

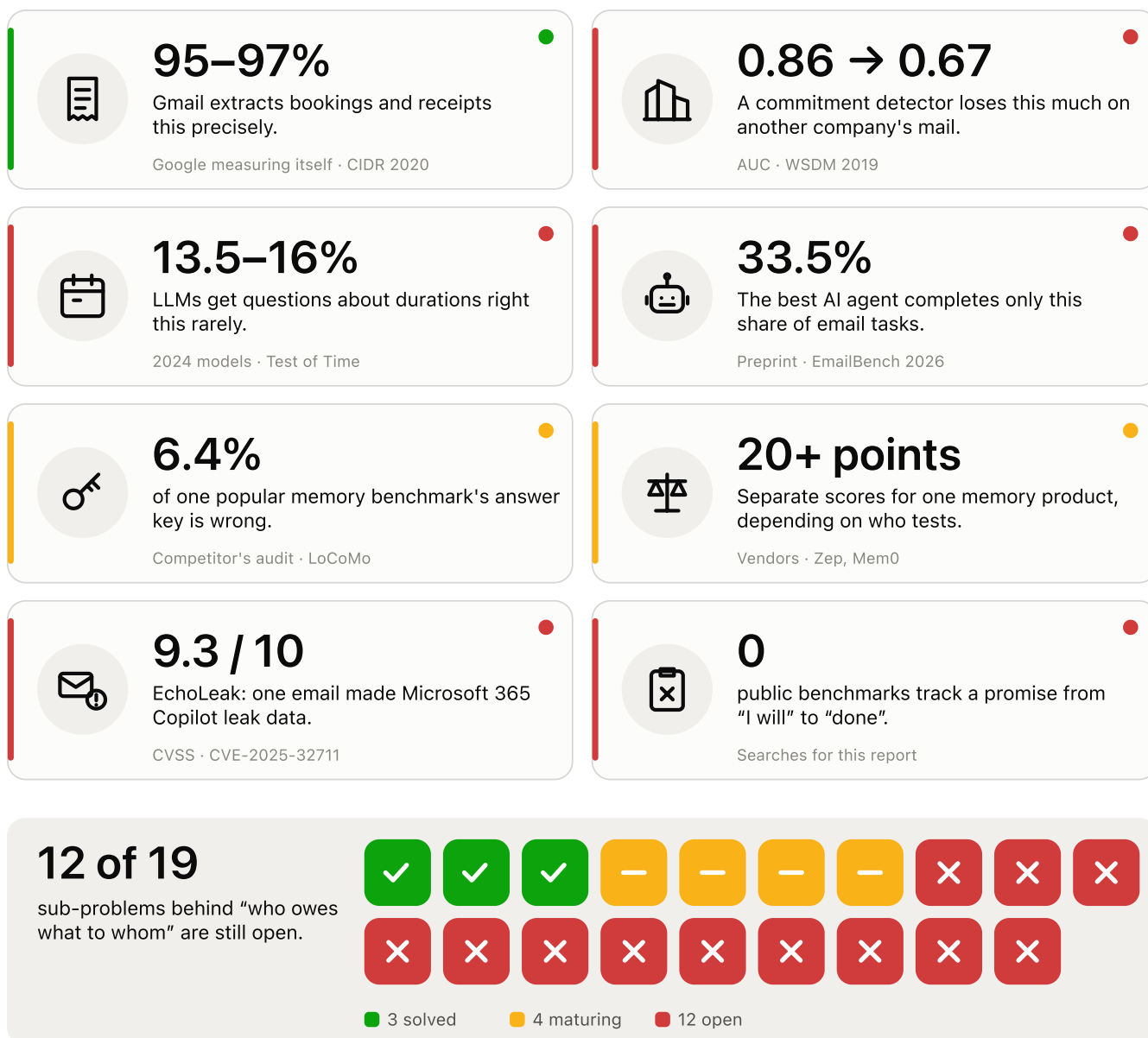


Figure 1. The field in numbers. Source: the table below links every number to its source; the pictogram counts the leaves of the problem tree in the solved, maturing and open section.

● Solved ● Maturing ● Open ○ Finding or principle

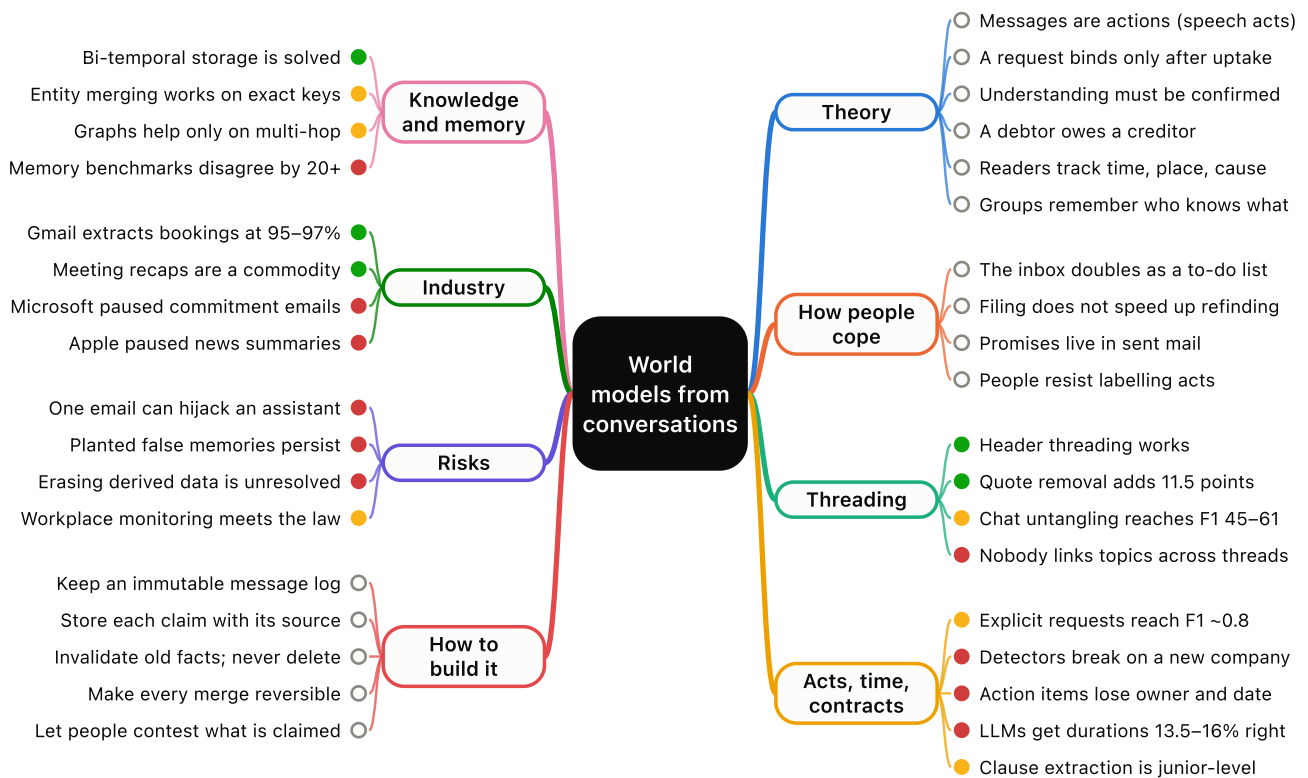


Figure 17. Mind map of the report. Source: the sections of this report; each leaf repeats a finding cited there.

24 named problem domains

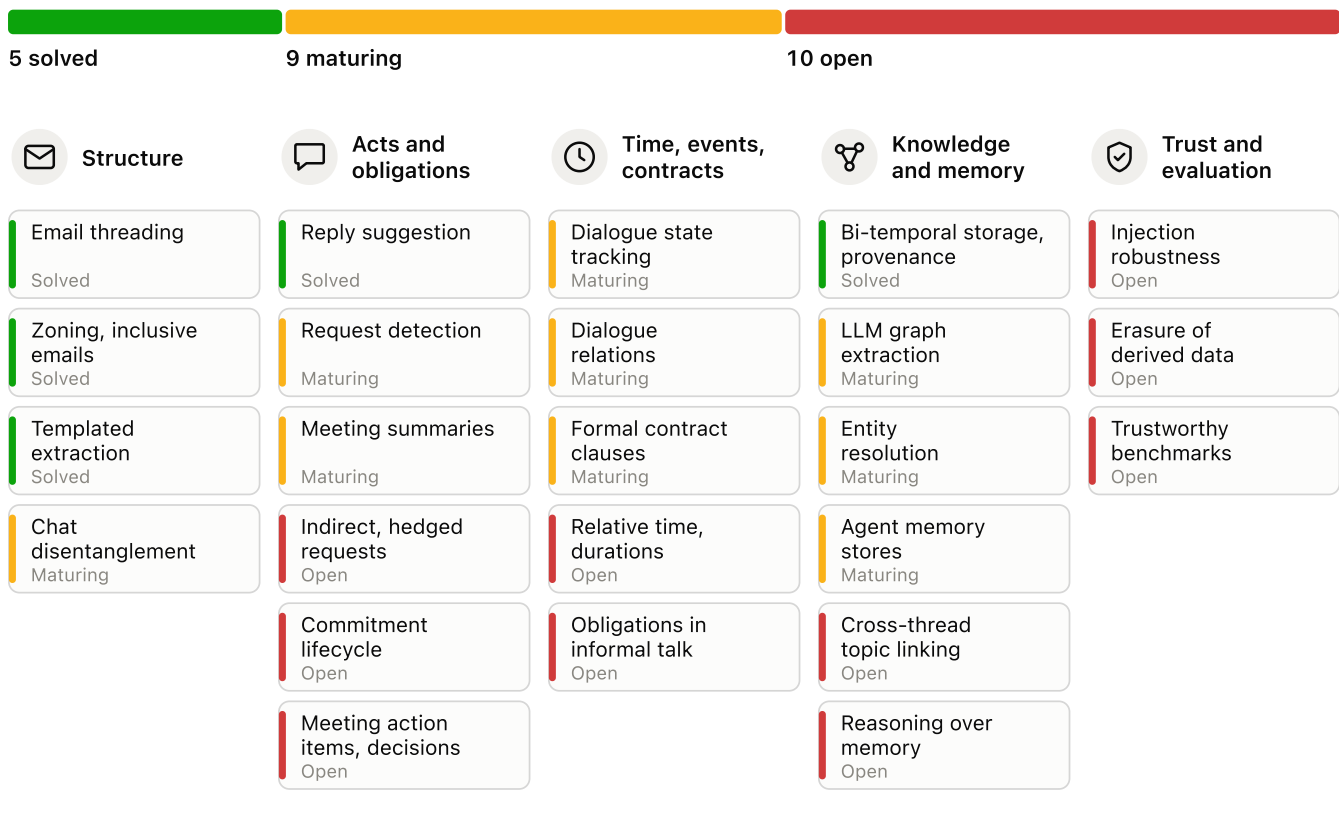


Figure 13. The problem domains and their status. Source: the table below. A domain is "open" when no method or benchmark shows it working on real conversations.

19 sub-problems behind one goal

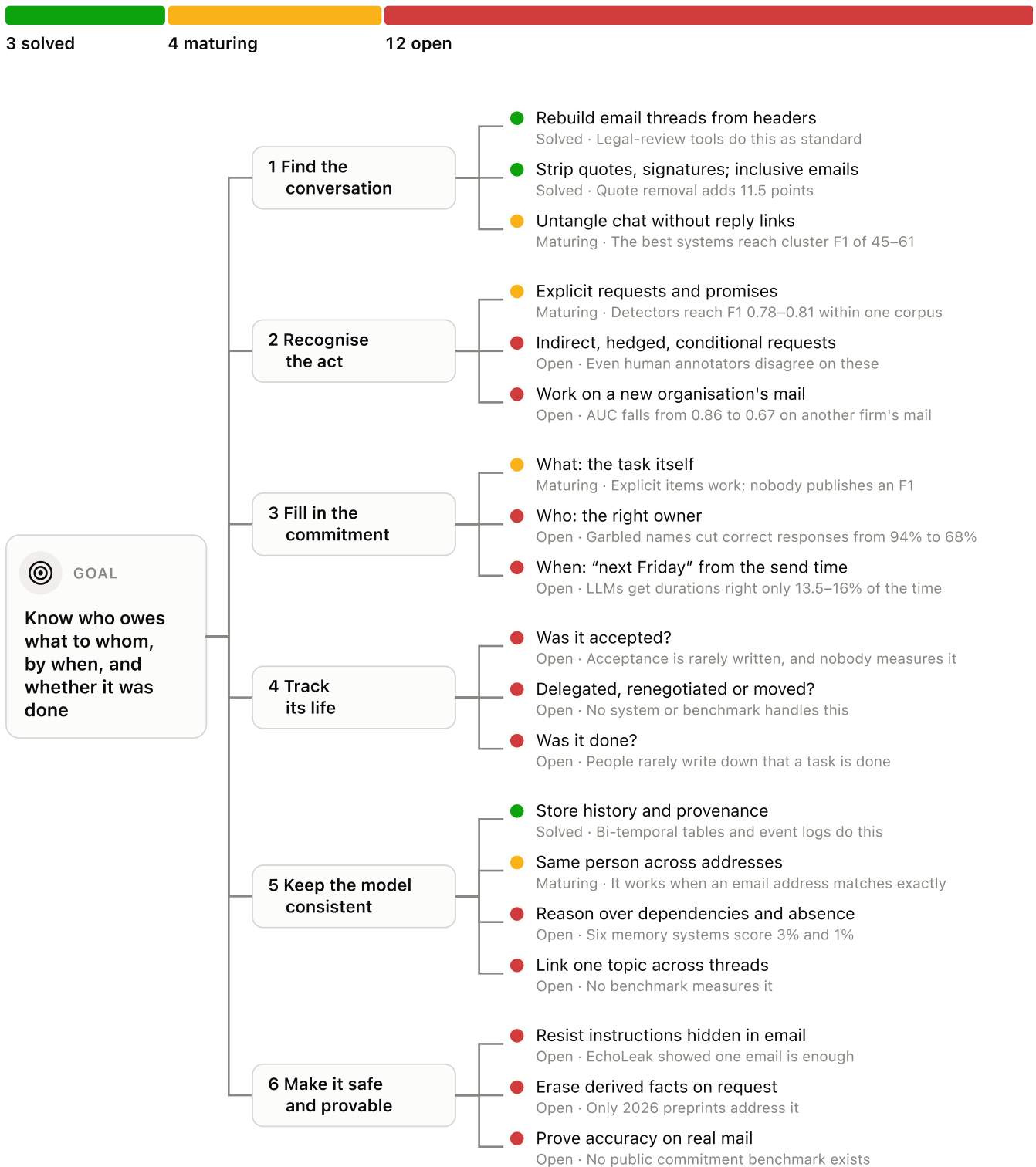


Figure 14. The problem tree behind "who owes what". Source: synthesis of this report; each leaf's evidence is cited in the sections above.



Schema-first

Define the types, then extract into them

EXAMPLES

Palantir Ontology · Gmail templated extraction · MultiWOZ slots · Symboleo contracts

STRENGTHS

- + Precise, consistent types
- + Easy to query, check and audit
- + Proven at scale on templated mail

WEAKNESSES

- Misses whatever the schema does not name
- New types need modelling work
- Brittle on entities it has not seen

RESULTS

- Gmail ML extraction: 95–97% precision
- Dialogue state: fixing labels adds 7–10% accuracy
- Unseen entities cut that accuracy by 29%



Schema-free

Let the model propose entities and relations

EXAMPLES

OpenIE · KGGen · GraphRAG · LightRAG

STRENGTHS

- + No upfront modelling
- + Captures the long tail
- + Helps multi-hop and summary questions

WEAKNESSES

- Inconsistent types and relations
- Duplicate or wrongly merged entities
- High token cost per query

RESULTS

- Fact capture: KGGen 66%, GraphRAG 48%, OpenIE 30% (author-run)
- Same method, different model: 44% to 73%
- Reasoning 53.4 vs plain RAG 42.9; lookups tie
- Up to ~100K prompt tokens per query (LightRAG)
- Mem0 removed graph memory from its open-source SDK



A hybrid · this report's synthesis

A schema for the core (people, commitments, events, time); open extraction for the long tail, promoted into the schema once a type keeps recurring; every claim keeps its source.



Open questions · no answers found

How should a schema grow from open extractions? What are real duplicate and merge-error rates on conversations? Does a graph still win against million-token context? Who approves a new type?

Figure 6. Two schools of graph construction. Sources: [survey arXiv 2510.20345](#), [KGGen](#), [GraphRAG-Bench](#), [Qian et al., CIDR 2020](#), [Mem0 docs](#). The hybrid is this report's synthesis.

Problem	 How people cope	 How machines try	 Known failure mode
Keeping promises	Rely on memory and flags; promises live in sent mail	Detect commitments in sent mail (Cortana, 2017)	Accuracy drops on another company's mail (AUC 0.86 → 0.67); Microsoft paused the Briefing email in 2023
Reading indirect requests	Infer intent from politeness, hedges and context	Classifiers and LLMs label the intent of each message	Hedged and conditional requests split even human annotators (kappa ~0.68)
Knowing it was accepted	Grounding: look for evidence the other side understood	Treat each detected request as a task	Uptake is rarely written down, and no benchmark measures it
Anchoring time	Read "next Friday" against when it was said	LLMs resolve dates from the text	LLMs get durations right only 13.5–16% of the time; no benchmark uses message timestamps
Finding things again	Scroll and search; heavy folder users do no better	Thread, search and auto-sort mail into bundles (Inbox)	Automatic sorting took control away from users; Inbox closed in 2019
Knowing who knows what	Transactive memory: ask the right person	Infer roles and experts from who writes to whom	Hierarchy inference was tested on one company (Enron) only
Summarising	Write meeting notes and recap emails	LLMs write recaps and notification summaries	Apple's summaries produced false headlines and were paused; GPT-4 Turbo spotted hallucinations with only 72% accuracy
Labelling the conversation	Leave acts implicit; people resisted declaring them (The Coordinator)	Infer acts automatically	It feels like surveillance (Suchman's critique, the Productivity Score backlash)

Figure 4. How people cope, how machines try, and how they fail. Source: the sections of this report on human practice, machine pipelines and industry; every entry is cited there.



Figure 8. Threading as a workflow. Source: the stage table above; each stage's colour follows the maturity evidence cited in its row.

Research 0001 v0.1 · Turning message streams into trustworthy world models

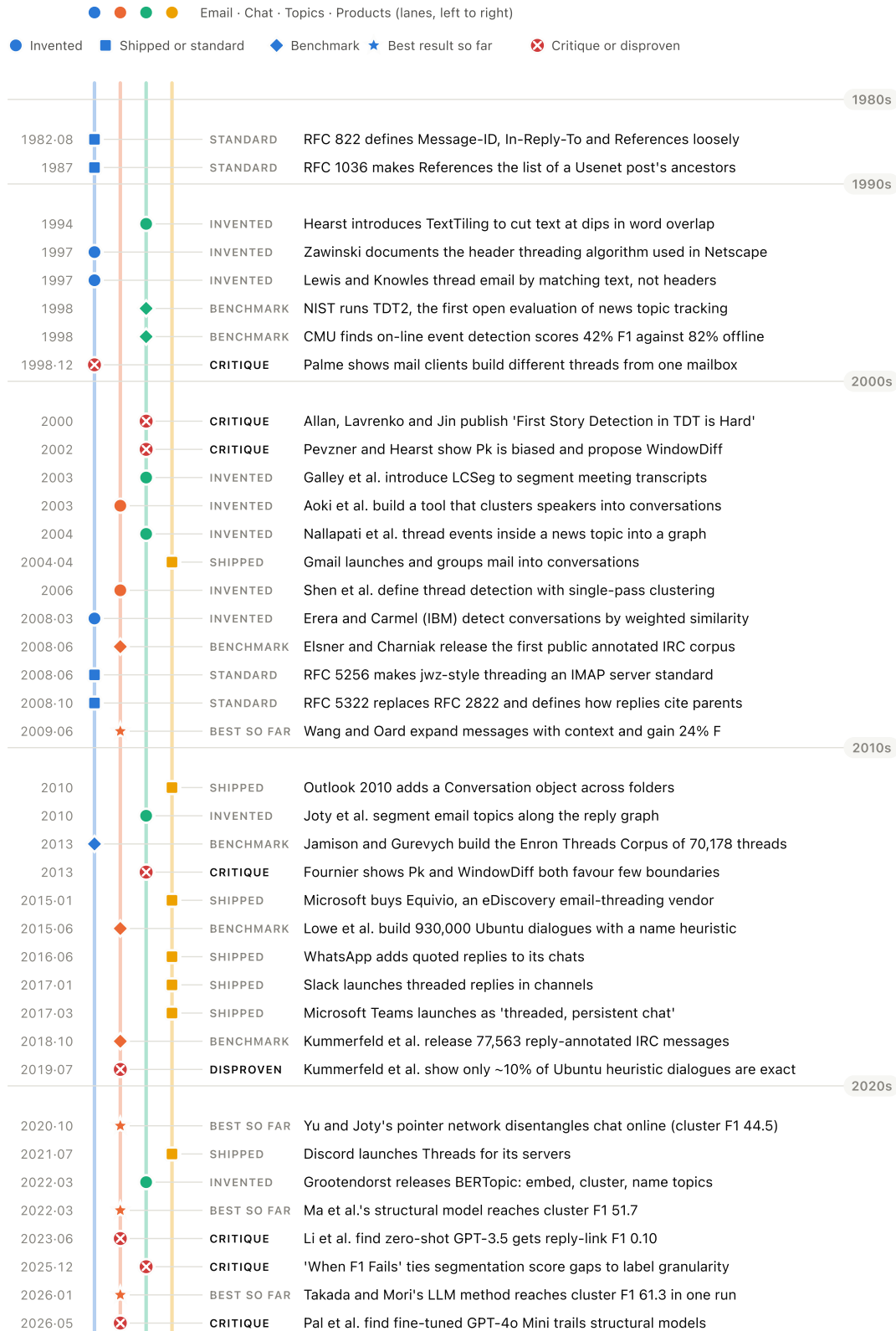


Figure 12. Timeline of threading research and products. Source: the three threading notes; every dated event carries its source in the figure's data file, `figures/threading_spec.json`.