**a. Why the Vector-Space model (VSM) is generally considered a better retrieval model than the Boolean model (BM)?**

- VSM is better than BM in various aspect:

1. It consider all the word (term) in the documents/query for calculation.
2. It can partially match documents
3. It can return the rank order

**b. It is pointed out in the text-book that the vocabulary terms in the postings are lexicographically ordered. Why is this ordering useful?**

VSM uses computation in high-dimensional spaces, this need to be fixed so that an standard vector space is identified for all the documents and query. Making all the terms in lexicographically order bring order to this fixed high dimensional space.

**c. Which Indexing structure is good for Phrase-Query? e.g. "Information Retrieval",give support to your choice with an example**.

Position Indexing will be an excellent choice for Phrase Query e.g. "Information Retrieval". The posting list will be retrieved for "Information" at a position x and "Retrieval" at the very next position x+1. There will be no false positive in this case.

**d. Do you agree with the statement: "Stemming Increases precision of information retrieval"- give arguments.**

Stemming can increasing both precision and recall. Precision is
precision = (relevant-retrieved) / (total-retrieved) if stemming maps all derive words to a single term that may comes up in the relevant (numerator of the equation), similarly query also look for this derivational form and match.

**e. Illustrate at least three problems while performing "Tokenization" of documents.**

Document processing is very challenging problem, particularly the process of "tokenization" - this is to get a core term for indexing, the entire retrieval model is significantly dependent of this.

1. Identifying the language of the documents, this will ensure how the tokens will be extracted.
2. How the language phrases will be handled? e.g. San Francisco
3. How numbers and other literals will be handled for tokenization? e.g.570A.D

**f. Why sometimes post-processing is required when wild-card queries are processed with k-gram indexing? give an example.**

Consider using the 3-gram index for the query red*. The process first issue the Boolean query $re AND red to the 3-gram index. This leads to a match on terms such as retired, which contain the conjunction of the two 3-grams $re and red, yet do not match the original wildcard query red*. A post-filtering step, in which the terms enumerated by the Boolean query on the 3-gram index are checked individually against the original query red*.

**g. Why Skip Pointers (in posting lists) are not useful for the queries like X OR Y? Where X and Y are terms of the documents for Boolean Retrieval Model.**
The query like X OR Y need to explore all posting list element, hence skip pointers of no use in this case. These pointers only helpful in determining the common document-ID for different posting list. Skip list is good for intersection not for union operation.

**h. It is observed that longer documents have higher term frequency and thus favorable for retrieval in VSM. Suggest a solution to this problem.**
This is true that if we our weighting scheme only assigned score based on term or term frequency, there will be higher scores for large documents. In order to handle this issue a normalized scheme is suggested, you can use document length |d| normalization to punished large documents.

**i. What are the three things that required for evaluating an Information Retrieval Implementation? Just enumerate them.**
1. A document collection
2. A test suite of information needs, expressible as queries
3. A set of relevance judgments, standardly a binary assessment of either relevant or non-relevant for each query-document pair.

**j. Must there always be a break-even point between precision and recall? Either show there must be or give a counter-example**.
If the highest ranked element of the result-set is not relevant, then at this point tp=0 and that is a trivial break-even point where both are equal. Otherwise, If the highest ranked element of the result-set is relevant: the number of false positive increases as you go down the list of result-set and number of false negative decreases. As you go down the list, if the item is R, then fn decreases and fp does not change. If the item is N, then fp increases and fn does not change. At the beginning of the list fp<fn. At the end of the list fp>fn. Thus, there has to be a break-even point. Conclusion: There is only one non-zero breakeven point. There can be additional breakeven points with precision=recall=zero.

**b. If the precision and recall for a given query is equal, what implication we can drive from it?**
It is believed that if precision and recall for an IR experiment are higher value and equal (close to each other) this mean system performed best efforts in retrieving all relevant docum1ents from the collection. Generally, F1 is used to report IR experiments which is harmonic mean to both precision and recall.

**c. Why precision and recall both are necessary for an IR experiment? explain?**
Precision and Recall both together serve very well for IR experiments evaluation. We can build system with 100% precision (system return 1 relevant document from the collection with best effort), and 100% recall (system return all documents to every query), both of these approaches are useless as they cannot satisfy the users. The effective system should have report both the precision and recall and should be high value and very close to each other.

**a. What are some of the drawbacks of Vector-Space Model (VSM)?**
- Order of the term from the document is lost when it is represented in VSM
- All terms are treated statistically independent to each other, human languages have a context and the use of words are generally based on the same context.
- Suffers from synonym and polysemy, documents that contains different vocabulary for the same context are treated as different documents.

**c. Illustrate at least three problems while performing "Tokenization" of documents.**
Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. It is very challenging aspect of information retrieval, the tokenization process need to decide about a lot of different aspect of a natural language like:
1. Direction of parsing for tokenization.
2. Should it treat space as separator for token e.g. Les Vegas a single or two tokens.
3. Treatment of punctuation characters like hyphen(-) co-ordinated.
4. If there is no space in between word boundary, how it will decide about tokens like in Japanese or German compound nouns.

**d. What do we mean by token normalization? How it is done? Give a simple example.**
Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens. Token normalization is the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens. The most standard way to normalize is to implicitly create equivalence classes, which are normally named after one member of the set.
Example: operator, operation, operational all mapped to a root word “operate”

**e. What is Truecasing? What are its benefits in IR?**
Truecasing is the process of restoring case information to badly-cased or noncased text, for index in IR systems. It is the process of restoring case information to raw text. The advantages of Truecasing is rEadaBiLITY and it also enhances the quality of case-carrying data.

**g. Define what do we mean by trailing wildcard query? Give an example. Suggest a data structures that best handle such queries.**
A query such as mon* is known as a trailing wildcard query, because the * symbol occurs only once, at the end of the search string. A search tree on the dictionary is a convenient way of handling trailing wildcard queries.

**h. Write down the entries in the permuterm index dictionary that are generated by the term "tata".**
The entries in permuterm index for term “tata” will be as follow: tata$, ata$t, ta$ta, a$tat, $tata.

**i. A bigram index is used to retrieve document for wildcard query “te*ti*al”. What Boolean query on a bigram index would be generated for this query? Give an example of term that may contain in the result-set.**

The Boolean query using bi-grams will be
$t AND te AND ti AND al AND l$
Example is: testimonial

**j. What is the limitation of isolated term correction approach to spelling correction?**
In isolated-term correction, we attempt to correct each term independent of the actual number of query terms. For example query like "flew form Heathrow" would be try to correct each term "flew" "form" and "Heathrow". This type of correction is applicable to all lexical terms. For example a single query term "carot" can easily be corrected to "carrot". There are two techniques for this method (i) Edit distance and (ii) k-gram overlap.

**a. What is a relevance feedback? Explain the general procedure of relevance feedback in information retrieval.**
The idea of relevance feedback (RF) is to involve the user in the retrieval process so as to improve the final result set. In particular, the user gives feedback on the relevance of documents in an initial set of results. The basic procedure is:
• The user issues a (short, simple) query.
• The system returns an initial set of retrieval results.
• The user marks some returned documents as relevant or non-relevant.
• The system computes a better representation of the information need based on the user feedback.
• The system displays a revised set of retrieval results.

**b. When does relevance feedback works? Give example situations.**
Relevance feedback works only with the following assumptions:
* Users has sufficient knowledge about what they are looking for, which enable them to put an initial query.
* The relevance feedback works in the situation when the relevant documents are similar each other

**c. Why is positive feedback likely to be more useful than negative feedback to an IR system?**
The idea of feedback system to tell the IR system which documents are relevant to the user and to maximize the return of such documents. Only the positive feedback (annotating the relevant documents) may help to optimize such a situation. If we also provide negative feedback(annotating non-relevant documents) this might pose an orthogonal optimization and thus reduce the precision of the system. Hence system only support positive feedback.

**d. What are the advantages of Probabilistic Model of information retrieval over Vector Space Model?**
In the Boolean or vector space models of IR, matching is done in a formally defined but semantically imprecise calculus of index terms. Given only a query, an IR system has an uncertain understanding of the information need. Given the query and document representations, a system has an uncertain guess of whether a document has content relevant

to the information need or not. Probability theory provides a principled foundation for such reasoning under uncertainty. The Probabilistic model of information retrieval estimate a probabilistic weight of each term with respect to a query, hence it model the impreciseness with probability estimates and it exploit this foundation to estimate how likely it is that a document is relevant to an information need.

**e. What are the main assumptions of Probability Ranking Principle (PRP)?**
A1: One Random variable for each term(word).
A2: d(w) are mutually independent given R.
A3: P(0/R=1)=P(0/R=0)=0
A4: If the word is not in the query, it is equally likely to occur in relevant and non-relevant populations(documents). practically: We only need to calculate probabilities of common words in query and documents.
A5: On average, a query word will occur in half the relevant documents. Practical: pw and (1-pw) will cancel out.
A6: Non-relevant set approximated by collection as a whole. (Most documents are non-relevant).

**f. How Language Model of Information Retrieval is different from Probabilistic Model?**
The language model of IR assume that every document is generated by a distinct model hence its terms probabilities are calculated as per this assumption. While probabilistic model use probabilities for the entire corpus. The language model is more impervious towards the query and better rank the documents for a given query. It is the probability that the query is generated keeping in mind the model of the document.

**g. Why a bi-gram language model is practically more challenging? Explain**.
A bi-gram language model used bi-gram phrases as a feature for information retrieval. let's assume that a document and query is parsed as a bi-gram feature we need to calculate the probabilities under a language model. for example P(t1,t2,t3) would be P(t1)*P(t2/t1)*P(t3/t2) finding these probabilities from the collection is quite challenging and computational expensive. Hence these models are not practical yet.

**b. In Rocchio's algorithm, what weight setting for α/β/γ does a "Find pages like this one" search correspond to?**
In order to find "Find pages like this one", we need to set β to a very high value as it correspond to relevancy of the search document. We can set α and γ to a minimal values. one such weight assignment would be α=0; β=1; and γ=0

**a. What is a relevance feedback? Explain the general procedure of relevance feedback in information retrieval.**
The idea of relevance feedback (RF) is to involve the user in the retrieval process so as to improve the final result set. In particular, the user gives feedback on the relevance of documents in an initial set of results. The basic procedure is:
• The user issues a (short, simple) query.

• The system returns an initial set of retrieval results.
• The user marks some returned documents as relevant or non-relevant.
• The system computes a better representation of the information need based on the user feedback.
• The system displays a revised set of retrieval results.

**b. Why relevance feedback mechanism is not popular among users? Explain.**
Relevance feedback is hard to explain to an average user. Relevance feedback is an extra computational phase in IR cycle, which seeks implicit or explicit judgment about the retrieved documents against a given query and thus delay the query response to a user. Relevance feedback generally increases recall while an average user interested in high precision.

**c. Discuss the pros and cons of implicit (indirect) vs. explicit (direct) feedbacks.**
Implicit(Indirect) feedback Explicit (Direct) feedback
- Implicit (indirect) feedback does not bother user for explicit actions. It is fast and can be possible for large IR system.
- It is less reliable and possibly introduce a problem of query drift.

- Explicit (Direct) feedback requires user to marks document relevant. It is slow process and does not scale to large systems.
- It is more reliable and generally save from query drift.

**d. In Rocchio algorithm what will be the condition (values of α, β and γ) for which original query (qo) is more close to centroid of relevant documents than modified query (qm).**
The original query (qo) is more close to centroid of relevant documents than modified query (qm), if the value β is very small and γ is very large and we keep α=1.

**e. What is probability ranking principle? Explain.**
Let Rd,q be an indicator random variable that says whether d is relevant with respect to a given query q. That is, it takes on a value of 1 when the document is relevant and 0 otherwise. The obvious order in which to present documents to the user is to rank documents by their estimated probability of relevance with respect to the information need: $P(R = 1|d, q)$. This is the basis of the Probability Ranking Principle (PRP).

**f. In Binary Independence Model (BIM) what does P(R=1/ x, q) + P(R=0/ x, q) =1 assumption represent?**
The assumption that a document “d” represented in vector space of terms as “x” is either belong to set of relevant document or set of non-relevant document given a fixed query. [A document is either relevant or non-relevant to a query]

**g. In a language model that uses uni-gram and bi-gram probabilities of features how can we calculate the probability of phrase "t1 t2 t3 t4" respectively. Only probabilities for calculating P (t1 t2 t3 t4) are required?**

The simplest form of language model simply throws away all conditioning context, and estimates each term independently. Such a model is called a uni-gram language model: P(t1t2t3t4) = P(t1)P(t2)P(t3)P(t4) bi-gram is a more complex model than uni-gram. This model condition the term based on the previous term hence keep track of order of terms in a limited sense. P (t1t2t3t4) = P(t1)P(t2|t1)P(t3|t2)P(t4|t3)

**h. What is "query expansion"?**

Query expansion is an autonomous process of reformulating a seed query (qo) to improve retrieval performance in information retrieval systems. It is generally perform to bridge the gap between user information need and the posed query. The process usually involves evaluating a user's input (query) to finds synonyms of query terms, morphological forms of the terms, and fixing spelling errors automatically, also re-weighting the terms in the given query to get more relevant documents to a given query. The query expansion add more terms to the query (qo) to expand the result-set.

**i. What does the assumption "a query term is equally likely to be present or absent from a randomly pick relevant document" in BIM signify?**

With this assumption we have the probability that a query term appears in a relevant document is P (t) =0.5, and a query term absent from relevant document is also Q (t) =0.5 where Q (t) = 1-P (t) [only for relevant collection against a query] which is practical for 1-P (t) and P (t) cancel out each other and thus simplify the expression for BIM.

**j. What does the assumption "if a term is not in a query, it is equally likely to occur in relevant and non-relevant collection" in BIM signify?**

With this assumption all non-query terms get equal likely value for both relevant and non-relevant sub-collections hence cancel out each other and only the query terms that appears in documents are used for actual calculation in BIM expression.

| Vector Space Model | Probabilistic Model | Language Model |
|---|---|---|
| **Idea:**<br>The document and query are represented in vector space of terms appears in both. | **Idea:**<br>Address the uncertainty in the query. Try to estimate a query term probability to | **Idea:**<br>It is also a variation of probabilistic model, it build a model for every document |

| | | |
|---|---|---|
| Similarity is define as the distance between document and query vectors. | appear in a relevant document. The documents are presented to user with decreasing probability of terms appears in the relevant documents. | and try to find the probability that a query is generated with the same model of a document. It is related to the idea that user already have some notion of documents in mind while coming up for the query. |
| **Opportunity:**<br>- Based on strong mathematical rigor.<br>- Rank documents based on the similarity of query and documents.<br>- Partial and full matching possible. | **Opportunity:**<br>- Based on general idea of probability theory.<br>- Rank documents based on their probability of relevance to a query.<br>- Partial and full match possible. | **Opportunity**:<br>- Based on the specific idea of probabilistic document model that generate the given query.<br>- Rank documents based on the probability of relevance of a document model and query.<br>- Partial and full match possible. |
| **Limitations:**<br>- No contextual information in the model<br>- Model tuning based on statistical term weighting<br>- High dimensionality | **Limitations:**<br>- Strong assumption of term independence<br>- Term with zero probabilities need some kind of smoothing | **Limitations:**<br>- Strong assumption about probability.<br>- Overcome theoretically between language mismatch of documents and query. |

**What do we mean by Relevance feedback? Where and when a relevance feedback is effective?**

The idea of relevance feedback (RF) is to involve the user in the retrieval process, using feedback on relevant or irrelevant result-set items for a query, use this knowledge to improve the next round of the retrieval against the same query from an IR system. When a user experiencing difficulty in formulating a good query, relevance feedback is quite effective.

**Explain how Rocchio's algorithm is used to fetch pages similar to a user's sample provided page (pages like this options)?**

In order to use Rocchio's algorithm for retrieving "find pages like this one". There is only on relevant feedback item and that is the sample page. We can completely ignore the query (q0), and we do not want to apply negative judgment and hence alpha ($\alpha$)= gamma($\gamma$) =0; This implies beta ($\beta$)= 1 for the modified query.

**Can Negative feedback improve relevant ranking? Give a situation, when a user can only provide negative relevance feedback?**

Sometime negative feedback is the only choice left, think about a difficult query and all top rank documents are non-relevant. Negative feedback can improve relevant ranking but it will be very slow, time consuming and requires many rounds of feedbacks.

**What do we mean by Boolean Retrieval Model? What are some of the draw backs it has?**
Boolean Model of IR is based on Boolean algebra (OR, AND and NOT) operators with features as term present in a document. The presence of exact term in a document qualify for the retrieval based on Boolean logic inferred from the expression.
Drawbacks:
Exact matching may retrieve too few or too many documents
Difficult to rank output, some documents are more important than others
Hard to translate a query into a Boolean expression
All terms are equally weighted

**There are 16 relevant documents in a collection for a given query "q". The precision of the query is 0.40 and recall of the query is 0.25, Find, how many documents are in the results set (number of documents response to the query "q" from the IR system)?**
we know,
precision = (relevant-retrieved) / (total-retrieved)
=> 0.4 = (relevant-retrieved) / (result-set) ---------- eq(i)
similarly,
recall = (relevant-retrieved)/ (total-relevant)
=> 0.25 = (relevant-retrieved)/ 16
=> relevant-retrieved= 0.25 * 16 = 4
hence
eq(i) => result-set = 4 /0. 4 = 10
Result-set has 10 documents.

**Consider the following set of documents:**
**D1: If you love life, life will love you back.**
**D2: If you love life, do not waste time.**
**D3: I love life and find fun all time.**
**Stop-word-list = {If, you, will, do, not, I, and, all}**
**a. Prepare a term-document matrix for above corpus. After removing terms from stop-word-list provided.**
D1 D2 D3
back 1 0 0
find 0 0 1
fun 0 0 1
life 1 1 1
love 1 1 1
time 0 1 1
waste 0 1 0

**b. Prepare a posting list for the above document- in sorted order of terms/documents, by using the stop word list provided**

Lexicon Doc-ID
back 1
find 3
fun 3
life 1;2;3
love 1;2;3
time 2;3
waste 2

**1. Assume a bi-word index is used in an IR system. Give an example of a document which will be returned for a query of "National University FAST" but is actually a false positive which should not be returned. Suggest a solution without a false positive solution.**
**Consider the following two text document:**
**D1: The leading computer science institution National University formally called University FAST, was very instrumental in fostering computer culture in Pakistan.**
**D2: National University FAST is a leading computer science school in Pakistan.**

It is clearly evident that D2 contains the query text and is the true positive of this query. If a system uses bi-word index it will hit the D1 as well which is a false positive the answer to this problem is using a positional indexing to support this type of long query.

**2. When can a wildcard query useful? Illustrate by giving 3 situational examples from search.**
**Wildcard queries are used in any of the following situations:**
(1) the user is uncertain of the spelling of a query term (e.g., Sydney vs. Sidney,
(2) the user is aware of multiple variants of spelling a term and (consciously) seeks documents containing any of the variants (e.g., color vs. colour);
(3) the user seeks documents containing variants of a term that would be caught by stemming, but is unsure whether the search engine performs stemming (e.g., judicial vs. judiciary, leading to the wildcard query judicia*);
(4) the user is uncertain of the correct rendition of a foreign word or phrase (e.g., the query Universit* Stuttgart).

**What is Boolean Retrieval Model? What are some of the advantages of this model?**
Boolean Model of IR is based on Boolean algebra (OR, AND and NOT) operators with features as term present in a document. The presence of exact term in a document qualify for the retrieval based on Boolean logic inferred from the expression.
Advantages:
Very simplified model with mathematical formalism easy to implement and test

**Given the following postings list sizes:**
**Term Postings size**

***eyes 213312***
***skies 271658***
***mountains 7712***
***trees 316812***
***Love 87123***

**What will be the best Boolean Model query processing order? (mountains OR trees) AND (love OR skies) AND eyes**

mountains or trees =total lookups 7712+316812 = 324524
love or skies= 87123+271658= 358781 eyes= 213312
We will do conservative estimate of the length of the union of postings lists, the recommended order: eyes AND (mountains OR trees) AND (love OR skies)

**Give at least three differences between Multinomial and Bernoulli model for text/document classification.**

| Multinomial model | Bernoulli model |
|---|---|
| - Its event model is generation of token (terms).<br>- Random variable X=t iff t occurs at given position.<br>- Can handle more features<br>- Work best for large documents. | - It event model is generation of document.<br>- Random variable Ut=1 iff t occurs in document.<br>- Best works for few features<br>- Work best for short documents. |

**Differentiate between Contiguity hypothesis and Cluster hypothesis.**

| Contiguity hypothesis | Cluster hypothesis |
|---|---|
| Contiguity hypothesis states that Documents in the same class form a contiguous region and regions of different classes do not overlap. It is very useful in vector space classification. | Cluster hypothesis states that Documents in the same cluster behave similarly with respect to relevance to information needs. It is very useful in clustering and its application |

.

**Illustrate at least three drawbacks of K-Mean Clustering.**

- you need to provide the number of clusters for this algorithm, which is need to be anticipated from the dataset.it is a hard requirement.
- it is very much sensitive to initial seeding.
- partitioned produced is independent of each other.
- it works on the notion of distance (Euclidean distance) to separate the objects.

**Illustrate the differences between Partitional clustering and Hierarchical clustering.**

1. Hierarchical and Partitional Clustering have key differences in running time, assumptions, input parameters and resultant clusters.
2. Partitional clustering is faster than hierarchical clustering.
3. Hierarchical clustering requires only a similarity measure, while partitional clustering requires stronger assumptions such as number of clusters and the initial centers.
4. Hierarchical clustering does not require any input parameters, while partitional clustering algorithms require the number of clusters to start running.
5. Hierarchical clustering returns a much more meaningful and subjective division of clusters but partitional clustering results in exactly k clusters.
6. Hierarchical clustering algorithms are more suitable for categorical data as long as a similarity measure can be defined accordingly.

**if we know that there are two types of documents in the given collection, where should we cut the dendrogram to get the results.**

if we know for sure that there are two types of documents in the given collection, we need to cut the resultant dendrogram from a point where we can get the two hierarchical groups, in the given case it will be cluster1(d1,d2) and cluster2(d3,d4,d5,d6). Hence, the two groups will have distinct type of documents, as HAC is an unsupervised approach there may be some noise documents that is mix type in each clusters.

**Differentiate between the k-Mean and Hierarchical agglomerative clustering.**

| k-Mean | Hierarchical agglomerative clustering |
|---|---|
| - partitioned the dataset into meaning full similar collection, this partitioning is independent with each other | - it produces a hierarchical arrangement of given collection. |
| - uses distance measure for performing clustering | - it works on a object to object similarity matrix |
| - running time is proportional to input size. | - there is no need for (number of clusters), as it produce all possible number of clusters in a given arrangement. |
| - need number of clusters to be produced as input parameter | - running time is O(n2) or higher |