• What is the relation between the Boolean retrieval model and the fuzzy retrieval model? Discuss the major similarities and differences.

Homework

• What are possible problems of using the Jaccard index for measuring term similarity? Give an example of two rather dissimilar terms that would typically yield a high Jaccard index.

Homework

• What is the basic idea underlying Ogawa’s approach to deriving fuzzy term weights? (Do not use any formulas in your answer!)

Homework

• Given a document collection that is stored on disk using an inverted index (with a term weight assigned to each term-document pair). What is the computational complexity of calculating the cosine similarity between two documents?

Homework

• What is the purpose of normalizing a document’s vector representation for document length?
What is the basic idea underlying the TF-IDF weighting scheme? Why should we care about how often a term occurs in the collection? Give an illustrating example of your own.

What is the difference between prior and posterior probability? How are both related to Bayes' Theorem?

Probabilistic IR models use $Pr(\text{document } d \text{ is useful for the user asking query } q)$ as underlying measure of similarity between queries and documents.

Advantages:
- Probability theory is the right tool to reason under uncertainty in a formal way
- Methods from probability theory can be re-used

Characterizing usefulness is really tricky, we will discuss this later…
Instead of usefulness we will consider relevance
Given a document representation $d$ and a query $q$, one can objectively determine whether $d$ is relevant with respect to $q$ or not
This means in particular:
- Relevance is a binary concept
- Two documents having the same representation are either both relevant or both irrelevant

Probabilistic information retrieval rests upon the Probabilistic Ranking Principle (Robertson, 1977)

"If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of usefulness for the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data."

Usefulness and Relevance
- Characterizing usefulness is really tricky, we will discuss this later…
- Instead of usefulness we will consider relevance
- Given a document representation $d$ and a query $q$, one can objectively determine whether $d$ is relevant with respect to $q$ or not
- This means in particular:
  - Relevance is a binary concept
  - Two documents having the same representation are either both relevant or both irrelevant
Denote the set of all document representations contained in our current collection by $C$.

For any query $q$, denote the set of relevant documents contained in our collection by $R_q$, i.e.

$$R_q = \{ d \in C \mid d \text{ is relevant with respect to } q \}$$

Our task then becomes:

- Input: The user’s query $q$ and a document $d$
- Output: $\Pr(d \in R_q)$

Precisely what does this probability mean?

- As we have defined it, it is either $d \in R_q$ or $d \notin R_q$

- Is $\Pr(d \in R_q)$ a sensible concept?

What does probability in general mean?

- Maybe we should deal with that first…

There are different interpretations of “probability,” we will look at the two most common ones:

- Frequentists vs. Bayesians

  - Frequentists:
    - Probability = expected frequency on the long run
    - Neyman, Pearson, Wald, …

  - Bayesians:
    - Probability = degree of belief
    - Bayes, Laplace, de Finetti, …

An event can be assigned a probability only if:
1. it is based on a repeatable(!) random experiment, and
2. within this experiment, the event occurs at a persistent rate on the long run, its relative frequency

An event’s probability is the limit of its relative frequency in a large number of trials

Examples:
- Events in dice rolling
- The probability that it will rain tomorrow in the whether forecast (if based on historically collected data)

Probability is the degree of belief in a proposition

The belief can be:
- subjective, i.e. personal, or
- objective, i.e. justified by rational thought

Unknown quantities are treated probabilistically

Knowledge can always be updated

Named after Thomas Bayes

Examples:
- The probability that there is life on other planets
- The probability that you pass this course’s exam

There is a book lying on my desk

I know it is about one of the following two topics:
- Information retrieval
- Animal health

What’s $\Pr(\text{"the book is about IR"})$?

That question is stupid! There is no randomness here!

That’s a valid question! I only know that the book is either about IR or AH! So let’s assume the probability is 0.5!
Frequentist vs. Bayesian (2) Detour

- Let’s assume that the book is lying on my desk due to a random draw from my bookshelf…
- Let X be the “topic result” of a random draw
- What’s Pr(“X is about IR”)

Frequentist

That question is valid!
This probability is equal to the proportion of IR books in your shelf.

Bayesian

This guy is strange…

Frequentist vs. Bayesian (3)

- Back to the book lying on my desk
- What’s Pr(“the book is about IR”?)

Even if I assume that you got this book by drawing randomly from your shelf, the question stays stupid.
I have no idea what this book is about.
But I can tell you what properties a random book has.

Frequentist

That question is valid!
This probability is equal to the proportion of IR books in your shelf.

Bayesian

This guy is strange…

Frequentist vs. Bayesian (4) Detour

- A more practical example: Rolling a dice
- Let x be the (yet hidden) number on the dice that lies on the table
  Note: x is a number, not a random variable!
- What’s Pr(x = 5)?

Silly question again.
As I told you, there is no randomness involved!

Frequentist

Since I do not know what x is, this probability expresses my degree of belief.
I know the dice’s properties, therefore the probability is 1/6.

Bayesian

This guy is strange…

Frequentist vs. Bayesian (5) Detour

- What changes if I show you the value of x?

Nothing changes. Uncertainty and probability have nothing in common.

Frequentist

Since I do not know what x is, this probability expresses my degree of belief.
I know the dice’s properties, therefore the probability is 1/6.

Bayesian

Now the uncertainty is gone.
The probability (degree of belief) is either 1 or 0, depending whether the dice shows a 5.

Probability of Relevance, Again

- How to interpret Pr(d ∈ R_q)?
  - Clearly: Bayesian (expressing uncertainty regarding R_q)
- Although there is a crisp set R_q (by assumption), we do not know what R_q looks like
- Bayesian approach: Express uncertainty in terms of probability
- Probabilistic models of information retrieval:
  - Start with Pr(d ∈ R_q) and relate it to other probabilities, which might be more easily accessible
  - On this way, make some reasonable assumptions
  - Finally, estimate Pr(d ∈ R_q) using other probabilities’ estimates

Lecture 3: Probabilistic Retrieval Models

1. The Probabilistic Ranking Principle
2. Probabilistic Indexing
3. Binary Independence Retrieval Model
4. Properties of Document Collections
Probabilistic Indexing

- Presented by Maron and Kuhns in 1960
- Goal: Improve automatic search on manually indexed document collections

Basic notions:
- A index terms
  - Documents = vectors over \([0, 1]\), i.e. terms are weighted
  - Queries = vectors over \([0, 1]\), i.e. binary queries
  - \(R_q\) = relevant documents with respect to query \(q\) (as above)

Task:
Given a query \(q\), estimate \(\Pr(d \in R_q)\), for each document \(d\)

Probabilistic Indexing (2)

- Let \(Q\) be a random variable ranging over the set of all possible queries
- \(Q\)'s distribution corresponds to the sequence of all queries asked in the past
  - Example (\(k = 2\)):
    Ten queries have been asked to the system previously:
    \[
    \begin{array}{cccc}
    (0, 0) & (1, 0) & (0, 1) & (1, 1) \\
    0 & 2 & 7 & 1 \\
    \end{array}
    \]
    Then, \(Q\)'s distribution is given by:
    \[
    \begin{array}{cccc}
    Pr(Q = (0, 0)) & Pr(Q = (1, 0)) & Pr(Q = (0, 1)) & Pr(Q = (1, 1)) \\
    0 & 0.2 & 0.7 & 0.1 \\
    \end{array}
    \]

Probabilistic Indexing (3)

- If \(Q\) is a random query, then \(R_Q\) is a random set of documents
- We can use \(R_Q\) to express our initial probability \(\Pr(d \in R_Q)\):
  \[
  \Pr(d \in R_Q) = \Pr(d \in R_Q \mid Q = q)
  \]

This means:
If we restrict our view to events where \(Q\) is equal to \(q\), then \(\Pr(d \in R_Q)\) is equal to \(\Pr(d \in R_Q \mid Q = q)\)

Probabilistic Indexing (4)

- Now, let's apply Bayes' Theorem:
  \[
  \Pr(d \in R_Q \mid Q = q) = \frac{\Pr(d \in R_Q) \cdot \Pr(Q = q \mid d \in R_Q)}{\Pr(Q = q)}
  \]

Combined:
  \[
  \Pr(d \in R_Q) = \frac{\Pr(d \in R_Q) \cdot \Pr(Q = q \mid d \in R_Q)}{\Pr(Q = q)}
  \]

Probabilistic Indexing (5)

- \(\Pr(Q = q)\) is the same for all documents \(d\)
- Therefore, the document ranking induced by \(\Pr(d \in R_Q)\) is identical to the ranking induced by \(\Pr(d \in R_Q) \cdot \Pr(Q = q \mid d \in R_Q)\)
- Since we are only interested in the ranking, we can replace \(\Pr(Q = q)\) by a constant:
  \[
  \Pr(d \in R_Q) = c(q) \cdot \Pr(d \in R_Q) \cdot \Pr(Q = q \mid d \in R_Q)
  \]

Probabilistic Indexing (6)

- \(\Pr(d \in R_Q)\) can be estimated from user feedback
  - Give the users a mechanism to rate whether the document they read previously has been relevant with respect to their query
  - \(Pr(d \in R_Q)\) is the relative frequency of positive relevance ratings
- Finally, we must estimate \(\Pr(Q = q \mid d \in R_Q)\)
Probabilistic Indexing (7)

- How to estimate $\Pr(Q = q | d \in R_Q)$?
- Assume independence of query terms:
  $$\Pr(Q = q | d \in R_Q) = \prod_{i \in \{1, \ldots, k\}} \Pr(Q_i = q_i | d \in R_Q)$$
- Is this assumption reasonable?
  - Obviously not (co-occurrence, think of synonyms)!

Probabilistic Indexing (8)

- What's next? Split up the product by $q_i$'s value!

$$\Pr(Q = q | d \in R_Q) = \prod_{i \in \{1, \ldots, k\}} \Pr(Q_i = q_i | d \in R_Q) \cdot \prod_{i \in \{1, \ldots, k\}} \Pr(Q_i = 1 | d \in R_Q)$$

- Look at complementary events:
  $$\Pr(Q = q | d \in R_Q) = \prod_{i \in \{1, \ldots, k\}} \left(1 - \Pr(Q_i = 1 | d \in R_Q)\right) \cdot \prod_{i \in \{1, \ldots, k\}} \Pr(Q_i = 1 | d \in R_Q)$$

Probabilistic Indexing (9)

$$\Pr(Q = q | d \in R_Q) = \prod_{i \in \{1, \ldots, k\}} \left(1 - \Pr(Q_i = 1 | d \in R_Q)\right) \cdot \prod_{i \in \{1, \ldots, k\}} \Pr(Q_i = 1 | d \in R_Q)$$

- Only $\Pr(Q_i = 1 | d \in R_Q)$ remains unknown
- It corresponds to the following:
  - Given that document $d$ is relevant for some query, what is the probability that the query contained term $i$?

Probabilistic Indexing (10)

- Given that document $d$ is relevant for some query, what is the probability that the query contained term $i$?
- Maron and Kuhns argue that $\Pr(Q_i = 1 | d \in R_Q)$ can be estimated by the weight of term $i$ assigned to $d$ by the human indexer
- Is this assumption reasonable? Yes!
  1. The indexer knows that the current document to be indexed definitely is relevant with respect to some topics
  2. She/he then tries to find out what these topics are
     - Topics correspond to index terms
     - Term weights represent degrees of belief

Reality Check

- $\Pr(d \in R_Q)$ models the "general relevance" of $d$
  - $\Pr(d \in R_Q)$ is proportional to $\Pr(d \in R_Q)$
  - This is reasonable
  - Think of the following example:
    - You want to buy a book at a book store
    - Book A's description perfectly fits what you are looking for
    - Book B's description perfectly fits what you are looking for
    - Book A is a bestseller
    - Nobody else is interested in book B
    - Which book is better?
Lecture 3: Probabilistic Retrieval Models

1. The Probabilistic Ranking Principle
2. Probabilistic Indexing
3. Binary Independence Retrieval Model
4. Properties of Document Collections

• Presented by van Rijsbergen in 1977
• Basic notions:
  – \( k \) index terms
  – Documents = vectors over \((0, 1)^k\), i.e. set of words model
  – Queries = vectors over \((0, 1)^k\), i.e. set of words model
  – \( R_q \) = relevant documents with respect to query \( q \)
• Task:
  Given a query \( q \), estimate \( \Pr(d \in R_q) \), for any document \( d \)

Binary Independence Retrieval (2)

• Let \( D \) be a uniformly distributed random variable ranging over the set of all documents in the collection
• We can use \( D \) to express our initial probability \( \Pr(d \in R_q) \):

\[
\Pr(d \in R_q) = \Pr(D \in R_q | D = d)
\]

• This means:
  If we restrict our view to events where \( D \) is equal to \( d \), then \( \Pr(d \in R_q) \) is equal to \( \Pr(D \in R_q) \).
• Note the similarity to probabilistic indexing:

\[
\Pr(d \in R_q) = \Pr(d \in R_q | Q = q)
\]

Binary Independence Retrieval (3)

\[
\Pr(d \in R_q) = \Pr(D \in R_q | D = d)
\]

• Again, let’s apply Bayes’ Theorem:

\[
\Pr(D \in R_q | D = d) = \frac{\Pr(D \in R_q)}{\Pr(D = d)} \cdot \Pr(D = d | D \in R_q)
\]

• Combined:

\[
\Pr(d \in R_q) = \frac{\Pr(D \in R_q)}{\Pr(D = d)} \cdot \Pr(D = d | D \in R_q)
\]

Binary Independence Retrieval (4)

\[
\Pr(d \in R_q) = \frac{\Pr(D \in R_q)}{\Pr(D = d)} \cdot \Pr(D = d | D \in R_q)
\]

• \( \Pr(D \in R_q) \) is identical for all documents \( d \)
• Since we are only interested in the probability ranking, we can replace \( \Pr(D \in R_q) \) by a constant:

\[
\Pr(d \in R_q) = c(q) \cdot \frac{1}{\Pr(D = d)} \cdot \Pr(D = d | D \in R_q)
\]

Binary Independence Retrieval (5)

\[
\Pr(d \in R_q) = c(q) \cdot \frac{1}{\Pr(D = d)} \cdot \Pr(D = d | D \in R_q)
\]

• \( \Pr(D = d) \) represents the proportion of documents in the collection having the same representation as \( d \)
• Although we know this probability, it basically is an artifact of our approach to transforming \( \Pr(d \in R_q) \) into something Bayes’ Theorem can be applied on
• Unconditionally reducing highly popular documents in rank simply makes no sense
How to get rid of Pr($D = d$)?

Instead of Pr($d \in R_q$) we look at its odds:

$$\text{Odds}(d \in R_q) = \frac{\Pr(d \in R_q)}{1 - \Pr(d \in R_q)} = \frac{\Pr(d \in R_q)}{\Pr(d \not\in R_q)}$$

As we will see on the next slide, ordering documents by this odds results in the same ranking as ordering by probability.

Applying Bayes’ Theorem on Pr($d \not\in R_q$) yields:

Again, $c(q)$ is a constant that is independent of $d$.

Putting it all together we arrive at:

It looks like we need an assumption.

Assumption of linked dependence:

(slightly weaker than assuming independent terms)

Is this assumption reasonable?

– No, think of synonyms…

Let’s split it up by term occurrences within $d$:

Replace Pr($D_i = 0 | \ldots$) by $1 - \Pr(D_i = 1 | \ldots$):
Let's split it up by term occurrences within $q$:

$$\text{Odds}(d \in R_q) = c(q) \prod_{t \in (l_{1...k} \setminus L_q)} \frac{1 - Pr(D_i = 1 | D \in R_q)}{Pr(D_i = 1 | D \in R_q)} \prod_{i \in L_q} \frac{Pr(D_i = 1 | D \in R_q)}{Pr(D_i = 1 | D \in R_q)}$$

- Let's split it up by term occurrences within $q$:

$$\text{Odds}(d \in R_q) = c(q) \prod_{t \in (l_{1...k} \setminus L_q)} \frac{1 - Pr(D_i = 1 | D \in R_q)}{Pr(D_i = 1 | D \in R_q)} \prod_{i \in L_q} \frac{Pr(D_i = 1 | D \in R_q)}{Pr(D_i = 1 | D \in R_q)}$$

Looks like we heavily need an assumption…

- Assume that $Pr(D_i = 1 | D \in R_q) = Pr(D_i = 1 | D \notin R_q)$, for any $i$ such that $q_i = 0$

- **Idea:** Relevant and non-relevant documents have identical term distributions for non-query terms

- **Consequence:** Two of the four product blocks cancel out

This leads us to:

- Multiply by 1 and regroup:

$$1 = \prod_{t \in (l_{1...k} \setminus L_q)} \frac{1 - Pr(D_i = 1 | D \in R_q)}{Pr(D_i = 1 | D \in R_q)} \prod_{i \in L_q} \frac{Pr(D_i = 1 | D \in R_q)}{Pr(D_i = 1 | D \in R_q)}$$

- Fortunately, the first product block is independent of $d$, so we can replace it by a constant:

$$\text{Odds}(d \in R_q) = c(q) \prod_{t \in (l_{1...k} \setminus L_q)} \frac{1 - Pr(D_i = 1 | D \in R_q)}{Pr(D_i = 1 | D \in R_q)} \prod_{i \in L_q} \frac{Pr(D_i = 1 | D \in R_q)}{Pr(D_i = 1 | D \in R_q)}$$

- How to estimate $Pr(D_i = 1)$?

- Since usually most documents in the collection will not be relevant to $q$, we can assume the following:

$$Pr(D_i = 1 | D \notin R_q) \approx Pr(D_i = 1)$$

- Reasonable assumption?
This leads us to the final estimate:

\[
\frac{1 - \Pr(D_i = 1 | D \in R_q)}{\Pr(D_i = 1 | D \notin R_q)} \approx \frac{1 - \frac{df(t_i)}{N}}{\frac{df(t_i)}{N}} = \frac{N - df(t_i)}{df(t_i)} \approx \frac{N}{df(t_i)}
\]

Pr(\(D_i = 1 | D \in R_q\)) cannot be estimated that easy…

There are several options:

– Estimate it from user feedback on initial result lists
– Estimate it by a constant (Croft and Harper, 1979), e.g. 0.9
– Estimate it by \(\frac{df(t_i)}{N}\) (Greiff, 1998)

Are there any other probabilistic models?

– Extension of the Binary Independence Retrieval model
– Learning from user feedback
– Different types of queries
– Accounting for dependencies between terms
– Poisson model
– Belief networks
– Many more…

Pros

– Very successful in experiments
– Probability of relevance as intuitive measure
– Well-developed mathematical foundations
– All assumptions can be made explicit

Cons

– Estimation of parameters usually is difficult
– Doubtful assumptions
– Much less flexible than the vector space model
– Quite complicated

Some data for two test collections:

<table>
<thead>
<tr>
<th>Newswire</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>1 GB</td>
</tr>
<tr>
<td>Documents</td>
<td>400,000</td>
</tr>
<tr>
<td>Posting entries</td>
<td>180,000,000</td>
</tr>
<tr>
<td>Vocabulary size (after stemming)</td>
<td>400,000</td>
</tr>
<tr>
<td>Index size (uncompressed, without word positions)</td>
<td>450 MB</td>
</tr>
<tr>
<td>Index size (uncompressed, with word positions)</td>
<td>800 MB</td>
</tr>
<tr>
<td>Index size (compressed, with word positions)</td>
<td>130 MB</td>
</tr>
</tbody>
</table>

Source: (Zobel and Moffat, 2006)
The vocabulary size is 30
4,000,000,000,000

• Term frequencies in Moby Dick:
  • Zipf's Law (3)

  - 65 Information Retrieval and Web Search Engines — Wolf-Tilo Balke and Joachim Selke — Technische Universität Braunschweig
  - Source: http://searchengineland.com/the-long-tail-of-search-12198

  • Example:

  - Looking at a collection of web pages, you find that there are 3,000 different terms in the first 10,000 tokens and 30,000 different terms in the first 1,000,000 tokens
  - Assume a search engine indexes a total of 20,000,000,000 pages, containing 200 tokens on average
  - What is the size of the vocabulary of the indexed collection as predicted by Heaps' law?
  - The vocabulary size is $30 \cdot 4,000,000,000,000^{0.5} = 60,000,000$
  - \[k = 30, \quad b = 0.5\]

  • Empirically verified for many different collections

  - Heaps' Law: \#terms = $k \cdot (#tokens)^b$

  - $k$ and $b$ are positive constants, collection-dependent
  - Typical values: $30 \leq k \leq 100, \quad b = 0.5$
  - Empirically verified for many different collections

  • How big is the term vocabulary?
    • Clearly, there must be an upper bound
      - The number of all "reasonable" words
      - ... Engines — Wolf-Tilo Balke and Joachim Selke — Technische Universität Braunschweig
        · Heaps' law: \#terms = $k \cdot (#tokens)^b$

  • Key insights:
    - Few frequent terms
    - Many rare terms
  - Zipf's law is an example of a power law:
    - $Pr(x) = a \cdot x^b$
      - $a$ is a normalization constant (total probability mass must be 1)
      - In Zipf's law: $b \approx -1$

  • Term frequencies in Moby Dick:

  • Zipf's own explanation:
    - Principle of least effort:
      - Do the job while minimizing total effort
    - Cognitive effort of reading and writing should be small
      - Pressure towards unification of vocabulary such that choosing and understanding words is easy (small vocabulary)
    - Diversity of language has to be high
      - Pressure towards diversification of vocabulary such that complex concepts can be expressed and distinguished
    - The "economy of language" leads to the balance observed and formalized by Zipf's law

  • Zipf analyzed samples of natural language
    - Letter frequencies
    - Term frequencies

  • Letter frequencies in English language:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>0.120</td>
</tr>
<tr>
<td>T</td>
<td>0.085</td>
</tr>
<tr>
<td>A</td>
<td>0.077</td>
</tr>
<tr>
<td>I</td>
<td>0.067</td>
</tr>
<tr>
<td>N</td>
<td>0.067</td>
</tr>
<tr>
<td>O</td>
<td>0.067</td>
</tr>
<tr>
<td>S</td>
<td>0.059</td>
</tr>
<tr>
<td>R</td>
<td>0.055</td>
</tr>
<tr>
<td>H</td>
<td>0.043</td>
</tr>
<tr>
<td>D</td>
<td>0.042</td>
</tr>
<tr>
<td>L</td>
<td>0.037</td>
</tr>
<tr>
<td>U</td>
<td>0.033</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Letter</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>0.024</td>
</tr>
<tr>
<td>M</td>
<td>0.024</td>
</tr>
<tr>
<td>W</td>
<td>0.023</td>
</tr>
<tr>
<td>Y</td>
<td>0.023</td>
</tr>
<tr>
<td>P</td>
<td>0.023</td>
</tr>
<tr>
<td>B</td>
<td>0.030</td>
</tr>
<tr>
<td>G</td>
<td>0.017</td>
</tr>
<tr>
<td>V</td>
<td>0.017</td>
</tr>
<tr>
<td>K</td>
<td>0.012</td>
</tr>
<tr>
<td>Q</td>
<td>0.007</td>
</tr>
<tr>
<td>J</td>
<td>0.005</td>
</tr>
<tr>
<td>X</td>
<td>0.004</td>
</tr>
<tr>
<td>Z</td>
<td>0.002</td>
</tr>
</tbody>
</table>

  • The same is true for many other languages...

  - Source: http://searchengineland.com/the-long-tail-of-search-12198
  - Information Retrieval and Web Search Engines — Wolf-Tilo Balke and Joachim Selke — Technische Universität Braunschweig
Similar relationships hold in many different contexts:
- Distribution of letter frequencies
- Distribution of accesses per Web page
- Distribution of links per Web page
- Distribution of wealth
- Distribution of population for US cities
- ...

Zipf's law: The $i$-th most frequent term has frequency proportional to $1/i$