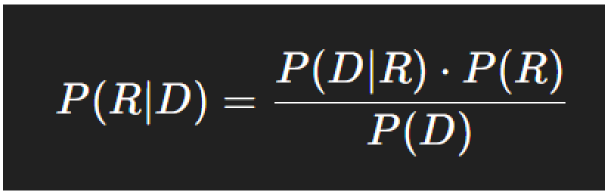
IR U 3

Probabilistic Retrieval: Review of Basic Probability Theory (8 Marks)

Probabilistic Retrieval is an information retrieval approach based on estimating the **probability that a document is relevant** to a user's query. The idea is that the system should rank documents in decreasing order of the probability of relevance. To understand this model, we must review the basic concepts of probability theory used to compute and update relevance scores.

The **sample space** is the entire document collection, and events such as "document is relevant (R)" or "document contains term t" are defined. **Conditional probability** is used to measure how likely a document is relevant given it contains a specific term: P(R|t)

A key tool is **Bayes' Theorem**, which allows rewriting the probability of relevance as:



$$P(R|D) = \frac{P(D|R) \cdot P(R)}{P(D)}$$

Since P(D) is constant across documents, ranking focuses on estimating P(D|R) and P(R). These probabilities are estimated from data, such as how often terms appear in relevant vs non-relevant documents.

An important assumption is **term independence**, meaning each term contributes separately to relevance. Although not perfect, it allows efficient computation and forms the basis of the **Binary Independence Model** and later models like **BM25**.

Thus, probability theory provides the mathematical foundation for estimating relevance, ranking documents, and improving retrieval through user feedback.

1. The Probability Ranking Principle (PRP): The 1/0 Loss Case

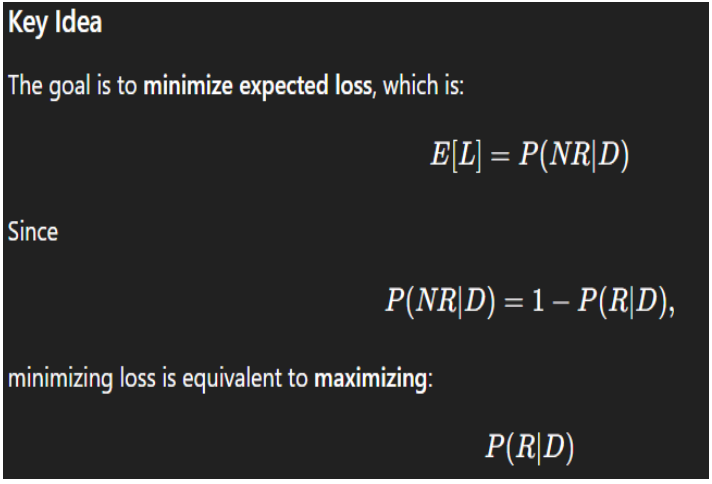
The **Probability Ranking Principle (PRP)** is a fundamental theory in Information Retrieval which states that:
"For optimal retrieval performance, documents should be ranked in decreasing order of their probability of relevance to the user's query."

In the **1/0 loss case**, the retrieval process assumes only two outcomes:

- **Loss = 0** → if a relevant document is retrieved
- **Loss = 1** → if a non-relevant document is retrieved

This means the system is punished only when it retrieves non-relevant documents, and there is no extra penalty for missing relevant ones.

Key Idea



Key Idea

The goal is to **minimize expected loss**, which is:

$$E[L] = P(NR|D)$$

Since

$$P(NR|D) = 1 - P(R|D),$$

minimizing loss is equivalent to **maximizing**:

$$P(R|D)$$

Thus, PRP dictates ranking documents by *decreasing* probability of relevance.

Why PRP Is Optimal

- It ensures that users see the most relevant documents first.
- It provides a mathematically sound ranking rule under uncertainty.
- It assumes perfect probability estimates, making the ranking theoretically optimal.

Assumptions

- Relevance is binary
- Loss for retrieving a relevant document = 0
- Loss for retrieving a non-relevant document = 1
- User prefers to avoid non-relevant documents

Significance

- Forms the foundation for probabilistic models like BIM and BM25.
- Provides a justification for why probability-based rankings outperform simple keyword matching.

2. The Probability Ranking Principle (PRP) with Retrieval Costs

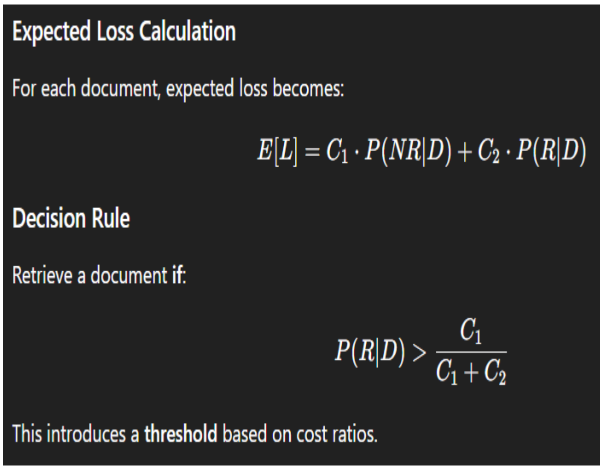
The basic PRP assumes equal loss for all retrieval errors. However, in real systems, **different types of errors have different costs**.

Realistic Scenario

- Missing a relevant document (false negative) may be very expensive → Example: medical diagnosis, legal information
- Retrieving a non-relevant document (false positive) may be less costly

Thus, losses are defined as:

- **C_1 = cost of retrieving a non-relevant document**
- **C_2 = cost of not retrieving a relevant document**



Expected Loss Calculation

For each document, expected loss becomes:

$$E[L] = C_1 \cdot P(NR|D) + C_2 \cdot P(R|D)$$

Decision Rule

Retrieve a document if:

$$P(R|D) > \frac{C_1}{C_1 + C_2}$$

This introduces a **threshold** based on cost ratios.

Interpretation

- If **C_2 is large** (missing relevant doc is costly) → Threshold becomes **smaller** → System retrieves more documents
- If **C_1 is large** (retrieving non-relevant doc is costly) → Threshold becomes **larger** → System becomes selective

Benefits

- Adapts retrieval to practical requirements.
- Allows designing retrieval systems for specific domains (healthcare, legal, finance).
- Provides a more flexible, cost-sensitive decision mechanism.

Significance

- Moves PRP beyond theoretical assumptions.
- Shows how probability-based retrieval can incorporate business or domain-specific priorities.

The Binary Independence Model (BIM)

The **Binary Independence Model (BIM)** is one of the earliest and most influential probabilistic models used in Information Retrieval. It provides a mathematical framework for ranking documents based on the **probability of relevance** to a user's query. The model makes two major assumptions: **terms are binary** (present or absent) and **terms occur independently** of one another.

1. Basic Idea of BIM

For any document *D* and query *Q*, BIM estimates:

P(R=1|D,Q)

A document is ranked higher if it has a higher probability of being relevant. Instead of calculating full probabilities, the model computes a **retrieval score** based on query terms.

2. Binary Representation of Terms

BIM uses **binary term incidence**, meaning:

- Term present → 1
- Term absent → 0

Thus, term frequency is not needed.

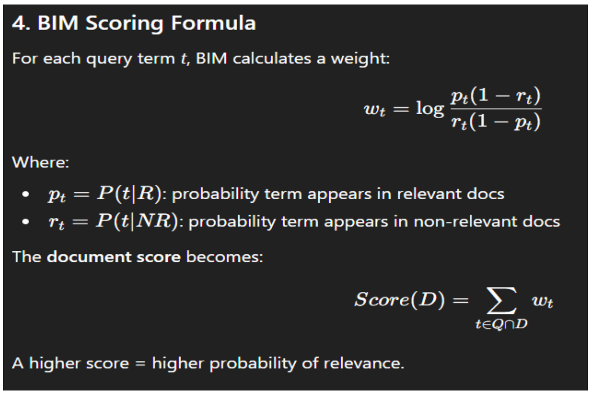
Example:
If a term appears 1 time or 10 times → treated the same (presence only).

3. Independence Assumption

Each query term is assumed to contribute independently to the relevance of a document.
Thus: P(D|R)=∏t∈DP(t|R)

This assumption greatly simplifies probability calculations.

4. BIM Scoring Formula



4. BIM Scoring Formula

For each query term *t*, BIM calculates a weight:

$$w_t = \log \frac{p_t(1 - r_t)}{r_t(1 - p_t)}$$

Where:

- $p_t = P(t|R)$: probability term appears in relevant docs
- $r_t = P(t|NR)$: probability term appears in non-relevant docs

The **document score** becomes:

$$Score(D) = \sum_{t \in Q \cap D} w_t$$

A higher score = higher probability of relevance.

5. Estimating Probabilities

Since we don’t know relevance beforehand, probabilities are estimated using:

- Past relevance judgments
- Relevance feedback
- Statistical smoothing (e.g., +0.5 adjustments)

This is often referred to as **“the Robertson & Sparck Jones (RSJ) weighting scheme.”**

6. Strengths of BIM

- **Mathematically sound** and probabilistic.
- **Simple to implement.**
- Directly supports **relevance feedback**, making weights more accurate.
- Provides the conceptual foundation for the widely used **BM25** ranking function.

7. Limitations of BIM

- Ignores **term frequency** (treats multiple occurrences as single).
- Assumes **independent terms**, which is unrealistic for correlated terms.
- Works best for short queries and simple collections.
- Cannot capture semantic relationships.

8. Importance and Applications

BIM is the theoretical root of modern retrieval models such as:

- **BM25**
- **BM25F**
- **Probabilistic relevance model variants**

Term Frequency (TF)

Term Frequency (TF) refers to the number of times a term appears in a document. It is an important factor in Information Retrieval because the more frequently a term appears in a document, the more likely it is that the document is relevant to that term or query.

Term Frequency is defined as:

TF(t,d)=number of times term t appears in document d

A higher TF value means the document discusses that term more prominently.

2. Importance in Retrieval Models

- TF helps differentiate between documents that merely **mention** a term and those that **focus heavily** on it.
- It improves ranking accuracy because a single occurrence does not carry the same importance as repeated occurrences.

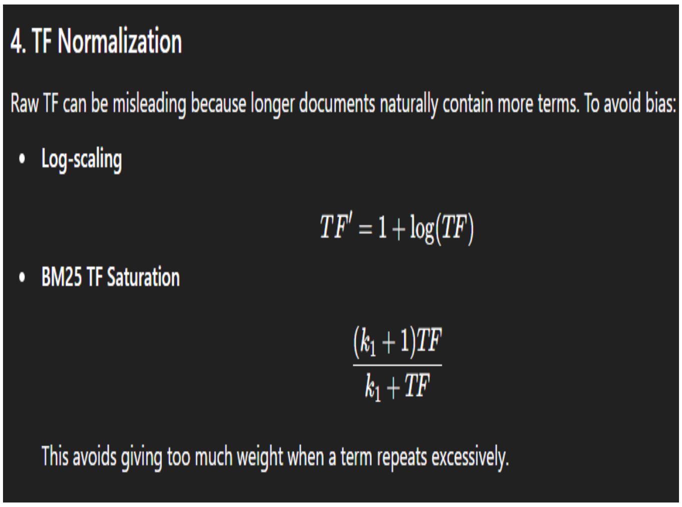
3. TF in Probabilistic Models

Early probabilistic models like the **Binary Independence Model** ignored term frequency by using only term presence/absence.
However, modern models (e.g., **BM25**) include TF because:

- Multiple occurrences strengthen evidence of relevance
- TF helps distinguish highly relevant documents from marginal ones

Thus TF became a core component in advanced probabilistic IR.

4. TF Normalization



4. TF Normalization

Raw TF can be misleading because longer documents naturally contain more terms. To avoid bias:

- Log-scaling

$$TF' = 1 + \log(TF)$$

- BM25 TF Saturation

$$\frac{(k_1+1)TF}{k_1+TF}$$

This avoids giving too much weight when a term repeats excessively.

5. Role in Ranking

Higher TF means:

- Higher document relevance
- Higher ranking score
- Better matching of query intent

TF is combined with IDF (Inverse Document Frequency) to form scoring models like **TF-IDF** and **BM25.**

1. An Appraisal of Probabilistic Models – 6 Marks

Probabilistic models are one of the strongest theoretical approaches in Information Retrieval because they rank documents based on the **probability of relevance**. However, they also have limitations that lead to further improvements.

Strengths

1. **Strong Theoretical Foundation**
   Based on probability theory and the Probability Ranking Principle (PRP), making them mathematically sound.
2. **Effective Ranking**
   Models like **BM25** provide high retrieval performance and are widely used in modern search engines.
3. **Supports Relevance Feedback**
   Probabilities can be refined as the user marks documents as relevant or not, improving accuracy.
4. **Clear Interpretability**
   Each term contributes to relevance in a measurable way (e.g., RSJ weights).

Weaknesses

1. **Term Independence Assumption**
   Probabilistic models assume terms occur independently, which is unrealistic (e.g., “machine learning”, “data mining”).
2. **Binary Term Representation**
   Early models like BIM ignored term frequency, losing important relevance information.
3. **Difficulty Estimating Probabilities**
   Impossible to know true relevance probabilities; models rely on approximations.
4. **Limited Semantic Understanding**
   They cannot capture synonyms, context, or deep linguistic meaning.

2. Tree-Structured Dependencies Between Terms

Early probabilistic models assume **independence between terms**, but in real text, terms often have meaningful relationships. Tree-structured dependency models were introduced to capture these relationships more accurately.

1. Motivation

Terms in a document are rarely independent. Examples:

- “machine” and “learning”
- “artificial” and “intelligence”

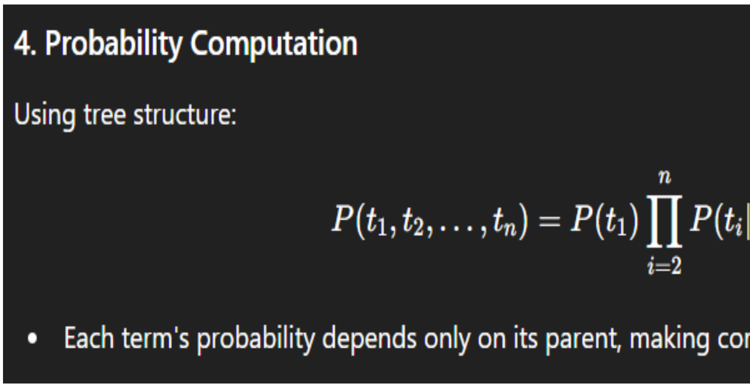
Ignoring such dependencies reduces retrieval accuracy.

2. Tree-Structured Model Concept

- Terms are represented as **nodes** in a tree.
- Edges represent **dependencies** (correlation or co-occurrence strength).
- The structure allows modeling how the presence of one term influences the probability of another.

3. How It Improves IR

- Captures relationships like synonymy, phrase structure, and co-occurrence patterns.
- Improves estimation of P(t|R) and overall relevance scoring.
- Reduces the unrealistic independence assumption of BIM.



4. Probability Computation

Using tree structure:

$$P(t_1, t_2, \ldots, t_n) = P(t_1) \prod_{i=2}^{n} P(t_i|$$

- Each term's probability depends only on its parent, making cor

5. Applications

- Query expansion
- Better relevance modeling
- Basis for more advanced models like **Bayesian networks** and **Markov Random Fields**

Okapi BM25: A Non-Binary Model

Okapi BM25 is one of the most widely used probabilistic ranking functions in Information Retrieval. It evolved from the Binary Independence Model (BIM) but removed its major limitations by incorporating **term frequency**, **document length normalization**, and **saturation effects**. BM25 is considered the best classical IR model and is used in engines like **Lucene, ElasticSearch, Solr, and Terrier**.

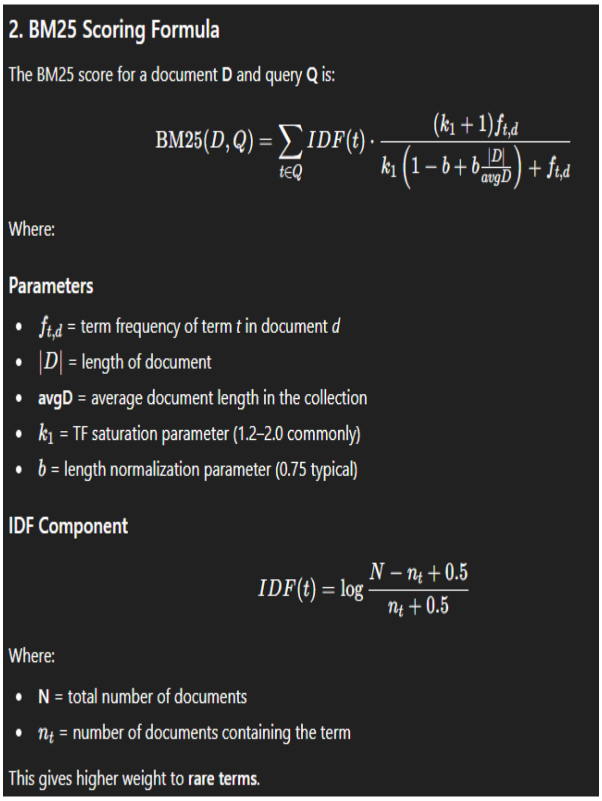
1. Motivation for BM25

Earlier models like BIM treated terms as **binary** (present/absent).
However, real-world retrieval needs:

- Term Frequency (TF) contribution
- Importance of rare terms (IDF)
- Document Length Normalization
- Controlled TF saturation (avoiding overweighting repeated terms)

BM25 was created to address these practical needs.

2. BM25 Scoring Formula



2. BM25 Scoring Formula

The BM25 score for a document **D** and query **Q** is:

$$\text{BM25}(D,Q) = \sum_{t\in Q} IDF(t) \cdot \frac{(k_1+1)f_{t,d}}{k_1\left(1-b+b\frac{|D|}{avgD}\right)+f_{t,d}}$$

Where:

Parameters

- $f_{t,d}$ = term frequency of term t in document d
- $|D|$ = length of document
- **avgD** = average document length in the collection
- k_1 = TF saturation parameter (1.2–2.0 commonly)
- b = length normalization parameter (0.75 typical)

IDF Component

$$IDF(t) = \log \frac{N - n_t + 0.5}{n_t + 0.5}$$

Where:

- **N** = total number of documents
- n_t = number of documents containing the term

This gives higher weight to **rare terms**.

3. Key Features of BM25

(a) Non-binary Term Frequency

Unlike BIM, BM25 uses TF:

- More occurrences of a term → higher relevance
- But saturation prevents TF from increasing score infinitely

The TF function is nonlinear and smooth.

(b) Document Length Normalization

Long documents naturally contain more terms. BM25 uses the **b parameter** to adjust for this:

- **b = 0 → no length normalization**
- **b = 1 → full normalization**

This balances length bias.

(c) TF Saturation

Repeated occurrences of a term are useful only to a point. The **k1k_1k1** parameter ensures:

- First few occurrences matter more
- Extra repetitions contribute less

This avoids over-favoring large or repetitive documents.

(d) IDF Weighting

IDF increases importance of **rare terms** and reduces weight of common terms. Thus, documents with rare but important query terms score higher.

4. Why BM25 Works So Well

- Captures real-world document behavior
- Smooth, balanced scoring mechanism
- Highly interpretable and tunable
- Works for both short keywords and long text queries
- Robust across domains and datasets

It is considered the **gold standard baseline** in IR research.

5. Advantages of BM25

1. Incorporates TF meaningfully
2. Length normalization prevents bias
3. No strict independence assumptions
4. Handles multiple query terms effectively
5. Fast to compute and easy to implement

6. Limitations

1. Assumes bag-of-words (no semantic understanding)
2. Parameters (k1, b) require tuning for best performance
3. Does not use contextual meaning or phrase-level matching.

Bayesian Network Approaches to Information Retrieval

A **Bayesian Network (BN)** is a probabilistic graphical model that represents variables and their conditional dependencies using a **directed acyclic graph (DAG)**. In Information Retrieval (IR), Bayesian networks are used to model relationships between **terms, documents, and relevance** in a more flexible way than traditional probabilistic models like BIM.

1. Motivation

Traditional probabilistic models assume:

- Independence between terms
- Binary term representation
- Limited structure in probability computation

These assumptions are unrealistic in real-world text, where terms are related (e.g., “artificial” + “intelligence”). Bayesian networks overcome these limitations by explicitly modeling dependencies.

2. Structure of a Bayesian Network in IR

The network consists of **nodes** and **directed edges**:

Nodes represent:

1. **Query terms (T_1, T_2, …)**
2. **Documents (D_1, D_2, …)**
3. **Relevance variable (R)**

Edges represent:

- Probabilistic dependencies between terms
- Influence of terms on documents
- Influence of document features on relevance

Example structure:
Terms → Document → Relevance

3. Probabilistic Representation

Each node has a **conditional probability table (CPT)** that quantifies relationships.

Example:

$P(D|T1,T2,...)$ $P(R|D)$

These probabilities combine to compute the final relevance score.

4. Inference for Ranking

Document ranking is based on computing:

$P(R=1|T1,T2,...,Tn)$

Using Bayesian inference:

$P(R|T)=\sum D\ \ P(R|D)\cdot P(D|T)$

Documents with higher probabilities are ranked higher.

5. Advantages of Bayesian Networks in IR

(a) Captures Term Dependencies

Unlike BIM, Bayesian networks model relationships like synonymy, co-occurrence, or semantic closeness.

(b) Handles Uncertainty Naturally

BNs provide a principled way to combine uncertain evidence.

(c) Supports Query Expansion

Dependencies help identify related or missing query terms.

(d) Flexible Model Structure

Terms, documents, fields, and user models can all be nodes.

(e) Integrates Multiple Evidence Sources

Metadata, link structure, user behavior, etc., can be added to the network.

6. Drawbacks

1. **High computational complexity** for large vocabularies.
2. **Requires learning the network structure**, which is difficult.
3. **Parameter estimation** for CPTs can be expensive.
4. Not as fast or scalable as BM25 or vector models.

7. Applications in IR

- Query expansion and refinement
- Modeling term-term correlations
- Filtering noisy results
- Expert and domain-specific retrieval systems
- Adaptive IR based on user feedback

Relevance Feedback in Information Retrieval

Relevance Feedback (RF) is a technique in Information Retrieval where the user’s judgments about which retrieved documents are relevant are used to improve the performance of the retrieval system. It is a feedback-based loop that refines the query representation and ranking function to produce better results in subsequent retrieval cycles.

1. Basic Concept of Relevance Feedback

After the user submits an initial query:

1. System retrieves a set of documents.
2. User marks some documents as **relevant** or **non-relevant**.
3. System uses this information to **recalculate term weights** or **modify the query model**.
4. New ranking is performed, usually yielding better results.

This process is iterative and continues until the user is satisfied.

2. Purpose of Relevance Feedback

- To refine the user’s query
- To learn which terms are truly useful
- To enhance recall and precision
- To reduce the effect of ambiguous or incomplete queries

RF transforms a **short, vague, or imprecise query** into a more effective one.

3. Types of Relevance Feedback

(a) Explicit Relevance Feedback

User directly selects relevant/non-relevant documents. Highly accurate but requires user effort.

(b) Implicit Relevance Feedback

System infers relevance automatically through:

- Click-through data
- Time spent on documents
- User interaction patterns
  Less accurate but easier to gather.

(c) Pseudo Relevance Feedback (Blind RF)

Assumes the top-k retrieved documents are relevant and updates query model accordingly.
Common in modern IR systems.

4. Relevance Feedback in Probabilistic Models

In probabilistic IR models, relevance feedback is used to **update the importance of terms** based on which documents the user marks as relevant or non-relevant. If a term appears often in documents judged relevant, the system increases its weight because the term is a good indicator of relevance. If a term appears mostly in non-relevant documents, its weight is reduced.

This updating process is called **RSJ weighting (Robertson–Sparck Jones weighting)**.
By adjusting term importance based on user feedback, the probabilistic model becomes more accurate and ranks future documents more effectively.

5. Relevance Feedback in Vector Space Models – Rocchio Algorithm

In vector space models, the **Rocchio Algorithm** modifies the original query using information from relevant and non-relevant documents.

- Terms from **relevant** documents are given more weight, pulling the query closer to useful content.
- Terms from **non-relevant** documents are reduced, pushing the query away from unwanted content.
- Part of the original query is kept to preserve user intent.

The result is an improved query that better represents what the user actually wants, giving more accurate and relevant search results.

6. Advantages of Relevance Feedback

1. **Improves Retrieval Quality**
   Usually increases both precision and recall.
2. **Reduces Query Ambiguity**
   Helps when the user’s initial query is vague or short.
3. **Learns User Intent**
   Adapts the system to user’s exact information need.
4. **Enables Query Expansion**
   Adds important terms from relevant documents.

7. Limitations of Relevance Feedback

1. Users may not want to mark documents manually.
2. Incorrect feedback can hurt performance.
3. Computationally expensive for large datasets.
4. Requires accurate UI and user cooperation.

Field Weights: BM25F (9 Marks)

BM25F is an extension of the popular BM25 ranking model, designed to handle **structured documents** that contain multiple fields such as *title*, *body*, *abstract*, *URL*, *headings*, etc.
Since different fields contribute differently to relevance, BM25F introduces **field weights** to give more importance to certain fields over others.

1. Need for Field Weighting

Documents on the web or digital libraries are not uniform.
A term appearing in the **title** is more important than the same term appearing in the **body**.
Examples:

- Title → very strong indicator of relevance
- Headings/subtitles → moderately important
- Body → normal importance
- Metadata/tags → high importance

BM25F solves this by applying **different weights to different fields**.

2. Core Idea of BM25F

Instead of using a single term frequency for the entire document, BM25F:

1. **Looks at each field separately**
2. **Applies field-specific weights**

3. **Combines them into a single score**

This allows the ranking model to respect the structure of the document.

3. Components Used in BM25F

BM25F extends BM25 by incorporating:

(a) Field Term Frequency (FTF)

Term frequency is computed separately for each field.

(b) Field Weights

Each field has a predefined importance value. For example:

- Title weight > Body weight
- Abstract weight > Body weight

(c) Field Length Normalization

Each field may have different lengths. Longer fields get lower weight; shorter fields get higher weight.

(d) Combined Score

The weighted and normalized frequencies from all fields are merged into a single score, similar to BM25's scoring mechanism.

4. Why Field Weights Improve Ranking

- **Title matches** are more meaningful than body matches.
- **Metadata**, **captions**, or **tags** can strongly indicate relevance.
- Helps avoid bias toward long fields (like large document bodies).
- Increases precision by using structural cues of the document.

BM25F is especially useful in web search engines and digital libraries.

5. Advantages of BM25F

1. **More accurate ranking** for structured documents
2. **Flexible system** where each field's importance can be tuned
3. **Better retrieval performance** than plain BM25 when documents have fields
4. Handles fields like title, URL, headings, keywords, etc.
5. Works extremely well for web search where structure matters

6. Limitations

1. Requires **manual tuning** of field weights (title vs body).
2. Slightly more computationally expensive than BM25.
3. Not effective if field weights are poorly chosen.
4. Cannot capture deeper semantic meaning—still bag-of-words.