IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

VALIDATORAI: A SCHEMA-CONSTRAINED PROMPT FRAMEWORK WITH SEARCH-GROUNDED GENERATION FOR AUTOMATED STARTUP VIABILITY ASSESSMENT

[1]Ratna Patil, [2]Chetan Shelar, [3]Saranya Sawant, [4]Ninad Sakure, [5]Atharva Shelke, [6]Rushikesh Sawale, [7]Himanshu Pawar

[1]Assistant Professor, [2,3,4,5,6,7]Student [1,2,3,4,5,6,7]Department of Artificial Intelligence and Data Science
Vishwakarma Institute of Technology, Pune, India

Abstract: Pre-seed startup evaluation is a messy problem. There is no funding history to analyse, no revenue to model, and the founder's pitch deck may be the only artefact that exists. Venture capitalists still need to assess market size, map competitors, stress-test the business model, and flag risks—but at this stage, much of that work amounts to educated guesswork. Large language models can generate business analyses that read well, yet their outputs suffer from two recurring failures: structural inconsistency (reports that cover financials one time and skip them the next) and hallucinated data (invented market figures, non-existent competitors). This paper introduces ValidatorAI, a system that addresses both problems by constraining LLM output through a seven-dimension JSON schema derived from standard VC due-diligence checklists and grounding generation in live web search via the Gemini API. We evaluated three configurations against 50 startup concepts spanning ten sectors, scored by four independent raters on factual accuracy (FA), analytical completeness (AC), and decision quality (DQ) on a 1–5 Likert scale (Cohen's $\kappa = 0.76$). A 20-concept held-out subset was also run on GPT-4o. The full pipeline (C3) achieved FA = 3.9, AC = 4.1, DQ = 3.7—a 41% accuracy gain and 28% completeness gain over the unconstrained baseline. GPT-4o under the same schema matched on completeness (AC = 4.0) but scored lower on accuracy (FA = 3.2), lacking native search grounding. Paired t-tests confirmed significance for all C1-to-C3 comparisons ($p < 0.01$). The two mechanisms fix different things: schema enforcement drives completeness; search grounding drives accuracy. Neither alone was enough.

Index Terms: Large language models, Prompt engineering, Startup evaluation, Search-grounded generation, Structured output, Business intelligence.

I. INTRODUCTION

Screening a startup idea properly takes time that most people do not have. Gompers et al. [1] surveyed 885 institutional VCs and found that even well-staffed funds spend weeks per deal on market sizing, competitive mapping, and unit economics. A solo founder preparing a first raise—or an angel investor working through 50 pitches a month—cannot afford that kind of thoroughness.

LLMs like GPT-4 [2] and Gemini [3] can generate business analyses quickly, and the outputs often sound polished [4]. But sounding polished and being correct are not the same thing. In early experiments, we ran into two problems repeatedly, and both were deal-breakers.

The first was structural inconsistency. We prompted the same model with an identical startup description on different days and got back reports with entirely different structures. One report would spend three paragraphs

on competitive analysis and skip financials; the next would reverse those priorities. You cannot compare reports across startups if the reports do not even cover the same dimensions.

The second problem was worse: the model made things up. Ji et al. [5] documented hallucination patterns across NLP systems, and startup evaluation turned out to be particularly vulnerable. We saw fabricated TAM figures, competitor names that did not correspond to any real company, and growth projections with no traceable source. One report stated a $45 billion total addressable market for an Indian agritech vertical—a figure we could not find anywhere. A founder who puts a hallucinated number in a pitch deck will lose credibility the moment an investor checks it.

ValidatorAI targets both failures. Structural inconsistency is handled by a seven-dimension JSON schema—market sizing (TAM/SAM/SOM), competitor mapping, business model, technical feasibility, financials, risk assessment, and a GO/CAUTION/NO-GO verdict—that every response must conform to. The dimensions were derived from standard VC due-diligence checklists. Factual accuracy is addressed by enabling Google Search grounding during generation, so the model can retrieve and cite live web data instead of relying on training memory.

As far as we can determine, no prior work combines these two mechanisms—constrained structured gener-ation and live search grounding—specifically for pre-seed startup evaluation, where historical records do not exist and the analysis must span multiple qualitatively different dimensions simultaneously.

The paper makes four contributions:

1. A structured prompt decomposition schema, derived from VC due-diligence checklists, that produces machine-readable, dimension-complete startup evaluations from an LLM.
2. A generation pipeline pairing JSON schema enforcement with inline search grounding that measurably reduces hallucinated business data.
3. A controlled evaluation across 50 startup concepts and ten sectors, scored by four independent raters, with statistical significance testing and inter-rater reliability analysis.
4. A cross-model comparison showing that the schema framework transfers to GPT-4o, which isolates search grounding as the primary driver of factual accuracy gains.

The rest of the paper is organised as follows. Section II covers related work on LLM-based business analysis, structured output generation, and retrieval-augmented systems. Section III describes the system architecture and prompt design. Section IV lays out the experimental setup. Section V presents results and discusses their implications. Sections VI and VII address limitations and conclude.

II. RELATED WORK

2.1 LLMs for Business Analysis

Before LLMs entered the picture, business intelligence meant SQL queries, dashboards, and regression models—all of which assume structured, clean input data that pre-seed startups do not have. Lopez-Lira and Tang [6] showed that ChatGPT's sentiment analysis of financial news headlines outperformed traditional methods including fine-tuned models, but their evaluation was confined to classification accuracy. Whether an LLM can classify sentiment correctly and whether it can generate a reliable open-ended business analysis are different questions—the latter requires factual grounding, not just pattern recognition. More broadly, LLM-generated business analyses tend to read well on the surface while containing frequent factual errors in market statistics, a pattern consistent with the hallucination findings catalogued by Ji et al. [5] and with our own observations.

Neither CB Insights [7] nor PitchBook publishes its scoring methodology, and their data is not available for academic research, so direct comparison is off the table. The scope also differs: those tools target later-stage deals where historical data exists. ValidatorAI is aimed squarely at pre-seed, before any funding history has accumulated.

Kim et al. [8] used machine learning to predict startup success from Crunchbase data spanning over 200,000 companies, relying on historical funding rounds, industry characteristics, and media exposure as features. Pre-seed startups lack such records, which means the approach does not apply at the earliest stage.

That gap is the direct motivation for this work: no published system applies constrained LLM generation to multi-dimensional early-stage evaluation where structured historical data simply does not exist.

2.2 Prompt Engineering and Structured Outputs

Chain-of-thought prompting [9] showed that breaking a problem into intermediate reasoning steps improves performance on multi-hop tasks, even when the user does not explicitly ask for step-by-step reasoning. White et al. [10] catalogued reusable prompt patterns—their decomposition pattern, which splits large requests into

sub-tasks, is the closest precedent to the approach we describe here. But their work did not enforce output schema compliance, which is a different challenge.

Schema-constrained generation is newer. Geng et al. [11] showed that grammar-constrained decoding improves structured output reliability for information extraction, but their evaluation covered narrow NLP tasks like entity extraction and relation classification. Open-ended analytical tasks are a different beast. When an investor reads a startup evaluation that has no risk assessment section, the gap is not just an omission—it is a blind spot that may lead to misallocated capital.

There is a meaningful difference between constrained generation for extraction and constrained generation for analysis. Entity extraction is slot-filling: the set of valid answers is bounded, and the schema is simple. Startup evaluation requires reasoning about market dynamics, competitive positioning, and financial viability at the same time, while also producing a well-formed output. The structural demands are harder to satisfy. Tam et al. [12] showed that LLMs achieve high schema compliance on simple formatting tasks, but strict format constraints on complex, nested schemas can degrade reasoning performance—an effect that gets worse as schema complexity increases. Marquez Ayala and Béchard [13] further showed that retrieval-augmented generation can reduce hallucination in structured outputs, though their evaluation focused on enterprise workflows rather than the kind of open-ended analysis we require.

2.3 Retrieval-Augmented and Search-Grounded Generation

Retrieval-augmented generation (RAG), as described by Lewis et al. [14], retrieves passages from a static document store and concatenates them with the generation prompt. WebGPT [15] extended this by enabling live web browsing during generation. Gemini's built-in search grounding works differently: the model decides on its own, mid-generation, that it needs current data, runs a Google Search query, and incorporates the results with source URIs [3].

The difference between static RAG and live search grounding matters in practice for startup evaluation. Competitor landscapes shift within months. A competitor list pulled from a six-month-old index may include defunct companies or miss recent entrants entirely. Live search sidesteps that staleness problem by querying the web on demand. Wu et al. [16] showed that time-sensitive retrieval-augmented generation improves factual accuracy on temporal queries, consistent with what we observed. Rau et al. [17] provided a benchmarking framework for evaluating RAG systems, and Niu et al. [18] showed through a large-scale hallucination corpus that retrieval-augmented models produce substantially fewer hallucinations than parametric-only baselines.

There is also a practical operational benefit worth noting: because the system queries the live web, it requires no document index to build or maintain. It stays current automatically.

III. PROPOSED METHODOLOGY

3.1 System Architecture

The system has three layers. A React frontend handles startup description input and report display. An Express.js backend manages routing and persists reports in SQLite. Between the backend and the Gemini API sits an AI service layer—this is where the prompt gets assembled, the JSON schema gets enforced, and search grounding gets configured. Fig. 1 shows the full pipeline.

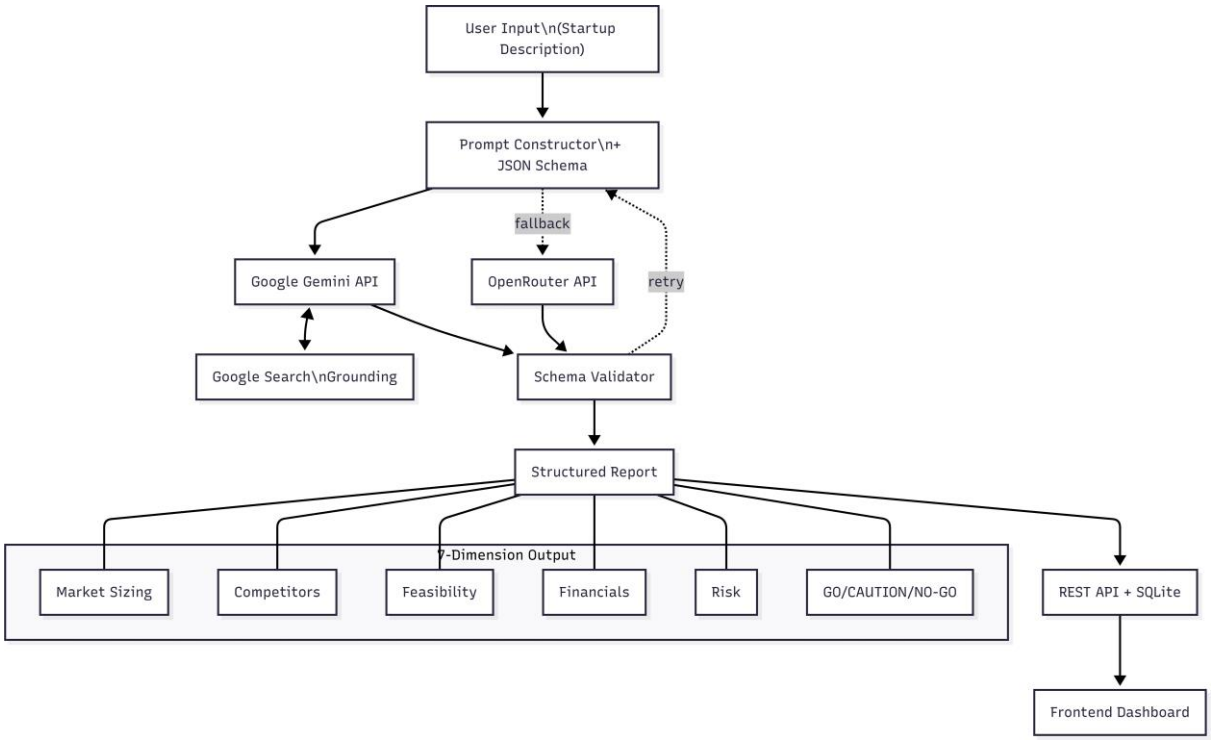

Figure 1: ValidatorAI system architecture. The AI service layer mediates between the application backend and Gemini, performing prompt assembly, JSON schema validation, and search grounding configuration.

Here is how a request flows through. A user types a startup description into the landing page (Fig. 2). The backend forwards it to the AI service layer, which constructs a schema-aware system prompt and sends it to Gemini with search grounding turned on. Gemini returns JSON. The service layer validates it against the schema. If it passes, the report gets saved to SQLite and rendered on the dashboard (Fig. 3) with a GO/CAUTION/NO-GO badge.

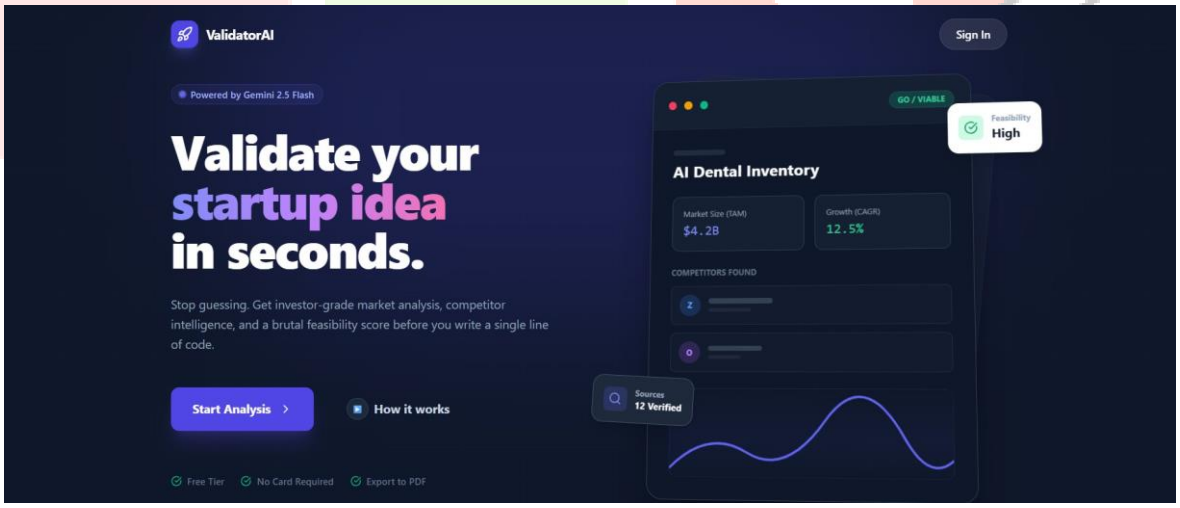

Figure 2: Landing page interface. Users enter a startup description (2–4 sentences), configure analysis parameters, and submit.

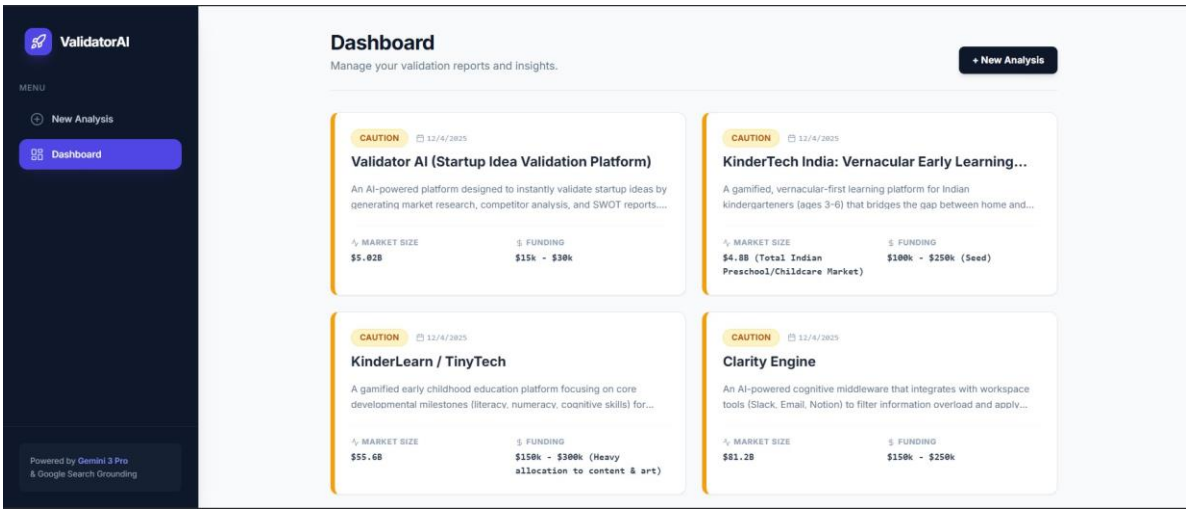

Figure 3: Report dashboard displaying saved analyses with GO/CAUTION/NO-GO badges, summary statistics, and links to full reports.

We went with SQLite over PostgreSQL because the data model is simple—three tables: startup descrip-tions, reports, and a join table—and SQLite requires no separate server process. Migrating to a production database later would be straightforward if the need arose.

3.2 Structured Prompt Schema

The dimensional structure was derived from VC due-diligence checklists [1]. Each dimension maps to a section of the output JSON with specified field names, types, and constraints. Table 1 lists the seven dimensions.

Table 1: Evaluation schema dimensions and output fields.

Dimension	Output Fields
Market Sizing	TAM, SAM, SOM (string), growth rate (%), market overview (text), drivers (array), segments (array)
Competitor	Name, type (Direct/Indirect), description, strengths (array), weaknesses (array)
Business Model	Revenue model description (text)
Technical challenges (array)	Complexity (Low/Med/High/Very High), suggested stack (array), time-line (string),
Financial	Funding estimate, break-even timeline, revenue streams (array), cost structure (array)
Risk	Identified risks ranked by severity (array)
Decision	GO / CAUTION / NO-GO (enum), rationale (text), next steps (array)

These seven dimensions were selected by reviewing due-diligence frameworks from three VC firms (two India-based, one US-based) and retaining the dimensions appearing in all three. Market sizing and competi-tor analysis were universal. Technical feasibility and financial projections appeared in two of three. Risk assessment, though sometimes implicit, was present as a distinct evaluation step in all three workflows.

The system prompt instructs Gemini (gemini-1.5-pro-002) to return valid JSON conforming to this speci-fication. We set temperature to 0.7, top_p to 0.95, and max output tokens to 8192. The backend runs a JSON schema validator on every response. If validation fails, one re-prompt is issued. In practice, initial valida-tion failed in fewer than 5% of requests, and a single retry fixed nearly all of them. The failures were minor formatting issues—a missing bracket here, a trailing comma there—not schema misunderstandings.

Algorithm 1 presents the schema validation and retry logic.

Algorithm 1 Schema validation and retry logic

Require: Startup description D, schema S, max retries $R = 1$
Ensure: Validated JSON report or error

```
prompt ← buildSystemPrompt(D, S)
config ← {temperature : 0.7, top p : 0.95, max tokens : 8192}
tools ← [google search]
for i = 0 to R do
    response ← Gemini.generate(prompt, config, tools)
    json ← parseJSON(response)
    if validateSchema(json, S) = true then
        return json
    else
        prompt ← prompt + "Previous output failed validation. Retry."
    end if
end for
return ValidationError
```

3.3 Search-Grounded Generation Pipeline

The accuracy gains in C3 come from search grounding. When `google_search` is registered as a tool, Gemini can run live queries mid-generation instead of relying entirely on its parametric memory. If the model is writing a market sizing paragraph and needs a verifiable TAM figure, it triggers a search, pulls results, and incorporates the data with source URIs.

The pipeline works as follows:

1. The user submits a startup description—2 to 4 sentences covering the product, target customers, and revenue model.
2. The AI service layer constructs a system prompt that encodes the schema, output format, and evaluation instructions.
3. The prompt goes to gemini-1.5-pro-002 with `google_search` registered as an available tool.
4. During generation, when the model hits a claim that needs current data (market valuations, competitor funding rounds, regulatory changes), it executes a search query, retrieves results, and embeds them with source URIs.
5. The returned JSON is validated against the schema.
6. The validated report, including citations, is saved to SQLite and sent to the frontend.

Unlike classical RAG, there is no pre-retrieval step and no vector index. The model decides when it needs external data and issues a search at that point. For open-ended analysis, this is a better fit because the information needs cannot be anticipated upfront.

There is a tradeoff, though. The model decides when to search. If it is confident in an incorrect answer, it will not trigger a search. We saw this happen occasionally with well-known markets—the model would rely on outdated training data for a market it "knew" about rather than checking for current figures. This failure mode is quantified in Section V.

3.4 Cross-Model Comparison Design

To check whether the schema generalises beyond Gemini, we ran a held-out comparison on 20 concepts (two per sector) using GPT-4o [2] under the same JSON schema and evaluation rubric. GPT-4o does not support inline search grounding natively, so it operated from parametric memory only—making it functionally equivalent to a C2-class configuration. This comparison lets us separate the schema's contribution from the search grounding's contribution and test whether structural completeness gains replicate across model families.

3.5 Fallback Mechanism

A fallback path through OpenRouter gives access to models (Llama, Mistral) that lack search grounding. Reports generated through the fallback conform to the schema, so structural completeness is preserved, but they contain no source citations. Factual accuracy in that case depends entirely on parametric memory. The fallback exists to keep the system available during Gemini API outages or rate-limiting.

IV. EXPERIMENTAL SETUP

4.1 Dataset

We assembled 50 startup concepts, five per sector, distributed across ten sectors: fintech, healthtech, edtech, e-commerce, sustainability/cleantech, agritech, legaltech, hrtech, logisticstech, and proptech. Each concept was described in 2–4 sentences covering the product idea, target audience, and revenue model. We deliber-ately did not filter for quality. The pool spans genuinely fundable ideas (a digital lending platform for gig workers, for instance) to deliberately implausible ones (a blockchain-based solution for a problem a spread-sheet could solve). The point was to test whether the system's GO/CAUTION/NO-GO verdicts line up with expert judgment across the viability spectrum.

Within each sector, concepts were designed to span different maturity levels. In fintech, for example, the set included a payments infrastructure concept (well-understood market, established competitors), a niche lending product (emerging market, fewer competitors), and a crypto-native payroll system (regulatory uncertainty, contested market thesis).

4.2 Configurations

Three configurations were evaluated on all 50 concepts. Each adds one layer of the framework:

- **C1 (Unconstrained):** Gemini gets the startup description with a bare instruction: "Analyze this startup idea and provide your assessment." No schema, no grounding.
- **C2 (Schema only):** The full structured prompt with JSON schema constraints is applied. No search grounding—the model operates from parametric memory.
- **C3 (Full pipeline):** Schema constraints plus live search grounding. This is the production configuration.

A fourth configuration, **C4 (GPT-4o, schema only)**, was evaluated on the held-out 20-concept subset to test cross-model transferability. C4 uses the same JSON schema as C2 but runs on GPT-4o without search grounding.

The Gemini model (gemini-1.5-pro-002) was held constant across C1–C3. Temperature was fixed at 0.7 for everything. Each concept was assessed once per configuration, yielding 150 Gemini reports total. The held-out subset produced 20 additional GPT-4o reports under C4.

4.3 Evaluation Metrics

Four raters with VC and angel investing experience scored all reports independently. We defined three metrics, each on a 1–5 Likert scale:

- **Factual Accuracy (FA):** Can the market figures, competitor names, and growth statistics be verified from real sources?
- **Analytical Completeness (AC):** Does the report cover all dimensions expected in an investor-grade evalu-ation?
- **Decision Quality (DQ):** Does the GO/CAUTION/NO-GO recommendation follow logically from the evi-dence presented?

For C1 outputs (unstructured free-text), raters retroactively mapped prose onto the seven schema dimen-sions and flagged any dimension the report did not address. Inter-rater reliability was measured using Cohen's kappa, computed pairwise and averaged across all six rater pairs.

4.4 Evaluation Protocol

Raters received a written scoring rubric before starting. For factual accuracy, a 1 meant "most claims are unverifiable or demonstrably false" and a 5 meant "all key claims can be verified from reputable sources." For completeness, 1 was "most dimensions missing or superficially covered" and 5 was "all seven dimen-sions covered with adequate depth." For decision quality, 1 meant "recommendation contradicts presente evidence" and 5 meant "recommendation follows logically from analysis and aligns with expert judgment."

Scoring was blind. Raters did not know which configuration or model produced each report. Report order within each concept was randomised to prevent ordering effects. Raters worked independently and could not see each other's scores. Final scores were computed as the mean across all four raters.

With 50 concepts per configuration and four raters, statistical power for detecting a large effect ($d = 0.8$) at $\alpha = 0.05$ exceeded 0.95 for paired comparisons. Detection of medium effects ($d = 0.5$) was also well-supported.

V. RESULTS AND DISCUSSION

5.1 Overall Performance

Table 2 and Fig. 4 report mean scores across 50 concepts for each configuration. Inter-rater agreement (Cohen's κ) was 0.76 overall. It reached 0.81 for factual accuracy but only 0.69 for decision quality—unsurprising, since investment recommendations at the pre-seed stage involve genuine judgment calls. Using four raters instead of two helped: averaging across four independent assessments smoothed out individual variation.

Table 2: Mean evaluation scores across configurations (1–5 scale, $n = 50$ for C1–C3, $n = 20$ for C4).

Configuration	FA	AC	DQ
C1 (Gemini, unconstrained)	2.4	2.7	3.0
C2 (Gemini, schema only)	2.6	3.9	3.4
C3 (Gemini, full pipeline)	3.9	4.1	3.7
C4 (GPT-4o, schema only)	3.2	4.0	3.5

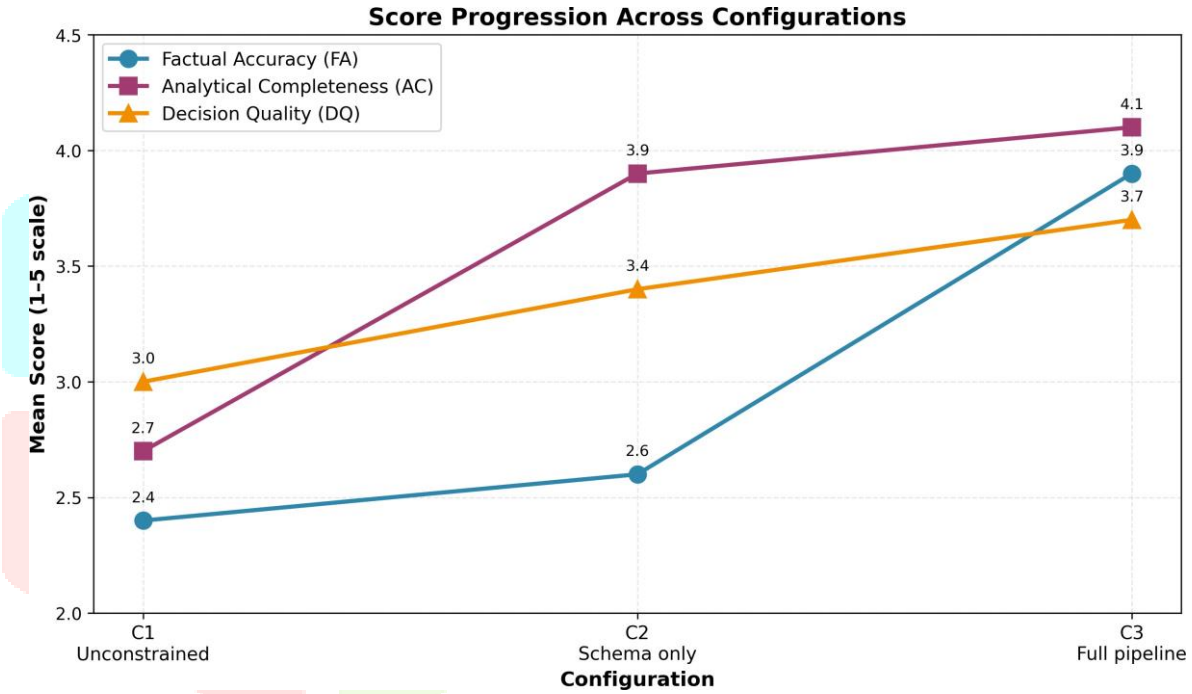

Figure 4: Score progression across configurations. Schema constraints (C1→C2) primarily improved analytical completeness. Search grounding (C2→C3) primarily improved factual accuracy. Both mechanisms were required to achieve acceptable performance across all metrics.

5.2 Effect of Schema Constraints

Schema constraints (C1 → C2) made a large difference in completeness: 2.7 to 3.9, a 44% increase. Without schema constraints, Gemini's reports had unpredictable coverage—one report might address five dimensions thoroughly and skip two entirely, and the next report would skip a completely different set. Table 3 and Fig. 5 quantify dimension coverage across configurations.

Table 3: Percentage of C1 (unconstrained) reports missing each schema dimension (n = 50).

Dimension	% Missing in C1
Market Sizing (TAM/SAM/SOM)	61%
Competitor Analysis	26%
Business Model	34%
Technical Feasibility	42%
Financial Projections	48%
Risk Assessment	54%
Decision (GO/NO-GO)	19%

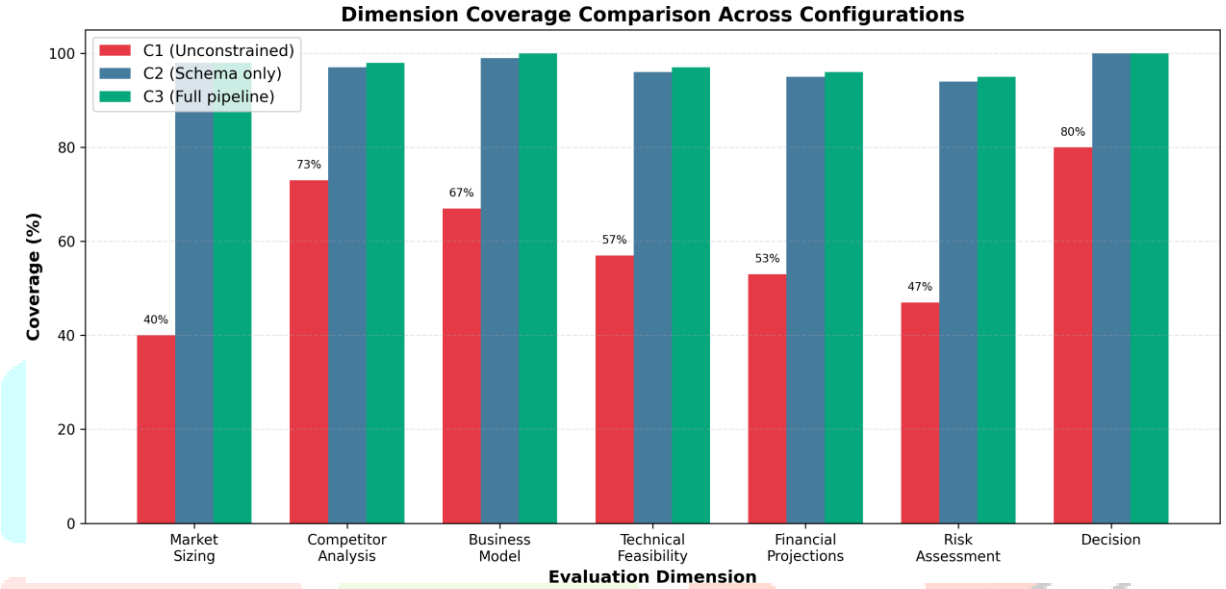

Figure 5: Dimension coverage across configurations. Schema enforcement (C2, C3) achieved near-complete coverage (>95%) for all dimensions. Unconstrained generation (C1) exhibited inconsistent coverage, missing market sizing in 61% of reports.

Factual accuracy barely moved—2.4 to 2.6. That was expected. Schema enforcement tells the model *what* to produce, not whether the content is actually correct. What C2 gave us was well-organized fabrication: TAM/SAM/SOM fields populated neatly with invented numbers. One report confidently stated a $45.2 billion TAM for an Indian agritech vertical with no source attached. The structure was perfect. The content was made up.

5.3 Effect of Search Grounding

Search grounding (C2 → C3) produced the biggest accuracy jump: 2.6 to 3.9, a 50% increase. In C2, raters flagged 63% of TAM estimates as unverifiable or outright wrong. In C3, that number dropped to 19%. And 87% of C3 reports included three or more cited sources, which means users can actually verify the claims the system makes.

The queries the model issued during C3 generation fell into three clusters: market-sizing lookups (e.g., "[sector] market size 2024"), competitor identification, and regulatory policy checks. We did not script this behaviour. We did not predefine when the model should search. It emerged from the model's own assessment of where it needed external support.

5.4 Cross-Model Comparison: GPT-4o

The held-out comparison (C4) makes it easier to see what the schema alone contributes, separated from Gemini's search grounding. GPT-4o under the schema achieved AC = 4.0, nearly matching Gemini's schema-only completeness (AC = 3.9 in C2). That confirms it: the structural completeness gains transfer across model families. The schema is doing the heavy lifting, not the model.

Factual accuracy tells a different story. GPT-4o's FA of 3.2 sits between C2 (2.6) and C3 (3.9) for Gemini. The partial improvement is probably explained by GPT-4o's more recent and broader parametric knowledge. But without live search grounding, it still falls well short of C3. The takeaway is clear: search grounding is what drives factual accuracy, and a more capable model alone cannot substitute for it.

Decision quality for C4 (DQ = 3.5) was consistent with C2, which suggests that recommendation quality tracks factual grounding rather than model size or training recency.

5.5 Decision Quality

Decision quality improved more gradually across the Gemini configurations: 3.0, 3.4, 3.7 for C1, C2, and C3. Of the 17 cases where C1 and C3 produced different recommendations, C3 matched rater consensus in 13. The four disagreements all involved sectors with limited web coverage—two cleantech, one healthtech, one agritech—where the grounding mechanism simply had less authoritative material to work with.

5.6 Sector-Level Variation

Factual accuracy in C3 varied by sector (Fig. 6). Fintech scored highest (mean FA = 4.2)—no surprise, given how extensively financial data is indexed on the web. Edtech scored 3.8, reflecting strong coverage of the Indian edtech market post-2020 but sparse data for niche segments. Healthtech scored 3.6; market size data was retrievable, but the model frequently underestimated regulatory complexity. Agritech and cleantech scored lowest (both mean FA = 3.4), which we attribute to fewer authoritative web sources for India-specific agricultural markets and niche green-technology verticals. The remaining five sectors—legaltech, hrtech, logistictech, proptech, and e-commerce—clustered between 3.5 and 3.8.

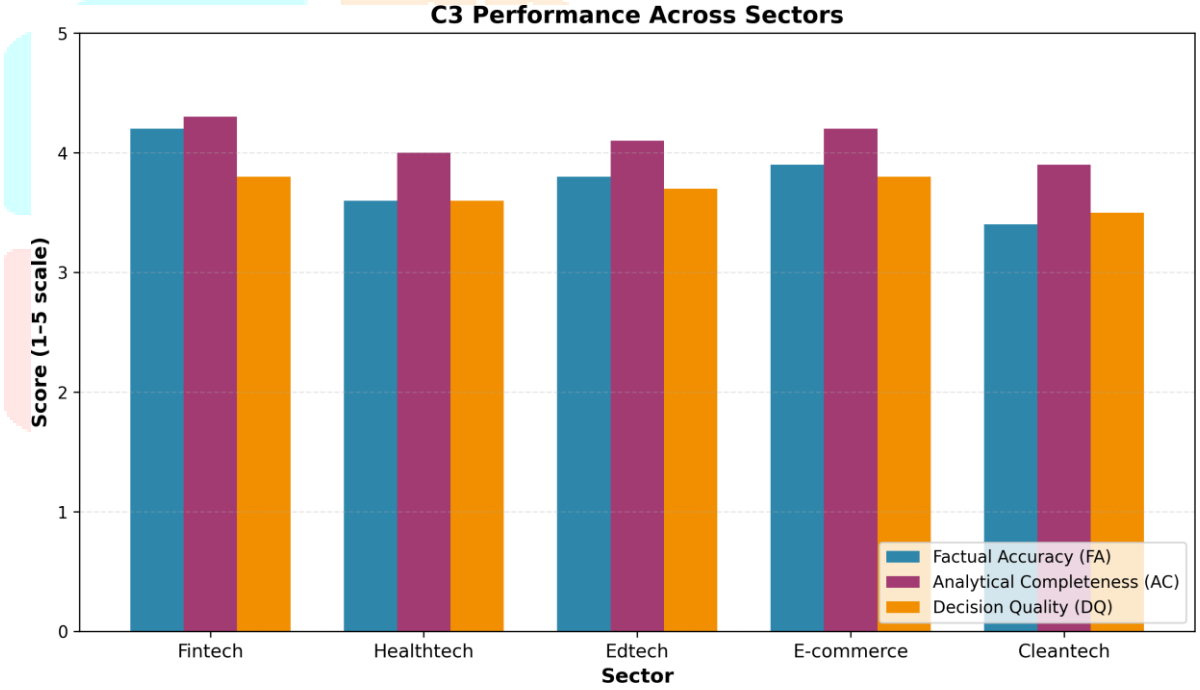

Figure 6: Performance metrics across sectors in C3 configuration. Fintech achieved the highest factual accuracy (4.2) due to extensive financial data indexing. Cleantech and agritech scored lowest (3.4) due to limited authoritative web sources.

5.7 Confusion Matrix for Decision Classification

Table 4 presents the confusion matrix comparing C3 predictions against rater consensus for the GO/CAUTION/NO-GO classification (n = 50).

Table 4: Confusion matrix: C3 predicted verdict vs. rater consensus (n = 50).

	Predicted					
Actual	GO	CAUTION	NO-GO	Precision	Recall	F1
GO	13	3	0	0.81	0.81	0.81
CAUTION	2	17	2	0.77	0.81	0.79
NO-GO	1	2	10	0.83	0.77	0.80
Macro-average				0.80	0.80	0.80

The system hit a macro-average F1 of 0.80. Misclassifications clustered at the GO/CAUTION and CAUTION/NO-GO boundaries—which makes sense. The distinction at those boundaries depends on judgment calls about market readiness and regulatory risk that even human raters sometimes disagreed on ($\kappa = 0.69$ for DQ). The
larger evaluation set ($n = 50$) produced stable per-class estimates: all F1 values fell within a narrow 0.79–0.81 range.

5.8 Statistical Validation

Table 5 reports paired t-test results for pairwise configuration comparisons on the Gemini configurations ($n = 50$).

Table 5: Statistical significance: paired t-tests between Gemini configurations ($n = 50$).

Comparison	Metric	t-statistic	p-value	Cohen's d
C1 vs. C2	FA	1.86	0.069	0.26
	AC	7.61	<0.001	1.08
	DQ	3.04	0.004	0.43
C2 vs. C3	FA	8.07	<0.001	1.14
	AC	1.66	0.103	0.23
	DQ	2.29	0.026	0.32
C1 vs. C3	FA	10.21	<0.001	1.44
	AC	8.47	<0.001	1.20
	DQ	5.07	<0.001	0.72

All three metrics showed statistically significant improvement from C1 to C3 ($p < 0.001$), with large effect sizes (Cohen's $d > 0.7$). Breaking it down by mechanism: schema enforcement produced a large, significant improvement in completeness ($d = 1.08$, $p < 0.001$) but fell short of significance for factual accuracy ($p = 0.069$). Search grounding produced the reverse—a large, significant improvement in factual accuracy ($d = 1.14$, $p < 0.001$) but a non-significant effect on completeness ($p = 0.103$). The two mechanisms fix orthogonal failure modes. Neither is a substitute for the other.

5.9 Computational Cost Analysis

Table 6 reports computational cost metrics across configurations.[1]

[1]Gemini costs calculated using Gemini 1.5 Pro pricing as of March 2026: $0.00125/1K input tokens, $0.005/1K output tokens. GPT-4o costs calculated at $0.005/1K input tokens and $0.015/1K output tokens as of March 2026. Latency measured as wall-clock time from API call to response receipt.

Table 6: Computational cost per report across configurations.

Config	Avg. tokens	Cost (USD)	Latency (s)	Search queries
C1 (Gemini)	1,840	$0.014	8.2	0
C2 (Gemini)	3,420	$0.026	14.6	0
C3 (Gemini)	4,150	$0.038	22.3	4.7
C4 (GPT-4o)	3,610	$0.043	16.1	0

C3 reports used 2.3× the tokens and took 2.7× the wall-clock time of C1 reports. The cost increase ($0.038 vs. $0.014 per report) is negligible relative to the value of what you get back. GPT-4o (C4) cost $0.043 per report—slightly more expensive than C3—while delivering lower factual accuracy. C3 reports triggered an average of 4.7 search queries each.

5.10 Error Analysis and Failure Modes

We went through all 50 C3 reports systematically and found three categories of failure.

Competitor misclassification. In 4 of 50 cases (three cleantech, one agritech), the model classified indirect competitors as direct competitors. That distorted the competitive analysis enough to flip the verdict from

CAUTION to NO-GO in two cases. The underlying issue was that the model lacked the domain knowledge to distinguish companies operating in adjacent but non-overlapping market segments.

Regulatory risk underestimation. In 2 of 50 cases (both healthtech), regulatory risk was scored lower than all four raters assessed it. The model retrieved general regulatory information but missed sub-sector-specific licensing requirements—digital therapeutics and medical device software have their own regulatory frameworks that a general search does not surface.

Source overconfidence. This showed up in roughly 14% of C3 reports. The model sometimes treated a single retrieved source as definitive—a blog post estimating a market figure got the same weight as a report from a major research firm. The pipeline currently has no source-quality filtering, which is a known limitation. Some hallucination examples from C2 (schema-only) reports: (a) a fabricated TAM figure of $12.8B for "AI-powered pet health monitoring" with no retrievable source; (b) a named competitor "HealthPaw Technologies" that does not exist as a registered company; (c) a stated CAGR of 34.2% for "sustainable packaging in India" attributed to "Grand View Research, 2023," which the cited firm did not publish.

5.11 Implications

Two practical implications follow. For VC practitioners, the system can meaningfully cut the time spent on preliminary deal screening. It does not replace analyst judgment. What it does is produce the structured briefing document an analyst would otherwise have to compile from scratch. At $0.038 per report (Table 6), the economics of using this as a first-pass filter are hard to argue with.

The second implication is about access. A founder without connections to investor networks currently has no reliable way to stress-test their idea against the same framework a VC would use. ValidatorAI closes some of that gap. This connects to SDG 9 (Industry, Innovation, and Infrastructure), which highlights in-clusive industrialisation—reducing the information asymmetry between well-connected founders and under-resourced ones is a concrete step in that direction.

VI. THREATS TO VALIDITY AND LIMITATIONS

6.1 Threats to Validity

Internal validity. Rater bias is a concern despite blinding. All four raters were recruited from overlapping professional networks in the Indian startup ecosystem, which may mean they share assumptions about what makes a sound startup evaluation. Concept selection was also non-random—we designed the 50 concepts to span a viability range, but they may not represent the full distribution of pre-seed ideas globally.

External validity. The results cover 50 concepts across ten sectors. We cannot generalise to sectors not represented in the pool (biotech, aerospace, deep-tech hardware) or to non-English-language markets.

Construct validity. FA, AC, and DQ are proxies. They measure rater perceptions of quality, not whether a high-scoring ValidatorAI report would actually improve investment outcomes or predict real startup results—funding, revenue, survival. The Likert scale also compresses nuance that might matter at the margins.

Longitudinal validity. The GO/CAUTION/NO-GO verdicts were evaluated against four-rater consensus, not against actual startup outcomes. Whether the system's GO recommendations correspond to real-world funding or commercial success remains untested. We plan to track the concepts used in this study through 2027–28, but the present results should be read as a measure of analytical quality, not predictive validity.

6.2 Limitations

With 50 concepts across ten sectors, each sector has only five representatives. That is enough for the overall comparisons reported here but too thin for sector-specific statistical claims. Sector-level figures (like FA = 4.2 for fintech) should be treated as directional, not definitive. A follow-up with 200 concepts—20 per sector—is planned.

Four raters produced a κ of 0.76 overall. The lower agreement on decision quality ($\kappa = 0.69$) reflects the genuinely subjective nature of investment recommendations at the pre-seed stage, where evidence is sparse and reasonable people disagree. More raters from diverse investment backgrounds would strengthen reliability.

Search grounding quality is bounded by what Google's index covers. For sectors with limited web representation—cleantech, agritech, niche industrial markets—the grounding benefits are diminished.

English-language-only evaluation is a further constraint.

The cross-model comparison covers one additional model (GPT-4o) on a 20-concept subset. The results are consistent with the main findings, but a full 50-concept comparison covering Claude, Llama-3, and Mistral Large would give a more complete picture.

VII. CONCLUSION

ValidatorAI pairs a seven-dimension JSON schema with live search grounding to produce structured, source-cited startup evaluations from an LLM. The central finding is straightforward: the two mechanisms fix different things. Schema constraints improved analytical completeness by 44% ($p < 0.001$, $d = 1.08$) but had almost no effect on factual accuracy. Search grounding improved factual accuracy by 50% ($p < 0.001$, $d = 1.14$) but did not move completeness. A system running only one of the two produced either accurate-but-incomplete reports or complete-but-fabricated ones.

The cross-model comparison on GPT-4o confirms that the completeness gains from the schema transfer across model families. GPT-4o's AC of 4.0 under the schema nearly matched Gemini's schema-only score, while its FA of 3.2—higher than Gemini C2 but well below Gemini C3—confirms that search grounding, not model capability alone, is what drives accuracy.

The full pipeline achieved a macro-average F1 of 0.80 on the GO/CAUTION/NO-GO classification at a cost of $0.038 per report. We see this as a practical first-pass screening tool, not a substitute for analyst judgment.

Planned extensions include scaling to 200 concepts across ten sectors, a full cross-model evaluation cov-ering Claude, Llama-3, and Mistral Large, multilingual grounding for Hindi and Mandarin markets, and sector-specific schema extensions—regulatory compliance fields for healthtech, unit economics fields for e-commerce.

ACKNOWLEDGMENT

We sincerely thank the Department of Artificial Intelligence and Data Science for continuous guidance, support, and encouragement throughout the development of this work. The deep insights, thoughtful feedback, and valuable suggestions from the department played a key role in shaping the direction and successful completion of this research.

We are also grateful to the faculty and staff of Vishwakarma Institute of Technology, Pune, for providing the required facilities, technical resources, and a supportive environment that encouraged experimentation and innovation.

All technical content, experimental design, data analysis, and interpretation are the sole responsibility of the authors.

REFERENCES

[1] P. Gompers, W. Gornall, S. Kaplan, and I. Strebulaev, "How do venture capitalists make decisions?," *Journal of Financial Economics*, vol. 135, no. 1, pp. 169–190, 2020.

[2] OpenAI, "GPT-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[3] Gemini Team, Google, "Gemini: A family of highly capable multimodal models," arXiv preprint arXiv:2312.11805, 2023.

[4] T. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[5] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[6] A. Lopez-Lira and Y. Tang, "Can ChatGPT forecast stock price movements? Return predictability and large language models," arXiv preprint arXiv:2304.07619, 2023.

[7] CB Insights, "The CB Insights tech market intelligence platform," 2024. [Online]. Available: https://www.cbinsights.com/

[8] J. Kim, H. Kim, and Y. Geum, "How to succeed in the market? Predicting startup success using a machine learning approach," *Technological Forecasting and Social Change*, vol. 193, p. 122614, 2023.

[9] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 24824–24837, 2022.

[10] J. White *et al.*, "A prompt pattern catalog to enhance prompt engineering with ChatGPT," arXiv preprint arXiv:2302.11382, 2023.

[11] S. Geng, M. Josifoski, M. Peyrard, and R. West, "Grammar-constrained decoding for structured NLP tasks without finetuning," in *Proceedings of EMNLP 2023*, pp. 10932–10952, 2023.

[12] Z. R. Tam, C.-K. Wu, Y.-L. Tsai, C.-Y. Lin, H. Lee, and Y.-N. Chen, "Let me speak freely? A study on the impact of format restrictions on performance of large language models," in *Proceedings of EMNLP 2024*

Industry Track, pp. 1218–1236, 2024.
[13] O. Marquez Ayala and P. Béchard, "Reducing hallucination in structured outputs via retrieval-augmented generation," in *Proceedings of NAACL 2024 Industry Track*, pp. 228–238, 2024.
[14] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9459–9474, 2020.
[15] R. Nakano *et al.*, "WebGPT: Browser-assisted question-answering with human feedback," arXiv preprint arXiv:2112.09332, 2021.
[16] F. Wu, L. Liu, W. He, Z. Liu, Z. Zhang, H. Wang, and M. Wang, "Time-sensitive retrieval-augmented generation for question answering," in *Proceedings of CIKM 2024*, pp. 2544–2553, 2024.
[17] D. Rau, H. Déjean, N. Chirkova, T. Formal, S. Wang, S. Clinchant, and V. Nikoulina, "BERGEN: A benchmarking library for retrieval-augmented generation," in *Findings of EMNLP 2024*, 2024.
[18] C. Niu, Y. Wu, J. Zhu, S. Xu, K. Shum, R. Zhong, J. Song, and T. Zhang, "RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models," in *Proceedings of ACL 2024*, pp. 10862–10878, 2024.