

This vs That: A Comparison System That Is Allowed to Say “It Depends”

Kabir Gupta
Woodstock School
kabirgupta@woodstock.ac.in

Abstract

Ask a language model whether one option is better than another and it will tell you. It will tell you even when the two options are good at different things, even when the sources disagree, and even when nobody actually knows. The answer sounds confident either way, which makes it hard to tell the useful answers from the rest.

This paper describes *This vs That*, a working system that is allowed to decline. It reads evidence for both sides of a comparison, breaks that evidence into individual claims, checks where the claims genuinely clash, and returns one of five answers — including “it depends” and “the sources contradict each other”. The answer is produced by fixed rules over the evidence, not by asking a model what it thinks, so it can be inspected and argued with.

We describe the design, the reasoning behind each part of it, and four things we changed after building it and measuring what it actually did. One of those changes deleted a component we had been confident about: a filter meant to save work, which turned out to save none and to throw away evidence. We also report a small pre-specified comparison against a plain language model, which found no clear winner and caught our own system hedging on a question it should have answered.

1 The problem with always having an answer

Most comparison questions do not have a winner. Renting or buying a home depends on how long you will stay. A camera phone is better at night and worse at zoom. Whether one programming language beats another depends entirely on what you are building.

Ask a language model anyway and you will get a recommendation. Ask it twice with the options in the opposite order and you may get the opposite recommendation, which is a known problem with using models as judges [68]. Worse, models handed conflicting evidence tend to pick a side and not explain why [20], and retrieved text can override what the model actually knows to be true [60].

There is a growing family of systems that research a topic properly before writing about it. STORM plans an article by imagining several kinds of reader and asking what each would want to know [48]. Co-STORM goes further: it runs a conversation between expert agents and a moderator, so a reader encounters things they did not know to ask about [19]. SLIDERS reconciles evidence across document sets [21]. All of them produce a report. None of them answers “which is better”, which is the question people actually type. Section 6 places the system against this and other related work in more detail.

We wanted a system that answers that question when the evidence supports an answer, and says so plainly when it does not. That turns out to be a design problem more than a modelling problem, and most of this paper is about the design.

2 What the system produces

Given a comparison (two options, optionally with a shared context like “for backend services”), the system returns one of five verdicts.

This / That Depends	The evidence consistently favours one side across the dimensions that matter. Each side wins on different dimensions. The system names the conditions that decide it.
Evidence conflicts	Credible sources genuinely contradict each other on something central.
Insufficient evidence	Too little was found, or from too few independent sources, to support any verdict.

Alongside the verdict it reports a *lean*: which way the evidence tilts and how strongly. These are separate on purpose. Two comparisons can both come out as “depends” while one is nearly balanced and the other is three dimensions to one. Collapsing those into the same output loses the more useful half of the answer.

It also reports what it did not check. A comparison generates hundreds of claim pairs and the system examines a bounded number of them, so the output carries lines like “*Only 11 of 635 opposing-claim pairs on health impact were checked, so a contradiction may remain unfound.*” This is the part readers find strange at first and the part we are least willing to remove. A system that searches a fraction of the evidence and reports “no conflicts found” is making a claim it has not earned.

3 How it works

Figure 1 shows the whole pipeline. Five stages, of which two are borrowed and three are ours.

3.1 Stage 1: understanding the comparison

The first job is to work out what is actually being compared. This sounds trivial and is not.

Take “Rust vs Go for backend services”. Split it on the word “vs” and you get *Rust* against *Go for backend services*. The scope has attached itself to one side by accident of word order. Everything downstream then compares unlike things: the search for one side pulls general results and the other narrow ones, and the final answer partly measures the phrasing rather than the evidence.

So the system separates three things: the first option, the second option, and any shared context that applies to both. When it is not confident about that separation it keeps the user’s wording untouched, because running without the scope is safer than guessing it wrong.

It then derives the *dimensions*, meaning the aspects on which the two options differ. These come from the comparison itself, never from a fixed checklist. Comparing cameras produces dimensions like night performance and lens range; comparing cities produces cost of living and commute. A stock list would flatten every comparison into the same shape.

3.2 Stages 2 and 3: perspectives and evidence, via Co-STORM

We did not build a research engine. Co-STORM already does this well, and rebuilding it would have been the least interesting part of the project.

Co-STORM discovers the perspectives relevant to a topic (a safety-focused engineer and a developer-productivity advocate will care about different things) and then runs a conversation between agents holding those perspectives, with a moderator, retrieving sources as they go. What comes back is a set of utterances with inline citations, each traceable to a page.

How This vs That Works

A research-first system that uses STORM / Co-STORM to explore multiple perspectives, compare the evidence, and reach an honest verdict — *not just a winner*.

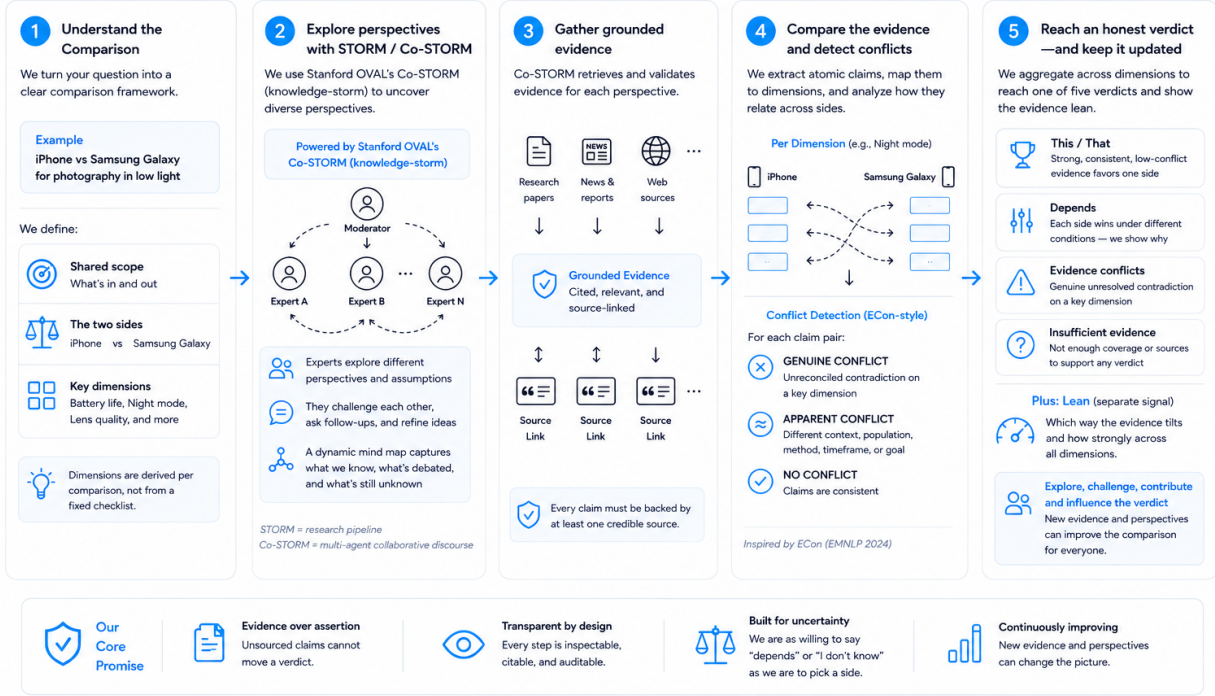


Figure 1: The pipeline. Stages 2 and 3 are Stanford OVAL’s Co-STORM, used as a dependency. Stages 1, 4 and 5 are this work.

We use it as a dependency and do not modify it. That boundary matters later.

3.3 Stage 4: claims and conflicts

The retrieved evidence arrives as prose, which is not something you can compute over. So the system breaks each utterance into *atomic claims*: single checkable assertions, each tied to the source that supports it and to the side and dimension it belongs to. A claim with no citation is discarded, because an unsourced claim must never be allowed to move a verdict.

Then the interesting part. For each dimension, the system pairs claims from one side against claims from the other and asks what kind of disagreement, if any, exists between them:

Genuine	The two claims cannot both be true. One of the sources is wrong.
Apparent	They look like they clash but both hold, because they differ in scope, timeframe, method, or what they are optimising for. The system names that difference, which we call the <i>conditioning variable</i> , and it is often the most useful thing in the output.
None	They do not bear on each other.

This distinction is the heart of the system. “Nuclear power emits 117g of CO₂ per kilowatt-hour” and “nuclear power produces no greenhouse gases” look like a flat contradiction until you notice one counts the whole plant lifecycle and the other counts generation only. That is an apparent conflict, and the conditioning variable

— what is being counted — is what a reader actually needs.

The model doing this classification must give reasons and quote the source passage it relied on. An adjudication that cites nothing has its confidence reduced. That rule exists because ECon’s central finding is that models resolve conflicts by picking a side and not justifying it [20], and we would rather catch that than assume it away.

A second, cheaper model then checks that quote. This step is recent. Section 4.4 explains how it came about and why it currently observes rather than intervenes, but it is part of the shipped pipeline, and it is the one place where the system asks a model to check another model’s work.

3.4 Stage 5: the verdict

The verdict is computed, not asked for. Five rules run in order:

1. If coverage is too thin or the sources too few, return **insufficient evidence**.
2. If a genuine, unreconciled conflict sits on a dimension that matters, return **evidence conflicts** and show the contradiction. Do not pick a winner.
3. If the sides win on different dimensions, return **depends** and name the conditions.
4. Otherwise, if one side wins a clear majority of the key dimensions by a clear margin, return **this** or **that**, with the residual uncertainty and the strongest opposing point attached.

A model writes the summary prose, but it cannot change the label. That separation is what makes the behaviour testable: we can feed the rules synthetic evidence and check that a genuine conflict really does block a winner, which is not something you can do to a prompt.

4 Four things we changed after building it

Building the system and then measuring what it did produced four corrections. We report them because each one was invisible until instrumented, and because three of the four are mistakes rather than improvements.

4.1 We deleted a filter that was doing nothing

Between claim extraction and conflict checking sat a screening step. A small local model scored each claim pair and dropped the obviously compatible ones before they reached the expensive check. The logic seemed sound: the expensive step costs money, so filter first.

It saved nothing at all. The expensive step is already capped at 40 checks per comparison, and every run we measured performed exactly 40 — out of anywhere between 174 and 973 candidate pairs. The cap was doing the saving. A filter in front of a cap cannot reduce the work; it can only change which items get the attention.

Once we understood it as a selector rather than a saver, it looked much worse. Roughly a quarter of what it discarded were real conflicts, and it improved the quality of what remained by about one percentage point. It was also slow: removing it took that stage from about 42 seconds per comparison to effectively nothing on our development machine, and on the small server we actually deploy to it had been consuming its entire four-minute budget before giving up. One further detail only surfaced during the timing: a cap intended to bound how long screening ran was also capping how many pairs could ever be considered, so 605 pairs across our test set were never candidates at all.

The general lesson we take from this: *if the expensive stage is already capped, a filter in front of it cannot save you anything, and has to justify itself purely by choosing better.* Ours did not.

4.2 Our tests were measuring the wrong distribution

That filter had not been guessed at. Its threshold was tuned against sixteen hand-written test cases, including ones designed to catch exactly the failure mode we were worried about. The tests passed.

The test cases are deliberately hard: pairs of claims worded almost identically but about different things. Measured properly, their word overlap is around 0.5 to 0.8. Real claim pairs from actual comparisons overlap at about 0.03. We had tuned a threshold on material twenty times more similar than anything the system encounters, and on real input the filter never fired at all.

The same effect shows up in how accurate our conflict classifier looks. On the hand-written test cases it scores 93.8%. On pairs drawn from real runs it scores 71.9%. Both numbers are real; only the second one describes the system.

4.3 Checking more pairs did not change any answer

If the cap limits the search, raising it should help. We checked every pair instead of forty — 3,141 checks instead of 360, across eleven comparisons — and re-ran the verdict rules. Not one verdict or lean changed.

Ten of the eleven comparisons contain no genuine cross-side contradiction at all, even at full coverage. That is partly a property of what we compare: lifestyle questions where sources emphasise different things rather than contradicting each other. A contested scientific question would likely behave differently. But it does mean the honest reading of our coverage disclosure is “we checked a sample” rather than “there is probably more hiding here”.

4.4 “Grounded” did not mean what we were reporting

Every conflict check must quote the source passage it relied on, and a simple check confirms the quote really appears in the retrieved sources. We had been treating a pass as evidence that the check was justified.

It is not. It only establishes that the sentence exists. A real sentence, from a real source, that says nothing whatsoever about how the two claims relate will pass it happily.

Finding out how often that happens needed a second opinion on every quote, which was too expensive to contemplate with a full language model — one call per check, on every comparison.

What made it affordable is a class of model that has appeared recently and is worth describing, because it changes what is worth building. These models do not write text. They take a situation and a fixed set of options and return one of them, with a probability attached that the vendor describes as calibrated. TypeSafe’s Jev [52], which we used, is reported to answer in well under a second, and its listed price (\$0.042 per million input tokens, with output not metered) is a small fraction of what a generative model charges for the same call. Stanford and Nvidia’s CLM-8B takes a different route to the same shape (it embeds the state and the candidate options separately and picks the closest match, which lets it cache and reuse those embeddings), and its authors report it running several times faster than Jev while scoring lower on some tool-use benchmarks [26]. We have not evaluated CLM ourselves and make no comparison between them.

For a system like ours the interesting thing about these models is not the speed but what the price makes possible. Asking a second model about every single adjudication is absurd at generative prices and routine at

these. Across 65 checks in four comparisons, the existing test passed 60. Of those 60, the second opinion judged 22 to have quotes that actually bore on the classification.

We are careful about what this shows. Two different instruments disagree by a factor of about two and a half, and we have not yet checked either against human judgement, so we cannot say which is closer to right. What we can say is that they are not measuring the same thing, and that the one we had been quoting is the weaker of the two. The system now records the second opinion on every check and deliberately does nothing with it — it changes no verdict — until it has been validated properly. Having just been caught trusting an unvalidated threshold, we were not keen to repeat it immediately.

4.5 What this says about where a small model helps

We tried the small structured model in four places. Three failed: it could not usefully replace the screening filter (which was inert anyway), it was slightly worse than the full model at classifying conflicts, and using it to check far more pairs changed nothing. The one that worked was the verification above, a check the pipeline had simply never performed, because nobody could afford it.

That is the pattern we would look for again. A cheap model does not obviously improve the calls you already make. It makes affordable the ones you had quietly skipped.

One practical caveat for research systems. Jev is a commercial service in early access, so a result that depends on it cannot be reproduced by someone without an account, which is an awkward property for a paper. CLM-8B is released under Apache 2.0 with open weights, which removes that problem, and we would use an open model for this role in future work for that reason alone. The architectural argument above does not depend on which one: it depends only on there being a model cheap enough that a per-item check stops being a budget question.

5 Does it work?

A companion evaluation, with the method committed to version control before any data was collected, compared the system against a plain language model on ten comparisons chosen in advance. Each comparison was put to the baseline twice with the options in opposite orders.

Three things came out of it. The baseline reversed its recommendation on 2 of 10 comparisons when the order was swapped; our rule-based verdict layer is invariant to option order by construction. The baseline named a winner in half its answers and ours in 3 of 10. But on the one comparison with a defensible winner, ours hedged and the baseline got it right. A system that will not commit on an easy question is being evasive, not careful, and we count that as a failure of the same kind as over-answering.

The third finding was about our own scoring. Our measure of “how uncertain does this answer sound” was built from hedging words like *generally* and *often*. It therefore scored “only 11 of 635 pairs were checked” as *less* uncertain than a vague qualifier — exactly backwards. An independent rater scored those cases the opposite way to our code, and on that measure agreed with it worse than chance ($\kappa = -0.30$).

Neither system came out better calibrated. We are not claiming otherwise: ten comparisons cannot settle it, and the two systems are not really comparable, since the baseline has no retrieval at all.

Table 1: How the closest systems handle a comparison or a conflict. The last column says what produces the final label.

System	What it returns	How disagreement is treated	Label produced by
CAM [45]	Scores for two objects on stated aspects, with supporting sentences	Sentences favouring each object are classified and scored	Aggregated sentence scores
LLM answers, as scored by CompQA [38]	A structured comparison ending in a choice	Arguments listed for both sides	The generating model
STORM, Co-STORM [19, 48]	A cited report; also a mind map in Co-STORM	Described in prose, if at all	No label
SLIDERS [21]	An answer from SQL over reconciled tables	Best-supported value kept, others removed	Model-written SQL, run deterministically
Cattan et al. [5]	A response styled to the conflict type	One conflict type per query	The generating model
AVeriTeC [46]	One of four labels for a single claim	A dedicated conflicting-evidence label	Varies by participating system
ProoFVer [25]	A verdict for a single claim, with a proof	Natural-logic operators over claim and evidence spans	An automaton run over the proof
ArgLLMs [13]	A verdict for a single claim	Supporting and attacking arguments with strengths	An argumentation semantics
This vs That	One of five verdicts, a lean, and a count of unchecked pairs	Claim pairs typed genuine, apparent or none, with a conditioning variable	Fixed rules over the typed evidence

6 Related work

This vs That draws on several lines of work that rarely cite each other: comparative question answering, multi-source research systems, conflicting evidence in retrieval, fact verification, abstention and expressed uncertainty, attribution, and the evaluation of language-model judges. We cannot cover each area in full. For each one we describe the work closest to ours, what we took from it, and where we differ. Table 1 lines up the nearest systems side by side.

6.1 Comparative question answering

Comparative questions were studied as a retrieval problem well before language models could answer them fluently. Bondarenko et al. studied comparative questions submitted to a web search engine [2], and later work split answering them into stages: recognising that a question is comparative, finding the objects and the aspect being compared, and classifying which object each retrieved sentence favours [1, 3, 40]. CAM [45] mined comparative sentences from a web crawl and scored the two objects against each other; in its user study, people answered comparative questions more accurately and faster with it than with keyword search. Later systems generated a written answer on top of the retrieved arguments [6, 47], and the Touché shared task turned comparative argument retrieval into a benchmark [4].

Two things separate this line of work from ours. The first is where the aspects come from. They are given by the user or pulled out of the question, and CAM’s authors note that the objects and aspects still had to be stated explicitly. Our dimensions are derived from the comparison itself (Section 3.1). The second is

that the output is built around a choice. CompQA [38], a recent framework for evaluating language-model answers to comparative questions, scores answers on 15 criteria and counts a distinct final choice as part of a well-structured answer, although one criterion does accept an inconclusive answer when the arguments are balanced. The same study reports one case, which its authors describe as rare, where GPT-4 was given arguments favouring Yahoo over Google, wrote a comparison favouring Google, and reworded the arguments to fit. Our design rules out that failure for the label, because the label is computed from the evidence rather than written by a model (Section 3.4).

Older still, opinion summarisation produced contrastive and comparative summaries of two entities from their reviews [18, 27], and aspect-based sentiment analysis made the aspect a standard unit of comparison [42]. Our dimensions do a similar job. What differs is the input: open web evidence about an arbitrary pair, not reviews of products in a known category.

6.2 Research and synthesis systems

STORM [48] and Co-STORM [19], which we use for retrieval, produce a cited long-form report, and Co-STORM also maintains a mind map of the discussion. Deep research agents follow the same broad pattern of planning, iterative search and report writing [17]. None of these returns a verdict. A report can mention that sources disagree without saying what the disagreement means for the question asked, and that gap is what our Stages 4 and 5 are for.

SLIDERS [21] is closer in one respect. It extracts facts from large document collections into a relational database, keeps provenance and an extraction rationale for every value, and runs a reconciliation agent that deduplicates records, merges partial ones and resolves competing values before questions are answered in SQL. When two values compete, SLIDERS keeps the one best supported by its provenance and removes the rest. That is the right move when the disagreement is an extraction error. It is the wrong one when two sources are both correct about different things, which is the case our apparent-conflict label exists for. SLIDERS also puts a relevance gate in front of extraction, and the contrast with our Section 4.1 is useful. Their gate is justified by what it prevents, namely extraction errors on chunks that contain nothing relevant, and its authors measured its false-negative rate on a sample. Ours was justified by a saving the downstream cap already provided, and we did not measure what it discarded until after it was built.

6.3 Conflicting evidence

A large body of work now studies what language models do when evidence disagrees. Xu et al. [63] divide knowledge conflicts into context–memory, inter-context and intra-memory conflicts. Ours are inter-context: conflicts between retrieved sources. Work on the context–memory kind still matters, because it shows how models behave under pressure. Depending on the setting, models adopt retrieved text even when it is wrong [60] or ignore it in favour of what they memorised [32]. They are highly receptive to coherent outside evidence, yet lean towards evidence that agrees with what they already believe [61], and they weigh documents mainly by how relevant they look to the query while largely ignoring the stylistic cues people use to judge credibility [54]. Chen et al. [7] found that contradictions among retrieved sources barely lowered model confidence, and proposed calibrating models so that they do not present a single answer when the evidence contains conflicting candidates.

Several benchmarks target conflict between sources directly. WikiContradict [16] uses real contradictions between Wikipedia passages. ConflictingQA [54] pairs contentious yes/no questions with evidence on both sides. DebateQA [62] scores answers to debatable questions partly by the perspectives they cover. QACC [31] and RAMDocs [55] cover open-domain questions whose retrieved contexts conflict, the latter adding ambiguity and misinformation. ECon [20] generates conflicts of controlled type and evaluates both

detection and resolution; its finding that models tend to favour one side without justification is why every adjudication in our system must quote its evidence (Section 3.3).

6.4 Genuine versus apparent conflict

The distinction at the centre of our system is not new. De Marneffe et al. [33] treated reference to the same event or entity as a precondition for contradiction, so two sentences about different events cannot contradict each other however opposed they look. Open-domain question answering reached a similar point from the other side: a question can have several correct answers under different readings [36, 49], or answers that change with time and place [66]. Our conditioning variable is a name for whatever separates the two readings.

The closest recent work is the conflict taxonomy of Cattan et al. [5]. It separates *complementary information*, meaning answers a single reader could accept together, from conflicting opinions or research outcomes, outdated information, and misinformation, and gives each type its own expected response style. Their complementary type corresponds to our apparent conflict. Outdated information falls on either side of our line: two figures from different years are an apparent conflict, with time as the conditioning variable, while two sources that both claim to give the current figure are a genuine one. Their remaining two types mostly correspond to our genuine conflict. We differ in granularity and in what the label is used for. They assign one type per query and use it to shape generated prose; we label individual claim pairs, record the conditioning variable for each apparent conflict, and pass the labels to fixed rules. Two of their numbers are worth setting beside ours, though neither is directly comparable. The models they tested predicted the conflict type with between 53% and 65% accuracy, and misinformation was rare among top search results in their curated set (5 of 458 queries). Both fit what we saw: the classification is hard, and genuine contradictions are uncommon in everyday comparisons (Section 4.3).

Data integration research studied conflicting sources long before language models. Truth discovery methods jointly estimate how reliable each source is and what the true value is when web sources disagree about a fact [28, 64]. They assume one true value per attribute. Our apparent-conflict label exists for the cases where that assumption fails.

6.5 Fact verification with more than two outcomes

Fact verification has long allowed a system to decline to decide. FEVER [51] labels a claim supported, refuted, or not enough info, and AVeriTeC [46] adds a label for real-world claims whose evidence conflicts or has been cherry-picked. Our verdicts are a close analogue: *insufficient evidence* corresponds to not enough info, and *evidence conflicts* to AVeriTeC’s conflicting-evidence label. The object being judged is different, though. A fact-checker judges one claim. We compare two options across several dimensions, and that needs an outcome fact verification has no counterpart for, *depends*, for when each side wins somewhere.

Computing a verdict with fixed rules also has precedent. ProoFVer [25] generates a natural-logic proof for a claim and gets the verdict by running a deterministic automaton over the proof’s operators, so its explanation is faithful by construction. Argumentative LLMs [13] have a model produce supporting and attacking arguments with strengths, then compute the verdict with a formal argumentation semantics, which makes the decision contestable. ProgramFC [39] turns a complex claim into a reasoning program that is executed step by step. We share their premise: a label the model cannot write directly is easier to inspect, test and argue with. Our rules are far simpler than a natural-logic calculus or an argumentation semantics. What they add is the set of outcomes a comparison needs, and an explicit record of the evidence the rules never saw.

Our atomic claims follow the practice of FActScore [37] and SAFE [58], which split long text into atomic facts and check each one. Wanner et al. [57] showed that such scores are sensitive to how the text is split,

because errors made while splitting are charged to the text being evaluated. The same risk applies here. Our conflict counts depend on how the evidence was split into claims, and this paper does not report a separate measurement of that step.

6.6 Abstention and expressed uncertainty

Letting a system decline has a long history in selective prediction, from Chow’s reject option [9] to selective classification [10] and selective question answering [23]. SQuAD 2.0 [43] put unanswerable questions into a mainstream reading-comprehension benchmark. For language models, Wen et al. [59] survey abstention methods, benchmarks and metrics; Feng et al. [12] use several models to find knowledge gaps and abstain; and AbstentionBench [24] finds abstention unsolved across 20 frontier models, including on underspecified and subjective questions, with reasoning fine-tuning making it worse. Most of this work treats abstention as a single outcome: the model does not answer. We split it into three different non-answers (*depends*, *evidence conflicts* and *insufficient evidence*) because each tells the reader to do something different next.

Work on verbal uncertainty asks whether a model’s confidence, stated or measured, tracks its accuracy [22, 29, 35], and how expressions of uncertainty in a prompt change what a model does [69]. Yona et al. [65] measure the gap between a model’s intrinsic confidence and how decisively its answer is worded, and penalise hedging too much as well as too little. That matches the position we took in Section 5, where hedging on an easy question counts as a failure just as over-answering does. It also helps explain our failed uncertainty score. Hedge detection has a lexical tradition built around cue words [11], and our score was lexical: it counted words like *generally* and missed a quantitative coverage statement entirely. Assertion-level measures, such as the one Yona et al. use, judge how decisive a statement is as a whole instead of counting cue words, and are a better fit for sentences like ours.

6.7 Attribution and verification

The problem in Section 4.4 sits on a ladder the attribution literature already describes. The weakest check is that a quoted passage exists in the source. Stronger is whether the passage supports the statement it is attached to, which AIS [44] defines, ALCE [14] benchmarks, and Liu et al. [30] measured in generative search engines, finding only about half of generated sentences fully supported by their citations. Stronger again is whether the model actually relied on the passage. Wallat et al. [53] show that a citation can be correct and still post-rationalised, with up to 57% of citations unfaithful in their experiments. Our string match sat on the lowest rung. The second model moves the check up one rung, towards support. It says nothing about faithfulness.

Small checkers for that middle rung already exist. MiniCheck [50] trains small models to check claims against grounding documents at close to GPT-4 accuracy for a fraction of the cost, and FrugalGPT [8] studies cascades that route work between cheap and expensive models. Our observation in Section 4.5 is narrower: the cheap model helped where it made possible a check the pipeline had been skipping, not where it replaced a check already being made. The newest models of this kind return a choice with a probability instead of text. They were commercial or very recent at the time of writing [26, 52], and we know of no independent evaluation of their calibration.

6.8 Language-model judges and the validity of evaluation

Order sensitivity is well documented. LLM judges change their verdict when the order of the candidates changes [56, 68], and multiple-choice accuracy moves with the order of the options [41, 67]. The usual fixes average over orders or calibrate the judge. Our rules are symmetric in the two options, which is why the

verdict layer is invariant by construction (Section 5). That guarantee covers the rules only. The evidence they receive comes from retrieval and dimension derivation, which could still be affected by order, and the summary prose is still written by a model.

Our Section 4.2 finding has precedent too. Contrast sets [15] and HANS [34] were built from minimally different, high-overlap examples to expose models that lean on surface heuristics. They are stress tests, not estimates of accuracy in use, and we had used a stress-test set as if it were a calibration set. ECon reports that factual-consistency models struggle with lexically similar conflicts [20], which suggests that high-overlap pairs behave as a regime of their own rather than as a harder version of the typical case.

6.9 What is new here

Each ingredient above has precedent: non-binary verdicts in fact verification, a conflict taxonomy for retrieval-augmented generation, rule-computed labels in natural-logic and argumentation systems, and multi-perspective retrieval in STORM and Co-STORM. To our knowledge, no earlier system combines them to answer two-option comparative questions, with a verdict that is allowed to be *depends*, conflicts typed at the level of claim pairs with a named conditioning variable, and a disclosure of how much evidence went unexamined. The measurements in Section 4, including the ones that went against us, are the other contribution.

7 Limitations

The system is only as good as what the retriever finds, and the retriever is one commercial search API.

The conflict classifier is right about 72% of the time on real claim pairs, measured against 48 hand-labelled examples. Of the labels drawn from real runs, only 11 carry recorded reasoning; the rest were quick judgements on obvious cases. That is a thin basis and we would not build anything further on it without extending the set.

The evaluation is ten hand-picked comparisons, mostly ones we expected to be conditional, with expected answers that are our own judgement rather than ground truth. The reliability check is one rater on ten outputs. The verification result in Section 4.4 rests on 65 checks and one model’s opinion.

Most of our comparisons are everyday questions (cities, diets, phones) where sources tend to emphasise different things rather than contradict each other. That is probably why so few genuine conflicts turn up, and it means the system’s conflict machinery has not really been tested on a contested empirical question.

None of this establishes that the system is effective. It establishes that it behaves the way the design intends, which is a weaker and more honest claim.

8 Conclusion

The interesting part of building this was not the modelling. It was deciding what the system is allowed to say, and then keeping it honest about what it actually checked. The five verdicts, the separation between rules and prose, and the disclosure of unexamined evidence are all attempts at the same thing: making a system whose confidence you can inspect rather than infer.

Three of the four corrections in Section 4 were cases where we had been confident and wrong, and none of them was visible from reading the code. We found them by measuring. That is an unglamorous conclusion

for a design paper, but it is the one we would pass on: a system that reports its own uncertainty needs its self-reports checked against something outside itself, because from the inside they look fine.

Availability

The system is open source under the MIT licence at <https://github.com/radcool22/this-vs-that>. The repository carries the full design history, the test suite (which runs offline, without API keys), and the measurement records behind Section 4 under `docs/research/`.

Statements

Ethics. No human-subjects research; one adult volunteer scored system outputs and consented to the use of their anonymous scores. **Conflict of interest.** The author built the system described here. **Funding.** None. **AI use.** Language models assisted with code, drafting, and finding related work. No language model scored the evaluation rubric. The verification described in Section 4.4 is itself a model judgement and is reported as such.

References

- [1] M. Beloucif, S. M. Yimam, S. Stahlhacke, and C. Biemann. Elvis vs. M. Jackson: Who has more albums? Classification and identification of elements in comparative questions. *LREC*, 2022.
- [2] A. Bondarenko et al. Comparative web search questions. *WSDM*, 2020.
- [3] A. Bondarenko, Y. Ajjour, V. Dittmar, N. Homann, P. Braslavski, and M. Hagen. Towards understanding and answering comparative questions. *WSDM*, 2022.
- [4] A. Bondarenko et al. Overview of Touché 2022: Argument retrieval. *CLEF*, 2022.
- [5] A. Cattan, A. Jacovi, O. Ram, J. Herzig, R. Aharoni, et al. DRAGged into conflicts: Detecting and addressing conflicting sources in search-augmented LLMs. arXiv:2506.08500, 2025.
- [6] V. Chekalina, A. Bondarenko, C. Biemann, M. Beloucif, V. Logacheva, and A. Panchenko. Which is better for deep learning: Python or MATLAB? Answering comparative questions in natural language. *EACL System Demonstrations*, 2021.
- [7] H.-T. Chen, M. J. Q. Zhang, and E. Choi. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. *EMNLP*, 2022.
- [8] L. Chen, M. Zaharia, and J. Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. arXiv:2305.05176, 2023.
- [9] C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1), 1970.
- [10] R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11, 2010.
- [11] R. Farkas, V. Vincze, G. Móra, J. Csirik, and G. Szarvas. The CoNLL-2010 shared task: Learning to detect hedges and their scope in natural language text. *CoNLL Shared Task*, 2010.

- [12] S. Feng et al. Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-LLM collaboration. *ACL*, 2024.
- [13] G. Freedman, A. Dejl, D. Gorur, X. Yin, A. Rago, and F. Toni. Argumentative large language models for explainable and contestable claim verification. *AAAI*, 2025.
- [14] T. Gao, H. Yen, J. Yu, and D. Chen. Enabling large language models to generate text with citations. *EMNLP*, 2023.
- [15] M. Gardner et al. Evaluating models' local decision boundaries via contrast sets. *Findings of EMNLP*, 2020.
- [16] Y. Hou et al. WikiContradict: A benchmark for evaluating LLMs on real-world knowledge conflicts from Wikipedia. *NeurIPS Datasets and Benchmarks*, 2024.
- [17] Y. Huang et al. Deep research agents: A systematic examination and roadmap. arXiv:2506.18096, 2025.
- [18] H. Iso, X. Wang, S. Angelidis, and Y. Suhara. Comparative opinion summarization via collaborative decoding. *Findings of ACL*, 2022.
- [19] Y. Jiang, Y. Shao, D. Ma, S. J. Semnani, and M. S. Lam. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations. *EMNLP*, 2024. arXiv:2408.15232.
- [20] C. Jiayang et al. ECon: On the detection and resolution of evidence conflicts. *EMNLP*, 2024. arXiv:2410.04068.
- [21] H. Joshi, P. Shethia, J. Dao, and M. S. Lam. SLIDERS: Systematic reviews via automated evidence synthesis and reconciliation. arXiv:2604.22294, 2026.
- [22] S. Kadavath et al. Language models (mostly) know what they know. arXiv:2207.05221, 2022.
- [23] A. Kamath, R. Jia, and P. Liang. Selective question answering under domain shift. *ACL*, 2020.
- [24] P. Kirichenko, M. Ibrahim, K. Chaudhuri, and S. J. Bell. AbstentionBench: Reasoning LLMs fail on unanswerable questions. *NeurIPS Datasets and Benchmarks*, 2025.
- [25] A. Krishna, S. Riedel, and A. Vlachos. ProofFVer: Natural logic theorem proving for fact verification. *Transactions of the ACL*, 10, 2022.
- [26] J. Kwok et al. Contrastive Language Models: CLM-v0.1-8B. Stanford University and NVIDIA, 2026. Model card at <https://huggingface.co/Contrastive-LM/CLM-v0.1-8B>. Open weights under Apache 2.0. Benchmark figures are the authors' own and are not independently verified here.
- [27] K. Lerman and R. McDonald. Contrastive summarization: An experiment with consumer reviews. *NAACL-HLT (Short Papers)*, 2009.
- [28] Y. Li et al. A survey on truth discovery. *SIGKDD Explorations*, 17(2), 2016.
- [29] S. Lin, J. Hilton, and O. Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022.
- [30] N. F. Liu, T. Zhang, and P. Liang. Evaluating verifiability in generative search engines. *Findings of EMNLP*, 2023.
- [31] S. Liu et al. Open domain question answering with conflicting contexts. *Findings of NAACL*, 2025.

- [32] S. Longpre, K. Perisetla, A. Chen, N. Ramesh, C. DuBois, and S. Singh. Entity-based knowledge conflicts in question answering. *EMNLP*, 2021.
- [33] M.-C. de Marneffe, A. N. Rafferty, and C. D. Manning. Finding contradictions in text. *ACL*, 2008.
- [34] R. T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *ACL*, 2019.
- [35] S. J. Mielke, A. Szlam, E. Dinan, and Y.-L. Boureau. Reducing conversational agents’ overconfidence through linguistic calibration. *Transactions of the ACL*, 10, 2022.
- [36] S. Min, J. Michael, H. Hajishirzi, and L. Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. *EMNLP*, 2020.
- [37] S. Min et al. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. *EMNLP*, 2023.
- [38] I. Nikishina et al. Argument-based comparative question answering evaluation benchmark. arXiv:2502.14476, 2025.
- [39] L. Pan et al. Fact-checking complex claims with program-guided reasoning. *ACL*, 2023.
- [40] A. Panchenko, A. Bondarenko, M. Franzek, M. Hagen, and C. Biemann. Categorizing comparative sentences. *Workshop on Argument Mining*, 2019.
- [41] P. Pezeshkpour and E. Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *Findings of NAACL*, 2024.
- [42] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar. SemEval-2014 Task 4: Aspect based sentiment analysis. *SemEval*, 2014.
- [43] P. Rajpurkar, R. Jia, and P. Liang. Know what you don’t know: Unanswerable questions for SQuAD. *ACL*, 2018.
- [44] H. Rashkin et al. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4), 2023.
- [45] M. Schildwächter, A. Bondarenko, J. Zenker, M. Hagen, C. Biemann, and A. Panchenko. Answering comparative questions: Better than ten-blue-links? *CHIIR*, 2019. arXiv:1901.05041.
- [46] M. Schlichtkrull, Z. Guo, and A. Vlachos. AVeriTeC: A dataset for real-world claim verification with evidence from the web. *NeurIPS Datasets and Benchmarks*, 2023.
- [47] A. Shallouf et al. CAM 2.0: End-to-end open domain comparative question answering system. *LREC-COLING*, 2024.
- [48] Y. Shao, Y. Jiang, T. A. Kanell, P. Xu, O. Khattab, and M. S. Lam. Assisting in writing Wikipedia-like articles from scratch with large language models. *NAACL*, 2024. arXiv:2402.14207.
- [49] I. Stelmakh, Y. Luan, B. Dhingra, and M.-W. Chang. ASQA: Factoid questions meet long-form answers. *EMNLP*, 2022.
- [50] L. Tang, P. Laban, and G. Durrett. MiniCheck: Efficient fact-checking of LLMs on grounding documents. *EMNLP*, 2024.
- [51] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. FEVER: A large-scale dataset for fact extraction and VERification. *NAACL*, 2018.

- [52] TypeSafe AI. Introducing System One models and Jev, 2026. <https://typesafe.ai/blog/introducing-system-one-models-and-jev>. Vendor announcement; latency, price and calibration claims are not independently verified here.
- [53] J. Wallat, M. Heuss, M. de Rijke, and A. Anand. Correctness is not faithfulness in RAG attributions. arXiv:2412.18004, 2024.
- [54] A. Wan, E. Wallace, and D. Klein. What evidence do language models find convincing? *ACL*, 2024.
- [55] H. Wang, A. Prasad, E. Stengel-Eskin, and M. Bansal. Retrieval-augmented generation with conflicting evidence. arXiv:2504.13079, 2025.
- [56] P. Wang et al. Large language models are not fair evaluators. *ACL*, 2024.
- [57] M. Wanner, S. Ebner, Z. Jiang, M. Dredze, and B. Van Durme. A closer look at claim decomposition. **SEM*, 2024.
- [58] J. Wei et al. Long-form factuality in large language models. *NeurIPS*, 2024.
- [59] B. Wen et al. Know your limits: A survey of abstention in large language models. *Transactions of the ACL*, 13, 2025.
- [60] K. Wu, E. Wu, and J. Zou. ClashEval: Quantifying the tug-of-war between an LLM’s internal prior and external evidence. arXiv:2404.10198, 2024.
- [61] J. Xie, K. Zhang, J. Chen, R. Lou, and Y. Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. *ICLR*, 2024.
- [62] R. Xu, X. Qi, Z. Qi, W. Xu, and Z. Guo. DebateQA: Evaluating question answering on debatable knowledge. arXiv:2408.01419, 2024.
- [63] R. Xu et al. Knowledge conflicts for LLMs: A survey. *EMNLP*, 2024.
- [64] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering*, 20(6), 2008.
- [65] G. Yona, R. Aharoni, and M. Geva. Can large language models faithfully express their intrinsic uncertainty in words? *EMNLP*, 2024.
- [66] M. J. Q. Zhang and E. Choi. SituatedQA: Incorporating extra-linguistic contexts into QA. *EMNLP*, 2021.
- [67] C. Zheng, H. Zhou, F. Meng, J. Zhou, and M. Huang. Large language models are not robust multiple choice selectors. *ICLR*, 2024.
- [68] L. Zheng et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *NeurIPS Datasets and Benchmarks*, 2023. arXiv:2306.05685.
- [69] K. Zhou, D. Jurafsky, and T. Hashimoto. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. *EMNLP*, 2023.