AI ANSWERS PANEL

Requires an AI subscription

You can open this panel by clicking the corresponding tab lower down on the right-hand side of the app.

What you can do here: Ask questions about your data in plain English and get AI-powered answers. Type questions like "What are the main barriers to education?" and the AI will search through your currently selected sources to provide relevant answers with supporting quotes. Perfect for exploring themes and getting quick insights from large amounts of text.
Main Features:

- **Query input** - Type your questions in plain English
- **Context (optional)** - Briefly describe what your sources are about (used to guide expansions and answers)
- **Expand Question (optional)** - Generate and edit expanded search phrases before asking
- **Automatic chunking** - Sources split into searchable pieces when needed
- **Similarity slider** - Control search precision (0.1-0.9)
- **Max Chunks slider** - Maximum number of the most relevant chunks to send to the AI
- **Enable checking (second AI pass)** - Optional checker reviews the draft answer, reports issues in a collapsed panel, and returns a corrected final answer (toggle is on by default).
- **Prompt history** - Navigate previous questions with prev/next buttons. See these tips

Bookmarking & restore:

- Each time you click **Ask**, the app automatically saves a **bookmark** for the current view (description: `AI Answers: <your question>`), and appends the bookmark link at the bottom of the answer.
- The bookmark footer also prints the **model name** and the **similarity/max-chunks settings** used for that answer.
- AI Answers settings are saved into the **URL state**, so bookmarks restore them (scope, sliders, model, thinking budget, checking toggle, and text fields).

Search Modes

AI Answers offers two search modes, automatically optimized based on your data size:

Full Sources Mode

Searches the complete text of your sources (documents/interviews).

How it works:

1. Type a question about the text of the currently selected sources
2. System automatically chunks sources into searchable segments (if not already done)
3. Searches through document chunks using AI embeddings and semantic similarity
4. Most relevant chunks are sent to AI for analysis
5. AI generates answers with supporting quotes from your sources

Question expansion and HyDE (Hypothetical Document Embeddings)

This is now **optional**:

- If you fill **Context**, it is included in both the expansion prompt and the final answer prompt (to reduce generic expansions/answers).
- If you click **Expand Question**, the app calls genAI to generate expansion phrases and shows them in an editable textbox (one per line). You can edit them, or use **Clear expansions** to hide/remove them.
- If you click **Ask** directly, the app **skips expansion** and searches using only your original question.

When expansions are provided, we match each phrase against chunks and sum scores per chunk, then select the top n by the max_chunks slider.

So for example if the user asks what is the connection between money and happiness, the AI produces question variants like:

- having money, being joyful
- being wealthy
- being happy
- connection between money and happiness

And answer variants like:

- financial security enables emotional wellbeing
- wealth contributes to life satisfaction
- economic resources support positive mental health outcomes

Best for: Exploratory questions about raw text, finding themes not yet coded, discovering new patterns.

Link Contexts Mode

Searches only through your coded causal links and their surrounding context (the quote + 3 sentences before/after).

How it works:

1. Gets filtered links from your current filter pipeline (respects Sources dropdown and all Source Groups filters)

2. For each link, extracts the selected quote plus surrounding context
3. Organizes contexts by source, with source metadata (title, custom columns)
4. For ≤500 links: Sends all contexts directly to AI
5. For >500 links: Same pattern as SRP — RPC + client-triggered embedding creation, no long-running Edge Function
    - Client gets query embedding (one call), then calls RPC `find_relevant_link_contexts(project_name, link_ids, query_embedding, top_k, threshold)`
    - RPC fetches links from DB, builds a **normalized key** per link (see Normalization note below), checks `embeddings`; if any missing (or `embedding IS NULL`) returns `needs_embeddings` + `missing_texts`
    - Client calls `DataService.createEmbeddingsBatch(missing_texts, ..., { useProvidedKeys: true })` (server-side generation + DB upsert), then retries RPC
    - RPC runs similarity in Postgres (pgvector), returns `relevant_link_ids` + `similarities` (and may include a `debug` block); no embedding vectors to frontend
    - Selection is **source-diversified** (round-robin: best link per source first), and avoids identical contexts within a source (exact de-dupe on the normalized key)
    - Prompt to AI uses full chunk_text (quote + surrounding 3 sentences) built on the client
6. AI analyzes contexts showing cause → effect relationships
7. AI is instructed to use the coded factor labels as the vocabulary (wrap labels in backticks like `this`) and to be sceptical about evidence coverage and alternative narratives.

Context format sent to AI:

```
'## Source: Interview with Participant 001
ID: ABC-123
custom_Country: Kenya | custom_Gender: Female | custom_Age: 34

Links from this source:

[ABC-123-1] Lack of resources → Poor school performance
Context: "We don't have enough books or supplies. The children struggle because..."

[ABC-123-2] Teacher training → Better outcomes
Context: "When teachers receive proper training, we see improvements in..."
```

Best for: Questions about causal relationships you've already coded, comparing patterns across sources, analyzing specific demographic groups using Source Groups filters.

Key advantages of Link Contexts mode:

- Uses your coded causal structure, not just raw text
- Respects all your filters (Sources dropdown + Source Groups)
- Includes source metadata in AI context (country, demographics, etc.)

- More focused and structured than full text search
- Automatically scales to large datasets (>500 links) using backend semantic search

etc etc

Sources, Links, and Factors subtabs

Use **Sources** for source rows, **Links** for link rows, and **Factors** for factor labels.

Why this exists: it lets you do fast, repeatable coding/scoring in bulk instead of editing one row at a time.

Each subtab now has two modes:

- **Answer mode**: asks a question over the currently selected payload fields and current filtered rows, then shows a text answer in the subtab (no row writes).
- **Create/Modify columns mode**: runs row-by-row writes.

How Create/Modify mode works:

1. Type an instruction in **Your Question** (for example, “score each row 0–3 for frustration”).
2. Choose a subtab and click **Run**.
3. The app only processes rows currently included by filters.
4. AI returns one value per requested target column per row.
5. The app writes those values back.

Per-subtab write behavior:

- **Sources**: writes to source fields/custom columns for filtered sources.
- **Links**: writes to link fields/custom columns for filtered links.
- **Factors**: does not create factor columns (there is no persisted factors table); it relabels matching values on currently filtered links, writing to either `cause`/`effect` (blank suffix) or `cause_<suffix>`/`effect_<suffix>` (suffix set).
- Related: Temporary Factor Labels filter.

Links + Factors (Create/Modify mode): target columns follow the **Label set** in the toolbar (there is no separate “target set” in Answers). Writes may overwrite existing non-blank values where the run applies; when targets are suffixed `cause_*` / `effect_*`, filtered-out rows may be backfilled from base `cause` / `effect` in the same run.

Choose payload columns (optional):

- All three subtabs use checkbox-pill choosers.
- Chosen fields are prepended to each row payload as `Field = value`.
- Default payloads:
  - **Sources**: `content`

- **Links**: `cause`, `effect`
- **Factors**: `label`

Safety behavior (Create/Modify mode):

- If a run would overwrite a standard (non-custom) column, the app shows one confirmation modal before processing.
- It is shown once per run (not once per batch).
- Protected/system columns are never overwritten.
- Non-filtered rows are unchanged.

Label-transform filters note:

- In **Answer mode**, transformed-label warnings/blocks are not shown.
- In **Factors → Create/Modify mode**, relabeling is blocked when label-transforming filters are active.