Fuzzy Index Terms

- **Observation:**
  - Not all index terms representing a document are equally important, or equally characteristic
    - Are there any synonyms to the document’s terms?
    - Does a term occur more than once in the document?
  - Can we assign weights to terms in documents?
- **Idea:**
  - Improve Boolean retrieval!
  - Describe documents by fuzzy sets of terms!
- **Advantage:** Fuzzy (i.e. ordered) results sets

Fuzzy Retrieval: Open Problems

- **Fuzzy sets:**
  - \{step, China, mountaineer\}
  - \{(step/0.4, China/0.9, mountaineer/0.8)\}
- **Open Problems:**
  - How to deal with fuzzy logic?
  - Where to get membership degrees from?

Fuzzy Logic

- Developed by Lotfi Zadeh in 1965
- Possible truth values are not just “false” (0) and “true” (1) but any number between 0 and 1
- Designed to deal with classes whose boundaries are not well defined
Zadeh Operators

- **How to translate** Boolean operators into fuzzy logic?
  - Propositional logic should be a special case
  - Fuzzy operators should have "nice" properties: commutativity, associativity, monotony, continuity, ...

- **Zadeh's original operators:**
  - Let $\mu(A)$ denote the truth value of the variable $A$
  - Conjunction: $\mu(A \land B) = \min\{\mu(A), \mu(B)\}$
  - Disjunction: $\mu(A \lor B) = \max\{\mu(A), \mu(B)\}$
  - Negation: $\mu(\neg A) = 1 - \mu(A)$

Example

- Document = \{step/0.4, China/0.9, mountaineer/0.8\}
- Query = "(step BUT NOT China) OR mountaineer"

- Document's degree of query satisfaction is 0.8

Intuitive?

- Zadeh operators indeed have "nice" properties
- But sometimes, they behave strange:

  - Document$_1$ = \{step/0.4, China/0.4\}
  - Document$_2$ = \{step/0.3, China/1\}
  - Query = "term$_1$ AND term$_2$"
  - Result = \{ Document$_1$/0.4, Document$_2$/0.3 \}

Fuzzy Index Terms

- **Second problem:** Where to get fuzzy membership degrees for index terms from?
- **Obvious solution:**
  - A lot of work ...

- **Better solution:**
  - Take crisp bag of words representation of documents, and convert it to a fuzzy set representation

Approach by Ogawa et al. (1991):

- **Idea:**
  - Extend each document's crisp sets of terms
  - Each document gets assigned:
    - Its crisp terms (use fuzzy degree 1)
    - Additional terms being similar to these crisp terms (use degree $\leq 1$)

1. Use the Jaccard index to get a notion of term similarity
2. Compute fuzzy membership degree for each term–document pair using this similarity
Fuzzy Index Terms (3)

- Jaccard index:
  - Measures which terms co-occur in the document collection
  - The Jaccard index $c(t, u)$ of the term pair $(t, u)$ is
    \[ \frac{\# \text{documents containing both } t \text{ and } u}{\# \text{documents containing at least one of } t \text{ and } u} \]
  - Also known as term-term correlation coefficient, although it is not a correlation in the usual sense
    - A usual correlation coefficient would be high, if most documents do not contain any of the two terms

- Computation of fuzzy membership weights usually is difficult
  - Main problem: All weights must be within $[0, 1]$
  - Lack of intuitive query processing
    - But: There are many other ways to define fuzzy conjunction and disjunction
      (using t-norms and t-conorms)

- Pros:
  - Supports non-binary assignment of index terms to documents
    - It is possible to find relevant documents that do not satisfy the query in a strict Boolean sense
  - Ranked result sets

Fuzzy Index Terms (4)

- Jaccard index:
  - Document $= \{ \text{step, man, mankind} \}$
  - Documents $= \{ \text{step, man, China} \}$
  - Documents $= \{ \text{step, mankind} \}$

- Example

- Given that usually occur together with the other document terms; those terms will capture the document’s topic best

Fuzzy Retrieval Model

- Cons:
  - Computation of fuzzy membership weights usually is difficult
    - Main problem: All weights must be within $[0, 1]$
  - Lack of intuitive query processing
    - But: There are many other ways to define fuzzy conjunction and disjunction
      (using t-norms and t-conorms)

- Pros:
  - Supports non-binary assignment of index terms to documents
    - It is possible to find relevant documents that do not satisfy the query in a strict Boolean sense
  - Ranked result sets

The Philosophy of Fuzzy Logic

- What’s the meaning of “x is contained in the set A with fuzzy degree of 0.25”?
  - Probability?
  - Missing knowledge?
  - Only a quarter of x?
  - Something else?
  - Complete nonsense?
Fuzzy sets describe “possibilities”
- “Joachim is 29 years old”
- “Joachim is young”
- What’s the degree of compatibility of “29” with “young”?

Focus on imprecision and vagueness, not on missing knowledge
Natural to human language

Possibility is different from probability!
Zadeh’s own example:
- “Hans ate X eggs for breakfast”

Another example:
Assume that I have some poison, which looks like water; a glass of it is just enough to kill you
- Probability theory:
The probability that the glass is full of poison is 20%,
the probability that it is full of water is 80%
- Possibility theory:
The glass contains 20% poison and 80% water

Propositional formulas are mathematically handy, but often hard to use for querying
“step AND ((China AND taikonaut) OR man)”

Alternative: Bag-of-words queries
- Queries are represented as a bag of words (“virtual documents”)
- Luhn’s idea: Let the user sketch the document she/he is looking for!
- Advantage: Comparing queries to documents gets simpler!

Many successful retrieval models are based on bag-of-words queries!

Coordination level matching (CLM) is a straightforward approach to bag-of-words queries
- Idea: Documents whose index records have n different terms in common with the query are more relevant than documents with n−1 different terms held in common
- The coordination level (also called “size of overlap”) between a query Q and a document D is the number of terms they have in common
- How to answer a query?
  1. Sort the document collection by coordination level
  2. Return the head of this sorted list to the user (say, the best 20 documents)
Example

- Document₁ = {step, man, mankind}
- Document₂ = {step, man, China}
- Document₃ = {step, mankind}

- Query₁ = {man, mankind}
  Result:
  1. Document₁ (2)
  2. Document₂ (1)

- Query₂ = {China, man, mankind}
  Result:
  1. Document₁, Document₂ (2)
  2. Document₃ (1)

Information Spaces

- Spatial structure of libraries:
  Topically related books are standing side by side
- Can we transfer this principle to information retrieval?
- Idea:
  Represent documents and queries as points in an abstract semantic space
  - Measure similarity by proximity

Vector Space Model

- The vector space model was proposed by Gerard Salton (Salton, 1975)
- Documents and queries are represented as points in n-dimensional real vector space \( \mathbb{R}^n \), where \( n \) is the size of the index vocabulary
  - Usually, \( n \) is very large: 500,000 terms (at least)
- Each index term spans its own dimension
- Obvious first choice:
  Represent documents by its incidence vectors

Lecture 2: More Retrieval Models

1. Fuzzy retrieval model
2. Coordination level matching
3. Vector space retrieval model
4. Recap of probability theory

Distance and Similarity

- How to define similarity/proximity?
  - A metric on a set \( X \) is a function \( d: X \times X \rightarrow \mathbb{R} \) having the following properties:
    - \( d(x, y) \geq 0 \), for any \( x, y \in X \) (non-negativity)
    - \( d(x, y) = 0 \) iff \( x = y \), for any \( x, y \in X \) (identity)
    - \( d(x, y) = d(y, x) \), for any \( x, y \in X \) (symmetry)
    - \( d(x, z) \leq d(x, y) + d(y, z) \), for any \( x, y, z \in X \) (triangle inequality)
  - Example: Euclidean distance
    \[
    d(x₁, \ldots, xₙ; y₁, \ldots, yₙ) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}
    \]
Euclidean Distance

- **Geometric meaning** of Euclidean distance:

  All documents on the line have a Euclidean distance of 1 from Document,

  \[
  d(x_1, \ldots, x_n, y_1, \ldots, y_n) = \sqrt{\sum (x_i - y_i)^2}
  \]

  \[
  \text{(aka Euclidean norm, } \|
  \|)
  \]

  - The vector pointing from the origin to \( x \)
  - The vector pointing from the origin to \( y \)

China

All documents on the circle have a Euclidean distance of 1 from Document,

Similarity

- A similarity measure on a set \( X \) is a function \( s: X \times X \to [0, 1] \) where
  - \( s(x, y) = 1 \) means that \( x \) and \( y \) are maximally similar
  - \( s(x, y) = 0 \) means that \( x \) and \( y \) are maximally dissimilar

  There is no general agreement on what additional properties a similarity measure should possess

  - **Example: Cosine similarity** in vector spaces
    \[
    s(x, y) = \cos(\alpha)
    \]
    - \( \alpha \) is the angle between these two vectors:
      - The vector pointing from the origin to \( x \)
      - The vector pointing from the origin to \( y \)

Cosine Similarity

- **Geometric meaning** of cosine similarity:

  All documents on the line have a cosine similarity of \( \cos(45°) \approx 0.71 \) to Document,

  \[
  s(x_1, \ldots, x_n, y_1, \ldots, y_n) = \cos(\alpha)
  \]

  - \( \cdot \) denotes the dot product (aka scalar product), i.e.
  - \( \| \cdot \| \) denotes the Euclidean norm (aka \( l^2 \)-norm), i.e.
  \[
  x \cdot y = \sum_{i=1}^{n} x_i \cdot y_i
  \]

Recap: Coordination Level Matching

- Let’s assume term vectors only contain binary term occurrences

  - Then, the scalar product of the query vector \( x \) and a document vector \( y \) is the coordination level of \( x \) and \( y \)

  \[
  x \cdot y = \sum_{i=1}^{n} x_i \cdot y_i
  \]

The “Right” Measure

- **Be careful!**
  - The choice of distance or similarity measure always depends on the current application!

  Different measures often behave similar, but not always …
  - Low Euclidean distance implies high cosine similarity, the converse is not true
Normalization

- Cosine similarity does not depend on the length of document and query vectors.
- But using other measures, this might make a difference.

Using e.g., Euclidean distance, are shorter documents more similar to the query than longer ones?

Normalization (2)

- There are many ways to normalize the vector representation of documents and queries.
- Most popular:
  - Divide each coordinate by the vector's length, i.e. normalize to length 1:
  \[ \frac{x}{\|x\|} \]
  - Divide each coordinate by the vector's largest coordinate:
  \[ \frac{x}{\max(x)} \]
  - Divide each coordinate by the sum the vector's coordinates:
  \[ \frac{x}{\sum x} \]

Normalization (3)

- Normalization to unit vectors, i.e. vectors of length/norm 1, is a special case:

- All documents and queries are located on the unit sphere.
- The rank ordering produced for a query is the same for Euclidean distance and cosine similarity.

Normalization (4)

- Often, longer documents cover a topic more in-depth.
- Therefore, accounting for document length might be reasonable.
  - There are several strategies how this can be done.
  - Straightforward:
    1. Compute query result on normalized documents and query.
    2. Give long documents a small boost proportional to their length (maybe you should apply a dampening factor to account for extremely large documents).
  - More advanced:
    - Measure the effect of document length on relevance within your current document collection.
    - Adjust the ranking according to these insights.

Vector Representation

- Are there any more advanced ways of representing documents in vector space than just copying their bag of words representation?
- Of course!
- Luhn's observation (1961):
  Repetition of words is an indication of emphasis.
  - We are already exploiting this by using the bag of words model.
  - The number of occurrences of a term in a document or query is called its "term frequency".
  - Notation:
    \[ tf(d, t) \] is the term frequency of term \( t \) in document \( d \).

Vector Representation (2)

- Discrimination:
  - Not every term in a collection is equally important.
  - For example, the term "psychology" might be highly discriminating in a computer science corpus.
  - In a psychology corpus, it doesn't carry much information.
  - Denote the discriminative power of a term \( t \) by \( \text{disc}(t) \).
  - There are many ways to formalize discriminative power:
    - Higher term frequency \( \Rightarrow \) Higher term weight.
    - Higher discriminative power \( \Rightarrow \) Higher term weight.
  - Term weight should be proportional to \( tf(d, t) \cdot \text{disc}(t) \).
TF-IDF

Karen Spärck Jones observed that, from a discrimination point of view, what we’d really like to know is a term’s specificity (Spärck Jones, 1972):

- In how many documents a given term is contained?
- The term specificity is negatively correlated with this number!
- The more specific a term is, the larger its discriminative power is

TF-IDF (3)

Spärck Jones: The relationship between specificity and inverse document frequency is logarithmic!

This leads to today’s most common form of TF-IDF, as proposed by Robertson and Spärck Jones (1976):

$$tf(d, t) \cdot \log \left( \frac{N + 0.5}{df(t) + 0.5} \right)$$

- $N$ is the number documents in the collection
- “$+ 0.5$” accounts for very frequent and very rare terms
- “$N / df(t)$” normalizes with respect to the collection size

TF-IDF (2)

The number of documents containing a given term $t$ is called $t$’s document frequency, denoted by $df(t)$

Karen Spärck Jones proposed the TF-IDF term weighting scheme:

- Define the weight of term $t$ in document $d$ as:

$$bf(d, t) = \frac{1}{df(t)}$$

- “IDF” ≡ “inverse document frequency”

Term Discrimination

A different approach to defining $disc(t)$ is motivated by looking at the document collection’s structure

- Let $s$ be some similarity measure between documents
- Let $C$ be a collection and let $N$ be its size
- Define $s_{avg}$ to be the average similarity across all documents:

$$s_{avg} = \frac{1}{N^2} \sum_{d \neq d'} s(d, d')$$

- Define $s_{avg, t}$ to be the average similarity across all documents, after removing the vectors’ dimension corresponding to term $t$
- Then, a measure for term $t$’s discriminative power is

$$s_{avg} - s_{avg, t}$$

Term Discrimination (2)

Underlying idea:

- Removing a highly discriminative term will lead to large changes in average document similarity
- Removing a non-discriminative term will not change the average document similarity significantly

Computation of average similarity is expensive but can be speeded up by heuristics

- For example, use average similarity to the average document instead of average similarity over all document pairs (linear runtime, instead of quadratic)

Retrieval Effectiveness

Salton et al. (1983) analyzed the retrieval effectiveness of Boolean retrieval, fuzzy retrieval, and vector space retrieval

<table>
<thead>
<tr>
<th>Collection</th>
<th>MEDLARS</th>
<th>ISI</th>
<th>INSPEC</th>
<th>CACM</th>
</tr>
</thead>
<tbody>
<tr>
<td>#documents</td>
<td>1033</td>
<td>1660</td>
<td>12684</td>
<td>3204</td>
</tr>
<tr>
<td>#queries</td>
<td>30</td>
<td>35</td>
<td>77</td>
<td>52</td>
</tr>
<tr>
<td>Boolean</td>
<td>0.21</td>
<td>0.11</td>
<td>0.12</td>
<td>0.18</td>
</tr>
<tr>
<td>Fuzzy</td>
<td>0.24</td>
<td>0.10</td>
<td>0.13</td>
<td>0.16</td>
</tr>
<tr>
<td>Vector space</td>
<td>0.55</td>
<td>0.16</td>
<td>0.23</td>
<td>0.30</td>
</tr>
</tbody>
</table>

- The table shows average precision using fixed recall, this will be explained in detail in one of the next lectures
- Rule of thumb: The larger the number, the more relevant documents have been retrieved
**Vector Space Model: Pros**

- **Pros:**
  - Simple and clear
  - Intuitive querying yields high usability
  - Founded on “real” document rankings, not based on result sets
  - Highly customizable and adaptable to specific collections:
    - Distance/similarity functions
    - Normalization schemes
    - Methods for term weighting
  - High retrieval quality
  - Relevance feedback possible (will be covered soon…)

**Vector Space Model: Cons**

- **Cons:**
  - High-dimensional vector spaces, specialized algorithms are required (next lecture…)
  - Relies on implicit assumptions, which do not hold in general:
    - Cluster hypothesis: “Closely associated documents tend to be relevant with respect to the same queries”
    - Independence/orthogonality assumption: “Whether a term occurs in a document, is independent of other terms occurring in the same document”

---

**Manual vs. Automatic Indexing**

- Libraries and classical IR:
  - Manually define a list of suitable index terms
  - Manually assign a list of index terms to each document
  - Rationale: “Effectiveness is more important than efficiency.”

- Modern IR and Web search:
  - Automatically assign index terms to documents
  - Every word in the document is an index term!
  - Rationale: “Efficiency is more important than effectiveness.”

**Manual vs. Automatic Indexing (2)**

- The situation around 1960:

**Manual vs. Automatic Indexing (3)**

- Research question:
  - How can we speed up and simplify the manual indexing process, without sacrificing quality?

**Manual vs. Automatic Indexing (4)**

- The Cranfield II research project (1963–1966):
  - Investigate 29 novel indexing languages
  - Most of them artificial and highly controlled
  - But also: Simple and “natural” ones
  - Find methods to evaluate IR systems

- Surprising result:
  - Automatic indexing is (at least) as good as careful manual indexing
This conclusion is so controversial and so unexpected that it is bound to throw considerable doubt on the methods which have been used. [...] A complete recheck has failed to reveal any discrepancies. [...] There is no other course except to attempt to explain the results which seem to offend against every canon on which we were trained as librarians.

SMART:
System for the Mechanical Analysis and Retrieval of Text
- Information retrieval system developed at Cornell University in the 1960s
- Research group led by Gerard Salton (born Gerhard Anton Sahlmann)
- "Gerry Salton was information retrieval" (From: In memoriam: Gerald Salton, March 8, 1927–August 28, 1995)
- SMART has been the first implementation of the vector space model and relevance feedback

SMART (2)
- Early hardware: IBM 7094
- "A basic machine operating cycle of 2 microseconds"

SMART (3)
- System was under development until the mid-1990s (up to version 11)
- The latest user interface:
  - `# indexes the document collection`
  - `$ smart index.doc.spec.file < doc_loc`
  - `# shows statistics on dictionaries, inverted files, etc`
  - `$ smprint -s spec.data rel_header file.above`
  - `# index the query collection`
  - `$ smart index.query.spec.file < query`
  - `# automatic retrieval run`
  - `$ smart retrieve.spec.atc`

SMART (4)
- Early versions of SMART have been evaluated on many test collections:
  - ADE: Publications from information science reviews
  - CACM: Computer science
  - Cranfield collection: Publications from aeronautic reviews
  - CISI: Library science
  - Medlars collection: Publications from medical reviews
  - Time magazine collection:
    - Archives of the generalist review Time in 1963

Lecture 2: More Retrieval Models
1. Fuzzy retrieval model
2. Coordination level matching
3. Vector space retrieval model
4. Recap of probability theory
**Probability Theory**

- Soon, we will discuss probabilistic retrieval models
- To prepare for this, we will have a quick look at some fundamental concepts needed:
  - Probability
  - Statistical independence
  - Conditional probability
  - Bayes’ theorem

**Probability**

- Probability is the likelihood or chance that something is the case or will happen
- Usually, used to describe the results of well-defined random experiments
- Example:
  - Let’s play the following game:
    - Roll a 6-sided dice
    - Then, roll it again
    - If you roll at least 9 in total or if your second roll is 1, you win
    - Otherwise, you lose

**Probability (2)**

- Would you play this game, if it costs you 10€ and you can win 20€?
- What can happen?
  - $6 \times 6 = 36$ different events

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

**Probability (3)**

- What’s the probability of rolling at least 9 in total?
  - Answer: $10/36 \approx 0.28$
- What’s the probability of getting 1 in the second roll?
  - Answer: $1/6 \approx 0.17$
- What’s the probability of winning?
  - Answer: $16/36 = 0.44$

**Statistical Independence**

- Two events are independent, intuitively means that the occurrence of one event makes it neither more nor less probable that the other occurs
- Standard definition:
  - Events A and B are independent, if and only if $\Pr(A \text{ and } B) = \Pr(A) \cdot \Pr(B)$
- Questions:
  - Are “3 in the first roll” and “4 in the second roll” independent?
    - Answer: Yes
  - Are “10 in total” and “5 in the second roll” independent?
    - Answer: No
  - Are “12 in total” and “5 in the first roll” independent?
    - Answer: No

**Conditional Probability**

- Conditional probability is the probability of some event A, given the occurrence of some other event B
  
  \[
  \Pr(AB) = \frac{\Pr(A \text{ and } B)}{\Pr(B)}
  \]

- What’s the probability of winning the game, given I got 4 in the first roll?
  - Answer: $3/36 / 1/6 = 1/2$
- What’s the probability of having had 4 in the first roll, given I won the game?
  - Answer: $3/36$ / $3/16 = 19/36 \approx 0.19$
Bayes’ Theorem

• After Thomas Bayes (1702–1761)
• It says:
  \[ \Pr(A|B) = \frac{\Pr(A)}{\Pr(B)} \cdot \Pr(B|A) \]
• What’s the probability of having had 4 in the first roll, given I won the game?
  – \( \Pr(\text{win} | 4 \text{ in first roll}) = \frac{1}{2} \)
  – \( \Pr(\text{win}) = \frac{16}{36} \)
  – \( \Pr(4 \text{ in first roll}) = \frac{1}{6} \)
  Answer: \( \frac{1}{6} / \frac{16}{36} = \frac{1}{2} \approx 0.19 \)

Bayes’ Theorem (2)

\[ \Pr(A|B) = \frac{\Pr(A)}{\Pr(B)} \cdot \Pr(B|A) \]

• \( \Pr(A) \) is called the prior probability of \( A \)
• \( \Pr(A|B) \) is called posterior probability of \( A \)
• Idea underlying these names: \( \Pr(A) \) gets “updated” to \( \Pr(A|B) \) after we observed \( B \)

Next Lecture

• Indexing
• Document normalization
  – Stemming
  – Stopwords
  – ...n
• Statistical properties of document collections