1 - Boolean Model

- incidence vector
- semantics of the query model (AND/OR/NOT, and other operators, e.g., /k, /S)
- inverted index, positional inverted index
- query processing methods for basic and advanced boolean queries (including phrase query, queries with /S operator, etc.)
- query optimization methods (list merge order, skip pointers)
- Not required: next-word index

1-1: Example for short answer questions

- Why we need document frequency?
- List merge order (heuristic for list merge)
- How to put skip pointers and why?
- /k /S query

1-2. Important Content

- x AND y, x OR y, x AND NOT y, x OR NOT y
- Gallop search and its time complexity

2 - Preprocessing

- typical preprocessing steps: tokenization, stopword removal, stemming/lemmatization

3 - Index Construction

- Why we need dedicated algorithms to build the index?
- BSBI: Blocked sort-based indexing
- SPIMI: Single-pass in-memory indexing
- Dynamic indexing: Immediate merge, no merge, logarithmic merge

3-1: Example for short answer questions

- Why we need dedicated algorithms to build the index: with the limitation of hardware conditions, index construction cannot be implemented in-memory due to large corpus size.
- How to do in-memory index construction:
  Sorting-based algorithm or hash-based in-memory based algorithm
- What the problems with the immediate merge or no merge?
  Immediate merge: merge a lot, inefficient for O/S, No merge: slow query performance.

3-2. Important Content

- BSBI and Dynamic indexing

4 - Vector Space Model

- What is/why ranked retrieval?
- raw and normalized tf, idf
- cosine similarity
- tf-idf variants (using SMART notation): e.g., lnc.ltc
- basic query processing method: document-at-a-time vs term-at-a-time
- exact & approximate query optimization methods (heap-based top-k algorithm, MaxScore algorithm, etc.)

4-1: Example for short answer questions

- What is / why ranked retrieval?

Good for expert users not good for the majority of users. Boolean model can only tell us if a document matches the query or not, results too few or too many.

In ranked retrieval models, the system returns an ordering over the (top) documents in the collection with respect to a query.

- What's the problem of jaccard coefficient // raw term frequency?

- Meaning and calculation of tf, idf?

- Does idf have effect on ranking one term query?

- Why distance is a bad idea // or why cosine similarity?

4-2. Important Content

- DAAT, TAAT, MAXSCORE

5 - Evaluation

- Existing method to prepare for the benchmark dataset, queries, and ground truth
- For unranked results: Precision, recall, F-measure
- For ranked results: precision-recall graph, 11-point interpolated precision, MAP, etc.
- Not required: NDCG, Kappa (ϰ) measure for inter-judge (dis)agreement

---

6 - Probabilistic Model and Language Model

- Probability ranking principle (intuitively, how to rank documents and when to stop)
- derivation of the ranking formula of the probabilistic model
- the BM25 method
- Query-likelihood unigram language model with Jelinek-Mercer smoothing.

- How to rank document? With cost and without cost.
- (probabilistic model) Binary independence model. (And BM25)
- What assumptions we made?
  - Independence Assumption: terms occur in documents independently.
  - for all terms not occurring in the query, pi = ri.
  - boolean representation of documents/queries/relevance
  - document relevance value are independent.
- Language model (n-gram) (slides 48 - 51, 54)

7 - Learning to Rank

- Motivation
- Setup, jargons, and basic ideas of Machine Learning
- List-wise L2R
- Not required: The details of the SVM L2R model and other advanced variations.

- Why weren’t early attempts very successful / influential? (slide 12)
- Why wasn’t ML much needed before and why is ML needed now? (slides 13 - 14)
- Learning to Rank (The ranking SVM) (slides 23 - 24)

8 - Link Analysis

- The pagerank algorithm
- Not required: Personalized PR

- The concept of PageRank, how it is calculated in practice? Why is it relevant for web search?
- How do we compute the pagerank scores? (slides 23-24)
- What problems may PageRank meet with?
    - Spider trap: definition and solution(random teleports)
    - Dead end: definition
- How to compute steady-state?

9 - Language Models

- Definition, usage, and evaluation (perplexity)
- n-gram LM: Parameter learning, including various smoothing
- Not required: Neural LM

- What is language model? (Slide 3)
- What is perplexity?
- Add-one smoothing (laplace smoothing)
- Definition of Backoff and interpolation. (slide 59)
- Based on interpolation, how to set the lambdas? (slide 61)
- Understand "stupid backoff" (slide 64) and Kneser-Ney smoothing (slides 68-74)

10 - Vector Semantics

- Motivation, taxonomy, and concepts
- Sparse vectors: PPMI weighting and its variants
- High-level understanding of word2vec skip-gram model
- Not required: Maths details of Word2vec

- Four kinds of vector models. (slide 7 of part I)
- Why we need PPMI or the problem with raw count? (slide 19 of part I)
- What are the the definitions of PMI and PPMI? (slides 20-21 of part I)
- Add-one smoothing and weighting PMI.
- Why dense vectors? (slide 4 of part II)
- Describe the skip-gram algorithms (slide 11 of part II)