The Generative Brand Mention Framework: A Formula-by-Formula Primer

A technical primer on MV, MA, and MS

Aiviara Research · Technical Primer Series

June 2026

How to use this document

This primer explains the formulas in the Generative Brand Mention Framework (GBMF) with worked examples. Each formula section covers the definition, intuition, step-by-step breakdown, a concrete worked example, and a small Python implementation. No prior familiarity with mathematical or academic notation is assumed. The worked example throughout is the synthetic brand Acme Analytics, which produces the headline scores MV 52 / MA 67 / MS +25 from a 30-prompt × 5-engine measurement. The full per-prompt per-engine mention rates are published as supplementary data alongside the canonical paper, so the MV-related figures here (per-engine MV, headline MV, reach, intensity, HHI) can be re-derived from an open dataset. MA and MS figures rely on response-level evaluator output that is not included in the published CSV.

> **Validation status.** GBMF is a proposed open specification. It has not yet been empirically validated. The calibration conventions used here (the $n \times k \geq 150$ collection rule, the eligible-cell floor of 10, the working evaluator-agreement targets) are stated as provisional pending field calibration. The framework is published to enable methodological scrutiny and field adoption, not as a fully validated operational system.

Framework overview

Before working through each formula, here is the entire framework in plain English.

MV (brand-mention visibility) answers: *How often does this brand appear in AI answers to relevant questions?*

A brand with a high MV appears consistently across many different prompts and many different AI engines. A brand with a low MV is rarely mentioned. MV is a detection measure over the full declared prompt set. It does not condition on anything.

MA (brand-mention alignment) answers: *When AI does mention this brand, how accurate is what it says?*

A high MA means AI describes the brand correctly: right category, right pricing, right differentiators. A low MA means AI gets the facts wrong, with wrong product tier, outdated positioning, or characteristics borrowed from a competitor. MA is a conditional measure; it conditions on the brand being mentioned.

MS (brand-mention sentiment) answers: *When AI does mention this brand, does it speak positively or negatively about it?*

A positive MS means AI characterisations favour the brand. A negative MS means AI characterisations criticise it. The unit of analysis is the brand-directed characterisation, not the overall tone of the response.

MS is the second conditional measure: it also conditions on the brand being mentioned. It does not condition on whether the description was accurate.

Why three measures and not one. A brand can be mentioned yet described inaccurately. A brand can be described accurately yet positioned unfavourably. A brand can be visible, accurate, and well-characterised, while another brand in the same category sits at every other corner of that cube. The three measures separate properties that prior instruments collapse or omit.

What the framework does not do. The formulas alone do not explain *why* a brand has a given score or prescribe how to improve it; those questions are handled through the diagnostic reports and practitioner interpretation built on the measures. The framework also does not establish that AI representation predicts commercial outcomes. It provides a reproducible measurement that brands, researchers, and agencies can use to track change over time and compare positions across competitors.

Notation primer: reading mathematical symbols

Mathematical notation is a compact shorthand. The notation is consistent: the same symbols appear throughout, and each formula is an instruction written in that shorthand. Here is a translation guide for every symbol used in this primer.

Symbols as variable names

Each symbol represents a value in the measurement:

Symbol	What it represents	Example value
B	The brand being measured	"Acme Analytics"
F_B	The Brand Profile for B (the ground-truth document)	"ACME-v2025-11-01"
P	The declared prompt set	30 questions
n	Total number of prompts in P	30
E	Number of AI engines in the declared engine set	5
k	Number of runs per prompt-engine pair	5
p	A single prompt in P	"best CRM for startups"
e	A single engine in E	"ChatGPT" (code: GPT)
s	A single run, $1 \leq \mathrm{s} \leq \mathrm{k}$	run 3 of 5
$r_e(p)$	Per-engine mention rate for engine e on prompt p	0.60
MV_e	Per-engine MV for engine e	62
$A(r, F_B)$	Alignment score of response r against Brand Profile F_B	0.80

Subscripts: which item in a list

When a symbol carries a subscript, the subscript is an index. $r_e(p)$ is "the mention rate when engine e answers prompt p", read like `rate[e][p]` in code. MV_e is the MV value computed for engine e. $c(p, e, s)$ is the binary mention outcome (0 or 1) on prompt p, engine e, run s, read like `mentioned[p][e][s]` in code.

Sets: collections of distinct items

A **set** is a bag of distinct values, like a Python `set()` or a list with no duplicates.

- `{1, 2, 3, ..., n}` = the integers from 1 to n
- `e ∈ {1, ..., E}` means "e is one of the integers from 1 to E"
- `|S|` means the number of elements in set S, like `len(S)` in Python
- `Q ⊆ M` means "every element of Q is also in M" (Q is a subset of M)

Σ (sigma): adding things up

Σ means "add all of these up". The notation:

```
Σ_{i=1}^{n}  f(i)   →   f(1) + f(2) + f(3) + ... + f(n)
```

The bottom of the Σ tells you where to start; the top tells you where to stop. In Python:

```
total = sum(f(i) for i in range(1, n + 1))
```

Example: Σ_{i=1}^{5} i = 1 + 2 + 3 + 4 + 5 = 15.

Everyday equivalent: "Add up the scores for every question from question 1 to question n."

Mean (average)

A horizontal bar over a symbol, or `(1/n) × Σ ...`, denotes the arithmetic mean: add up all the values and divide by how many there are. The mean of {0.6, 0.8, 0.4, 0.6} is (0.6 + 0.8 + 0.4 + 0.6) / 4 = 0.60.

Indicator: c(p, e, s) in {0, 1}

`c(p, e, s)` is either 0 or 1. It is 1 if engine e mentioned brand B in response to prompt p on run s, and 0 otherwise. Adding up indicators counts the cases where the condition holds. `Σ_s c(p, e, s)` counts the number of runs on which engine e mentioned brand B in response to prompt p.

Shares and net (for MS)

`%pos`, `%neu`, `%neg` are the percentage of eligible mentioning cells classified positive, neutral, and negative. They sum to 100% per engine. The **net** is `%pos - %neg`, ranging from −100 (everything negative) to +100 (everything positive). For Acme Analytics on engine GPT: %pos = 50, %neu = 25, %neg = 25, net = 50 − 25 = +25.

HHI: the concentration index

The Herfindahl-Hirschman Index is the sum of squared shares: `HHI = Σ_i s_i²` where `s_i` is the share contributed by item i. If all shares are equal at 1/n, HHI = n × (1/n)² = 1/n. So `1/n` is the uniform baseline; a multiple of uniform (e.g., 1.4× uniform) is the readable diagnostic.

Part 1: MV (brand-mention visibility)

MV measures how consistently a brand is mentioned across a declared set of prompts posed to a declared set of AI engines. It answers: *"Across the questions buyers actually ask, how often is this brand appearing in AI answers?"*

MV is a detection measure. It operates over the full prompt set and conditions on nothing.

Formula 1.1. Per-engine mention rate per prompt: $r_e(p)$

For a single prompt and a single engine, the fraction of k runs in which the engine mentioned the brand.

Intuition. You ask one colleague the same question 5 times: "Which analytics platform do you recommend for B2B teams?" They mention Acme Analytics in 3 of those 5 answers. The per-engine mention rate for

that prompt is 3 / 5 = 0.60. If you ran the same prompt through a different engine, that engine's rate might be 0.20 or 0.80; you record each engine separately and aggregate at the next step.

The formula:

$$r_e(p) = \frac{1}{k}\sum_{s=1}^{k} c(p,e,s) = \frac{\text{runs on which engine } e \text{ mentioned } B \text{ on prompt } p}{k}$$

where $c(p,e,s) \in \{0,1\}$ is 1 if engine e mentioned brand B on prompt p in run s, and 0 otherwise.

Breakdown:

- The numerator counts the runs on which engine e produced a mention
- Dividing by k converts the count to a proportion between 0 and 1
- A higher k reduces sampling noise; the framework requires k $\geq$ 3 for all measurement

Worked example. k = 5 runs. On the prompt "best analytics platform for B2B SMBs", engine GPT's outcomes are:

Run	Mentioned?
1	yes
2	yes
3	no
4	yes
5	no

```
r_GPT(p) = (1 + 1 + 0 + 1 + 0) / 5 = 3 / 5 = 0.60
```

Python:

```python
def per_engine_rate(outcomes):
    # outcomes: list of 0/1 values, length k
    return sum(outcomes) / len(outcomes)

per_engine_rate([1, 1, 0, 1, 0])  # → 0.60
```

$r_{GPT}(p) = 0.60$ means engine GPT mentioned the brand in 60% of its runs on this prompt. r = 0 means it never mentioned. r = 1 means it mentioned every time. The same prompt produces different r values for each of the declared engines.

Minimum runs. k $\geq$ 3 applies to all measurement, including monitoring scans. Single-run measurements (k = 1) are too sensitive to prompt-temperature and sampling noise to be stable, and are not admitted as framework-compliant measurement.

Formula 1.2. Per-engine MV: MV_e

The mean per-engine mention rate across all n prompts, expressed on a 0–100 scale. This is the atomic comparable unit of the framework: two studies measuring the same engine over the same period compare directly on this score.

The formula:

$$MV_e = 100 \times \frac{1}{n}\sum_{p=1}^{n} r_e(p)$$

Intuition. For each engine, take its rate on every prompt in the declared set, average those rates, and multiply by 100. An MV_e of 62 means engine e mentioned the brand in an estimated 62% of its responses to prompts in the declared set.

Worked example. Engine GPT, n = 30 prompts. After running every prompt 5 times, the 30 per-prompt rates $r_{GPT}(p)$ sum to 18.6.

$$MV_{GPT} = 100 \times \frac{1}{30} \times 18.6 = 100 \times 0.62 = 62$$

The remaining four engines in the worked example:

Engine	Sum of 30 $r_e(p)$	MV_e
GPT	18.6	62
CLD	17.4	58
PPX	13.8	46
GEM	16.2	54
COP	12.0	40

Per-engine scores are reported alongside the headline; engine differences are systematic, not sampling noise.

Python:

```python
def per_engine_mv(per_prompt_rates):
    # per_prompt_rates: list of r_e(p) values, length n
    n = len(per_prompt_rates)
    return 100 * sum(per_prompt_rates) / n
```

Formula 1.3. Headline MV

The equal-weighted mean of per-engine MV values across the declared engine list.

The formula:

$$MV = \frac{1}{E} \sum_{e=1}^{E} MV_e$$

Intuition. Take the per-engine MV scores; average them. The headline reconciles exactly with its parts because it is the arithmetic mean of those parts.

Why equal weighting and not market-share weighting? Two reasons. First, the score should change because AI mention behaviour changed, not because market shares among the declared engines shifted between measurement periods. Equal weighting holds the engine list constant and lets the engine-level signal show through. Second, equal weighting is engine-neutral: no engine is privileged in the headline, and the engine list is itself disclosed as part of every published score. Market-share-weighted overlays are permitted as documented non-canonical variants; the canonical headline is equal-weighted.

Worked example.

```
MV = (62 + 58 + 46 + 54 + 40) / 5 = 260 / 5 = 52
```

A headline MV of 52 means the brand appears in an estimated 52% of AI responses to category-relevant prompts, averaged across the declared engines.

Python:

```python
def headline_mv(per_engine_mvs):
    # per_engine_mvs: list of MV_e values, length E
    return sum(per_engine_mvs) / len(per_engine_mvs)
```

Companion statistics: reach, intensity, concentration

MV is a level measure. Two brands with the same MV can have very different mention shapes. Three companion statistics carry the shape information. They are reported alongside the headline, never merged into it.

Reach: percentage of prompts with a mention

Per-engine reach. The fraction of prompts on which engine e mentioned the brand at least once across its k runs.

$$\text{reach}_e = \frac{|\{p : r_e(p) > 0\}|}{n}$$

Overall reach. The fraction of prompts on which at least one engine mentioned the brand at least once.

$$\text{reach} = \frac{|\{p : \exists e, r_e(p) > 0\}|}{n}$$

Worked example. For Acme Analytics, overall reach = 26 / 30 = 87%. Per-engine reach: GPT 87%, CLD 80%, PPX 67%, GEM 77%, COP 70%.

Intensity: mean mention rate among reached prompts

Per-engine intensity. Among the prompts engine e reaches (where $r_e(p) > 0$), the mean mention rate on those prompts.

$$\text{intensity}_e = \text{mean of } r_e(p) \text{ over } \{p : r_e(p) > 0\}$$

Headline intensity. The equal-weighted mean of per-engine intensities, following the same per-engine-first aggregation as MV.

$$\text{intensity} = \frac{1}{E} \sum_{e=1}^{E} \text{intensity}_e$$

Worked example. Per-engine intensities for Acme Analytics: GPT 0.72, CLD 0.73, PPX 0.69, GEM 0.70, COP 0.57. Headline intensity = (0.72 + 0.73 + 0.69 + 0.70 + 0.57) / 5 = **0.68**.

Reach and intensity together describe the mention distribution. An MV of 40 from "70% reach, intensity 0.57" is a broad-but-shallow position. An MV of 40 from "30% reach, intensity 1.00" is narrow and deep. The headline MV cannot distinguish these states; reach and intensity do.

Concentration: HHI over prompt contribution shares

The Herfindahl-Hirschman Index measures how concentrated mention is across the prompt set.

Let total_mentions = $\sum_p \sum_e \sum_s c(p, e, s)$ be the total number of mention observations across all prompts, engines, and runs. The contribution share of prompt p is:

$$\text{share}_p = \frac{\sum_e \sum_s c(p, e, s)}{\text{total\_mentions}}$$

The HHI is then:

$$\text{HHI} = \sum_p \text{share}_p^2$$

Uniform baseline. If every prompt contributed equally, each share would be 1/n and HHI = n × $(1/n)^2$ = 1/n. For n = 30, uniform baseline HHI = 0.033.

Interpretation. A multiple of uniform is the readable quantity. HHI 1.0× uniform indicates broadly distributed visibility. HHI 3× uniform or higher indicates mention concentrated on a small number of prompts (fragile, narrow visibility). For Acme Analytics, HHI = 0.047, approximately 1.4× uniform, indicating broadly distributed visibility close to the uniform baseline.

Why HHI and not Gini or coefficient of variation? HHI has an interpretable uniform baseline (1/n); a multiple of uniform is the immediately readable diagnostic. The Gini coefficient and coefficient of variation describe distribution shape but lack the direct "multiple of uniform" interpretation.

Python:

```python
def hhi(contribution_shares):
    # contribution_shares: list of share_p values, summing to 1.0
    return sum(s ** 2 for s in contribution_shares)

def uniform_baseline(n):
    return 1.0 / n
```

Error quantification: confidence intervals

$r_e(p)$ is a binomial proportion over k trials. Per-engine MV is the mean of n such proportions. Analytic or bootstrap 95% confidence intervals are reported on per-engine and headline MV.

Benchmark publication condition. For published benchmark figures, the collection rule is n × k ≥ 150 observations per engine. At n = 30 and k = 5, this threshold is met exactly. The worst-case 95% CI half-width on a per-engine MV at this threshold is approximately ±8 points on the 0–100 scale.

Below threshold. n and k are tradeable above the k ≥ 3 floor: n = 50 at k = 3 also clears the n × k ≥ 150 threshold. Only scans feeding published benchmark figures must clear it; monitoring scans use k ≥ 3 throughout but may have lower n × k.

A lower bound on uncertainty. Binomial confidence intervals are a lower bound on real uncertainty, because within-window engine drift adds variance beyond the binomial sampling model. The control arm specified in the framework (§7.5 of the canonical paper) is the dedicated instrument for isolating that protocol-level variance.

Optional: Rogan-Gladen correction

When mention detection has known false-positive and false-negative rates (for example, an abbreviation that sometimes resolves to a different brand, or an entity-resolution rule that occasionally misses a valid mention), the observed mention rate can be corrected using the method of Rogan and Gladen (1978):

```
r_corrected = (r_observed + specificity - 1) / (sensitivity + specificity - 1)
```

where `sensitivity` is the probability that a true mention is correctly detected (true positive rate) and `specificity` is the probability that a non-mention is correctly identified as such (true negative rate). Both are estimated from manual audit of a random sample of responses.

Edge case. When `sensitivity + specificity < 1` (a worse-than-random detector), the formula produces values outside [0, 1]. The corrected estimate is clamped to [0, 1] and flagged in the reporting block. In practice this should not occur for brand name detection with reasonable entity resolution; its occurrence signals a misconfigured detection rule, not a regime the correction was designed to handle.

The uncorrected estimate is the default. Correction is documented when applied.

Worked example: full MV computation

Setup:

- Brand: Acme Analytics (synthetic)
- Prompt set: B2B-ANALYTICS-v2025-11-01, n = 30 prompts
- Engines: E = 5 (ChatGPT [GPT], Claude [CLD], Perplexity [PPX], Gemini [GEM], Microsoft Copilot [COP])
- Runs: k = 5 per prompt-engine pair (n × k = 150 per engine, meets benchmark threshold)
- Measurement period: 2026-06

Per-engine MV table:

Engine	MV_e	Reach	Intensity
GPT	62	87%	0.72
CLD	58	80%	0.73
PPX	46	67%	0.69
GEM	54	77%	0.70
COP	40	70%	0.57

Headline:

```
MV       = (62 + 58 + 46 + 54 + 40) / 5 = 52
Reach     = 26 / 30                      = 87%
Intensity = (0.72 + 0.73 + 0.69 + 0.70 + 0.57) / 5 = 0.68
HHI       = 0.047 (1.4× uniform baseline of 0.033)
```

The brand appears in an estimated 52% of category-relevant AI responses across the declared engines, with broad reach across the prompt set (87%) and mention rates well above zero on the prompts it reaches (mean 0.68). Concentration is close to the uniform baseline: visibility is broadly distributed, not narrow.

Part 2: MA (brand-mention alignment)

MA measures whether what AI systems say about the brand is accurate. It operates over the same prompts as MV, but it scores only AI responses that say something evaluable about the brand.

MA is a conditional measure. It conditions on the brand being mentioned. It does not condition on whether the mention is positive or negative.

Setup: the Brand Profile (F_B)

Before computing MA, the brand assembles a **Brand Profile**: a structured, version-controlled document listing the brand's declared facts. The Brand Profile is the ground truth against which AI responses are evaluated. It is held constant across measurement periods; any update creates a marker on the MA series.

The three tiers:

Tier	Nature	Examples	MA treatment
1	Externally verifiable facts	Founding year, company type, pricing entry point, primary product category	Full weight
2	Brand-stated facts (operationally authoritative)	Feature descriptions, use cases, named integrations	Full weight
3	Positioning claims	"Best for enterprise", "leading platform", "most intuitive"	Excluded from scoring

Tier 1 minimum (normative). A Brand Profile used for benchmark MA publication must contain at least three Tier 1 fields. Tier 1 fields are the only externally auditable component of MA's ground truth, since a Brand Profile without externally verifiable fields is unauditable and the MA score it produces is unanchored to any external reality.

Example Brand Profile (Acme Analytics, version ACME-v2025-11-01):

Field	Declared value	Tier
Primary category	B2B analytics and reporting software	1
Pricing entry point	$89/month	1
Key claim 1	Integrates with all major CRM platforms	2
Key claim 2	No-code dashboard builder	2
Disambiguation	Acme Analytics is not a CRM; it does not store contact data	2
Founding year	2019	1
Company type	Private	1

An AI response saying "Acme Analytics is a free CRM for enterprise teams starting at $0/month" would score 0 on every field it triggered. An AI response saying "Acme Analytics is a B2B analytics platform with no-code dashboards starting at $89/month" would score close to 1.0. The Brand Profile makes the scoring unambiguous.

Formula 2.1. Alignment score of a single response: $A(r, F_B)$

A score between 0 and 1 for one AI response: how accurately the response described the brand compared to the Brand Profile.

Intuition. A fact-checker reads an AI-generated paragraph about the brand, compares every factual claim to the Brand Profile, and assigns a score to each triggered field. Everything correct: 1.0. Everything wrong: 0.0. Partially outdated or imprecise: 0.5.

The formula:

$$A(r, F_B) = \frac{1}{|T|} \sum_{f \in T} \text{score}(f, r, F_B)$$

where T is the set of fields triggered by response r (fields the response makes claims about), and $\text{score}(f, r, F_B)$ is the rubric score for field f (0, 0.5, or 1.0) per the published rubric in Appendix C of the canonical paper.

Eligibility rule. A response is eligible for MA scoring only if $|T| \geq 2$ (it triggers at least two evaluable Brand Profile fields). Bare-name mentions in lists (responses that include the brand but say nothing about it) are excluded. This eligibility rule ensures near-zero-information observations do not distort the score.

Field types and grade definitions (summary): - **Binary fields** (e.g., company type): score 0 or 1. - **Thresholded fields** (e.g., pricing, founding year): score 0, 0.5, or 1.0 depending on tolerance bands (pricing: $\pm 5\% \rightarrow 1.0$, $\pm 15\% \rightarrow 0.5$, outside $\rightarrow 0$; founding year: exact $\rightarrow 1.0$, ± 2 years $\rightarrow 0.5$, otherwise $\rightarrow 0$). - **Graded fields** (e.g., primary category, key claims, disambiguation): score 0, 0.5, or 1.0 based on vocabulary match and contradiction.

Non-triggered fields are excluded from the computation. Silence about pricing is not misalignment; stating the wrong price is.

Handling repeated runs (k > 1). Each run of an engine on a prompt may produce a separate response; each response is scored individually, and the eligible scores are averaged within the mentioning cell. Let `R_{e,p}` be the set of eligible scored responses from engine e on prompt p (one cell, k runs). The per-cell per-engine alignment is:

```
A_{e,p} = (1 / |R_{e,p}|) × Σ_{r  R_{e,p}} A(r, F_B)
```

Worked example, p1 (GPT response from the canonical paper):

> "Acme Analytics is a B2B analytics platform offering no-code dashboards and CRM integrations, starting from around $120/month."

Field	Triggered	AI claim	Brand Profile	Score
Primary category	Yes	"B2B analytics platform"	"B2B analytics and reporting software"	1.0
Pricing entry point	Yes	"$120/month"	"$89/month"	0.0 (35% above ±15% tolerance)
CRM integration	Yes	"CRM integrations"	Confirmed	1.0
No-code dashboard	Yes	"no-code dashboards"	Confirmed	1.0
Disambiguation	Not triggered	N/A	N/A	excluded
Founding year	Not triggered	N/A	N/A	excluded
Company type	Not triggered	N/A	N/A	excluded

```
A(r, F_B) = (1.0 + 0.0 + 1.0 + 1.0) / 4 = 0.75
```

Worked example, p2 (CLD response from the canonical paper):

> "Acme Analytics provides no-code reporting tools for B2B teams, with plans starting at $89/month. It connects to major CRMs and was founded in 2019."

Field	Triggered	Score
Primary category	Yes	0.5 (partial; "no-code reporting tools for B2B teams" is broader than "analytics and reporting software")
Pricing entry point	Yes	1.0
CRM integration	Yes	1.0

Field	Triggered	Score
No-code dashboard	Yes	0.5 (implied via "no-code reporting" but not explicit dashboard)
Founding year	Yes	1.0

```
A(r, F_B) = (0.5 + 1.0 + 1.0 + 0.5 + 1.0) / 5 = 0.80
```

Python:

```python
def alignment_score(triggered_field_scores):
    # triggered_field_scores: list of 0, 0.5, or 1.0 values
    # for each Brand Profile field the response addressed
    if len(triggered_field_scores) < 2:
        return None  # response not eligible
    return sum(triggered_field_scores) / len(triggered_field_scores)
```

Formula 2.2. Per-engine MA

The mean alignment score across that engine's eligible mentioning cells on the full prompt set, expressed on a 0–100 scale.

The formula:

$$MA_e = 100 \times \frac{1}{|R_e|} \sum_{r \in R_e} A(r, F_B)$$

where R_e is the set of all eligible mentioning cells from engine e across all n prompts (one cell per prompt-engine pair, after within-cell averaging from Formula 2.1).

Eligible-cell floor. A per-engine MA based on fewer than 10 eligible mentioning cells is excluded from benchmark tables, rankings, and headline aggregation. It may appear in full diagnostic reporting, flagged as indicative, with eligible count and confidence interval. At N = 10 and realistic score dispersion (SD 0.25), the 95% CI half-width is approximately ±15 points on the 0–100 scale: wide but interpretable. Below 10, the interval spans most of the scale.

The floor sits at the conservative end of small-cell suppression practice in official statistics. The value 10 is a working assumption pending calibration.

Intuition. For each engine, take the alignment scores of its eligible mentioning cells, average them, multiply by 100. An MA_e of 71 means engine e's descriptions of the brand align with the Brand Profile 71% of the time on average.

Worked example. For Acme Analytics across 30 prompts, engine GPT produced 16 eligible mentioning cells (out of 26 mentioning cells; the other 10 mentioning cells were bare-name list inclusions that did not trigger two Brand Profile fields). The mean cell-level alignment score across those 16 eligible cells is 0.71, so $MA_{GPT} = 71$.

Engine	Eligible cells	Per-engine MA
GPT	16	71
CLD	15	76
PPX	12	63
GEM	16	68

Engine	Eligible cells	Per-engine MA
COP	11	59

All five engines clear the 10-cell floor.

Python:

```python
def per_engine_ma(eligible_scores):
    # eligible_scores: list of A(r, F_B) values from one engine
    # across the prompt set (one entry per eligible mentioning cell)
    if len(eligible_scores) < 10:
        return None  # below floor; not admitted to headline
    return 100 * sum(eligible_scores) / len(eligible_scores)
```

Formula 2.3. Headline MA

The equal-weighted mean of per-engine MA values across engines that clear the eligible-cell floor.

The formula:

$$MA = \frac{1}{|E^*|} \sum_{e \in E^*} MA_e$$

where $E^* \subseteq E$ is the subset of engines clearing the eligible-cell floor. This mirrors the per-engine-first aggregation used for MV: the headline is the arithmetic mean of the per-engine scores, no engine is privileged, and the headline reconciles exactly with its parts.

Worked example.

```
MA = (71 + 76 + 63 + 68 + 59) / 5 = 337 / 5 = 67
```

A headline MA of 67 means the average eligible mentioning cell receives an alignment score of 67/100 against the Brand Profile.

Python:

```python
def headline_ma(per_engine_mas):
    # per_engine_mas: list of MA_e values for engines clearing the floor
    return sum(per_engine_mas) / len(per_engine_mas)
```

Coverage: a first-order diagnostic

Coverage is the proportion of mentioning cells that are eligible for MA scoring.

$$\text{coverage} = \frac{\text{eligible mentioning cells}}{\text{total mentioning cells}}$$

Worked example. Acme Analytics has 114 mentioning cells in total (the sum across engines: 26 GPT + 24 CLD + 20 PPX + 23 GEM + 21 COP). Of those, 70 mentioning cells were eligible (triggering at least two Brand Profile fields). Coverage = 70 / 114 = **61%**.

Why coverage is a first-order diagnostic, not a footnote. If most mentions are bare-name list inclusions, the brand is named but never described in AI responses. This is a distinct representational state with its own remediation path: the brand needs describable, citable claims in circulation. Low coverage means a substantial portion of AI mentions are passing references, useful for visibility (counts toward MV) but uninformative for alignment.

Caution threshold. Coverage below 50% is treated as a caution condition: the MA score is reported, but its interpretation should note that a majority of mentions are bare-name or list mentions without substantive description. The score is informative for the eligible mentions; it is not informative for the bare-name share.

Cross-family evaluator and shared pre-training bias

The evaluator scoring $A(r, F_B)$ must be an LLM from a different model family than the engine that produced the response. The technical reason is shared pre-training bias: an evaluator from the same model family may replicate identical training-data errors as the evaluated engine, so a response containing a training-data-derived misalignment could be scored as aligned by an evaluator that holds the same misalignment, artificially inflating the score.

Cross-family selection reduces but does not eliminate the residual risk that overlapping pre-training corpora across different model families share some errors. The human calibration protocol addresses this residual: a stratified gold-standard sample double-coded by humans and adjudicated, with the LLM evaluator validated against the adjudicated labels at Cohen's kappa or Krippendorff's $\alpha \geq 0.75$ and LLM-human correlation r ≥ 0.80.

When the standard engine list contains an engine from the same family as the evaluator (a structural condition, not an edge case), responses from that engine are scored by a designated second evaluator from a different family. Same-family scoring is permitted only as a documented exception, with explicit flagging in the reporting conditions block and an elevated human-audit sample.

The production evaluator runs as an ensemble: median of three independent evaluations per response. The evaluator model version is pinned and documented in all published reporting; an unpinned evaluator silently recalibrates and is prohibited by the specification.

Part 3: MS (brand-mention sentiment)

MS measures whether AI responses speak positively or negatively about the brand. It operates over the same mention stream as MA but with a different eligibility criterion and a different output structure.

MS is the second conditional measure: it conditions on the brand being mentioned, and does not condition on whether the description was accurate.

Setup: brand-directed sentiment, not text polarity

The unit of analysis is the sentiment expressed toward the brand, not the overall tone of the text in which the brand appears.

The critical case. Consider this response:

> "Most analytics tools in this category are bloated and overpriced for small teams. Acme Analytics is the exception: it keeps the interface lean and the entry price is reasonable."

The text is broadly critical (it criticises "most tools"), but the brand-directed sentiment is positive: Acme Analytics is specifically distinguished from the negatively described category norm. The correct MS classification is **positive**.

An evaluator trained on text-level sentiment will mis-score this case as negative. This is the single most common failure mode in brand-directed sentiment evaluation. The instruction to the evaluator must state explicitly: classify the sentiment expressed toward the named brand, not the sentiment of the passage overall. The published evaluator prompt in Appendix E of the canonical paper makes this distinction operational.

The closest technical literatures are **target-dependent sentiment analysis** and **stance detection**: perspective or opinion directed toward a specific target entity, with stance and sentiment treated as separable dimensions of evaluation against the target. These are adjacent constructs rather than synonyms for MS; the

technical terms are included here for implementer reference, and the canonical paper uses "brand-directed sentiment" in reader-facing sections.

Eligibility for MS evaluation

A response is **eligible for MS** if it contains substantive evaluative characterisation of the brand beyond bare-name inclusion. A response listing the brand in a set ("top tools include X, Y, Acme, and Z") with no evaluative content is not eligible.

The MS eligibility criterion may yield a different count than the MA criterion. MA requires at least two triggerable Brand Profile fields; MS requires any evaluative content. A response that says "Acme Analytics is a good choice for small teams" is MS-eligible (evaluative) but might not be MA-eligible (no specific fields triggered). A response that says "Acme Analytics has CRM integrations and was founded in 2019" is MA-eligible (two fields triggered) but not MS-eligible (no evaluative characterisation).

In the Acme Analytics worked example the two eligible counts both equal 70, but this is a property of the synthetic dataset, not a general feature of the framework.

Formula 3.1. Per-engine shares

For each engine, the percentage of eligible cells classified positive, neutral, and negative.

Let `N_e` = number of eligible mentioning cells from engine e. Let `pos_e`, `neu_e`, `neg_e` be the counts of cells classified positive, neutral, and negative respectively (`pos_e + neu_e + neg_e = N_e`).

The formulas:

$$\%\text{pos}_e = 100 \times \frac{\text{pos}_e}{N_e}, \quad \%\text{neu}_e = 100 \times \frac{\text{neu}_e}{N_e}, \quad \%\text{neg}_e = 100 \times \frac{\text{neg}_e}{N_e}$$

Each set sums to 100% per engine.

Eligible-cell floor. The same floor as MA applies: a per-engine MS based on fewer than 10 eligible mentioning cells is excluded from benchmark tables and headline aggregation. At N = 10, shares move in 10-point steps, which the mandatory confidence intervals make visible.

Worked example. Engine GPT, N = 16 eligible mentioning cells. 8 are classified positive, 4 neutral, 4 negative.

```
%pos_GPT = 100 × 8/16 = 50%
%neu_GPT = 100 × 4/16 = 25%
%neg_GPT = 100 × 4/16 = 25%
```

The full per-engine table for Acme Analytics:

Engine	Eligible	%pos	%neu	%neg
GPT	16	50%	25%	25%
CLD	15	60%	20%	20%
PPX	12	42%	25%	33%
GEM	16	50%	25%	25%
COP	11	55%	18%	27%

Python:

```python
def per_engine_shares(classifications):
    # classifications: list of 'positive' | 'neutral' | 'negative'
    # for one engine's eligible mentioning cells
    n = len(classifications)
    if n < 10:
        return None
    pos = sum(1 for c in classifications if c == 'positive')
    neu = sum(1 for c in classifications if c == 'neutral')
    neg = sum(1 for c in classifications if c == 'negative')
    return {
        'pos_pct': 100 * pos / n,
        'neu_pct': 100 * neu / n,
        'neg_pct': 100 * neg / n,
    }
```

Formula 3.2. Per-engine net sentiment

The headline figure for a single engine, computed from that engine's shares.

The formula:

$$\text{net}_e = \%\text{pos}_e - \%\text{neg}_e$$

Range. -100 to $+100$. A net of $+100$ means every eligible mentioning cell was positive. A net of -100 means every eligible mentioning cell was negative. A net of 0 may arise from many distributional shapes; it is the shares table that tells the practitioner what the net hides (see §6.3 of the canonical paper).

Worked example.

```
net_GPT = 50 - 25 = +25
net_CLD = 60 - 20 = +40
net_PPX = 42 - 33 = +9
net_GEM = 50 - 25 = +25
net_COP = 55 - 27 = +28
```

Python:

```python
def per_engine_net(shares):
    return shares['pos_pct'] - shares['neg_pct']
```

Formula 3.3. Headline shares and headline net

The equal-weighted mean of per-engine shares (and per-engine net) over engines clearing the eligible-cell floor.

The formulas:

$$\%\text{pos} = \frac{1}{|E^*|}\sum_{e\in E^*}\%\text{pos}_e \qquad \%\text{neu} = \frac{1}{|E^*|}\sum_{e\in E^*}\%\text{neu}_e \qquad \%\text{neg} = \frac{1}{|E^*|}\sum_{e\in E^*}\%\text{neg}_e$$

$$\text{net} = \frac{1}{|E^*|}\sum_{e\in E^*}\text{net}_e$$

The per-engine-first aggregation mirrors MV and MA. Equivalently, since $\text{net}_e = \%\text{pos}_e - \%\text{neg}_e$, the headline net equals the headline positive share minus the headline negative share, so the four values reconcile.

Worked example.

```
Headline %pos = (50 + 60 + 42 + 50 + 55) / 5 = 51%
Headline %neu = (25 + 20 + 25 + 25 + 18) / 5 = 23% (after rounding)
Headline %neg = (25 + 20 + 33 + 25 + 27) / 5 = 26% (after rounding)

Headline net  = (25 + 40 + 9 + 25 + 28) / 5 = +25
```

A net of +25 means that, across the declared engines, positive characterisations of the brand outnumber negative ones by 25 points of share.

Python:

```python
def headline_ms(per_engine_shares_list):
    # per_engine_shares_list: list of share dicts from engines
    # clearing the eligible-cell floor
    n = len(per_engine_shares_list)
    pos = sum(s['pos_pct'] for s in per_engine_shares_list) / n
    neu = sum(s['neu_pct'] for s in per_engine_shares_list) / n
    neg = sum(s['neg_pct'] for s in per_engine_shares_list) / n
    return {
        'pos_pct': pos, 'neu_pct': neu, 'neg_pct': neg,
        'net': pos - neg,
    }
```

Display bands: a presentation convention

The featured figure encodes MS as point colour using four display bands, evaluated in this precedence order:

1. **Mixed:** positive share $\geq 25\%$ AND negative share $\geq 25\%$
2. **Positive:** net $\geq +20$
3. **Negative:** net ≤ -20
4. **Neutral:** otherwise

These are presentation conventions for figure colour encoding. They are not metric parameters and do not enter any computation. The underlying shares and net remain continuous.

Why precedence matters. For Acme Analytics, headline shares are 51% positive / 23% neutral / 26% negative, with net +25. The Mixed condition triggers because both positive and negative shares clear 25%, even though the net is above zero. Mixed takes precedence over Positive: the precedence rule encodes the practitioner reading that "polarised coverage netting positive" is a different brand situation than "uniformly positive coverage", and warrants different attention.

The ±20 threshold. The Positive and Negative bands trigger at net $\geq +20$ and net ≤ -20. The threshold exceeds the approximate ±15 CI half-width at the eligible-cell floor (10 eligible mentioning cells, SD 0.25), so a colour assignment is unlikely to be a pure sampling artefact at the floor.

Mixed band reliability at the floor. At 10 eligible mentioning cells, a 30% share is 3 cells, so the Mixed trigger (25% $\geq$ both pos and neg) is sensitive to single-cell movements. Mixed band classifications drawn at the floor require either elevated k or human review before downstream use.

Worked example: full MS computation

70 eligible mentioning cells across five engines.

Engine	Eligible	Positive	Neutral	Negative	Net
GPT	16	8 (50%)	4 (25%)	4 (25%)	+25
CLD	15	9 (60%)	3 (20%)	3 (20%)	+40

Engine	Eligible	Positive	Neutral	Negative	Net
PPX	12	5 (42%)	3 (25%)	4 (33%)	+9
GEM	16	8 (50%)	4 (25%)	4 (25%)	+25
COP	11	6 (55%)	2 (18%)	3 (27%)	+28

Headline shares: 51% positive / 23% neutral / 26% negative. Headline net: (25 + 40 + 9 + 25 + 28) / 5 = **+25**. Display band: **Mixed** (positive 51% $\geq$ 25% and negative 26% $\geq$ 25%).

Sentiment drivers (from the canonical paper's worked example): positive characterisations highlight integration breadth and ease of use. Negative characterisations focus on pricing uncertainty (AI responses stating the wrong price, which then read as "expensive relative to alternatives"). Because the negative sentiment tracks the pricing misalignment identified in MA, fixing the price facts in cited third-party sources is likely to improve both MA and MS simultaneously.

Part 4: Full computation pipeline

```
INPUT: P (n prompts), E engines, k runs per engine per prompt,
       brand B, Brand Profile F_B, evaluator model M

-----------------------------------------------------------------
STEP 1: Data collection
  For each prompt p  P, each engine e  E, each run s = 1..k:
    Query engine e with prompt p (run s of k).
    Record c(p, e, s)  {0, 1}  // 1 if brand B was mentioned
    Record response text r(p, e, s) if mentioned.

STEP 2: Compute MV
  a) For each engine e, each prompt p:
     r_e(p) = (1/k) × Σ_s c(p, e, s)                  [Formula 1.1]
  b) For each engine e:
     MV_e = 100 × (1/n) × Σ_p r_e(p)                 [Formula 1.2]
  c) Headline:
     MV = (1/E) × Σ_e MV_e                           [Formula 1.3]

STEP 3: Companion statistics for MV
  reach_e        = |{ p : r_e(p) > 0 }| / n
  reach          = |{ p : any r_e(p) > 0 }| / n
  intensity_e    = mean of r_e(p) over { p : r_e(p) > 0 }
  intensity      = (1/E) × Σ_e intensity_e
  share_p        = (Σ_e Σ_s c(p, e, s)) / total_mentions
  HHI            = Σ_p share_p²

STEP 4: Compute MA
  a) For each mentioning response r in the mention stream:
       Determine triggered fields T from response r.
       If |T| < 2: response is INELIGIBLE; skip.
       Else: compute A(r, F_B) using the rubric.   [Formula 2.1]
  b) (If k > 1) Average within engine-prompt:
       A_{e,p} = mean of A(r, F_B) across eligible runs of e on p
  c) For each engine e:
       N_e = number of eligible mentioning cells from engine e
       If N_e < 10: engine excluded from headline (flag in report).
       Else: MA_e = 100 × mean of A_{e,p}            [Formula 2.2]
  d) Headline:
```

```
      MA = (1 / |E*|) × Σ_{e  E*} MA_e            [Formula 2.3]
      where E* is the set of engines clearing the floor.
  e) Coverage:
      coverage = (total eligible mentioning cells) / (total mentioning cells)

STEP 5: Compute MS
  a) For each mentioning response r in the mention stream:
      Determine MS eligibility (substantive evaluative
      characterisation beyond bare-name inclusion).
      If eligible: classify as positive | neutral | negative
      per Appendix D codebook of the canonical paper.
  b) For each engine e:
      N_e' = number of MS-eligible mentioning cells from engine e
      If N_e' < 10: engine excluded from headline (flag in report).
      Else: compute %pos_e, %neu_e, %neg_e            [Formula 3.1]
            net_e = %pos_e - %neg_e                   [Formula 3.2]
  c) Headline shares and net:
      %pos, %neu, %neg = equal-weighted mean of per-engine shares
      net = equal-weighted mean of per-engine nets   [Formula 3.3]
  d) Display band:
      if %pos $\geq$ 25 and %neg $\geq$ 25: Mixed
      elif net $\geq$ +20: Positive
      elif net $\leq$ -20: Negative
      else: Neutral

-----------------------------------------------------------------
STEP 6: Reporting conditions block (mandatory)
  - Engine list (three-character codes, alphabetical, hyphen-joined)
  - Prompt set version identifier
  - Measurement period (year-month)
  - n, k
  - Evaluator model and version (for MA and MS)
  - Eligible response counts and coverage statistics
  - Any same-family evaluator exception
  - Any Rogan-Gladen correction applied

-----------------------------------------------------------------
OUTPUT:
  MV (0-100) + per-engine MV table + reach + intensity + HHI
  MA (0-100) + per-engine MA table + coverage
  MS net (-100 to +100) + shares + per-engine table + display band

  Example block (from the worked example):
  "MV 52 / MA 67 / MS +25 (CLD-COP-GEM-GPT-PPX,
   B2B-ANALYTICS-v2025-11-01, 2026-06, n=30, k=5, evaluator: claude-opus-4-6,
   eligible MA=70, coverage=61%, eligible MS=70)"
```

The reporting conditions block is part of the result, not a footer. A score without its conditions cannot be compared to another score: two studies with different engine lists, different prompt-set versions, or different evaluators are measuring different things even if both report a number called MV.

Part 5: The Visibility–Alignment quadrant with MS overlay

The Visibility–Alignment map places every brand in one of four positions, with sentiment shown as point colour using the display bands. MA sits on the horizontal axis and MV on the vertical, because MV is the more consequential dimension, so the gradient reads top-to-bottom for visibility and left-to-right for alignment. Quadrant names are factual (position only) and make no claim about sentiment. In the table

below, *Low MA* means inaccurate description and *High MA* means accurate; *Low MV* means invisible and *High MV* means visible.

	Low MA	**High MA**
High MV	**Visible & Misaligned**	**Visible & Aligned**
Low MV	**Unseen & Misaligned**	**Aligned but Unseen**

Brand-representation states. Each positional quadrant combines with the sentiment band to give a named brand-representation state. The state name has two parts. A *reception root* names where the brand stands when an answer engine assembles candidates: AI Champion for positive reception (engine advocates), AI Contender for neutral (consideration set without advocacy), AI Wildcard for mixed (dividing opinion), AI Pariah for negative (characterised unfavourably). A *position prefix* names how the brand's standing departs from the visible-and-aligned ideal, with one prefix per quadrant: none, Undiscovered, Misrepresented, or Misrepresented & Undiscovered. The reception root is determined solely by the MS display band, one root per band: Positive → AI Champion, Neutral → AI Contender, Mixed → AI Wildcard, Negative → AI Pariah. Sixteen named states result (four quadrants × four roots).

Position quadrant	Prefix	Positive	Neutral	Mixed	Negative
Visible & Aligned	*(none)*	AI Champion	AI Contender	AI Wildcard	AI Pariah
Aligned but Unseen	Undiscovered	Undiscovered AI Champion	Undiscovered AI Contender	Undiscovered AI Wildcard	Undiscovered AI Pariah
Visible & Misaligned	Misrepresented	Misrepresented AI Champion	Misrepresented AI Contender	Misrepresented AI Wildcard	Misrepresented AI Pariah
Unseen & Misaligned	Misrepresented & Undiscovered	Misrepresented & Undiscovered AI Champion	Misrepresented & Undiscovered AI Contender	Misrepresented & Undiscovered AI Wildcard	Misrepresented & Undiscovered AI Pariah

From scores to descriptor. Two lookups give the descriptor: position (MV and MA each read against the 50 boundary) sets the prefix, sentiment (the MS band) sets the reception root, and the descriptor is prefix plus root. Acme Analytics (MV 52 visible, MA 67 aligned, shares 51/23/26 → Mixed band) resolves to **AI Wildcard**, a brand that is visible, accurately described, and received with divided opinion.

What each quadrant means in practice:

Visible & Aligned (high MV, high MA). AI mentions the brand often and describes it accurately. AI Champion is the target state: maintain prompt-set coverage and keep the Brand Profile current. AI Contender (neutral reception) signals an accurate, visible brand the engine carries without advocacy; the work is content that earns advocacy. AI Wildcard signals a divided reception over an accurate baseline; AI Pariah signals visible accurate description with negative reception. The three off-Champion states need positioning and sentiment-driver work, not fact correction.

Visible & Misaligned (high MV, low MA). AI mentions the brand frequently but gets the facts wrong. A Misrepresented AI Pariah is the commercially exposed state: high reach, inaccurate descriptions, adverse reception that often tracks the misrepresentation. The field-level MA report identifies which facts AI gets wrong; fixing the cited sources that carry the misaligned claims is the primary action, and the negative reception frequently resolves with the facts. A Misrepresented AI Wildcard sits in the same quadrant with polarised reception; the same alignment-first sequence applies.

Aligned but Unseen (low MV, high MA). AI describes the brand accurately when it mentions it, but rarely does so. The information quality is in place; the reach is not. An Undiscovered AI Champion can

scale visibility safely; an Undiscovered AI Contender likewise (neutral reception, no sentiment blocker). An Undiscovered AI Wildcard or Undiscovered AI Pariah signals that scaling visibility without first resolving the reception drivers amplifies the negative coverage.

Unseen & Misaligned (low MV, low MA). Rarely mentioned and inaccurately described when mentioned. Both dimensions need improvement; reach is usually the first priority because there is limited value in correcting accuracy for prompts where the brand almost never appears. The framework's motivation for measuring MA separately from MV is that a brand that becomes visible while remaining inaccurate has merely converted one problem into another.

The sentiment overlay.

MS is encoded as point colour across all four quadrants. The four bands (positive, neutral, negative, mixed) sit on the dot and tell the practitioner the reception layer that the position alone cannot. Neutral and mixed are distinct bands and are not merged: neutral indicates absent characterisation; mixed indicates co-present praise and criticism (both shares $\geq 25\%$). A high MV with mixed sentiment carries different commercial risk than a high MV with uniformly positive sentiment, even though both might net to the same number; the mixed band surfaces what net alone hides.

Insufficient data and AI Absent. A brand that clears the eligible-cell floor (10 eligible mentioning cells per engine, see Formula 2.2) on both conditional measures resolves to one of the sixteen brand-representation states. A brand below the floor on alignment, on sentiment, or on both carries insufficient data for the affected measure. It takes no state label, since the position requires a readable MA and the reception root a readable MS, and is reported by its readable metrics with the shortfall flagged, in any quadrant. The shortfall is per-measure: a brand may have readable MA but insufficient MS, or the reverse, or both insufficient. A brand with MV = 0 produces no mentioning cells. MA and MS are undefined, and the brand is reported as **AI Absent**. AI Absent is not a reception root, carries no prefix, and sits off the teaching grid.

Separability, not independence.

MV, MA and MS are separable measures: each addresses a distinct property, and all four combinations of accurate/inaccurate and positive/negative are live brand states. Separability is a property of the constructs. It does not assume the measures are statistically independent in field data; the same content ecosystem can drive all three, so they may correlate. The framework's incremental-validity tests (in the canonical paper §9.3) measure that empirical redundancy: if $\rho(\text{MV}, \text{MA}) > 0.80$, MA's incremental diagnostic value over MV is limited and the framework's recommendation revises accordingly; the same test exists for MS regressed on (MV, MA).

Quadrant boundaries. Boundaries at MV = 50 and MA = 50 are presentation conventions. They are declared in figure notes and labelled as such. They are not metric parameters and do not enter any computation; the underlying scores remain continuous.

Further reading

The canonical paper, *Brand Representation in AI Answers: An Open Specification for Measuring Visibility, Alignment and Sentiment* (Khan, 2026, Aiviara Research), contains:

- The full formal specification (§§4–6)
- The Brand Profile and three-tier claim model (§5.2)
- The evaluator architecture, including the determinism-tested control arm (§7)
- The governance and comparability rules (§8)
- The published MA scoring rubric (Appendix C)
- The published MS sentiment codebook with the worked counter-example (Appendix D)
- The published evaluator prompt templates (Appendix E)
- The pre-stated falsifiability criteria and proposed validation study (§9)
- The end-to-end worked example computation (Appendix B)

The full per-prompt per-engine mention rates for the worked example are published as supplementary data (`gbmf-worked-example-mention-rates.csv`, 30 prompts × 5 engines). The MV-related headline figures

in this primer reproduce from that dataset; MA and MS figures are illustrative unless the response-level evaluator output is also released.

Aiviara Research is a research and publishing initiative. For more information visit aiviara.com/research.