

MODULE 1

Introduction & User Interfaces for Search

MODULE 1: Introduction & User Interfaces for Search

Textbook Reference: Ch.1 (1.1-1.4), Ch.2 (2.1-2.5)

1. The Information Retrieval (IR) Problem

Information Retrieval is the science of searching for information within documents, or searching for documents themselves, based on an information need expressed as a query. Unlike Data Retrieval (DR), which deals with structured data and exact matching (e.g., SQL queries on a database), IR deals with unstructured or semi-structured natural-language text and returns results ranked by relevance, not exact match.

1.1 Data Retrieval vs Information Retrieval

Aspect	Data Retrieval	Information Retrieval
Matching	Exact match	Partial / best match, ranked
Query language	Formal (SQL, relational algebra)	Free text / natural language / keywords
Data model	Structured, well-defined schema	Unstructured / semi-structured text
Result	Objects satisfying query exactly	Documents ordered by estimated relevance
Error tolerance	Small errors are significant	Small errors tolerated, still useful

1.2 The IR System — Building Blocks

- Text Operations (preprocessing): tokenization, stopword removal, stemming — converts raw text into index terms.
- Indexing: builds an efficient internal representation (typically an inverted index) of the collection.
- Searching: retrieves documents that satisfy the query using the index.
- Ranking: orders retrieved documents by estimated relevance, using a retrieval model.
- User Interface: manages query specification, results display, and relevance feedback.
- Query Operations: query expansion, spelling correction, feedback processing.

Exam Tip: Draw the block diagram: User -> Query Operations -> Searching <- Indexing <- Text Operations <- Text Database, with Ranking feeding results back to the User Interface. A very common 'draw and explain' question.

1.3 The IR Problem — Formal View

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

Given a collection of documents D and a query q representing the user's information need, the system returns a ranked subset of D estimated to be relevant. Relevance is subjective, situational, and a matter of degree — the central reason IR ranks results instead of merely filtering them.

- Information need: the underlying, often vague need in the user's mind.
- Query: the formal expression of that need submitted to the system.
- Relevance: correspondence between a document and the need — may be topical, user, or situational.

1.4 The Web as a Special IR Environment

- Huge and rapidly growing, with no central editorial control over content or quality.
- Highly dynamic — pages are created, updated, and removed continuously, requiring re-crawling.
- Heterogeneous — mixed formats, languages, quality, duplication (mirrors, near-duplicates).
- Hyperlinked — link structure carries relevance/authority signals (exploited by PageRank, HITS).
- Adversarial — subject to spam and search-engine-optimization manipulation.

2. User Interfaces for Search

2.1 Introduction

The user interface (UI) bridges the user's information need and the retrieval engine. It must support query formulation, iterative refinement, and effective presentation of results, since search is an interactive, iterative process rather than a single query-answer exchange.

2.2 How People Search

- Problem/need recognition: the user senses a gap in knowledge (Belkin's Anomalous State of Knowledge — ASK).
- Query formulation: translating the need into keywords — often imprecise on the first attempt.
- Results scanning: users typically scan only the first page/first few results (position bias).
- Relevance judgment and reformulation: the user refines the query — broadening, narrowing, or pivoting.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- Berry-picking model (Bates): information needs and queries evolve continuously as the user learns from results; search is a sequence of small, evolving queries, not one static query.
- Information foraging theory: users follow 'information scent' — cues such as titles, snippets, and links — to decide which path is worth pursuing.

2.3 Search Interfaces Today

- Query box with autocomplete and query suggestions.
- Result list with title, URL, and a query-biased snippet (summary highlighting matched terms in context).
- Faceted navigation panels — filters by category, date, price, etc., for structured drill-down.
- Spelling correction ('Did you mean...') and related/expanded query suggestions.
- Rich results: images, maps, knowledge panels, direct answers.

2.4 Visualization on Search Interfaces

- Starfield displays / scatterplots: plot documents on two attributes (e.g., date vs relevance).
- Cluster/graph visualizations (e.g., Scatter/Gather): group similar documents so users browse by topic instead of rank.
- TileBars / term-distribution visualizations: show where query terms occur inside a document.
- Tag clouds: frequency-weighted terms from a collection or result set.
- Timeline visualizations: show result distribution over time — useful for temporal/news queries.

2.5 Design and Evaluation of Search Interfaces

Design principles (from usability/HCI heuristics):

- Give informative feedback (result counts, corrections, terms used).
- Reduce short-term memory load — keep the query visible, highlight matched terms.
- Support easy reversal of actions (return to a previous query/result set).
- Ensure consistency; minimize user effort to reach relevant information.

Evaluation methods:

- System-based/quantitative: task completion time, number of queries issued, click-through rate.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- User-based/qualitative: think-aloud protocols, satisfaction surveys, usability testing, eye-tracking.

Important Definitions: Information scent = perceived relevance of a link/snippet based on visual/textual cues. ASK (Anomalous State of Knowledge) = Belkin's model: users search because they recognize a knowledge gap they cannot fully specify. Berry-picking = Bates' model of evolving, iterative search.

3. Worked Example — A Complete Search Session

To connect the theory above, trace a realistic session using the Berry-picking model:

- Step 1 (ASK): A student has a vague sense that 'search engines don't just match keywords' but cannot articulate the exact question — an anomalous state of knowledge.
- Step 2 (Formulation): They type the query 'how search engines rank pages'.
- Step 3 (Scan): The result snippets mention 'PageRank' repeatedly — this term acts as strong information scent.
- Step 4 (Pivot): The user reformulates the query to 'PageRank algorithm explained' — the query has evolved based on what was learned, not simply corrected.
- Step 5 (Berry-picking continues): Reading the PageRank page, a link to 'HITS algorithm' provides new scent; the user follows it, gathering information incrementally from multiple documents rather than a single 'final' answer — this incremental gathering is the essence of the Berry-picking model.

Why this matters for exams: Examiners frequently ask you to 'illustrate the Berry-picking model with an example' — always show the query evolving across at least 2-3 steps driven by information scent, not just define the model abstractly.

4. Summary of Key Points — Module 1

- IR differs from DR in matching type (best-match vs exact), result (ranked vs exact set), and tolerance to error.
- An IR system = Text Operations + Indexing + Searching + Ranking + Query Operations + User Interface.
- The Web adds scale, dynamism, heterogeneity, hyperlink structure, and adversarial spam to the classical IR problem.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- Search is iterative (Berry-picking) and driven by information scent (Information Foraging Theory), not a single query-answer step.
- Good interface design reduces memory load, gives feedback, and supports reversible actions; evaluation combines system metrics with user studies.

5. Glossary of Key Terms

Term	Definition
Information need	The underlying, often vague, gap in knowledge motivating a search.
Query	The formal expression of an information need submitted to the system.
Relevance	The degree to which a document satisfies an information need (topical/user/situational).
ASK	Anomalous State of Knowledge — Belkin's model of why users search.
Berry-picking	Bates' model: search needs and queries evolve continuously during a session.
Information scent	Perceived value of a link/snippet in leading to relevant information.
Faceted search	Filtering results using structured attribute categories (date, price, type).

6. Quick-Revision Question List — Module 1

- Differentiate Data Retrieval and Information Retrieval.
- Draw and explain the block diagram of an IR system.
- Explain the formal view of the IR problem (information need, query, relevance).
- List and explain the challenges the Web poses for IR.
- Explain the ASK model and the Berry-picking model with an example.
- Explain Information Foraging Theory and information scent.
- List the features of modern search interfaces.
- Explain any four visualization techniques for search results.
- Explain design principles and evaluation methods for search interfaces.

7. Detailed Model Answers — Module 1 (Exam-Ready)

The following are full, exam-length model answers to the Quick-Revision questions above. Use these as a template for how much detail a 20-mark VTU-style answer should contain.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML**Q1. Differentiate between Data Retrieval and Information Retrieval.**

Data Retrieval (DR) and Information Retrieval (IR) both aim to satisfy a user's need for information, but they differ fundamentally in philosophy. DR systems, such as relational databases, operate over structured, well-defined data using formal query languages like SQL. A DR query is evaluated using exact, Boolean logic: a record either satisfies the query condition or it does not, and there is no notion of 'partial' correctness — an off-by-one error in a WHERE clause produces a completely wrong (and unacceptable) answer set. IR systems, by contrast, operate over unstructured or semi-structured natural-language text where the user's information need is inherently imprecise. IR queries are matched using best-match/ranked retrieval: a document can be more or less relevant to a query, and the system's job is to estimate and order documents by their probability or degree of relevance. Consequently, IR systems tolerate small errors gracefully — a slightly imperfect ranking still delivers a useful, browsable list of results — whereas DR systems cannot tolerate even small logical errors. This difference cascades into every layer of system design: indexing strategy, ranking algorithms, evaluation metrics (precision/recall vs correctness), and user interface (a ranked list with snippets vs a table of exact rows).

Q2. Draw and explain the block diagram of an IR system.

A generic IR system is organized as a pipeline with a feedback loop. The Text Database supplies raw documents to the Text Operations module, which performs tokenization, stopword removal, and stemming to produce clean index terms. These terms flow into the Indexing module, which builds an inverted index (vocabulary + postings). On the query side, the user submits a request through the User Interface; this raw query passes through the same (or a lighter) set of Query Operations — including spelling correction and possible query expansion — before being handed to the Searching module. Searching consults the index to retrieve the candidate set of documents containing the query terms, and the Ranking module then scores and orders these candidates using a retrieval model (Boolean, Vector, Probabilistic, etc.). The ranked list is returned to the User Interface for display, and — crucially — the user's interaction with these results (clicks, relevance judgments) can loop back into Query Operations as relevance feedback, closing the loop and enabling iterative query refinement across a search session.

Q3. Explain the formal view of the IR problem.

Formally, an information need is the internal, often vague and evolving, gap in a user's knowledge that motivates them to search — it exists in the user's mind before any words are typed. A query is the external, formal translation of that need into a form the system can process (keywords, a Boolean expression, a natural-language sentence). These two are not identical: the same information need can be expressed by many different queries, and a poorly chosen query may fail to capture the true need at all.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

Relevance is the (subjective) correspondence between a specific document and the information need — not merely the query. A document can match a query's keywords perfectly yet be irrelevant to the underlying need (e.g., a query for 'Java' matching a document about the island of Java when the user meant the programming language), which is why relevance judgments in evaluation are made by humans assessing the actual need, not by simple keyword-overlap. This is the theoretical reason IR is fundamentally a ranking problem rather than an exact-match problem.

Q4. List and explain the challenges the Web poses for IR.

The Web stresses classical IR techniques along five main dimensions. First, scale: with tens of billions of pages, indexes must be distributed across thousands of machines, and even simple operations like computing global term statistics become large engineering problems. Second, dynamism: pages are created, edited, and deleted continuously, so a crawler's snapshot is always partially stale, requiring careful freshness/re-crawl scheduling. Third, heterogeneity: content spans many languages, formats, and quality levels, and near-duplicate/mirrored pages must be detected to avoid wasting index space and confusing rankings. Fourth, hyperlink structure: unlike a classical flat document collection, Web pages are richly interconnected, and this structure itself is a valuable relevance signal (exploited by PageRank and HITS) that classical models never had access to. Fifth, adversarial behaviour: because ranking well brings commercial value, the Web is subject to deliberate spam and search-engine-optimization manipulation, forcing search engines to defend their ranking functions against gaming — a concern that essentially does not exist in closed, curated collections like a library catalog.

Q5. Explain the ASK model and the Berry-picking model with an example.

Belkin's Anomalous State of Knowledge (ASK) model explains the origin of an information need: a user searches because they have recognized an anomaly or gap in their state of knowledge relative to some goal, but — crucially — they often cannot fully or precisely specify what is missing, which is precisely why their first query is frequently imprecise or exploratory. Bates' Berry-picking model then explains what happens after that first query is issued: rather than a single query producing a single final, complete answer (the traditional view), real search is a sequence of small steps. At each step the user's need itself evolves as they encounter new information, encountering new vocabulary, related topics, and useful side-paths — much like a berry-picker moving through a field, gathering a bit at a time from many bushes rather than harvesting one bush completely. For example, a user researching 'ranking algorithms' may start there, discover 'PageRank' in the results, refine their query to 'PageRank algorithm', and then, prompted by a link, pivot again to 'HITS vs PageRank' — the final understanding is assembled from fragments gathered across this whole evolving trail, not from one single query-result pair.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML**Q6. Explain Information Foraging Theory and information scent.**

Information Foraging Theory, adapted from animal foraging behaviour in ecology, models a searcher as trying to maximize the value of information gained per unit of effort/time spent. At each decision point (which result to click, which link to follow), the user relies on 'information scent' — the perceived cues available before actually visiting a page, such as a link's anchor text, a result's title, or a highlighted snippet — to predict how valuable that path is likely to be. Strong scent (a snippet that closely and specifically matches the need) encourages the user to follow that path; weak or misleading scent causes the user to abandon it and try another. This theory explains observed search behaviour such as users rarely scrolling past the first page of results (position bias) and heavy reliance on snippet quality, and it directly motivates interface design choices like query-biased snippet generation, since improving scent accuracy measurably improves user efficiency.

Q7. List the features of modern search interfaces.

Modern web search interfaces share a common set of features that together minimize user effort. The query box typically offers autocomplete and query suggestions, reducing typing effort and helping users formulate better queries before they even submit. The results list displays a title, a URL/breadcrumb, and a query-biased snippet — a short extract from the page selected and highlighted specifically because it contains the query terms in context, rather than simply the page's opening sentence. Faceted navigation panels let users narrow results using structured attributes such as date, category, or price without re-typing a query. Spelling correction ('Did you mean...') gracefully handles typos, and related-query suggestions help users explore adjacent topics. Finally, rich results — images, maps, direct 'answer box' responses, and knowledge panels — increasingly let simple informational needs be satisfied without even clicking through to a page, reflecting the interface's evolution from a mere link-list to an integrated answer-delivery system.

Q8. Explain any four visualization techniques for search results.

Starfield displays plot each retrieved document as a point on a 2-D scatterplot using two meaningful attributes (e.g., publication date on one axis, relevance score on the other), letting users spot clusters and outliers visually rather than scanning a flat list. Cluster-based visualizations, such as the classic Scatter/Gather technique, group similar documents together and let the user browse by selecting whole topic clusters, which is especially useful when a query is ambiguous or broad and a single ranked list would mix several distinct senses together. TileBars display, for each retrieved document, a compact bar showing exactly where in the document each query term occurs, letting a user judge at a glance whether a long document discusses all query terms together in one place or scattered separately. Tag clouds show

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

frequency-weighted terms (larger font for more frequent terms) drawn either from the whole collection or from a specific result set, giving a quick visual summary of dominant topics without reading any individual document.

Q9. Explain design principles and evaluation methods for search interfaces.

Good interface design follows established HCI heuristics adapted to search: give informative feedback (always show result counts, applied corrections, and which query was actually run); minimize the user's short-term memory load (keep the current query visible at all times, and highlight matched terms directly in result snippets so the user doesn't need to re-read to check relevance); support easy reversal of actions (a visible, working back button to a previous query or result set, since query reformulation is the norm, not the exception); and maintain consistency across the interface so learned behaviour transfers between sessions. Evaluation combines two complementary approaches. System-based/quantitative evaluation measures objective behavioural signals such as task completion time, number of queries needed to satisfy a need, and click-through rate on top results — these are cheap to collect at scale (via logs) but do not directly capture subjective satisfaction. User-based/qualitative evaluation instead studies real users performing realistic tasks using methods like think-aloud protocols (users verbalize their reasoning while searching), post-task satisfaction surveys, formal usability testing, and eye-tracking studies — these are more expensive but reveal why users struggle, not just that they struggled.

8. Real-World Application — How Search Engines Apply These Concepts

Consider how a commercial engine such as Google or Bing operationalizes the theory above. The gap between Data Retrieval and Information Retrieval is visible the moment you compare a SQL query against a company database (exact rows only) to a web search query (a ranked, best-effort list with 'About X results' shown, acknowledging uncertainty in the count itself). The IR system's block-diagram pipeline maps directly onto the engine's architecture: text operations and indexing happen offline across a massive crawl, while searching, ranking, and the user interface happen online, in real time, within a strict latency budget (often under a few hundred milliseconds). The Berry-picking model is why engines keep a visible search history and 'related searches' — they are explicitly designed to support session-level, evolving search rather than assuming one query fully satisfies the user. And information scent directly motivates continuous investment in snippet-generation quality, since a snippet that accurately previews a page's relevance reduces wasted clicks and improves perceived search quality even when the underlying ranking algorithm is unchanged.

9. Common Mistakes & Exam Tips — Module 1

- Do not confuse 'precision' in the general sense with the IR metric of the same name introduced in Module 3 — in Module 1, discussions of interface quality are qualitative unless a specific metric is asked for.
- When asked to 'explain the IR system with a diagram', always show the feedback loop from Ranking/User Interface back into Query Operations — omitting the feedback loop is the most common mark-losing mistake.
- Berry-picking and ASK are frequently confused: ASK explains why the very first query is issued (a knowledge gap), while Berry-picking explains what happens across the whole session afterward (evolving queries). An answer that only defines one of them when both are asked will lose marks.
- When listing visualization techniques, name the technique AND briefly explain what specific problem it solves — a bare list of names without explanation earns partial credit only.

10. Additional Solved Practice Problems — Module 1

Problem 1: Classify the scenario

Scenario: A librarian runs the query 'SELECT * FROM books WHERE author="Baeza-Yates" AND year=2011' against a catalog database, versus a student typing 'best book on information retrieval' into a web search box. Classify each as DR or IR and justify.

Solution: The librarian's query is Data Retrieval: it uses a formal query language (SQL) against structured, schema-defined data, and expects an exact set of matching rows with no notion of partial correctness. The student's query is Information Retrieval: it is free natural-language text against an unstructured web collection, and the system must return a ranked, best-effort list of pages since no page can be proven 'exactly correct' the way a database row can.

Problem 2: Identify the IR system component

For each task, name which IR system block is responsible: (a) removing 'the', 'is', 'of' from a document before indexing; (b) computing a cosine similarity score between a query and a document; (c) accepting a user's typed query and displaying snippet results.

Solution: (a) Text Operations (stopword elimination). (b) Ranking (applying a retrieval model's scoring function). (c) User Interface (query input and results display).

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML**Problem 3: ASK vs Berry-picking**

A user cannot articulate why a piece of code fails, so they search 'python function not returning value'. After reading a result about the 'return' keyword, they reformulate to 'python return vs print difference'. Identify which part of this scenario illustrates ASK and which illustrates Berry-picking.

Solution: The user's initial inability to fully articulate the problem (a vague sense that 'something about how the function ends' is wrong) illustrates the Anomalous State of Knowledge (ASK) — they recognize a gap but cannot precisely specify it. The reformulation from the first query to the second, driven by what was learned from the first set of results, illustrates the Berry-picking model — the information need itself evolved mid-session rather than being fixed from the start.

Problem 4: Design critique

A search interface shows results with no visible query box (it disappears after search) and no highlighting of matched terms in snippets. Identify which design principles this violates and propose fixes.

Solution: This violates 'reduce short-term memory load' (the user must remember what they searched for since the query box vanished) and 'give informative feedback' (unhighlighted snippets force the user to re-read carefully to judge relevance instead of scanning quickly). Fixes: keep the query box populated and visible at all times, and bold/highlight query terms within every snippet.

11. Algorithm/Process Reference — Module 1

Pseudocode: Basic IR System Query Pipeline

```
Pseudocode: function processQuery(rawQuery): cleanQuery = textOperations(rawQuery) // tokenize,
stopword removal, stemming expandedQuery = queryOperations(cleanQuery) // spelling correction,
expansion candidates = search(index, expandedQuery) // fetch postings for query terms ranked =
rank(candidates, expandedQuery, model) // score using Boolean/Vector/Probabilistic model display(ranked)
// show snippets, facets, highlighting on userFeedback: update(queryOperations) // relevance feedback
loop
```

12. Historical Context & Evolution

Early IR systems in the 1960s-70s (the era of the Cranfield experiments) worked over small, curated bibliographic collections using Boolean queries entered by trained librarians — search was a specialist skill, not a consumer activity. The shift toward ranked retrieval and the Vector/Probabilistic models through the 1970s-80s made systems usable without Boolean expertise. The explosive growth of the Web through the 1990s forced an entirely new set of concerns — scale, link structure, spam, heterogeneous content quality

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

— that pre-Web IR theory had never needed to address, directly motivating the Web-specific material developed later in Module 5 (PageRank, HITS, crawling). User-interface research evolved in parallel: the single-shot 'query box + result list' paradigm of early search engines has been progressively enriched with autocomplete, faceted filtering, rich results, and personalization, all grounded in the human-search-behaviour theories (ASK, Berry-picking, Information Foraging) that this module covers.

13. Frequently Asked Questions — Module 1

- Q: Is a database keyword search (e.g., searching a product name in an e-commerce site) DR or IR?
A: It depends on the matching used — if it does exact/prefix string matching with no ranking, it leans DR; if it ranks products by relevance/popularity even for partial text matches, it is functioning as IR despite sitting on structured data.
- Q: Can a system use more than one visualization technique at once? A: Yes — commercial engines often combine several (e.g., snippet highlighting plus a knowledge panel plus related-search suggestions) simultaneously, each addressing a different aspect of the user's need.
- Q: Why doesn't Boolean search dominate the consumer web anymore? A: Because it requires query-formulation skill most users lack, gives no ranking (so result order conveys no relevance information), and cannot gracefully handle vocabulary mismatch — all problems the Vector/Probabilistic models and their descendants directly solve.
- Q: Is 'information scent' the same as a snippet? A: No — the snippet is the interface ELEMENT; information scent is the user's PERCEPTION of relevance triggered by that element (and by titles, URLs, and other cues together).

14. Self-Assessment Checklist — Module 1

- I can state at least three differences between Data Retrieval and Information Retrieval.
- I can draw the IR system block diagram from memory, including the feedback loop.
- I can define information need, query, and relevance, and explain why they are distinct.
- I can list at least four ways the Web complicates classical IR.
- I can explain ASK and Berry-picking with a fresh example of my own (not textbook).
- I can explain information scent and information foraging theory.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- I can name and explain at least four search-interface visualization techniques.
- I can list interface design principles and both categories of evaluation method.

15. Multiple Choice Review Questions — Module 1

Attempt each question, then check the answer and explanation below it.

- Q1. Which of the following best distinguishes IR from DR? (a) IR uses SQL (b) IR returns a ranked, best-match result set (c) DR tolerates small errors (d) DR has no schema. Answer: (b) — IR's defining feature is best-match, ranked retrieval rather than exact matching; the others describe DR properties or are false.
- Q2. In the IR system block diagram, which block is directly responsible for building the inverted index? (a) Text Operations (b) Indexing (c) Searching (d) Ranking. Answer: (b) Indexing — Text Operations prepares clean terms, but Indexing is the block that structures them into the searchable index.
- Q3. The Anomalous State of Knowledge (ASK) model was proposed by: (a) Bates (b) Belkin (c) Salton (d) Kleinberg. Answer: (b) Belkin — Bates proposed the related but distinct Berry-picking model.
- Q4. Information scent is best described as: (a) a ranking algorithm (b) a perceived cue predicting a link's relevance (c) a stopword list (d) a compression technique. Answer: (b) — it is a perceptual/cognitive cue from Information Foraging Theory, not an algorithm.
- Q5. Which visualization technique specifically shows WHERE query terms occur within a single long document? (a) Tag cloud (b) TileBars (c) Starfield display (d) Timeline. Answer: (b) TileBars — designed exactly for intra-document term-distribution visualization.
- Q6. Faceted navigation primarily supports: (a) spelling correction (b) structured filtering by attributes (c) relevance feedback (d) stemming. Answer: (b) — facets let users filter by structured attributes like date/category/price.
- Q7. The Berry-picking model's central claim is that: (a) one query always suffices (b) queries evolve continuously across a search session (c) Boolean queries are superior (d) users never reformulate. Answer: (b) — this is the model's defining, exam-tested claim.
- Q8. 'Reducing short-term memory load' as a design principle is best supported by: (a) hiding the query box (b) keeping the query visible and highlighting matched terms (c) removing snippets (d) requiring Boolean syntax. Answer: (b).

16. Extended Case Study — Redesigning a Weak Search Interface

Consider a fictional internal company knowledge-base search that currently returns a flat, unranked alphabetical list of document titles with no snippets, no query box retained after search, and no spelling correction. Employees report frequently giving up before finding what they need. Applying this module's principles: first, the ranking problem is addressed by replacing alphabetical ordering with a Vector or Probabilistic ranking model (Module 2) so the most relevant documents surface first rather than requiring alphabetical luck. Second, query-biased snippets should be generated and query terms highlighted within them, directly reducing memory load and strengthening information scent, so employees can judge relevance without opening each document. Third, the query box should remain populated after search (supporting easy reformulation, core to the Berry-picking model) and spelling correction should be added, since internal terminology (product codenames, acronyms) is exactly the kind of vocabulary users often mistype. Fourth, adding lightweight facets (document type, department, date) would let employees narrow a broad initial query without retyping — directly reflecting the 'search interfaces today' feature set covered earlier. This case study illustrates how nearly every concept in Module 1 — ranking, snippets, information scent, Berry-picking, facets — maps onto concrete, testable interface improvements, which is exactly the kind of applied reasoning exam answers should demonstrate.

17. Cross-Module Connections

Module 1 sets up problems that later modules solve concretely: the 'ranking' need identified here in the IR system block diagram is fully developed in Module 2's IR models (Boolean, Vector, Probabilistic, BM25); the 'evaluation' methods introduced briefly here (system-based vs user-based) are formalized rigorously in Module 3's retrieval metrics (Precision, Recall, MAP); and the Web-specific challenges flagged in section 1.4 are addressed in full in Module 5 (crawling, PageRank, HITS). Recognizing these connections is valuable for synthesis-style exam questions that span multiple modules.

18. Diagram Walkthroughs — What to Draw and How

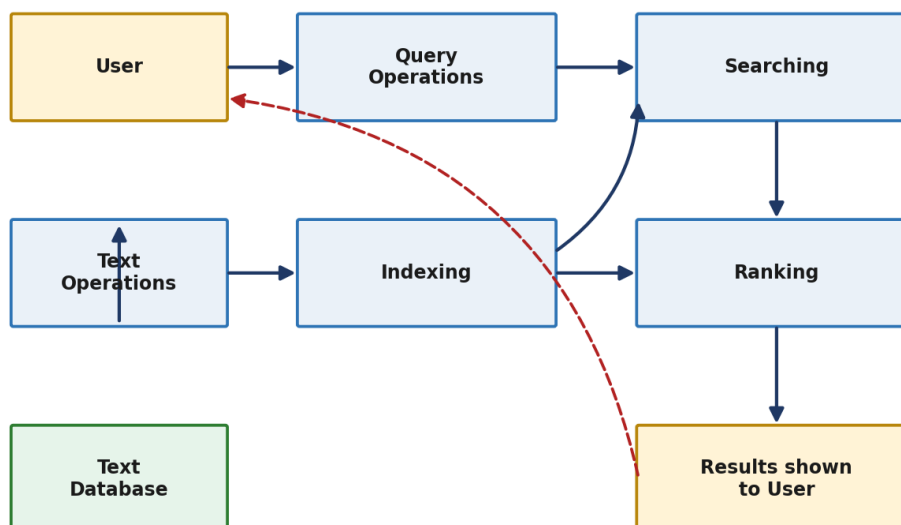
Diagram 1: The IR System Block Diagram (step-by-step drawing guide)

- Step 1: Draw a box labeled 'Text Database' on the middle-left of the page — this is the source of all documents.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- Step 2: From it, draw an arrow rightward into a box labeled 'Text Operations' (tokenization, stopwords, stemming).
- Step 3: From Text Operations, draw an arrow into a box labeled 'Indexing', producing the inverted index.
- Step 4: On the top of the page, draw a box labeled 'User', with an arrow down into a box labeled 'Query Operations' (spelling correction, expansion).
- Step 5: Draw arrows from BOTH Indexing and Query Operations converging into a central box labeled 'Searching'.
- Step 6: From Searching, draw an arrow into 'Ranking', which applies a retrieval model to score/order documents.
- Step 7: From Ranking, draw an arrow back up to the 'User' box (via the User Interface) — this delivers results.
- Step 8: CRITICAL — draw a dashed feedback arrow FROM the User box BACK INTO 'Query Operations', labeled 'relevance feedback' — this closes the loop and is the single most commonly forgotten element.

Figure 1.1 — Information Retrieval System: Block Diagram



Dashed red arrow = Relevance Feedback loop (Results -> Query Operations)

Figure 1.1 — The IR System block diagram rendered from the steps above.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

Diagram 2: The Berry-Picking Search Session (step-by-step drawing guide)

- Step 1: Draw a small circle labeled 'ASK' on the far left, representing the user's initial vague knowledge gap.
- Step 2: Draw an arrow to a box labeled 'Query 1', then an arrow down to a 'Results 1' box.
- Step 3: From Results 1, draw a branching arrow labeled 'information scent' leading to a new box 'Query 2' (reformulated, not identical to Query 1).
- Step 4: Repeat the pattern for 'Query 3', showing at least 3 total query-result cycles.
- Step 5: Draw a large dashed oval around all the Query/Results boxes together, labeled 'Evolving Information Need' — this visually communicates that the need itself changed across the whole diagram, not just the wording of the query.

Figure 2.1 — Berry-Picking Model of an Evolving Search Session

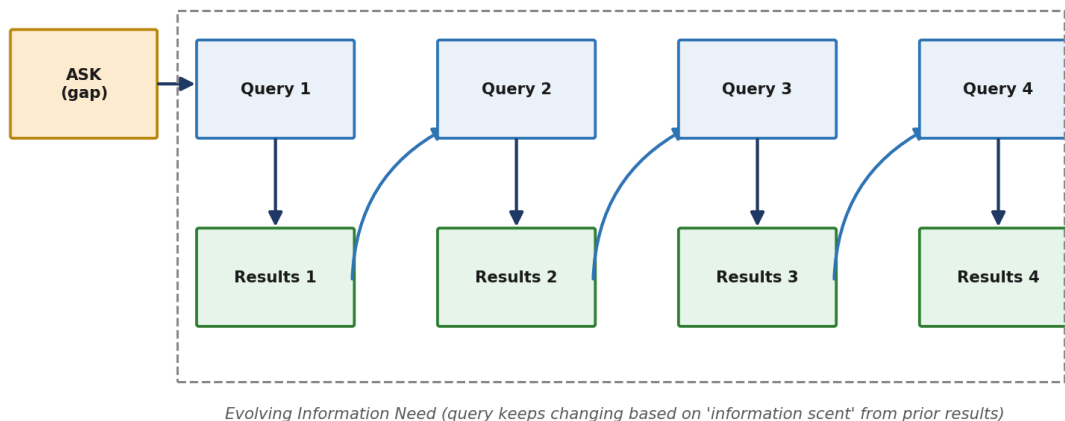


Figure 2.1 — The Berry-Picking model rendered as an evolving query/results sequence.

19. Unsolved Practice Problems (Self-Study, with Hints)

- 1. Classify: A user asks a librarian a vague question about 'books like the one I read last summer.' Is the librarian's subsequent search process closer to DR or IR? (Hint: consider whether the librarian's search criteria are exact or approximate.)
- 2. Draw the IR system block diagram from memory without looking back at this document, then compare against Section 2's model answer.
- 3. Give an original (non-textbook) example of the ASK model from your own recent experience.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

- 4. List three specific ways a mobile search interface's design constraints (small screen, touch input) might change how the design principles in this module are applied.
- 5. Explain why 'number of results found' displayed at the top of a search page is an example of 'informative feedback' as a design principle.
- 6. A user's Berry-picking session involves five successive queries. Sketch (in words) what information scent might have driven each transition.
- 7. Compare a starfield display and a tag cloud: what does each show that the other does not?
- 8. Explain, using the Web characteristics from Section 1.4, why a search engine indexing only academic journals would face far fewer challenges than one indexing the open Web.

20. Full Sample 20-Mark Exam Answer — Module 1

Question: 'Explain the building blocks of an IR system with a neat block diagram, and relate them to how a user's search session actually unfolds.' (20 marks)

Model Answer: An Information Retrieval system is organized as a pipeline of cooperating blocks, each responsible for one stage of transforming raw text and a raw query into a ranked, useful result list. The Text Database supplies the raw document collection. These documents first pass through Text Operations, where tokenization splits text into words, stopword elimination removes very common low-information terms (justified statistically by Zipf's Law), and stemming reduces morphological variants to a common root so that, for example, a query for 'search' can also match a document containing only 'searching'. The cleaned index terms then flow into Indexing, which builds an inverted index — a sorted vocabulary of terms, each pointing to a postings list of the documents containing it — enabling fast term lookups instead of a slow linear document-by-document scan. On the query side, when a user submits a raw query through the User Interface, it passes through an analogous, often lighter, Query Operations stage, which may apply spelling correction or query expansion before the query reaches Searching. The Searching block consults the inverted index to retrieve the candidate documents containing the query's terms (for example, intersecting postings lists for a Boolean AND query), and the Ranking block then applies a chosen retrieval model (Boolean, Vector, Probabilistic, or a variant such as BM25) to score and order these candidates by estimated relevance. The final ranked list returns to the User Interface, where it is presented with titles, snippets, and highlighting. Crucially, this is not the end of the process: as the user scans the results, their reaction — a click, a reformulated query, an explicit relevance judgment — feeds back into Query Operations, closing a feedback loop that enables exactly the kind of iterative, evolving

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

search session described by the Berry-picking model, where a single query rarely satisfies the full information need and the system must support graceful, informed reformulation across several rounds. This tight correspondence between the system's architecture and the theory of how people actually search is precisely why both halves of Module 1 — system components and user behaviour models — are taught together.

21. Extended Comparison Reference — Module 1

Aspect	Classical Library Search	Modern Web Search
Query style	Controlled vocabulary, catalog cards	Free natural-language keywords
User expertise assumed	Trained librarian intermediary	General public, no training
Result presentation	Sorted card catalog / shelf list	Ranked list with snippets, rich results
Feedback loop	Manual reference interview	Autosuggest, click behaviour, session history
Scale	Thousands to millions of items	Tens of billions of pages
Structure exploited	Subject headings, call numbers	Hyperlinks, page structure, user behaviour

This table is useful for 'compare classical vs modern search' style questions, and directly ties together the DR/IR distinction, the IR system block diagram, and the Web-specific challenges covered earlier in this module.

22. Additional Worked Scenario — Interpreting a Search Log

Consider a short session log: Query 1 = 'python error' (2 results clicked, both abandoned quickly); Query 2 = 'python TypeError string int' (1 result clicked, session ends). Applying this module's concepts: the transition from Query 1 to Query 2 is a textbook Berry-picking reformulation — the user's information need sharpened based on what they learned (likely the exact error message) from the first round of results, rather than the first query simply being 'wrong'. The quick abandonment of both Query 1's results suggests weak information scent: the snippets likely did not clearly signal whether they addressed the SPECIFIC error type the user was facing, so the user could not efficiently judge relevance before clicking, wasting effort on unhelpful pages. The successful single click on Query 2 suggests that once the query became specific enough, snippet-level scent was strong enough to guide the user directly to a satisfying result. A system designer reviewing this log might conclude that snippet generation should more aggressively surface exact error-message matches, directly addressing the scent weakness observed at Query 1 — demonstrating how the theoretical concepts of this module (Berry-picking, information scent) translate into concrete, actionable interface engineering decisions derived from real usage data.

23. Comprehensive Quick-Reference Sheet — Module 1

Concept	One-Line Reference
DR vs IR	DR = exact match, structured data; IR = ranked best-match, unstructured text
IR system blocks	Text DB -> Text Ops -> Indexing -> Searching -> Ranking -> UI (+ feedback loop)
Information need	Vague internal gap in knowledge motivating a search
Query	Formal expression of the information need submitted to the system
Relevance	Subjective, graded correspondence between a document and the need
ASK	Belkin: users search due to an Anomalous State of Knowledge they can't fully specify
Berry-picking	Bates: information need evolves continuously across a search session
Information scent	Perceived cue (title/snippet/link) predicting a path's value
Faceted search	Structured filtering by attributes (date, category, price)
Starfield display	2-D scatterplot of documents on two meaningful attributes
TileBars	Visualization of WHERE query terms occur inside a document
Tag cloud	Frequency-weighted term visualization of a collection/result set

24. Chapter Summary Essay — Module 1

Module 1 establishes the conceptual foundation for everything that follows in this course. It begins by drawing a sharp line between Data Retrieval and Information Retrieval — a distinction that is not merely academic but shapes every subsequent design decision in an IR system, from how documents are represented to how results are ranked and evaluated. The IR system's building blocks (Text Operations, Indexing, Searching, Ranking, Query Operations, User Interface) form a pipeline that reappears, in more technical detail, across the rest of the syllabus: Module 2 elaborates the Ranking block into a full taxonomy of retrieval models; Module 3 provides the metrics to evaluate that ranking and the feedback mechanisms to improve Query Operations; Module 4 details exactly how the Indexing and Searching blocks are physically implemented; and Module 5 extends the whole pipeline to web scale and to structured documents. The second half of the module — human search behaviour (ASK, Berry-picking, Information Foraging) and interface design/evaluation — grounds all of this technical machinery in the reality that IR systems exist to serve real people with imprecise, evolving needs, not to solve a single, static, well-specified query. A strong grasp of Module 1 therefore pays off throughout the course: nearly every later technical concept can be traced back to a problem first identified here, whether it is the need for ranking (motivating Module 2's models), the need for measurable quality (motivating Module 3's metrics), the need for speed at scale (motivating Module 4's structures), or the unique demands of the open Web (motivating Module 5).

25. Final Revision Mnemonic — Module 1

A useful memory aid for this module's core theories: 'A-B-I-F' — ASK explains why a search Begins (a recognized knowledge gap); Berry-picking explains how it continues (evolving queries across a session); Information scent explains how each step is chosen (perceived link/snippet value); and Foraging theory ties it together as an effort-vs-value optimization process. For the system architecture, remember the pipeline order alphabetically by function: Text operations CLEAN the text, Indexing ORGANIZES it, Searching FINDS candidates, Ranking ORDERS them, and the User Interface DELIVERS and COLLECTS feedback — five verbs (clean, organize, find, order, deliver) that map directly onto the five core blocks and are easy to recall under exam pressure even if the exact diagram layout is momentarily forgotten.

26. One More Worked Comparison — Interfaces Across Device Types

Comparing a desktop search interface against a mobile/voice search interface reinforces this module's design principles concretely. A desktop interface can afford a dense results list with multiple facets visible simultaneously (large screen, mouse precision), directly supporting the 'give informative feedback' principle through visible result counts and filter chips. A mobile interface, constrained to a small screen, must compress this same feedback into fewer, more prioritized signals (e.g., a single top 'best answer' card rather than ten links), which is precisely why rich results and direct-answer boxes have grown in importance as mobile search overtook desktop search in volume. A voice-only interface goes further still: there is no screen at all, so information scent (normally conveyed visually via snippets and titles) must instead be conveyed through carefully ordered spoken summaries, and the Berry-picking model's support for iterative reformulation becomes even more critical, since a user cannot visually scan multiple results at once and must rely entirely on a conversational back-and-forth to refine their need. This progression — desktop to mobile to voice — is a useful lens for any exam question asking you to 'apply search interface design principles to a new context,' since it demonstrates the same underlying principles (informative feedback, reduced memory load, supporting reformulation) manifesting in very different concrete interface choices depending on the device's constraints.

27. Final Note — Why This Module Matters Beyond the Exam

Beyond exam preparation, Module 1's core ideas are the ones practicing search engineers, UX researchers, and information scientists actually reach for daily. Every time a product team debates 'why

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

are users abandoning search so quickly,' the diagnostic vocabulary is exactly this module's: is it a ranking problem (wrong documents surfaced — a Module 2 concern), a scent problem (right documents surfaced, but users can't tell from the snippet — a Module 1 concern), or a reformulation-support problem (users need to iterate but the interface fights them — again Module 1). Recognizing which category a real search-quality complaint falls into, using exactly the vocabulary and models covered here (ASK, Berry-picking, information scent, the IR system pipeline), is a genuinely transferable skill well beyond this specific course — which is also precisely why examiners return to these concepts across nearly every assessment in this subject, in various combinations and framings, rather than treating them as a one-time introductory topic to be memorized and set aside.

As a final consolidation exercise before moving to Module 2, try writing out — without referring back to this document — a two-minute spoken explanation covering: (1) the DR vs IR distinction with one example each, (2) the IR system pipeline in order, (3) the ASK and Berry-picking models with a fresh example, and (4) two visualization techniques with the specific problem each solves. If any of these four feels shaky, revisit the corresponding section above before proceeding, since Module 2 onward assumes this foundational vocabulary is already fluent and will build on it without re-explaining it.

A final consolidated timeline view, for quick recall of how the field arrived at today's interfaces, is worth memorizing: 1960s-70s — Boolean, librarian-mediated catalog search; 1970s-90s — ranked retrieval models emerge (Module 2 material) enabling non-expert users; 1990s — the Web arrives, forcing new scale and interface thinking (search engines as we recognize them today); 2000s — visualization, faceted search, and formal usability evaluation mature as distinct sub-disciplines; 2010s-present — mobile-first and voice interfaces, rich/direct-answer results, and personalization become the norm, all while the underlying ASK/Berry-picking/information-scent theories from decades earlier remain fully valid explanations of user behaviour, since human search psychology has not fundamentally changed even as the technology delivering results has been transformed many times over.