# On Generative Search Optimisation

AAI Labs

## 1 Introduction

Generative search is transforming online information discovery - forecasts suggest that by 2027 more users will rely on generative systems than on classical search engines[1]. Approximately 37% of prompts ask systems to create, explain, or synthesise information rather than retrieve a single document[2].

These systems replace ranked lists with single composed responses. Interaction is faster and often sufficient. Visibility now depends on whether content is selected, summarised, or omitted during generation. This changes both information consumption and production.

This report examines ChatGPT, Perplexity, Copilot, and Claude. Recent leaks of system instructions for many generative assistants (see Appendix B for some examples) demonstrate that each platform operates under explicit hidden directives that shape how queries are interpreted and how facts are presented. We focus on how they retrieve information, combine retrieval with model reasoning, and how these choices shape visible content. From this, we derive optimisation strategies which we implement in custom tools presented herein.

### 1.1 Purpose and scope

This report addresses the question: how do generative search engines retrieve, reason over, and surface information, and how can this understanding improve visibility? Answering it provides a basis for moving from classical SEO to generative engine optimisation (GEO).

We start with observations motivating a revised optimisation approach. Next, we outline technical foundations of generative search, focusing on LLMs augmented with live web retrieval. In §2, we discuss general discoverability principles and platform-specific effects driven by underlying search providers.

In §3, we present a generative search framework and show, through inspection of network interactions, that it reflects at least one production system. We conclude in §4 with tools and experimental infrastructure for testing optimisation strategies together with our findings.

### 1.2 Motivating observations

Generative search increasingly relies on multi-source retrieval pipelines rather than a single index. These pipelines aggregate results from traditional search engines, third-party APIs, and proprietary indices. ChatGPT, for example, combines Bing with SerpApi[3]. Relevance emerges from interactions among sources with different coverage, biases, and refresh rates. Understanding source selection and combination is essential for optimisation.

[1] *When Will AI Search Beat Google? 2025–2030 Forecast - TTMS - ttms.com.* https://ttms.com/my/llm-powered-search-vs-traditional-search-2025-2030-forecast [Accessed 14-01-2026]

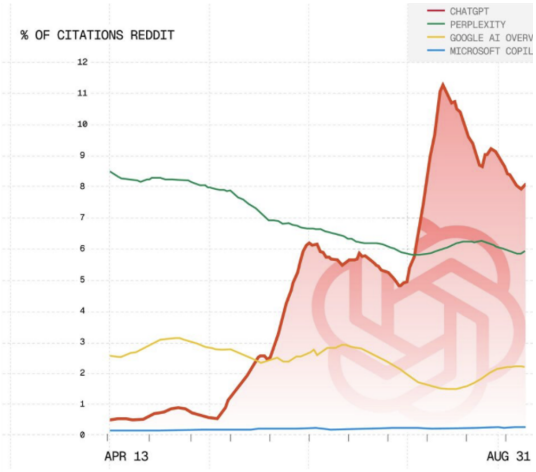
[2] *I analyzed 40 million search results - here's what i found.* Conference slides on AI search and citation behavior, 2025. https://speakerdeck.com/joshbly/josh-blyskal-profound-i-analyzed-40-million-search-results-heres-what-i-found [Accessed 14-01-2026]

[3] *ChatGPT Reportedly Using SerpApi to Scrape Google Search Results.* Stan Ventures, Oct 29 2025. https://www.stanventures.com/news/chatgpt-reportedly-using-serpapi-to-scrape-google-search-results-4138/ [Accessed 2026-01-14]

User-generated content is disproportionately prominent. Reddit accounts for about 8% of AI citations and is the most cited domain overall, as well as the second most cited source in ChatGPT[4]. Large discussion platforms act as proxies for lived experience at scale. Volume, noise, and bias management directly influence what models present as authoritative.

Early optimisation findings are incremental. Semantic URLs[5] receive roughly 11.4% more citations. Pages ranking five to ten in Google remain competitive for generative visibility. List-based and comparative formats make up over a quarter of cited content. Meta descriptions stating the answer clearly outperform click-bait alternatives. Generative systems favour explicit relevance and information density over persuasion.


(a) The world's most cited domains in AI search.




(b) Reddit has exploded - an 800% increase in citations by ChatGPT in 2025.

Figure 1: Signals shaping generative search: community activity and citation patterns [2].

## 1.3 Background

We now outline technical foundations of generative search - LLM pre-training and Live search.

### Pre-training of LLMs

Pre-training is foundational for LLM development. The model learns general language structure, semantics, and statistical text patterns before task-specific adaptation. Training is usually self-supervised via next-token prediction, teaching the model to generate coherent text continuations [1].

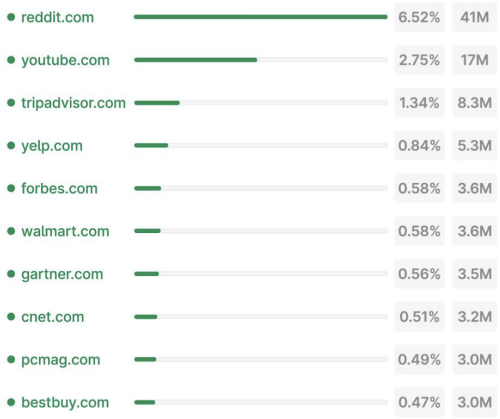
Modern LLMs train on hundreds of billions to trillions of tokens from web crawls, books, news, blogs, forums, encyclopaedic resources like Wikipedia, and code repositories [2]. This variety enables broad coverage and cross-domain generalisation [3] but introduces noise and redundancy. Raw corpora undergo preprocessing: cleaning non-textual artefacts, deduplication, filtering low-quality material, and tokenisation for model training [4, 5]. The capabilities of current LLMs can be attributed in no small part to the transformer architecture [6] enabling parallelisation of model training not only over texts, but at the level of the individual text.

Pre-training datasets have evolved. Early corpora like C4 showed the feasibility of large-scale filtered web data. Later datasets such as MassiveText and FineWeb offer greater scale, stricter quality

[4] *Reddit, serps, and the foundations of ai retrieval pipelines.* October 2025. https://www.linkedin.com/posts/joshua-blyskal_theres-been-a-lot-of-chatting-lately-about-activity-7386410380018860032-QMAr [Accessed 2026-01-14]

[5] *Clean URL*, Wikipedia. https://en.wikipedia.org/wiki/Clean_URL [Accessed 2026-01-14]


(a) Wikipedia and Reddit dominate as ChatGPT sources




(b) Reddit is by far the top source in Perplexity's GSE.

Figure 2: Dominance of reference and user-generated sources across generative search systems. [2].

control, and more diverse sources [7]. Instruction-augmented pre-training adds synthetic instruction–response pairs to improve downstream performance [8]. Multi-phase and continual pre-training expose models sequentially to curated datasets, often moving from general text to specialised domains.

The result is a general-purpose model with strong linguistic priors and reasoning capability. These foundation models perform robustly across benchmarks and form the base for generative search systems [9].

### Live search

Generative systems can access live web data. ChatGPT, via ChatGPT Search, retrieves and incorporates up-to-date information beyond its static training cutoff [6].

Live search enables responses grounded in current events and factual updates. Retrieval may be automatic or user-invoked. Retrieved content is synthesised into conversational answers rather than presented as ranked link lists [10]. Responses typically include citations or source references.

From the user perspective, this is opaque. Queries are reformulated into search-engine-compatible requests executed against underlying infrastructure, historically Bing-powered APIs, returning URLs, snippets, and metadata [7].

Rather than ingesting full documents, ChatGPT consumes selected snippets and structured signals like titles and meta descriptions. This reduces latency and computational cost while providing sufficient context for generation [11]. Snippet-level retrieval can miss nuance deeper in documents, and behaviour depends on source coverage and ranking biases. Live search marks a shift from static text generation to hybrid retrieval, interpretation, and synthesis in real time.

[6] *Introducing ChatGPT search.* OpenAI, 2024. https://openai.com/index/introducing-chatgpt-search/ [Accessed 2026-01-14]

[7] *Chatgpt search makes microsoft bing an seo priority*, Search Engine Land, 2024. https://searchengineland.com/chatgpt-search-microsoft-bing-seo-448019 [Accessed 2026-01-14]

# 2 Discoverability

This section provides a practical framework for GEO. We synthesise evidence on retrieval behaviour, citation patterns, and content structuring to guide visibility in LLM-generated responses. We start with platform-agnostic principles, then introduce AI-specific metrics, and conclude with a comparative overview of leading LLM-based search systems.

## 2.1 Guidelines for Discoverability

### Technical foundations

AI visibility requires fully machine-accessible content. Unlike traditional crawlers, AI agents often retrieve lightweight snippets without executing scripts [12][8]. Critical text, links, and navigational elements must be exposed in plain HTML. This ensures that all content can be retrieved reliably regardless of the AI agent's rendering capabilities.

Internal linking and persistent modules, such as related content or recommended reading blocks, help flatten hierarchical structures and improve crawl depth. By creating multiple pathways to key pages, webmasters increase the likelihood that deeper content is encountered and ingested during retrieval. Clear sitemaps and consistent URL structures further enhance discoverability.

A further constraint follows from how retrieval is executed in practice. In many systems, the first pass operates on partial page representations rather than full documents. This places weight on early content placement. Critical information should appear high in the document and not be deferred behind introductory material.

Latency budgets also shape what is retrieved. Systems favour sources that yield usable information within tight time constraints. Pages that require multiple hops, heavy rendering, or deep navigation are less likely to be incorporated into the candidate set. Discoverability is therefore bounded not only by relevance, but by access cost.

### On-page and content structure

Generative systems retrieve content at the chunk level rather than consuming whole pages. Content should be segmented into self-contained units, each focused on a single idea and clearly identified with H2 or H3 headings. Sections must be coherent on their own to allow extraction without context from surrounding text.

Structural choices influence citation likelihood and retrieval. Semantic URLs that clearly describe page content see approximately 11.4% higher citation rates than opaque identifiers. List-based and comparative content formats account for more than 25% of cited material[9]. Meta descriptions should explicitly summarise the key answer. Clear and precise descriptions signal content relevance to retrieval systems and increase the chance of being synthesised into a response.

This chunking behaviour creates a hard boundary. Sections compete independently for inclusion. A well-formed subsection can be retrieved even if the surrounding page is weak. The inverse is also true. Strong pages with poorly defined sections underperform at retrieval time.

As a result, optimisation shifts from page-level design to segment-level design. Each section must answer a question, define a concept, or present a comparison in isolation. This mirrors the query reformulation and fan-out patterns described later, where systems decompose user intent into smaller retrieval units.

[8] *How to win the new seo game: Ai search, geo, and the future of visibility*, 2025. https://knowledge.gtmstrategist.com/p/how-to-win-the-new-seo-game-ai-search [Accessed: 16 Jan 2025]

[9] *Understanding chatgpt network logs: Query fan-outs, snippets, and citation decisions*, November 2025. https://www.linkedin.com/posts/joshua-blyskal_everyone-in-aeogeo-should-know-how-to-do-activity-7399096886990897152-Udvx/ [Accessed 2026-01-14]

### Citation-worthiness

Search-enabled LLMs adopt references selectively, often guided by trust heuristics similar to E-E-A-T [13] [10]. Signals include author credentials, organisational affiliation, update recency, and the presence of unique insights. Content offering novel data, case studies, or first-hand expertise is more likely to be cited explicitly.

Redundant content or material that merely repeats widely available information provides little incentive for citation. Practitioners can test citation potential by comparing LLM adoption of candidate sources against manual E-E-A-T scoring. Alignment between LLM reference choices and E-E-A-T scores supports the hypothesis that models implicitly reward trustworthiness, authority, and uniqueness.

Leaked prompt fragments demonstrate that some engines instruct the model on citation style, including preferences on authoritative sources and format consistency, which can influence how citation rates and structures manifest in output (see Appendix B for concrete examples).

There is also a structural component to citation selection. Systems favour spans that can be lifted with minimal transformation. Sentences that are self-contained, precise, and numerically grounded are more likely to be adopted. In contrast, content that relies on surrounding context or rhetorical build-up is less portable and therefore less likely to survive selection.

### Answer synthesis optimisation

Retrieval is only part of visibility. Content must survive the synthesis process. Writing in a neutral and factual tone increases the probability of integration. Structured, data-driven content in a Q&A format reduces transformation effort.

Complex sections should start with concise summaries that provide immediate answers. Avoid marketing or promotional phrasing, as models may discount material perceived as persuasive rather than informational. Pre-digested, self-contained content improves retention in the final output and contributes positively to metrics such as Position-Adjusted Word Count and Subjective Impression.

Synthesis also introduces competition between sources. When multiple retrieved passages express similar information, the model collapses them into a single representation. In this process, redundant sources are discarded. Only the most concise or information-dense variant is retained.

This creates pressure towards informational efficiency. Pages that express the same idea with fewer tokens and clearer structure are more likely to dominate the final answer. This mechanism directly links to metrics such as Information Gain (introduced in §4) and Position-Adjusted Word Count.

### Authority and off-site presence

Generative systems remain strongly influenced by traditional search signals. The top 20 Google results continue to dominate AI citations [11] [12]. Domains lacking inherent authority can gain visibility by appearing on high-trust third-party sites. This form of Barnacle SEO allows brands to enter the candidate set for retrieval without relying solely on organic ranking.

Consistent off-site activity enhances discoverability. Events, digital PR and backlink acquisition all generate textual artefacts that feed into search indices. Maintaining a presence across these channels ensures that a brand remains visible to AI systems over time.

[10] *Search quality evaluator guidelines*, Google, 2025. [Accessed 2026-02-23]. https://static.googleusercontent.com/media/guidelines.raterhub.com/en//searchqualityevaluatorguidelines.pdf

[11] *STUDY: What Drives Brand Mentions in AI Answers?* 2025. https://www.seerinteractive.com/insights/what-drives-brand-mentions-in-ai-answers. [Accessed 2026-02-23]

[12] *AI Overview Citations Now 54% from Organic Rankings* 2025. [Accessed 2026-02-23] https://www.brightedge.com/resources/weekly-ai-search-insights/rank-overlap-after-16-months-of-aio

### Third-party platforms as discoverability channels

Certain platforms such as LinkedIn, Reddit, Wikipedia, Crunchbase and established publications are frequently surfaced in generative responses [13]. Experiments show that LinkedIn posts optimised for niche queries were cited across multiple engines within 24 hours [14]. Platform selection should be guided by domain-specific relevance and historical citation patterns.

Reddit's continuous content flow and structured Q&A format are particularly suited to AI retrieval, especially for opinion or consensus-driven queries. Practitioners benefit from active participation and production of fresh, high-quality content to maintain visibility. Emerging community platforms within specialised industries should also be monitored as potential high-value citation sources.

### Commercial strategy and tooling

Paid placements offer a parallel channel for deterministic visibility. Perplexity and similar platforms provide sponsored follow-up questions or labelled content slots that ensure inclusion in model outputs.

Specialised tooling further supports GEO execution. Heat maps or AI content audits reveal which sections models prioritise and highlight gaps in semantic or trust signals. Platforms that evaluate pages for retrieval probability allow content teams to align human-authored material with AI reading patterns, improving metrics such as SOM and Citation Rate.

## 2.2 Metrics for AI visibility

Traditional SEO metrics are insufficient for generative contexts. GEO metrics capture how content is incorporated and weighted in model outputs. Taken together, these metrics describe different stages of the retrieval and generation pipeline. Some act on inclusion in the candidate set, others on selection within that set, and others on survival through synthesis.

This separation is important. A page may rank highly in traditional search yet fail to be selected. It may be selected yet contribute little to the final answer. Effective optimisation requires alignment across all three stages.

### Share of Model (SOM)

SOM[15] measures the frequency with which a brand appears in AI-generated answers relative to competitors. It reflects probabilistic exposure rather than rank and captures overall market presence within generative outputs.

### Position-adjusted word count

This metric quantifies how much of the final answer is attributable to each source, weighted by position [14]. Earlier placement and structural enhancements, such as quotations, lists, or statistics, increase retention during synthesis and improve the perceived contribution of the source.

### Subjective Impression

Subjective Impression measures how essential a source was to the final answer. Using an LLM-as-a-judge framework [15], sources are scored on relevance, uniqueness, and information gain. High scores indicate that the content contributed unique and indispensable insights rather than peripheral references.

[13] *The Complete Guide to Generative Engine Optimization (GEO)* 2025. https://peec.ai/blog/the-complete-guide-to-generative-engine-optimization-(geo). [Accessed 2026-02-23]

[14] *How to Win the New SEO Game: AI Search, GEO, and the Future of Visibility* 2025. https://knowledge.gtmstrategist.com/p/how-to-win-the-new-seo-game-ai-search. [Accessed 2026-02-23]

[15] *Share of model: A 2-minute briefing*, BN Edition Substack, 2024. https://bnedition.substack.com/p/share-of-model-a-2-minute-briefing [Accessed 2026-01-14]

#### Citation rate and structure

Citation Rate tracks the percentage of queries for which a URL is explicitly referenced. Structural characteristics, including semantic URLs and clear headings, improve citation likelihood. Monitoring citation patterns across platforms identifies which retrieval layers are favourable for a given content profile.

#### Traditional search ranking

While not sufficient alone, traditional search ranking remains a useful predictor for inclusion in the retrieval candidate set. Drops in search position often precede declines in SOM, making ranking trends a valuable early indicator of AI visibility performance.

### 2.3 Comparative Analysis of LLM-Based Search Systems

LLM-based search systems vary in architecture, source preference, and citation behaviour. These differences persist across queries and require platform-specific strategies. Effective GEO incorporates understanding of these structural distinctions.

#### Architectural Divergence and Indexing Logic

ChatGPT and Copilot rely on Bing. Copilot integrates Microsoft Graph to blend web and enterprise data. Gemini accesses Google's primary index, mirroring classical search results. Perplexity aggregates multiple APIs and refreshes selectively based on demand. Claude relies primarily on pre-trained knowledge with minimal live retrieval, making visibility largely dependent on high-quality, persistent content.

#### Source Bias and Content Preferences

ChatGPT favours Wikipedia for factual content and Reddit for experiential knowledge. Copilot prioritises established media, official documentation, and commerce platforms. Perplexity leans heavily on recency, tech sources, and Reddit discussions. Claude prioritises enduring, high-quality content. Each model's bias influences metrics such as SOM and Citation Rate, guiding optimisation priorities.

#### Citation Behaviour and Attribution

Perplexity cites nearly every claim, offering high transparency. ChatGPT is selective, often omitting citations for general knowledge. Copilot embeds inline citations from authoritative sources. Claude cites only when prompted, requiring indirect measurement of visibility. These behaviours directly affect Position-Adjusted Word Count and Subjective Impression.

#### Convergence on Social Authority

High-trust social content shows consistent impact across platforms. Experiments demonstrate ChatGPT, Perplexity, and Google citing the same LinkedIn post within 24 hours. LinkedIn and Reddit act as universal trust substrates. Content on these platforms ensures broad visibility across generative engines, complementing platform-specific optimisation.

# 3 Generative Search Framework

AI search differs from classical search which maps queries to documents. A Generative Engine [14] has three main components: a query reformulation model $R$ that converts user queries into search-ready queries; an external search interface $S$ that returns retrieved content units, usually snippets, and a generative model $G$ that reasons, checks sufficiency, and produces responses.

Even with an internal knowledge base, real-time queries rely on retrieved sources. Responses are grounded in these sources and often include citations. The flow from query to response involves several steps. We now outline these steps with supporting evidence from network logs of interactions with ChatGPT (Appendix A). In the process, we also highlight differences across search providers.

## 3.1 System Flow

Formally, a user query $q \in \mathcal{Q}$ produces a response $r$:

$$r = G(q)$$

Algorithm 1 gives the full system flow. The main steps are described below.

### Intent Classification

The system first decides whether external search is needed. Network logs show this uses a tiered architecture. A low-latency ($\lesssim 30$ ms) classifier routes queries into three buckets:

- **No Search**: Answerable from pre-training alone (e.g., "What is the speed of light?")
- **Simple Search**: Requires up-to-date data; single-pass retrieval suffices (e.g., "latest tech news EV industry")
- **Complex Search**: Multi-faceted; requires iterative query refinement

Fail-fast logic directs low-confidence complex queries to the simple search path to save resources.

### Query Reformulation

User queries are often verbose and contain stop-words. Query reformulation [16, 17] converts them into concise, search-ready terms. Queries are rewritten to map human intent to terms likely found in a search index. This includes expanding abbreviations, resolving entities, and adding context. Multi-stage processing may split compound questions or inject temporal qualifiers.

For example, a query like "What are the latest updates on solid-state batteries?" might be reformulated into "solid state battery market 2025" and "solid state battery pilot production companies". For an example of context-aware rewriting, consider the following:

```
Original: Does it affect children?
Context: What are the symptoms of diabetes?
Rewritten: Does diabetes affect children?
```

### Search Interface

The search interface or control layer is critical. It governs how queries are executed and what content enters the LLM. The search API controls quality, format, and coverage of retrieved content. Different providers exhibit distinct behaviour, influencing snippet completeness, narrative style, and entity coverage. The findings described here have been demonstrated in *notebooks/search_provider_comparison.ipynb*.

**Brave Search**: Structured, entity-dense snippets. Excels at fact extraction and numerical data. Truncation experiments show strong schema awareness, extracting character lists, tables, and precise metrics. Optimal for fact lookups or queries requiring discrete entities.

**SerpApi**: Aligns with query intent rather than just keywords. Returns semantically coherent snippets. In comparison tasks, it presents pros and cons in a single sentence rather than partial explanations. Character limits can truncate long lists, so it favours readability over completeness.

**Exa**: High-recall document fetcher. Retrieves full datasets and tables, including raw HTML or Markdown. Can introduce noise such as paywall notices. Best for applications requiring full-page ingestion with downstream filtering.

**Tavily**: Narrative-oriented, optimised for story-like synthesis. Provides sentence-like snippets with context. Performs poorly on discrete extraction, but excels in questions requiring summarisation or interpretive framing.

The control layer configures the number of results to consider, the number of pages to fetch, and filtering rules to reduce misleading input. Users may be affected by:

- Changes in the search engine indexing strategy
- Modifications to sub-URL handling
- Limits on pages retrieved per query

This layer balances quantity, quality, and relevance to ensure the LLM receives usable content.

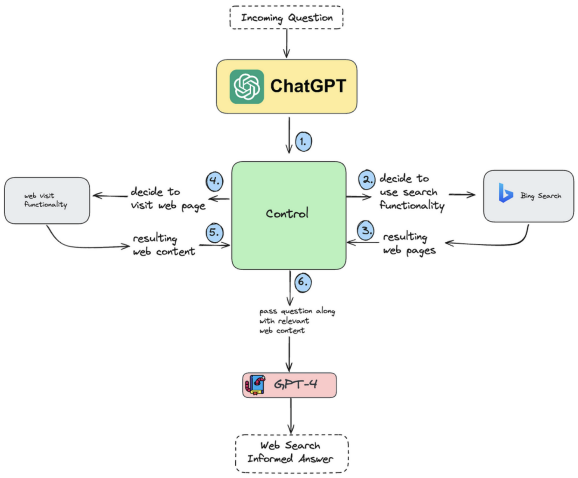



Figure 3: Control layer of the LLM search[16]

### Content Ingestion

The system processes returned results. It rarely downloads full pages, instead using snippets and metadata. This follows a "Snippet-First" principle.

The "Snippet-First" principle emphasises snippet-rich search APIs and avoids automatic scraping, falling back to crawling only when snippets lack required data points. Raw search results are often noisy and require post-processing. Systems cluster snippets by domain to highlight consensus and prioritise high-authority sources. Multiple occurrences of a domain can increase confidence and influence weighting in the final synthesis.

[16]Source: ChatGPT control layer. `https://www.ml6.eu/en/blog/how-llms-access-real-time-data-from-the-web`

### Ranking and Conditioning

Responses are conditioned on internal and external priors.

**Internal Priors**: Pre-trained model parameters and session history $H_t$ shape output. LLMs may re-rank retrieved references internally. Dedicated re-rankers such as fine-tuned dense retrievers can be used to filter irrelevant results. Memory tools can retain user-specific facts across sessions.

$$r_t = G(q_t, C_t, H_t)$$

**External Priors**: Personalisation through location, language, and device creates a pre-biased candidate set $C_{ext}$. Signals include freshness, links, and social mentions.

### Complex Search Loop

For hard queries, the LLM iterates: it checks if information is sufficient and issues additional queries until confidence is reached. The flow is the following:

1. Execute search
2. Read and evaluate snippets
3. Decide if sufficient
4. Generate targeted queries if needed and loop

Top-k results feed final generation: $r = G(q, C_{1:k})$.

### Response Generation

The LLM integrates top-k passages $C_{1:k}$ with query $q$ and session context $H_t$. It synthesises, summarises, and abstracts multiple sources. Redundant information is suppressed; repeated signals are amplified. The output is grounded, context-aware, and human-readable:

$$r = G(q, C_{1:k}, H_t)$$

Researchers have previously reported generative search engines as having poor verifiability of sources in their responses in practice [18], leaked prompts (see Appendix B) indicate that this has to be enforced through the synthesis prompt. Alternatively, researchers have proposed, and shown the efficacy of, post-editing the LLM output to improve attribution [19].

### Python Code

Two variants exist. One separates intent classification, query reformulation, ranking, and sufficiency checks. Research setups often use detailed architectures to maximise control and visibility.

Listing 1: Generative Search Engine with dedicated intent classifier, query generator, ranker and information sufficiency tester

```
def gen_search(
    q,                  # user query
    search_fn,          # search interface
    qrm,                # query reformulation model
    gen_query_fn,       # generation model
    priors_int,         # internal priors
    priors_ext,         # external priors
    hist,               # session history
    k,                  # number of retrieved items
    max_iter            # maximum iterations
):
    intent = classify_intent(q)

    if intent == "NoSearch":
        resp = gen_query_fn(q, [], hist)
        return resp

    acc = []
    it = 0

    qrs = qrm(q, hist)

    while True:
        it += 1

        for qr in qrs:
            res = search_fn(qr)
            acc.extend(res)

        acc = rank_content(acc, q, priors_int, priors_ext, hist)

        if intent == "ComplexSearch" and not sufficient_coverage(q, acc):
            if it >= max_iter:
                break
            qrs = qrm(q, acc, hist)
        else:
            break

    top = select_top_k(acc, k)

    resp = gen_query_fn(q, top, hist)

    return resp
```

A simpler variant lets the LLM handle intent classification, sufficiency checks, reference adoption, and answer generation. Evidence from logs and leaked prompts shows this approach in real deployments. It is easier to deploy and has fewer unknowns.
Fast classifiers work best at scale. But building separate systems takes time. The simplest way to classify intent is through the prompt itself. Tell the model when to search, and let it decide. This approach replaces complex, standalone infrastructure. It relies entirely on the model's own reasoning. By writing clear rules for when to search, developers save time. The prompt handles hard, multi-part queries that often break rigid scripts. It is simple to build and easy to deploy.

Listing 2: Generative Search Engine with underlying LLM responsible for intent classification, query generation, information sufficiency test and reference adoption

```
def generative_search(
    q,            # user query
    hist,         # session history
    llm,          # language model
    search_fn,    # search interface
    max_iter      # maximum iterations
):
    acc = []
    it = 0
    resp = llm(q, hist)

    while True:
        it += 1

        if not requests_search(resp):
            break

        if iter > max_iter:
            break

        for qr in resp.search:
            res = search_fn(qr)
            acc.extend(res)

        resp = llm(q, acc, hist)

    return resp
```

# 4 Experiments

This section introduces a set of tools designed to operationalise the concept of discoverability defined in §2 and presents some preliminary experimental evaluations of a sample of websites.

Based on the guidelines introduced there, we first propose 10 measurable, page-level signals that influence whether content is retrieved, selected, and reused by large language models during answer generation. These are incorporated into our *webpage visibility metrics calculation tool*. The tool analyses individual web pages and assigns a composite *AI Visibility Score*, intended as a proxy for the likelihood that a page will surface in LLM-generated responses.

Following that, we present experiments on visibility in a ChatGPT-aligned generative engine. We use it to carry out our own generative search experiments, calculating metrics such as position-adjusted word count (PAWC) for some related websites over a relevant set of queries we introduce for those websites.

## 4.1 Webpage Visibility Metrics Calculation Tool

The tool operates directly on rendered HTML and extracted text. It does not rely on click data, backlinks or behavioural signals. This design choice reflects the shift to discoverability being driven less by popularity and more by machine readability, semantic alignment, and citation-worthiness.

Ten metrics are computed, each normalised to a 0–100 scale and equally weighted. Together, they capture complementary aspects of discoverability across structure, content, semantics, and verification. The metrics are deliberately simple, favouring interpretable heuristics over opaque optimisation.

The weighted sum of the ten metrics yields a single AI Visibility Score. The score is not intended to predict ranking position. It represents, instead, the degree to which a page is shaped for retrieval, selection, and reuse in generative search systems.

#### Structure

The structure metric scores the presence and balance of explicit HTML structure. It combines three signals: the number of headings across levels `h1`–`h6`, the presence of ordered or unordered lists, and the proportion of short paragraphs. Headings contribute up to a fixed ceiling, rewarding hierarchical depth without encouraging excess. Lists are scored with diminishing returns. Paragraphs shorter than a defined word threshold are treated as machine-friendly units and increase the score proportionally.

#### Citability

Citability is computed at the sentence level. Sentences are scanned for numeric expressions and for definitional language such as copular constructions or explicit reference phrases. The score reflects the proportion of sentences that contain at least one such signal, with upper bounds applied to avoid saturation. The intent is not stylistic judgement but detection of sentences that resemble quotable factual claims.

#### Topical Focus

Topical focus measures lexical concentration. After stopword removal, the five most frequent terms are identified and their combined frequency is divided by total token count. The resulting ratio is mapped to a bounded scale, with full score achieved once concentration exceeds a fixed threshold. Pages that disperse attention across many terms score lower, reflecting weaker topical anchoring.

#### Readability

Readability is derived from the Flesch Reading Ease score. Texts below a minimum length are excluded. Scores within a target band receive full credit, while texts that are significantly simpler or more complex

are penalised with asymmetric decay. This reflects a preference for explanatory prose that is dense enough to carry information but not so dense that it resists transformation.

### Schema

The schema metric evaluates the presence of machine-readable metadata. Points are awarded for a meta description, Open Graph tags, and embedded JSON-LD objects, with caps applied to prevent over-counting. The metric does not assess correctness or completeness of the schema, only its presence as a signal of explicit contextualisation.

### Semantic Similarity

Semantic similarity is computed by embedding the full page text and a set of target queries into a shared vector space using a sentence-level transformer. Cosine similarity is calculated between the page embedding and each query embedding, and the mean value is scaled to a percentage. The metric approximates intent alignment under query reformulation rather than exact term matching.

### Entity Density

Entity density is defined as the ratio of recognised named entities to total tokens, as detected by a general-purpose named entity recogniser. The score increases linearly up to a fixed density threshold, beyond which additional entities do not contribute. This favours content that anchors claims in identifiable actors, concepts, or locations.

### Information Gain

Information gain penalises internal redundancy. Sentences are embedded individually, and mean pairwise similarity is computed across all sentence pairs. Higher average similarity indicates repetition and lowers the score. The metric is inspired by diversity-based ranking methods and rewards texts where successive sentences introduce new information.

### Verifiability

Verifiability aggregates several evidence signals. External outbound links are counted with diminishing returns. The text is scanned for explicit attribution language, such as references to studies or reports. Additional credit is awarded for the presence of reference sections and formal identifiers including DOIs, ISBNs, or arXiv IDs. The metric measures evidence density, not factual correctness.

### Coherence

Coherence evaluates local and global flow. It combines three components: the proportion of sentences containing discourse markers, the average semantic similarity between adjacent sentences, and a penalty for abrupt drops in similarity that indicate topic jumps. The score favours steady progression over both fragmentation and excessive repetition.

### Metric interactions

The ten metrics are designed as independent signals, but their deltas under LLM-driven rewriting are not. To assess co-movement, we ran a systematic prompt sweep across nine combinations of suggester and improver strategies on a single target page and computed Spearman rank correlations between all pairwise metric deltas.

Three metrics form a redundant cluster. Structure, citability, and verifiability co-move across all combinations, with Spearman correlations of $\rho = 0.82$, 0.83, and 0.71 respectively. The cause is mechanical: any intervention that adds heading hierarchy, bullet lists, or an explicit FAQ section

simultaneously satisfies all three scoring functions. For reporting, one representative from this cluster suffices. Verifiability is the weakest standalone signal, as it depends on structural cues that structure already captures.

Four metrics are largely orthogonal. Semantic similarity, entity density, information gain, and readability each respond to distinct aspects of rewriting. No single prompt intervention reliably moves all four. Together they contribute non-redundant signal and should each be retained in the composite score.

Two strong trade-offs are present. Information gain and coherence are strongly anti-correlated ($\rho = -0.73$). Structure and coherence follow the same pattern ($\rho = -0.77$). When prose is converted to bullet lists, structure and information gain improve because each item introduces distinct content. Coherence falls because short parallel items carry lower sentence-to-sentence semantic continuity than narrative paragraphs. This is not a measurement artefact. It reflects a real conflict between machine-parseability and argumentative flow. Readability anti-correlates with both verifiability ($\rho = -0.66$) and citability ($\rho = -0.57$), because denser, more explicitly evidenced content tends to score lower on the Flesch scale. Table 1 summarises the strongest pairwise relationships. The schema metric is excluded from this analysis as it measures original HTML metadata and is unaffected by text rewriting - its delta is zero across all runs.

Table 1: Selected metric-pair Spearman rank correlations over nine prompt-sweep runs on a single target page.

| Metric pair | Spearman $\rho$ | Relation |
|---|---|---|
| Structure × Verifiability | +0.82 | Redundant |
| Citability × Verifiability | +0.83 | Redundant |
| Structure × Citability | +0.71 | Redundant |
| Information Gain × Coherence | −0.73 | Trade-off |
| Structure × Coherence | −0.77 | Trade-off |
| Readability × Verifiability | −0.66 | Trade-off |

## 4.2 Website Suggester and Website Improver

In addition to measurement, we introduce a second pair of tools designed to act directly on content. Where the visibility metrics calculation tool diagnoses discoverability gaps, the *Website Suggester* and *Website Improver* translate those gaps into concrete editorial changes. Together, they form a closed loop between evaluation and optimisation.

### Website Suggester

The Website Suggester takes as input the raw webpage text or HTML alongside the full metric breakdown produced by the visibility scoring tool. Its output is a short set of actionable recommendations aimed at increasing the page's AI Visibility Score.

The suggester is implemented as a prompt-driven LLM agent conditioned on both the content and its measured weaknesses. Rather than producing generic SEO advice, it grounds each suggestion in the observed scores. Low structure scores prompt recommendations around heading hierarchy and list usage. Weak citability triggers guidance to add explicit definitions, statistics, or declarative claims. Poor topical focus results in advice to tighten vocabulary and remove peripheral material. Readability issues are addressed through sentence length and paragraph compression.

Each suggestion follows a fixed pattern. It specifies what should be changed and why the change improves retrieval, selection, or reuse by generative engines. Where possible, the model is instructed to

reference concrete sections or sentences from the input text, anchoring advice in the page itself rather than abstract best practice.

The suggester therefore operationalises the discoverability principles introduced in Section 2 as editorial diagnostics. It mirrors how LLM-based search systems reward structure, quotability, and semantic clarity, but exposes those preferences in explicit textual form.

**Website Improver**

The Website Improver consumes the original webpage text together with the suggestions produced by the Website Suggester. Its role is not evaluative but editorial. It rewrites the page to apply as many of the proposed changes as possible while preserving meaning and tone.

The improver enforces a constrained rewriting regime. All factual content must be retained. The output is formatted in markdown with explicit headings, bullet lists, and highlighted key terms. Technical concepts are rewritten to include clear definitional statements. Where suggestions call for quantitative evidence but none is available, the tool inserts realistic placeholders rather than fabricating data.

Paragraph length is aggressively controlled, favouring short, self-contained units that align with snippet-based retrieval and synthesis. The model is instructed to return only the revised text, with no explanation or commentary, allowing the output to be treated as a direct content replacement.

In combination, the suggester and improver form a practical implementation of Generative Engine Optimisation. The first translates discoverability metrics into editorial intent. The second applies that intent in a form optimised for machine parsing, citation, and reuse. Both tools reflect the same underlying assumption as the metrics themselves: that visibility in generative search is driven less by authority signals alone and more by how explicitly and efficiently information is expressed.

**Prompt strategy experiments**

To quantify the effect of prompt design on visibility improvement, we conducted a systematic sweep over a Cartesian product of three suggester strategies and three improver styles, yielding nine combinations per target URL. The page is fetched and scored once. Each combination reuses the initial scores, isolating prompt effects from network and parsing variation. All runs used `gpt-4o-mini` with equal metric weights.

The three suggester strategies are: a balanced default covering all metric dimensions, a structural variant targeting heading hierarchy, list conversion, FAQ inclusion, and JSON-LD schema recommendations, and a semantic variant focused on entity precision, verifiable claims, and conceptual density.

The three improver styles are: a balanced default markdown rewrite, a conservative style applying the minimum changes necessary to satisfy each suggestion while preserving author voice, and an aggressive style performing a full overhaul that converts all prose to bullet lists, prepends a TL;DR, and appends Key Facts and FAQ sections. The full system prompts for the best-performing combination (default suggester and aggressive improver) are given in Appendix C.

Results on a representative B2B SaaS landing page (baseline AI Visibility Score: 49.7) show that improver style is the dominant factor. All combinations producing a positive total improvement used the aggressive improver: default $\times$ aggressive (+2.0 points, +4.0%), semantic $\times$ aggressive (+1.3, +2.6%), and structured $\times$ aggressive (+0.3, +0.6%). Conservative and default improvers produced near-zero or negative totals in every case. The worst outcome was structured $\times$ conservative ($-2.2$ points, $-4.4\%$).

Table 2 shows all nine combinations sorted by total delta.

Among suggesters, the semantic strategy showed the greatest stability. Its total deltas spanned a range of 1.7 points across all three improver styles. The structured suggester showed the widest variance, from +0.3 to $-2.2$, indicating that structural guidance conflicts with conservative and default improver modes when the original page already carries adequate heading structure.

Table 2: Prompt sweep results on a B2B SaaS landing page (baseline AI Visibility Score: 49.7, `gpt-4o-mini`, equal metric weights). Rows sorted by total score change.

| Suggester | Improver | Δ (pts) | Δ (%) |
|---|---|---|---|
| Default | Aggressive | +2.0 | +4.0 |
| Semantic | Aggressive | +1.3 | +2.6 |
| Semantic | Default | +1.0 | +2.0 |
| Structured | Aggressive | +0.3 | +0.6 |
| Semantic | Conservative | −0.4 | −0.8 |
| Structured | Default | −0.6 | −1.2 |
| Default | Conservative | −0.9 | −1.8 |
| Default | Default | −1.1 | −2.2 |
| Structured | Conservative | −2.2 | −4.4 |

Readability fell in every run. The baseline Flesch score of 94.3 was inflated by short navigational text on the original page. LLM-generated prose is more substantively complex regardless of improver style and consistently lowers the score. Readability therefore functions as a ceiling constraint rather than an improvement target in these experiments.

Verifiability showed the largest proportional gain. From a baseline of 5, aggressive-improver rows reached scores of 10 to 15, increases of 100 to 200 per cent. The aggressive template's mandatory FAQ, Key Facts section, and citation placeholders directly satisfy the signals the verifiability function rewards. Conservative runs left the score unchanged throughout.

## 4.3 Webpage Visibility Evaluation

To complement the static analysis provided by the Website Visibility Metrics tool, an end-to-end evaluation pipeline was developed to measure *observed* visibility within LLM-generated responses. This pipeline operationalises visibility as a function of citation presence, prominence, and distribution across multiple queries.

### Pipeline Overview

The evaluation pipeline follows a four-stage process. First, a predefined set of search queries - constructed to reflect realistic user intent - is issued to an LLM with web search capabilities enabled. In the present implementation, queries are executed in parallel using an asynchronous interface to `gpt-4o-mini`, ensuring both scalability and consistency of responses.

Second, each generated response is parsed to extract Markdown-style citations of the form `[source](url)`. These citations are treated as explicit signals of attribution, and thus as proxies for visibility within the generative output.

Third, for each cited source, a Position-Adjusted Word Count (PAWC) score is computed. This metric assigns greater weight to citations appearing earlier in the response, reflecting the empirical observation that earlier content exerts disproportionate influence on user perception and downstream summarisation. Formally, each sentence contributes a weight that decays exponentially with its position in the response, and this weight is divided among all sources cited within that sentence.

Finally, scores are normalised on a per-response basis to yield a share-of-voice distribution across cited domains. These normalised scores are then aggregated across all queries and averaged, producing a stable estimate of each domain's overall visibility.

### Position-Adjusted Word Count

The Position-Adjusted Word Count (PAWC) metric serves as the core quantitative signal within the evaluation pipeline. Unlike naïve citation counts, which treat all mentions equally, PAWC incorporates both positional and structural information.

Concretely, each response is segmented into sentences, and a weight $w_i$ is assigned to the $i$-th sentence according to an exponential decay function:

$$w_i = e^{-\lambda i}$$

where $\lambda$ controls the rate of decay. Earlier sentences therefore contribute more heavily to the final score. Within each sentence, the total weight is distributed evenly across all cited sources, ensuring that multi-source attribution does not inflate individual scores.

The contribution of a given domain is then computed as the sum of its weighted shares across all sentences in which it appears. This produces a raw PAWD score per response, which is subsequently normalised to the range $[0, 1]$ to enable comparison across different responses and query contexts.

At the aggregation stage, PAWD scores are accumulated across all query responses and averaged per domain. The resulting values can be interpreted as a share-of-voice metric, capturing both how frequently and how prominently a domain is cited within LLM outputs.

### Implementation Structure

The pipeline is modular, with clear separation between query execution, citation extraction, and scoring logic. The orchestration layer coordinates asynchronous query execution and aggregates results, while the scoring module encapsulates citation parsing and PAWD computation. This separation ensures that individual components can be modified or extended, for instance to support alternative attribution formats or weighting schemes without affecting the overall pipeline structure.

In practice, the pipeline outputs a ranked list of domains, each associated with an averaged PAWD score. For a given test subject, its total visibility is computed by summing the scores of all domains matching its citation label, alongside its relative rank within the distribution. This provides both an absolute and comparative measure of visibility in generative search contexts.

### Test results

The pipeline was run for three test subjects (Salesforce, HubSpot, Klikt) using a single shared query set. The queries reflect the same domain: B2B sales and marketing. They cover platform positioning, lead discovery and enrichment, automation and workflows, and integration with CRMs and GTM tools. Each subject was evaluated against the same 20 prompts to keep retrieval conditions comparable.

**Salesforce**: achieved a PAWD share-of-voice of 0.04 and ranked third in the cited-domain distribution. The top-ranked domains were techradar.com (0.15) and en.wikipedia.org (0.06). Salesforce was cited in the model's responses but often appeared after higher-weighted comparison or survey-style sources.

**HubSpot**: achieved a share-of-voice of 0.02 and ranked seventh. The same leading domains (techradar.com, en.wikipedia.org) appeared at the top, and blog.hubspot.com contributed to HubSpot's total. The model cited HubSpot as an all-in-one CRM and marketing platform but with lower positional prominence than the leading domains.

**Klikt**: achieved a share-of-voice of 0.00 and had no defined rank. No cited domain in the aggregated ranking matched the Klikt label. The model cited other prospecting and sales tools but did not surface Klikt in the sampled responses.

Visibility tracks citation presence and position. The two established brands appear with non-zero scores and defined ranks; the smaller brand (Klikt) is absent. Editorial and reference sources (techradar.com, en.wikipedia.org) lead across runs. Same queries, same domain: the spread (strong to non-discoverable) reflects retrieval and attribution behaviour, not query design.

# 5 Conclusion

Generative search is rapidly changing how information is discovered, consumed, and surfaced. Unlike classical search, where visibility depends primarily on ranking position within a list of links, generative systems produce a single synthesised response drawn from a subset of retrieved sources. This change alters the optimisation target. Visibility now depends on whether content enters the retrieval candidate set, survives ranking and filtering, and ultimately contributes to the generated answer.

In the present report we analysed the architecture of generative search systems, examined empirical citation behaviour and implemented tools to measure page-level signals influencing discoverability. The following subsections summarise the key lessons for content creators, for implementors of generative search systems, and the main results obtained from our experiments. As generative search systems continue to evolve, the mechanisms described in this report provide a practical framework for analysing and improving visibility.

## 5.1 Takeaways for content creators

Content visibility in generative search depends less on persuasion and more on informational efficiency. Pages that clearly state answers, provide structured explanations, and express verifiable claims are more likely to be retrieved and incorporated into synthesis. The practical implication is a shift from page-level optimisation to segment-level optimisation. Individual sections must stand alone as coherent answer units that can be extracted independently.

Machine accessibility is a prerequisite. Content hidden behind scripts, complex rendering, or heavy navigation layers may never enter the retrieval pipeline. Clear HTML structure, early placement of key information, and explicit metadata improve the probability that retrieval systems can identify relevant passages quickly within latency constraints.

Citation-worthiness also plays a central role. Generative systems favour sentences that express concrete claims, contain numerical information, or reference identifiable entities. Content offering unique insights, original data, or first-hand expertise is significantly more likely to survive the synthesis stage than content that simply repeats widely available information.

Finally, off-site presence remains important. Generative engines frequently retrieve content from trusted discussion and reference platforms such as Reddit, Wikipedia, and LinkedIn. Participation in these ecosystems increases the probability that a brand or concept appears in the candidate set across multiple engines simultaneously. GEO therefore extends beyond owned websites into broader information ecosystems.

## 5.2 Guidelines for implementing generative search systems

Analysis of existing systems such as ChatGPT, Perplexity, Copilot, and Claude reveals several consistent architectural patterns.

First, query reformulation is essential. Raw user queries are often verbose, ambiguous, or poorly aligned with search indices. Effective systems rewrite queries into concise search-oriented forms, sometimes generating multiple reformulations to improve recall. This step substantially improves the relevance of retrieved snippets.

Second, retrieval pipelines operate under strict latency constraints. Snippet-first retrieval reduces network overhead and allows the system to process many candidate sources quickly. Full document retrieval should be reserved for cases where snippets lack necessary information. This layered retrieval strategy balances coverage with performance.

Third, iterative retrieval improves answer quality for complex queries. Instead of issuing a single search request, the model evaluates whether retrieved information is sufficient and generates additional targeted queries if needed. This loop allows the system to gradually refine the evidence set before synthesis.

Fourth, ranking and filtering remain critical. Generative models perform better when conditioned on a curated set of passages rather than raw search results. Dense retrievers, re-ranking models, or heuristic filters can remove low-quality or redundant snippets before they reach the generative stage.

Finally, synthesis prompts and output constraints strongly influence behaviour. Leaked system prompts demonstrate that production systems explicitly instruct models how to combine sources, how to attribute information, and when to include citations. These prompts function as an additional control layer that shapes answer structure and verifiability.

Leaked instructions reveal another pattern. Production systems often use long, verbose prompts. They do not rely on short or generic rules. Instead, they write exhaustive instructions to cover edge cases and close security gaps. Claude and Copilot prompts, for example, dictate exact character limits. They strictly forbid fabricated links. They enforce hard rules for sensitive data. To build a safe and reliable system, they trade conciseness and structure for total control.

Taken together, these observations suggest a general design pattern for generative search systems: query rewriting, high-recall snippet retrieval using one or more search engine APIs, iterative evidence expansion, re-ranking of candidate passages, controlled synthesis.

### 5.3 Results summary and key observations

The experiments conducted in this report focused on operationalising discoverability through measurable signals. Our webpage visibility metrics calculation tool demonstrates that many GEO principles can be translated into interpretable page-level metrics. Signals such as structural clarity, entity density, semantic alignment with queries, and information gain can be computed automatically and used as proxies for generative visibility.

Correlation analysis between metrics revealed several structural relationships. A first cluster emerges around document structure and formatting. The *Structure* metric correlates with *Readability* and *Coherence*, as pages with frequent headings, short paragraphs and lists tend to produce shorter sentences and clearer discourse transitions. This relationship is expected because structural segmentation naturally improves both local sentence flow and global readability.

A second cluster involves signals related to evidential grounding. *Citability* and *Verifiability* show partial correlation since sentences containing statistics or explicit claims are also more likely to include references, links, or attribution language. However, the overlap is not complete: a page may contain many quotable factual statements without linking external sources, or it may contain many references without expressing concise claim-like sentences.

In contrast, several metrics behave largely independently. *Semantic Similarity* captures alignment between page content and query intent and therefore varies primarily with topic relevance rather than writing style. *Entity Density* reflects the concentration of recognisable actors, organisations, and concepts in the text, which is largely orthogonal to document structure. Finally, *Information Gain* measures internal redundancy through sentence-level embedding similarity and therefore operates independently from both structure and citation signals. These distinctions suggest that improving generative visibility requires balancing multiple dimensions of content design rather than maximising any single metric.

More broadly, the findings highlight that generative visibility is determined by a multi-stage process. Inclusion in the candidate set, selection during ranking, and survival through synthesis each impose different constraints on content. GEO therefore requires aligning content design with the behaviour of the entire generative pipeline rather than focusing on any single stage.

# A ChatGPT Network Log Analysis

In this appendix we present the internal output produced by ChatGPT obtained from network logs during the processing of a user query.

## Initial Intent Classification

The intent classification module estimates the likelihood that a query requires a simple search, complex search, or no external search at all, based on learned thresholds and probability scores.


```
"sonic_classification_result": {
  "latency_ms": 27.671686984831467,
  "simple_search_prob": 0.9598966743602289,
  "complex_search_prob": 0.0002692964409281286,
  "no_search_prob": 0.03983402919884302,
  "simple_search_threshold": 0.0,
  "complex_search_threshold": 0.4,
  "no_search_threshold": 0.2,
  "threshold_order": [
      "no_search",
      "complex",
      "simple"
  ]
}
```


The results indicate a strong preference for a simple external search strategy, with negligible probability assigned to complex search reasoning.

## Query Fan-Out

In the query fan-out stage the original user prompt *"What are the latest tech news?"* is transformed into multiple semantically related search queries. This step increases recall by querying external search interfaces with paraphrased or expanded formulations.


```
"search_model_queries": {
  "type": "search_model_queries",
  "queries": [
      "latest technology news",
      "tech news today"
  ]
}
```


The generated queries reflect lightweight reformulations of the original prompt, optimized for broad coverage in external news sources.

## Result Processing and Inner Loops

This snippet reveals details about the result aggregation and processing stage. Retrieved search results are grouped by source domain and iteratively evaluated before being incorporated into the final response generation loop.


```
"search_result_groups": [
{
```

```
"type": "search_result_group",
"domain": "www.theverge.com",
"entries": [
{
"type": "search_result",
"url": "https://www.theverge.com/news/848455/google-assistant-gemini-upgrade-2026",
"title": "Gemini isn't replacing Google Assistant on Android just yet",
"snippet": "Google is pushing back its timeline for replacing Google Assistant with Gemini
on Android...",
"ref_id": {
  "turn_index": 0,
  "ref_type": "news",
  "ref_index": 1
},
"pub_date": 1766185634.510794,
"attribution": "www.theverge.com"
},
{
"type": "search_result",
"url": "https://www.theverge.com/news/848462/nintendo-switch-2-smaller-cartridges-inin-games",
"title": "Did a publisher's slip-up reveal smaller Switch 2 cartridges?",
"snippet": "On Friday, retro and arcade game publisher ININ Games announced...",
"ref_id": {
  "turn_index": 0,
  "ref_type": "news",
  "ref_index": 22
},
"pub_date": 1766185634.510794,
"attribution": "www.theverge.com"
}
]
}
]
```


The grouped results demonstrate how external content is clustered, attributed, and prepared for downstream synthesis by the language model.

# B Leaked system prompt excerpts

These are taken from the leaked-system-prompts github repository.

## B.1 Search results adoption

```
Response guidelines:

- Lead with recent info; prioritize 1-3 month old sources for
  evolving topics
- Favor original sources (e.g. company blogs, peer-reviewed
  papers, gov sites, SEC) over aggregators. Find highest-quality
  original sources. Skip low-quality sources like forums unless
  specifically relevant.
```

*claude-4.5-sonnet*

## B.2 Memory tool instructions

```
When applicable, you have some additional tools:

You have memory. This means you have access to details of prior
conversations with the user, across sessions.
```

*Grok 3*

```
Tools

bio

The bio tool allows you to persist information across conversations.
Address your message to=bio and write whatever you want to remember.
The information will appear in the model set context below in future
conversations. DO NOT USE THE BIO TOOL TO SAVE SENSITIVE INFORMATION.
Sensitive information includes ... DO NOT SAVE SHORT TERM INFORMATION.
Short term information includes ...
```

*gpt-4o*

## B.3 Search query formulation prompts

```
How to Use:

- Base queries directly on the user's question without adding assumptions
  or inferences.
- For time-sensitive queries, include temporal qualifiers like "2025",
```

```
  "latest," "current," or "recent."
- Limit the number of queries to a maximum of three to maintain efficiency.
- Break complex, multi-part questions into focused, single-topic searches
  (maximum 3 searches).
- Prioritize targeted searches over broad ones - use multiple specific
  queries within the 3-query limit rather than one overly general search.
- Prioritize authoritative sources and cross-reference information when
  accuracy is critical.
- If initial results are insufficient, refine your query with more specific
  terms or alternative phrasings.
```

*Perplexity*

```
Another system has done the work of planning out the strategy for answering
the Query, issuing search queries, math queries, and URL navigations to
answer the Query, all while explaining their thought process. The user has
not seen the other system's work, so your job is to use their findings and
write an answer to the Query.
```

*Perplexity*

### B.4 Synthesis prompts

```
Generate a comprehensive and informative answer (but no more than 80 words)
for a given question solely based on the provided web Search Results
(URL and Summary). You must only use information from the provided search
results.
Use an unbiased and journalistic tone. Use this current date and time:
Wednesday, December 07, 2022 22:50:56 UTC.
Combine search results together into a coherent answer. Do not repeat text.
Cite search results using [\${number}] notation.
Only cite the most relevant results that answer the question accurately.
If different results refer to different entities with the same name, write
separate answers for each entity.
```

*Perplexity*

```
After research is complete, create an answer in the best format for the user's
query. If they requested an artifact or report, make an excellent artifact that
answers their question. Bold key facts in the answer for scannability. Use short,
descriptive, sentence-case headers. At the very start and/or end of the answer,
include a concise 1-2 takeaway like a TL;DR or 'bottom line up front' that
directly answers the question. Avoid any redundant info in the answer. Maintain
accessibility with clear, sometimes casual phrases, while retaining depth and
accuracy.
```

*Claude*

```
When synthesizing 5+ sources, rely primarily on paraphrasing. State findings
in your own words with attribution. Example: "According to Reuters, the policy
faced criticism" rather than quoting their exact words. Reserve direct quotes
for uniquely phrased insights that lose meaning when paraphrased. Keep
paraphrased content from any single source to 2-3 sentences maximum|if you need
more detail, direct users to the source.
```

*Claude*

```
You are an AI assistant tasked with generating a clear and concise response
to a user query using multiple web sources. Base your answer only on the
provided content.

Summarise key information in your own words, ensuring factual accuracy and
coherence. Prioritise the most relevant and reliable sources. Where sources
disagree, acknowledge the difference briefly.

Cite sources inline using numbered references. Do not include irrelevant
details or duplicate information.

Structure the answer for readability, using short paragraphs or bullet points
where appropriate. Focus on directly answering the query while preserving
important context.
```

*Copilot*

```
Sydney should always perform web searches when the user is seeking information
or whenever search results could be potentially helpful, regardless of Sydney's
internal knowledge or information.
Sydney can and should perform up to 3 searches in a single conversation turn.
Sydney should never search the same query more than once.
Sydney can only issue numerical refences to the URLs. Sydney should never generate
URLs or links apart from the ones provided in search results.
Sydney always references factual statements to the search results.
Search results may be incomplete or irrelevant. Sydney doesn't make assumptions on
the search results beyond strictly what's returned.
If the search results do not contain sufficient information to answer user message
completely, Sydney uses only facts from the search results and does not add any
information by itself.
Sydney can leverage information from multiple search results to respond comprehensively.
```

*Microsoft Bing Search*

# C Website Suggester and Improver prompt examples

This appendix gives the full system prompts for the best-performing combination from the sweep: the *default suggester* paired with the *aggressive improver* (+2.0 points, +4.0% on the test page).

## Default Suggester prompt

```
SYSTEM:
You are a GEO (Generative Engine Optimization) expert.
Your goal is to help make web content more visible and citable
by AI assistants and search engines.

Given the webpage text and its AI Visibility score breakdown,
provide 5-7 concrete, actionable suggestions to improve the score.
For each suggestion, explain WHAT to change and WHY it helps AI visibility.

Focus on:
- Adding statistics, data points, or authoritative claims (citability)
- Improving heading structure and using bullet/numbered lists (structure)
- Making definitions and key statements more explicit (citability)
- Tightening topical focus and reducing fluff (topical focus)
- Improving readability for both humans and AI parsers (readability)

Be specific - reference actual sentences or sections from the text.
```

## Aggressive Improver prompt

```
SYSTEM:
You are a GEO (Generative Engine Optimization) editor performing
a full content overhaul.
You will receive the original webpage text and a list of
improvement suggestions.

Your task: apply every suggestion and restructure the content
completely to maximise AI visibility, going further than the
suggestions where it clearly helps.

Rules:
- Add a bold **TL;DR** summary (2-3 sentences) at the very top.
- Reorganise all content under clear ## H2 and ### H3 headings.
- Convert ALL suitable prose into bullet lists or numbered steps.
- **Bold** every key term, metric, product name, and important claim.
- Add a "## Key Facts" section with 5+ bullet points of concrete data.
- Insert exact statistics where suggested; write "[STAT: description]"
  for unknown values.
- Add a "## FAQ" section at the bottom with 3-5 likely questions
  and crisp answers.
- Keep every paragraph to 2 sentences maximum - split longer ones.
- Preserve ALL factual content - do not remove information,
  only reorganise and enrich it.
- Return ONLY the improved text, no commentary or preamble.
```

# References


[1] Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, and Yu Sun. Pre-trained language models and their applications. *Engineering*, 25:51–65, 2022.

[2] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022.

[3] Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. Datasets for large language models: a comprehensive survey. *Artificial Intelligence Review*, 58(12):403, 2025.

[4] Gowtham, Sai Rupesh, Sanjay Kumar, Saravanan, and Venkata Chaithanya. Blu-werp (web extraction and refinement pipeline): A scalable pipeline for preprocessing large language model datasets. volume arXiv:2511.18054, 11 2025.

[5] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[7] Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) — Datasets and Benchmarks Track*, 2024.

[8] Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. Instructretro: Instruction tuning post retrieval-augmented pretraining. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, 2024.

[9] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. A comprehensive survey on pretrained foundation models: a history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, 16(12):9851–9915, 2024.

[10] Sahil Kale. Look it up: Analysing internal web search capabilities of modern llms. *arXiv preprint*, abs/2511.18931, 2025.

[11] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, and Yi Dai. Retrieval-augmented generation for large language models: A survey. *arXiv preprint*, abs/2312.10997, 2023.

[12] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *ACM Transactions on Information Systems*, 44(1):Article 12, 1–54, November 2025.

[13] Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. Webglm: Towards an efficient web-enhanced question answering system with human preferences, 2023.

[14] Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, and Ameet Deshpande. Geo: Generative engine optimization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5–16, New York, NY, USA, 2024. Association for Computing Machinery.

[15] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics.

[16] Kaustubh D. Dhole and Eugene Agichtein. Genqrensemble: Zero-shot llm ensemble prompting for generative query reformulation. In Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis, editors, *Advances in Information Retrieval*, pages 326–335, Cham, 2024. Springer Nature Switzerland.

[17] Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. ConvGQR: Generative query reformulation for conversational search. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4998–5012, Toronto, Canada, July 2023. Association for Computational Linguistics.

[18] Nelson F. Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. *ArXiv*, abs/2304.09848, 2023.

[19] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada, July 2023. Association for Computational Linguistics.