

# 1) High-level steps ( $\leq 1$ page)

## 1. Bronze: Ingest & standardize

- Start from your existing Databricks table; keep rows where `comment is not null` (and ideally not blank).
- Store standardized fields (`survey_key`, `response_ts`, `response_date`, `visn`, `facility_id`, `comment`, etc.).

## 2. Bronze: Preprocess flags

- Compute deterministic flags (empty, NA-like, punctuation-only).
- Set `is_processable` to skip junk before LLM.

## 3. Silver: LLM extraction (1 comment per call)

- For each processable comment, call GPT-4o with the locked prompt.
- Store raw JSON per `survey_key` (audit trail). If skipped  $\rightarrow$  `{"topics":[]}`.

## 4. Silver: Explode JSON $\rightarrow$ theme rows

- Explode `topics[]` and `themes[]` to 1 row per theme mention.
- Create a stable **theme row key** (since one comment  $\rightarrow$  multiple themes).

## 5. Silver: Filter to actionable + enrich

- Apply fixed quality thresholds.
- Add deterministic `sentiment_score`.
- (Optionally combine with embedding columns initially null.)

## 6. Silver: Embeddings

- Compute embeddings for `theme_text` (sentence-transformers).

- Store vectors (in same Silver enriched table or separate, your choice).

## 7. Gold: Canonicalization

- For each **topic**, cluster theme embeddings using **HDBSCAN**.
- Create `canonical_theme_id` and `canonical_theme_label`.
- Map each theme row to a canonical theme.

## 8. Gold: Cadence-agnostic metrics + signals

- Join `response_date` to `ref_dim_calendar` to support iso-week/biweek/month/quarter/etc.
- Aggregate to `(time_grain, period_id, canonical_theme_id)` metrics.
- Compute: Top concerns, compliments, emerging, trends, spikes, outliers.
- Produce a “leadership-ready” **signals table** + example snippets.

---

## 2) JSON Response Format (LOCKED)

```
{
  "topics": [
    {
      "topic": "Appointment Scheduling",
      "topic_type": "standard",
      "themes": [
        {
          "theme_text": "appointment was canceled twice",
          "sentiment": "very_negative",
          "sentiment_confidence": 0.97,
          "theme_confidence": 0.90
        },
        {
          "theme_text": "rescheduled three weeks later",
          "sentiment": "negative",
          "sentiment_confidence": 0.94,
```

```
        "theme_confidence": 0.86
      }
    ]
  }
]
```

## Rules

- Max **3 topics** per comment
  - `topic_type`  $\in$  {standard, new}
  - Max **3 themes per topic**
  - Sentiment labels: {very\_negative, negative, neutral, positive, very\_positive}
  - Confidence scores: 0–1
  - If no actionable content: {"topics":[]}
  - No topic-level sentiment/confidence
- 

## 3) Prompt (copy/paste)

SYSTEM:

You are an expert analyst for the U.S. Department of Veterans Affairs (VA), specializing in veteran voice-of-the-customer survey comment analysis. Your task is to extract clear, actionable topics and themes from exactly ONE veteran free-text survey comment. Be faithful to the text. Do not invent issues that are not stated or strongly implied. This output is used for recurring (weekly/monthly/quarterly) operational reporting and trend detection.

USER:

Analyze exactly ONE VA survey comment below and extract up to 3 topics and up to 3 themes per topic.  
Return ONLY valid JSON in the exact schema provided.  
If more than one comment is provided, return {"topics": []}.

VA SURVEY COMMENT:

```
""  
  
{{comment}}  
""
```

REQUIRED JSON SCHEMA:

```
{  
  "topics": [  
    {  
      "topic": "<string>",  
      "topic_type": "<standard|new>",  
      "themes": [  
        {  
          "theme_text": "<string>",  
          "sentiment":  
            "<very_negative|negative|neutral|positive|very_positive>",  
          "sentiment_confidence": <number 0.0-1.0>,  
          "theme_confidence": <number 0.0-1.0>  
        }  
      ]  
    }  
  ]  
}
```

RULES:

- Max 3 topics per comment
- Prefer standard topics; use topic\_type="new" only if needed
- Max 3 themes per topic
- Themes must be short (3-12 words), specific, and grounded in the text
- Use full confidence range (0-1)
- If no actionable content, return {"topics":[]}
- Return only JSON; no extra keys

#### STANDARD TOPICS:

Appointment Scheduling, Wait Time, Access / Getting Care,  
Communication (Calls, Messages),  
Staff Courtesy / Respect, Provider Interaction / Bedside Manner,  
Care Quality / Clinical Experience, Pharmacy / Medications,  
Billing / Benefits / Eligibility, Referrals / Specialty Care,  
Community Care, Facility / Environment,  
Technology / Portal / My HealtheVet,  
Transportation / Parking, Mental Health,  
Pain Management, Follow-up / Care Coordination,  
Test Results / Lab / Imaging,  
Discharge / After Visit Summary,  
Emergency / Urgent Care, Other

---

## 4) Detailed steps for each step (engineering-ready)

### Bronze

#### B0 — `bronze_comments`

- Input: your existing Databricks table.
- Filter: `comment IS NOT NULL` (recommend also `trim(comment) <> ''`).
- Keep: `survey_key, response_ts, response_date, visn, facility_id, comment, survey_name, department='VA'`.

#### B1 — `bronze_comment_preprocess`

- Compute flags:
  - `is_empty`: blank after trim
  - `is_na_like`: matches `n/a, none, no comment`, etc.
  - `is_punct_only`: punctuation/emojis only

- `is_processable = NOT(is_empty OR is_na_like OR is_punct_only)`
  - Persist `skip_reason`.
- 

## Silver

### S1 — `silver_llm_output`

- For each `survey_key`:
  - If `is_processable=false` → store `{"topics":[]}`, `llm_status=SKIPPED`.
  - Else call GPT-4o using the prompt and store raw JSON + metadata (`model_name`, `prompt_version`, `run_ts`).

### S2 — `silver_theme_fact_raw` (explode)

- Parse JSON.
- Explode `topics[ ]` and each topic's `themes[ ]`.
- Add `topic_rank` (1..3) and `theme_rank` (1..3) from array positions.
- Create stable theme row key:
  - `theme_row_id = hash(survey_name, survey_key, topic, theme_rank, prompt_version)`
  - (This prevents collisions because 1 comment can create multiple themes.)

### S3 — `silver_theme_fact_filtered` (quality gate)

- Filter:
  - `theme_confidence >= 0.65`

- `sentiment_confidence >= 0.70`
- Add deterministic `sentiment_score`:
  - `very_negative=-2, negative=-1, neutral=0, positive=+1, very_positive=+2`

#### S4 — embeddings

- Compute embedding for `theme_text` using sentence-transformers.
  - Practical design:
    - Either separate table `silver_theme_embeddings`
    - Or same table as S3 with nullable columns populated later:
      - `embedding_model, embedding_vector, embedding_ts`
  - Embedding job updates rows where `embedding_vector IS NULL`.
- 

### Gold

#### G0 — `ref_dim_calendar` (Gold/reference)

- One row per date.
- Provides `iso_week_id, iso_biweek_id, semimonth_id, month_id, quarter_id` (+ optional fiscal).

#### G1 — canonicalization (HDBSCAN per topic)

- For each topic:
  - cluster embeddings with HDBSCAN
  - assign `canonical_theme_id`

- canonical label = medoid theme\_text or summarized cluster label
- Output:
  - gold\_canonical\_theme\_dim
  - gold\_theme\_fact\_canonical (filtered fact + canonical id)

## G2 — cadence-agnostic metrics

- Join response\_date → ref\_dim\_calendar.
- Aggregate per (time\_grain, period\_id, canonical\_theme\_id) into gold\_period\_theme\_metrics.

## G3 — leadership signals

- Using volume-aware thresholds per period, produce:
  - gold\_period\_smoke\_signals (Top concerns, compliments, emerging, spikes, trends, outliers)
  - gold\_period\_theme\_examples (3–5 representative snippets per top theme)

---

# 5) How to compute Top N, Emerging, Trend, Spike, Outliers + leadership view

## Shared definitions

For a chosen cadence (e.g., time\_grain='iso\_week', period\_id='2026-W02'):

- N\_total = total comments in period
- N\_negative = distinct comments having ≥1 negative theme
- N\_positive = distinct comments having ≥1 positive theme

- For theme T:  $N\_theme(T) = \text{distinct comments with canonical\_theme\_id}=T$

### Fixed quality thresholds (Silver)

- $theme\_confidence \geq 0.65$
- $sentiment\_confidence \geq 0.70$

### Volume-aware thresholds (Gold) — works for weekly/monthly/quarterly/multi-survey

- $MIN\_COUNT(period) = \max(5, \text{round}(0.003 \times N\_total))$
- $MIN\_COUNT\_EMERGING(period) = \max(5, \text{round}(0.002 \times N\_negative))$
- For VISN/facility slice:  $MIN\_COUNT\_OUTLIER(slice) = \max(5, \text{round}(0.01 \times N\_slice))$

### A) Top N Concerns (fires now)

Filter negative themes:

- $sentiment\_score < 0$   
Eligibility:
- $N\_theme(T) \geq MIN\_COUNT(period)$

Rank by:

$ConcernScore(T) =$   
 $N\_theme(T)$   
 $\times \text{avg}(|sentiment\_score|)$   
 $\times \text{avg}(sentiment\_confidence)$   
 $\times \text{avg}(theme\_confidence)$

Output: Top 5 (or Top N)

---

## B) Top N Compliments

Filter positive themes:

- `sentiment_score > 0`  
Eligibility:
- `N_theme(T) >= MIN_COUNT(period)`

Rank by:

```
PraiseScore(T) =  
N_theme(T)  
× avg(sentiment_score)  
× avg(sentiment_confidence)  
× avg(theme_confidence)
```

---

## C) Emerging themes (smoke before fire)

Not the same as top concerns. Smaller volume but fast growth.

Filter negative themes and require:

- `N_theme(T) >= MIN_COUNT_EMERGING(period)`
  - `Rate(T) = N_theme(T)/N_negative >= 0.003` (0.3% of negative)
  - `rate_pct_change >= 0.30` ( $\geq +30\%$  vs prior period)
  - `avg_sent_conf >= 0.80` and `avg_theme_conf >= 0.70`
  - `visn_count >= 2` (helps avoid single-site noise)
-

## D) Trend (slow burn)

Sustained change over K periods (4–8):

Eligibility:

- $N\_theme(T) \geq MIN\_COUNT(period)$  for most of the K windows

Flag:

- consistent increase, or slope > threshold, or cumulative growth  $\geq 40\text{--}50\%$
- 

## E) Spike (sudden event)

Eligibility:

- $N\_theme(T) \geq MIN\_COUNT\_EMERGING(period)$

Flag:

- $Rate(T, current) \geq 2 \times Rate(T, previous)$  OR z-score  $\geq 3$
- 

## F) Outliers (VISN / facility only — why not “all”?)

Outliers require **peer comparison**. Global anomalies are handled by trend/spike/emerging.

Compute per slice S (VISN or facility):

- $SliceRate(T, S) = N\_theme(T, S) / N\_negative(S)$

Eligibility:

- $N\_theme(T, S) \geq MIN\_COUNT\_OUTLIER(S)$

Flag outlier:

- $\text{SliceRate} > \text{mean}(\text{peer rates}) + 3 \times \text{std}$  (or robust MAD)

---

## What leadership sees (example layout)

Period: 2026-W02 (ISO Week)

### Top 5 Concerns

- Appointment Scheduling → “Clinic canceled appointments” (6.2% of negative; +12% WoW; 12 VISNs)

### Top 5 Compliments

- Staff Courtesy → “Staff were kind/respectful” (4.8% of positive; 15 VISNs)

### Emerging (Watch List)

- Communication → “No callback after voicemail” (0.7% of negative; +80% WoW; 6 VISNs)

### Spikes

- Technology → “Portal login outage” (+220% WoW)

### Outliers

- VISN 6: Wait Time → “Phone hold time” 3× system average

### Trends

- Pharmacy → “Medication refill delays” rising 6 straight weeks

And each theme includes **3–5 representative snippets** (not all comments).

---

## 6) Example data process flow (Bronze → Silver → Gold)

### Example Bronze input (**bronze\_comments**)

survey_key	response_date	visn	comment
SK001	2026-01-06	6	"My appointment was canceled twice and I waited 45 minutes on the phone."
SK002	2026-01-07	6	"Clinic canceled my visit last minute. Staff were very kind though."

### Bronze preprocess (**bronze\_comment\_preprocess**)

survey_key	is_processable	skip_reason
SK001	true	null
SK002	true	null

### Silver LLM output (**silver\_llm\_output**)

survey_key	llm_status	llm_json
SK001	SUCCESS	{ "topics": [ ... ] }
SK002	SUCCESS	{ "topics": [ ... ] }

### Silver exploded (**silver\_theme\_fact\_raw**)

theme_row_id	survey_key	topic	theme_rank	theme_text	sentiment	sent_conf	theme_conf
H1	SK001	Appointment Scheduling	1	appointment canceled twice	very_negative	0.96	0.91
H2	SK001	Wait Time	1	waited 45	negative	0.94	0.89

				minutes on phone			
H3	SK002	Appointment Scheduling	1	canceled last minute	very_negati ve	0.93	0.88
H4	SK002	Staff Courtesy / Respect	1	staff were very kind	very_positi ve	0.92	0.84

**Silver filtered + enriched (silver\_theme\_fact\_filtered or combined enriched table)**

(add sentiment\_score, apply thresholds)

theme_row_id	topic	theme_text	sentiment_score	sent_conf	theme_conf
H1	Appointment Scheduling	appointment canceled twice	-2	0.96	0.91
H2	Wait Time	waited 45 minutes on phone	-1	0.94	0.89
H3	Appointment Scheduling	canceled last minute	-2	0.93	0.88
H4	Staff Courtesy / Respect	staff were very kind	+2	0.92	0.84

**Silver embeddings (embedding\_vector filled later)**

theme_row_id	embedding_model	embedding_vector
H1	all-MiniLM-L6-v2	[ ... ]
H2	all-MiniLM-L6-v2	[ ... ]

**Gold canonicalization (gold\_theme\_fact\_canonical)**

theme_row_id	topic	theme_text	canonical_theme_id
H1	Appointment Scheduling	appointment canceled twice	APPT_C001
H3	Appointment Scheduling	canceled last minute	APPT_C001
H2	Wait Time	waited 45 minutes on phone	WAIT_C001
H4	Staff Courtesy / Respect	staff were very kind	COURT_C001

### Gold period metrics (**gold\_period\_theme\_metrics**)

time_grain	period_id	canonical_theme_id	theme_comment_count	theme_rate_neg	rate_pct_change
iso_week	2026-W02	APPT_C001	2	1.00	n/a
iso_week	2026-W02	WAIT_C001	1	0.50	n/a

### Gold signals for leadership (**gold\_period\_smoke\_signals**)

time_grain	period_id	signal_type	canonical_theme_id	topic	theme_comment_count	theme_rate	notes
iso_week	2026-W02	TOP_CONCERN	APPT_C001	Appointment Scheduling	2	1.00	highest negative
iso_week	2026-W02	TOP_COMPLIMENT	COURT_C001	Staff Courtesy / Respect	1	1.00 (pos)	strongest positive

Now the data is fully ready to compute **Top N**, **Emerging**, **Trend**, **Spike**, **Outliers** at any cadence simply by choosing **time\_grain + period\_id**.

