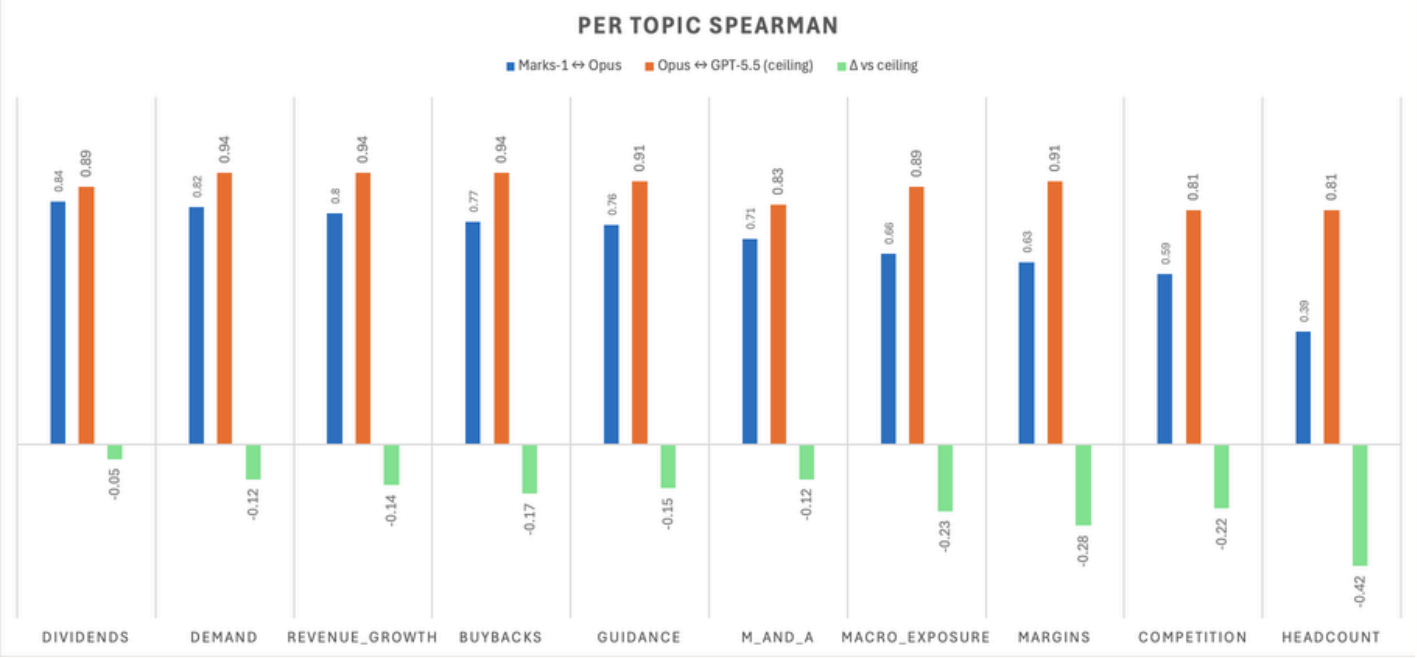
Per topic mention F1

Topic	Mention F1
revenue_growth	0.95
guidance	0.95
demand	0.94
margins	0.93
dividends	0.92
buybacks	0.91
headcount	0.9
m_and_a	0.89
macro_exposure	0.89
competition	0.88

Even the weakest mention head (competition at 0.88) is well above the threshold where a binary classifier becomes useful as a gating signal. In practice, the mention heads are reliable enough that consumers can use them as hard gates.

Per-topic Spearman vs Claude Opus 4.7

The model is strongest on topics with the clearest direction-of-growth signals (dividend changes, demand commentary, revenue growth) and weakest on topics that require disentangling subtle structural language (margins, competition, headcount).



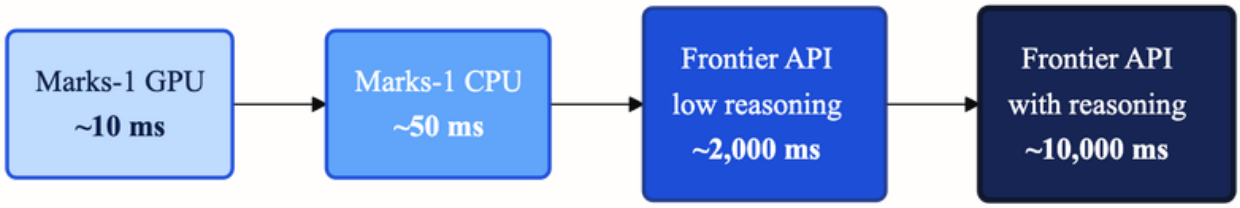
Why headcount is hard

- It is often discussed indirectly. Phrases like “evaluating our cost structure” or “right-sizing the organization” are euphemisms that require interpretation rather than direct mapping.
- Hiring is frequently mentioned in passing rather than as a formal disclosure, with weak and strong signals mixed in the same transcript.
- Headcount language is highly context-dependent across sectors; what is normal in software can be meaningful in a mature retailer.

The model card explicitly flags headcount as the weakest dimension. Future iterations will target this with more focused data and harder negative examples.

Cost Savings

Approach	Latency / call	Cost / call	Determinism	Offline?
Frontier LLM API (Opus / GPT-5.5 / Grok at low reasoning)	2–10 seconds	$0.01–$0.05	No (sampling)	No
Marks-1 on a modern CPU	~50ms	~$0 (electricity)	Yes	Yes
Marks-1 on a single A100/H100/B200 GPU (bf16)	~10ms	~$0	Yes	Yes
Marks-1 batched on a single GPU	~12 calls/sec/instance	~$0	Yes	Yes



Total cost of ownership scenarios

For a quant fund scoring its full coverage universe quarterly:

Universe	Calls / quarter	Frontier API ($0.09/call)	Marks-1 (single CPU instance)
US large-cap (S&P 500)	500	$45	~$0
US all-cap (3,000 names)	3,000	$180	~$0
Global developed (5,000 names)	5,000	$450	~$0
Global all-listed (15,000 names)	15,000	$1,350	~$0
Full historical backfill (50,000+ calls)	50,000+	$4,500+	~$0

These numbers assume one signal generation per call. A realistic backtest workflow that re-scores transcripts as the schema evolves (for example when adding a new dimension) multiplies the API cost line accordingly while leaving the marks-1 line essentially flat.

The cost ratio is roughly two orders of magnitude. The latency ratio is similar. Determinism - same input always produces the same 23 numbers - is the third factor; it makes downstream backtesting reproducible without freezing API responses or managing version pins.

Time-to-signal in a live pipeline

For a pipeline that scores transcripts as they land, for example processing the US earnings calendar shortly after market close:

Approach	Wall-clock for 200 calls landing in a 30-minute window
Frontier LLM API (sequential)	~30 minutes (one call per ~10 seconds)
Frontier LLM API (parallelized, 16-way)	~3–4 minutes plus retry overhead
Marks-1 on a single CPU instance	~10 seconds
Marks-1 on a single GPU instance	~2 seconds

For a desk that needs the signal block quickly to inform positioning, the API approach requires careful parallelization and capacity management. Marks-1 completes essentially in real time relative to transcript arrival.

Use Cases

Marks-1 outputs are features, not trades. They are designed to be consumed by existing quant infrastructure. This section covers (1) how to integrate the model, (2) deployment patterns, and (3) the application categories where marks-1 outputs are most useful.

Convenience helper:

```
MarksScorer(
    model_id="BinomialTechnologies/binomial-marks-1",    # HF Hub or local path
    device="cuda",                  # auto-detects: cuda > mps > cpu
    dtype=torch.bfloat16,           # default: bf16 on GPU, fp32 on CPU
    max_length=16384,               # tokenizer truncation cap
    mention_threshold=0.5,          # sigmoid threshold on the mention heads
)
```

Batched scoring

```
from binomial_marks import MarksScorer

scorer = MarksScorer()                                  # loads once
results = scorer.score_batch([
    {"transcript": "...", "ticker": "NVDA", "sector": "Technology", "year": 2025, "quarter": 4},
    {"transcript": "...", "ticker": "AAPL", "sector": "Technology", "year": 2025, "quarter": 1},
    {"transcript": "...", "ticker": "MSFT", "sector": "Technology", "year": 2025, "quarter": 2},
])
```

via transformers

```
from transformers import AutoTokenizer, AutoModel
import torch

tok = AutoTokenizer.from_pretrained("BinomialTechnologies/binomial-marks-1")
model = AutoModel.from_pretrained(
    "BinomialTechnologies/binomial-marks-1",
    trust_remote_code=True,
    torch_dtype=torch.bfloat16,
).eval().cuda()

prefix = "[SECTOR: Technology] [COUNTRY: US] [TICKER: NVDA] [QUARTER: Q4 2025]\n\n"
inputs = tok(prefix + transcript_text, return_tensors="pt",
             truncation=True, max_length=16384).to("cuda")

with torch.no_grad():
    out = model.predict(**inputs)
```

Convenience helper

```
from binomial_marks import score

result = score(
    transcript=transcript_text,
    ticker="NVDA", sector="Technology", country="US",
    year=2025, quarter=4,
)
```

Adjusting the mention threshold: The default mention_threshold = 0.5 is calibrated against the panel. For applications that prefer higher precision (fewer false-positive mentions, tighter gating) raise it to 0.65–0.7. For applications that prefer higher recall (catch every possible mention) lower it to 0.3–0.35. In a factor-model context, threshold is often less important than a probability-weighted signal: multiply the topic score by the mention probability before feeding it into the factor pipeline.

Deployment patterns: Marks-1 is a regular transformers model. Standard deployment patterns apply:
Inline in a research notebook. from binomial_marks import score, drop into a pandas pipeline. Suitable for ad-hoc analysis on tens to thousands of calls. On a CPU this runs at roughly 50ms per call, so 5,000 calls take ~5 minutes wall-clock.

Batch job: For a quarterly batch over 10,000+ new transcripts, a single GPU instance completes in 10–15 minutes wall-clock. FastAPI microservice. Wrap MarksScorer in a simple HTTP endpoint, deploy on internal infrastructure as a shared signal-generation service. Memory footprint is roughly 3 GB peak with a 16k-token context.

HuggingFace Inference Endpoints: The model loads cleanly with trust_remote_code=True so any standard HF deployment surface works. For external-facing or shared-research-team deployments, this is the fastest path to a managed endpoint. Subprocess in a data pipeline. For environments where the parent process cannot import torch (for example an Airflow worker without GPU dependencies), run marks-1 as a child process that reads transcripts on stdin and writes JSON on stdout. This decouples the heavy ML dependency from the orchestrator.

Screening filters

Pre-filter the universe by topic-direction signs:

```
SELECT ticker FROM marks_signals
WHERE quarter = '2026Q1'
  AND guidance_score > 1.0
  AND mgmt_confidence > 4.5
  AND analyst_skepticism < 2.5
  AND mention_prob_guidance > 0.7
```

This isolates a particular flavor of post-call setup ("management raised, analysts believed") without any return-data dependency. The screen can be combined with standard fundamentals filters (valuation, leverage) to produce a tight watchlist.

Event-study triggers

Use mention flags to gate event-study buckets:

- All calls where m_and_a was mentioned and m_and_a_score > 1.0 — explicit announcement of strategic acquisitions; useful for buyside merger-arb event studies.
- All calls where headcount_score < -0.5 — explicit layoff disclosures; useful for cost-cycle event studies.
- All calls where mgmt_defensiveness > 4.0 — calls where management was visibly defensive, which historically correlates with negative forward returns even when the headline numbers are mixed.

The output is a clean, reproducible event boundary with a built-in direction signal.

Sentiment indices

Aggregate marks-1 outputs across an index or sector to build a real-time sentiment series:

```
sector_sentiment = (
    df.groupby(["sector", "quarter"])
      .agg({"guidance_score": "mean", "mgmt_confidence": "mean"})
)
```

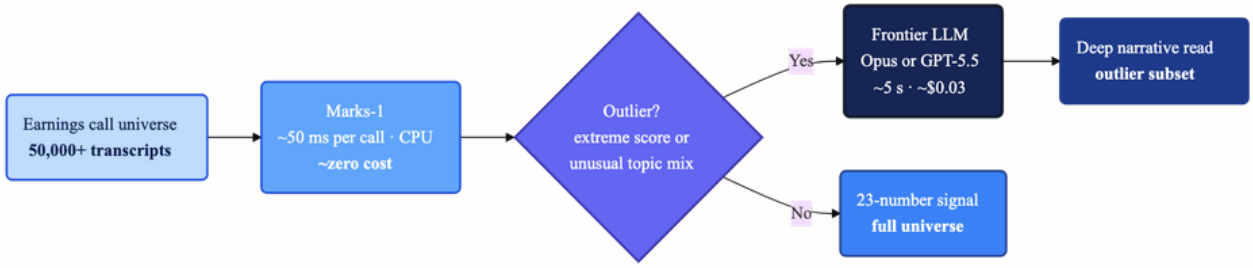
Because the model is deterministic and the schema is stable, the index can be backfilled across the full 2012–2026 corpus and updated daily as new calls land. The resulting series is a useful overlay on traditional macro sentiment indices such as ISM or consumer confidence, but at a sector-specific resolution.

Risk model overlays

The tone outputs, particularly mgmt_defensiveness and analyst_skepticism, are useful as risk-model overlays. A quarterly portfolio risk decomposition that includes a defensive-management factor will surface idiosyncratic risks that volatility-only frameworks miss.

Cross-call comparison

Because marks-1 scores are deterministic and on a fixed scale, consecutive quarters are directly comparable:



Limitations

- **Tone has rank-order signal but absolute levels drift:** The teacher models that generate tone labels are partially mode-collapsed. Confidence tends to cluster around 4–5, defensiveness around 1–2. The model captures relative ordering well, but absolute values are not directly comparable across sectors or time.

- **English transcripts only:** Training data is heavily skewed toward English-language transcripts, primarily US issuers with additional coverage from Europe and parts of Asia. Non-English transcripts in translation work but degrade, especially if the translation is automated.

- **16,384-token context:** Covers roughly the 99th percentile of earnings calls. Extremely long calls, such as extended analyst sessions, exceed this limit and lose middle content due to truncation. This affects a small fraction of the universe.

- **Distillation framing:** The model is a language-imitation system. It inherits both strengths and biases from its teacher panel. Agreement metrics capture differences between models but not shared miscalibration.

- **Deterministic but not adversarially robust:** Outputs are deterministic for standard inputs, but the model is not designed to resist adversarial manipulation such as injected text.

Where to find it

- Model weights & card: https://huggingface.co/BinomialTechnologies/binomial-marks-1
- PyPI package: https://pypi.org/project/binomial-marks/
- Source repository: https://github.com/Binomial-Capital-Management/binomial-ai-research
- Labels dataset (forthcoming): BinomialTechnologies/marks-labels-v1