Chapter IR:V

V. Retrieval Models

- Overview of Retrieval Models
- Empirical Models
- Boolean Retrieval
- Vector Space Model
- Probabilistic Models
- Binary Independence Model
- Okapi BM25
- Hidden Variable Models
- Latent Semantic Indexing
- Explicit Semantic Analysis
- Generative Models
- Language Models
- Divergence From Randomness
- Combining Evidence
- Web Search
- Learning to Rank

Empirical Models [Probabilistic Models] [Hidden Variable Models] [Generative Models]

Basic empirical retrieval models abstract over a document $d \in D$ by treating it as a "bag of words" comprising the index terms derived from d.

A document representation $\mathbf{d}$ is composed of weighted index terms of d.

Discriminating factors of empirical models:

1. Term weighting method to compute the weight w_i of an index term t_i.
2. Construction method of the query representation $\mathbf{q}$.
3. Computation method of the relevance function $\rho(\mathbf{q}, \mathbf{d})$.
4. Composition method of the result set R.

Boolean Retrieval

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ [Generic Model] [VSM] [BIM] [BM25] [LSI] [ESA] [LM]

Document representations $\mathbf{D}$.

The set of index terms $T = \{t_1, \ldots, t_m\}$ is composed of nouns as lemmatized word stems.

The representation $\mathbf{d}$ of a document d is a function from T to $\{0, 1\}$, where $\mathbf{d}(t_i) = 1$ is interpreted as "term t_i present in d", and $\mathbf{d}(t_i) = 0$ as "term t_i absent from d".

Query representations $\mathbf{Q}$.

A query representation $\mathbf{q}$ corresponds to a logical formula with alphabet $\Sigma = T$, where the logical operators $\wedge$, $\vee$, $\neg$, and brackets can be used.

Relevance function ρ.

The document representation $\mathbf{d}$ of a document d induces an interpretation $\mathcal{I}_{\mathbf{d}}$ for $\mathbf{q}$, yielding $\rho(\mathbf{q}, \mathbf{d}) = \mathcal{I}_{\mathbf{d}}(\mathbf{q})$.

If $\rho(\mathbf{q}, \mathbf{d}) = 1$, the document d is an element of the result set R.

Boolean Retrieval

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ [Generic Model] [VSM] [BIM] [BM25] [LSI] [ESA] [LM]

Document representations $\mathbf{D}$.

The set of index terms $T = \{t_1, \ldots, t_m\}$ is composed of nouns as lemmatized word stems.

The representation $\mathbf{d}$ of a document d is a function from T to $\{0, 1\}$, where $\mathbf{d}(t_i) = 1$ is interpreted as "term t_i present in d", and $\mathbf{d}(t_i) = 0$ as "term t_i absent from d".

Query representations $\mathbf{Q}$.

A query representation $\mathbf{q}$ corresponds to a logical formula with alphabet $\Sigma = T$, where the logical operators $\wedge$, $\vee$, $\neg$, and brackets can be used.

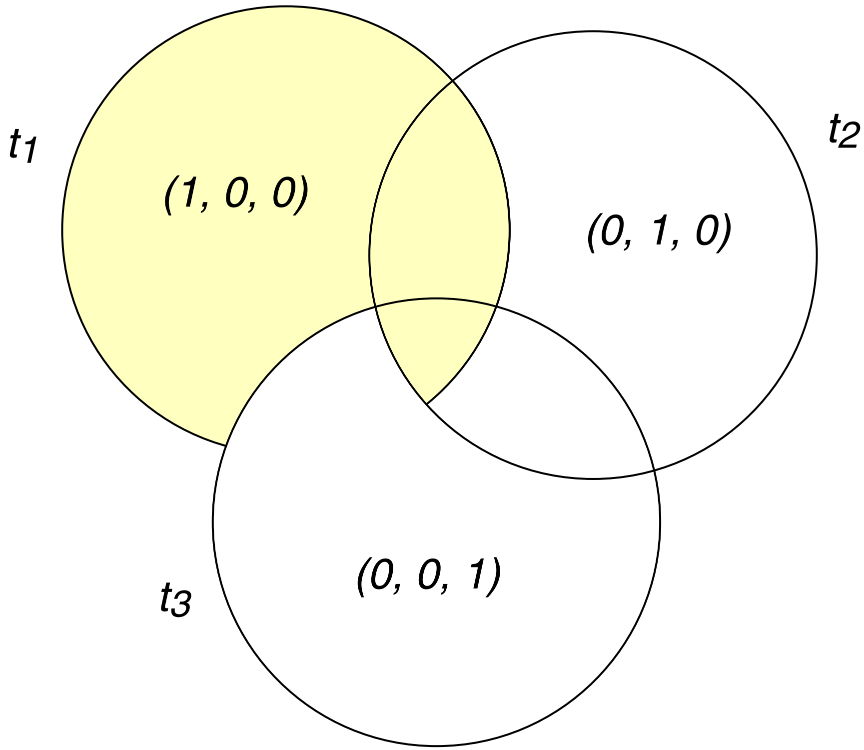
Relevance function ρ.

The document representation $\mathbf{d}$ of a document d induces an interpretation $\mathcal{I}_\mathbf{d}$ for $\mathbf{q}$, yielding $\rho(\mathbf{q}, \mathbf{d}) = \mathcal{I}_\mathbf{d}(\mathbf{q})$.

If $\rho(\mathbf{q}, \mathbf{d}) = 1$, the document d is an element of the result set R.

Boolean Retrieval

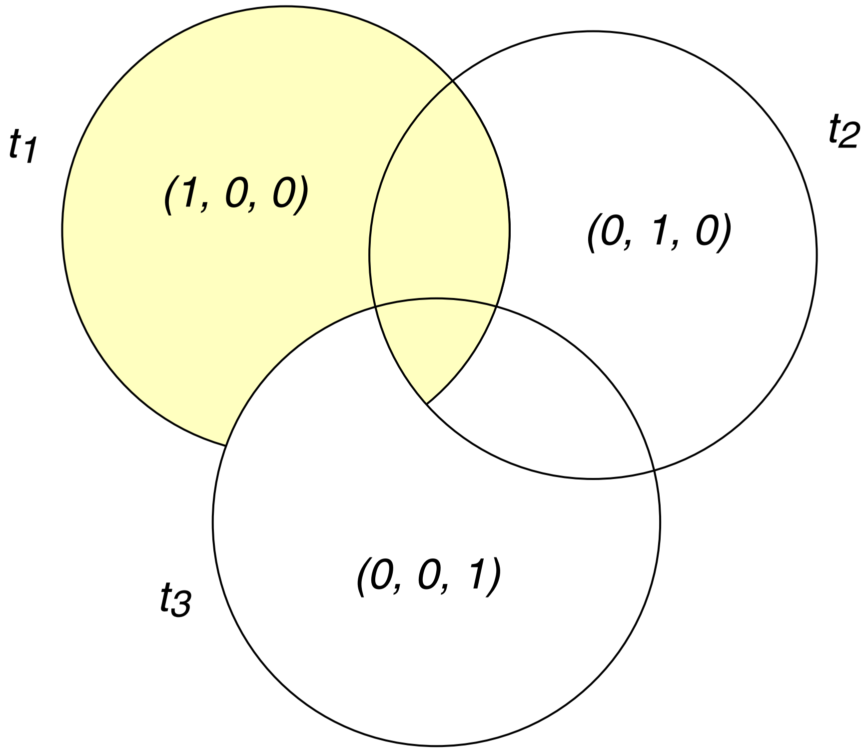
Relevance Function ρ



What is the query illustrated?

Boolean Retrieval

Relevance Function ρ



What is the query illustrated?

$$\mathbf{q} = t_1 \wedge (t_2 \vee \neg t_3) \equiv (t_1 \wedge \neg t_2 \wedge \neg t_3) \vee (t_1 \wedge t_2 \wedge \neg t_3) \vee (t_1 \wedge t_2 \wedge t_3)$$

Boolean Retrieval

Example

Document representation:

$$\mathbf{d} = \{\ (\texttt{chrysler}, 1),\ (\texttt{deal}, 1),\ (\texttt{usa}, 1),\ (\texttt{china}, 0),\ (\texttt{cat}, 0),\ (\texttt{sales}, 1),\ (\texttt{dog}, 0),\ \ldots\ \}$$

Query representation:

$$\begin{aligned} \mathbf{q} &= \texttt{usa} \wedge (\texttt{dog} \vee \neg\texttt{cat}) \\ &\equiv (\texttt{usa} \wedge \texttt{dog}) \vee (\texttt{usa} \wedge \neg\texttt{cat}) \\ &\equiv (\texttt{usa} \wedge \neg\texttt{dog} \wedge \neg\texttt{cat}) \vee \\ &\quad\ (\texttt{usa} \wedge \texttt{dog} \wedge \neg\texttt{cat}) \vee \\ &\quad\ (\texttt{usa} \wedge \texttt{dog} \wedge \texttt{cat}) \end{aligned}$$

Induces interpretation:

$\mathcal{I}_{\mathbf{d}}(\mathbf{q}) = 1$, since $\mathcal{I}_{\mathbf{d}}(\texttt{usa}) = 1, \mathcal{I}_{\mathbf{d}}(\texttt{dog}) = 0$, and $\mathcal{I}_{\mathbf{d}}(\texttt{cat}) = 0$.

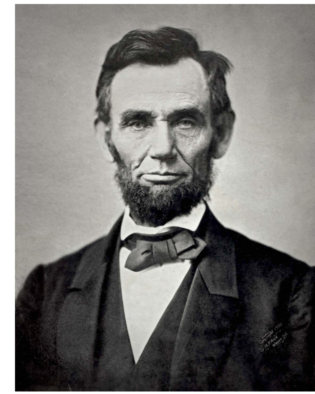
Remarks:

- The symbol "$\equiv$" denotes "is logically equivalent with".
- What does logical equivalence mean?
- A Boolean query in disjunctive normal form can be answered straightforward using an inverted index in parallel for each conjunction.
- A Boolean query in canonical disjunctive normal form will retrieve each document only once.

Boolean Retrieval

Query Refinement: "Searching by Numbers"

Best practice in Boolean retrieval: (re)formulate queries until the number of documents retrieved is manageable. Example: pages about President Lincoln.



1. `lincoln`
   Result: many pages about cars, places, people

2. president $\wedge$ lincoln
   Result: "Ford Motor Company today announced that Darryl Hazel will succeed Brian Kelley as president of Lincoln Mercury."

3. president $\wedge$ lincoln $\wedge$ $\neg$automobile $\wedge$ $\neg$car
   Not in result: "President Lincoln's body departs Washington in a nine-car funeral train."

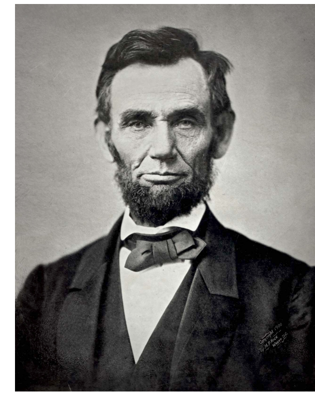
4. president $\wedge$ lincoln $\wedge$ $\neg$automobile $\wedge$ biography $\wedge$ life $\wedge$ birthplace $\wedge$ gettysburg
   Result: $\emptyset$

5. president $\wedge$ lincoln $\wedge$ $\neg$automobile $\wedge$ (biography $\vee$ life $\vee$ birthplace $\vee$ gettysburg)
   Top result might be: "President's Day – Holiday activities – crafts, mazes, word searches, . . . 'The Life of Washington' Read the entire book online! Abraham Lincoln Research Site"

Boolean Retrieval

Query Refinement: “Searching by Numbers”

Best practice in Boolean retrieval: (re)formulate queries until the number of documents retrieved is manageable. Example: pages about President Lincoln.

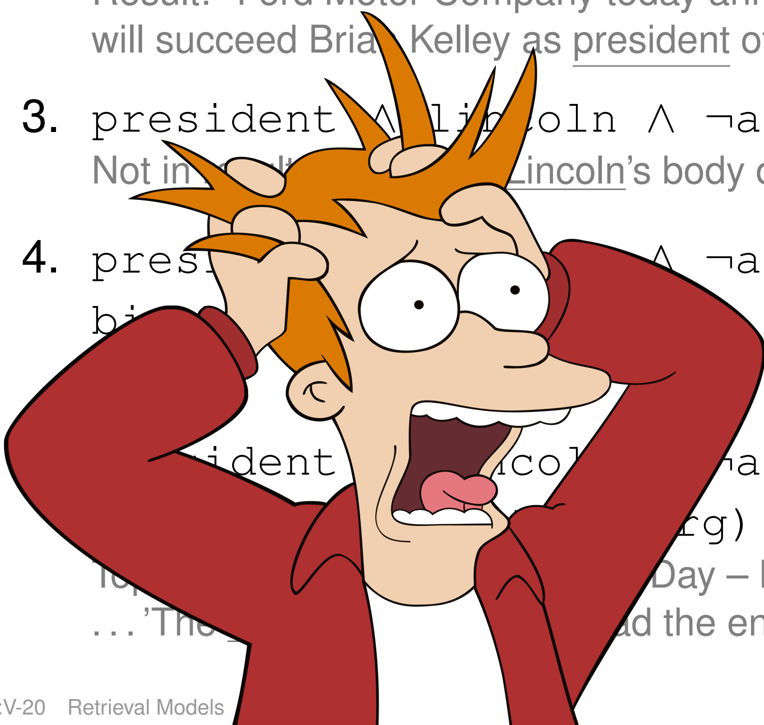


1. `lincoln`
   Result: many pages about cars, places, people

2. `president ∧ lincoln`
   Result: “Ford Motor Company today announced that Darryl Hazel will succeed Bria Kelley as president of Lincoln Mercury.”

3. `president ∧ lincoln ∧ ¬automobile ∧ ¬car`
   Not in Lincoln’s body departs Washington in a nine-car funeral train.”

4. `pres ∧ ¬automobile ∧ biography ∧ life ∧`
   `b`



`ident col ¬automobile ∧ (biography ∨ life ∨`
`rg)`
Day – Holiday activities – crafts, mazes, word searches,
. . . ’The d the entire book online! Abraham Lincoln Research Site”

Boolean Retrieval

Discussion

Advantages:

- Precision: in principle, any subset of documents from a collection can be designated by a Boolean query
- as in data retrieval, other fields are possible (e.g., date, document type, etc.)
- simple, efficient implementation

Disadvantages:

- retrieval effectiveness depends entirely on the user
- cumbersome query formulation (e.g., expertise required)
- no possibility to weight query terms
- no ranking; binary relevance scoring is too restrictive for most practical purposes (exceptions: systematic reviews, patent prior art, legal cases)
- the size of the result set is difficult to be controlled

Vector Space Model

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ [Generic Model] [Boolean Retrieval] [BIM] [BM25] [LSI] [ESA] [LM]

Document representations $\mathbf{D}$.

The set of index terms $T = \{t_1, \ldots, t_m\}$ is typically composed of the word stems of the vocabulary of a document collection, excluding stop words.

The representation $\mathbf{d}$ of a document d is a $|T|$-dimensional vector, where the i-th vector component of $\mathbf{d}$ corresponds to a term weight w_i of term $t_i \in T$, indicating its importance for d. Various term weighting schemes have been proposed.

Query representations $\mathbf{Q}$.

A query representation $\mathbf{q}$ is constructed like a document representation.

Relevance function ρ.

Document representations and query representations are interpreted as points in a vector space spanned by unit vectors for each term in T, assuming their orthogonality.

Distance and similarity functions defined for vector spaces serve as relevance functions ρ. The Euclidean distance and the cosine similarity are important examples.

Vector Space Model

Retrieval Model $\mathcal{R} = \langle \mathbf{D}, \mathbf{Q}, \rho \rangle$ [Generic Model] [Boolean Retrieval] [BIM] [BM25] [LSI] [ESA] [LM]

Document representations $\mathbf{D}$.

The set of index terms $T = \{t_1, \ldots, t_m\}$ is typically composed of the word stems of the vocabulary of a document collection, excluding stop words.

The representation $\mathbf{d}$ of a document d is a $|T|$-dimensional vector, where the i-th vector component of $\mathbf{d}$ corresponds to a term weight w_i of term $t_i \in T$, indicating its importance for d. Various term weighting schemes have been proposed.

Query representations $\mathbf{Q}$.

A query representation $\mathbf{q}$ is constructed like a document representation.

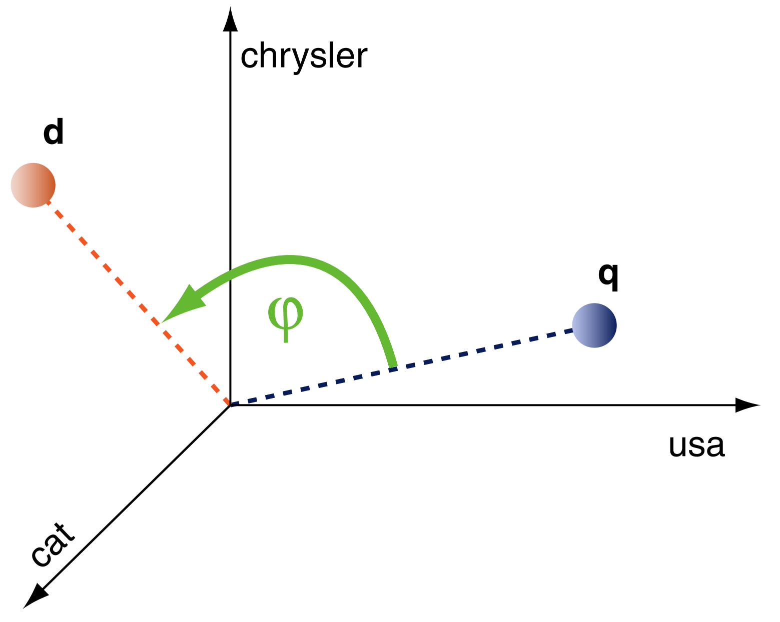
Relevance function ρ.

Document representations and query representations are interpreted as points in a vector space spanned by unit vectors for each term in T, assuming their orthogonality.

Distance and similarity functions defined for vector spaces serve as relevance functions ρ. The Euclidean distance and the cosine similarity are important examples.

Vector Space Model

Relevance Function ρ: Cosine Similarity



Vector Space Model

Relevance Function ρ: Cosine Similarity

The scalar product $\mathbf{a}^T\mathbf{b}$ between two n-dimensional vectors $\mathbf{a}$ and $\mathbf{b}$, where φ denotes the angle between them, is defined as follows:

$$\begin{aligned} \mathbf{a}^T\mathbf{b} &= ||\mathbf{a}|| \cdot ||\mathbf{b}|| \cdot \cos(\varphi) \\ \Leftrightarrow \quad \cos(\varphi) &= \frac{\mathbf{a}^T\mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||}, \end{aligned}$$

where $||\mathbf{x}||$ denotes the L2 norm of vector $\mathbf{x}$:

$$||\mathbf{x}|| = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$$

Let $\rho(\mathbf{q}, \mathbf{d}) = \cos(\varphi)$ the relevance function of the vector space model.

Vector Space Model

Example

$$\mathbf{d} = \begin{pmatrix} \texttt{chrysler} & w_1 \\ \texttt{usa} & w_2 \\ \texttt{cat} & w_3 \\ \texttt{dog} & w_4 \\ \texttt{mouse} & w_5 \end{pmatrix} = \begin{pmatrix} \texttt{chrysler} & 1 \\ \texttt{usa} & 4 \\ \texttt{cat} & 3 \\ \texttt{dog} & 7 \\ \texttt{mouse} & 5 \end{pmatrix}$$

$$\mathbf{d}' = \begin{pmatrix} \texttt{chrysler} & 0.1 \\ \texttt{usa} & 0.4 \\ \texttt{cat} & 0.3 \\ \texttt{dog} & 0.7 \\ \texttt{mouse} & 0.5 \end{pmatrix}, \qquad \mathbf{q}' = \begin{pmatrix} \texttt{chrysler} & 0.5 \\ \texttt{usa} & 0.5 \\ \texttt{cat} & 0.5 \\ \texttt{dog} & 0.5 \\ \texttt{elephant} & 0.5 \end{pmatrix}$$

The angle φ between $\mathbf{d}'$ and $\mathbf{q}'$ is about $41°$, $\cos(\varphi) \approx 0.75$.

Vector Space Model

Term Weighting: *tf* · *idf* [BIM Relevance Function]

To compute the weight w for a term t from document d under the vector space model, the most commonly employed term weighting scheme $\omega(t)$ is *tf* · *idf*:

- *tf*(t, d) denotes the normalized *term frequency* of term t in document d.
  The basic idea is that the importance of term t is proportional to its frequency in document d. However, t's importance does not increase linearly: the raw frequency must be normalized.

- *df*(t, D) denotes the *document frequency* of term t in document collection D.
  It counts the number of documents that contain t at least once.

- *idf*(t, D) denotes the *inverse document frequency*:

$$\mathit{idf}(t, D) = \log \frac{|D|}{\mathit{df}(t, D)}$$

  The importance of term t in general is inversely proportional to its document frequency.

A term weight w for term t in document $d \in D$ is computed as follows:

$$\omega(t) = \mathit{tf}(t, d) \cdot \mathit{idf}(t, D).$$

Vector Space Model

Term Weighting: *tf* · *idf* [BIM Relevance Function]

To compute the weight w for a term t from document d under the vector space model, the most commonly employed term weighting scheme $\omega(t)$ is *tf* · *idf*:

- *tf* (t, d) denotes the normalized *term frequency* of term t in document d.
  The basic idea is that the importance of term t is proportional to its frequency in document d. However, t's importance does not increase linearly: the raw frequency must be normalized.

- *df* (t, D) denotes the *document frequency* of term t in document collection D.
  It counts the number of documents that contain t at least once.

- *idf* (t, D) denotes the *inverse document frequency*:

$$\mathit{idf}(t, D) = \log \frac{|D|}{\mathit{df}(t, D)}$$

  The importance of term t in general is inversely proportional to its document frequency.

A term weight w for term t in document $d \in D$ is computed as follows:

$$\omega(t) = \mathit{tf}(t, d) \cdot \mathit{idf}(t, D).$$

Vector Space Model

Term Weighting: *tf · idf* [BIM Relevance Function]

To compute the weight w for a term t from document d under the vector space model, the most commonly employed term weighting scheme $\omega(t)$ is *tf · idf*:

- *tf*(t, d) denotes the normalized *term frequency* of term t in document d.
  The basic idea is that the importance of term t is proportional to its frequency in document d. However, t's importance does not increase linearly: the raw frequency must be normalized.

- *df*(t, D) denotes the *document frequency* of term t in document collection D.
  It counts the number of documents that contain t at least once.

- *idf*(t, D) denotes the *inverse document frequency*:

$$\mathit{idf}(t, D) = \log \frac{|D|}{\mathit{df}(t, D)}$$

  The importance of term t in general is inversely proportional to its document frequency.

A term weight w for term t in document $d \in D$ is computed as follows:

$$\omega(t) = \mathit{tf}(t, d) \cdot \mathit{idf}(t, D).$$

Vector Space Model

Term Weighting: *tf* · *idf* [BIM Relevance Function]

To compute the weight w for a term t from document d under the vector space model, the most commonly employed term weighting scheme $\omega(t)$ is *tf* · *idf*:

- *tf*(t, d) denotes the normalized *term frequency* of term t in document d.
  The basic idea is that the importance of term t is proportional to its frequency in document d. However, t's importance does not increase linearly: the raw frequency must be normalized.

- *df*(t, D) denotes the *document frequency* of term t in document collection D.
  It counts the number of documents that contain t at least once.

- *idf*(t, D) denotes the *inverse document frequency*:

$$\mathit{idf}(t, D) = \log \frac{|D|}{\mathit{df}(t, D)}$$

  The importance of term t in general is inversely proportional to its document frequency.

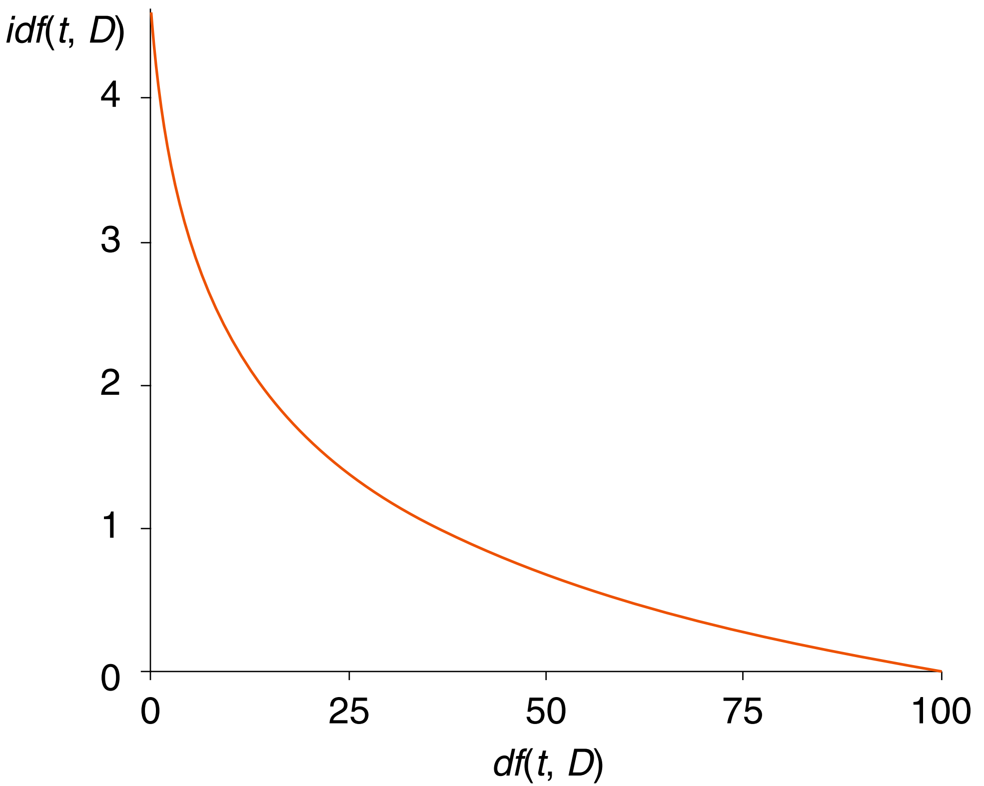
A term weight w for term t in document $d \in D$ is computed as follows:

$$\omega(t) = \mathit{tf}(t, d) \cdot \mathit{idf}(t, D).$$

Vector Space Model

Term Weighting: *tf* · *idf*

Plot of the function $\mathit{idf}(t, D) = \log \frac{|D|}{\mathit{df}(t, D)}$ for $|D| = 100$.



Remarks:

- Term frequency weighting was invented by Hans Peter Luhn: “There is also the probability that the more frequently a notion and combination of notions occur, the more importance the author attaches to them as reflecting the essence of his overall idea.” [Luhn 1957]

- The importance of a term t for a document d is not linearly correlated with its frequency. Several normalization factors have been proposed [Wikipedia]:
  - $\mathit{tf}(t,d)/|d|$
  - $1+\log(\mathit{tf}(t,d))$ for $\mathit{tf}(t,d) > 0$
  - $k + (1-k)\dfrac{\mathit{tf}(t,d)}{\max_{t' \in d}(\mathit{tf}(t',d))}$, where k serves as smoothing term; typically $k = 0.4$

- Inverse document frequency weighting was invented by Karen Spärck Jones: “it seems we should treat matches on non-frequent terms as more valuable than ones on frequent terms, without disregarding the latter altogether. The natural solution is to correlate a term’s matching value with its collection frequency.” [Spärck Jones 1972]

- Spärck Jones gives little theoretical justification for her intuition. Given the success of *idf* in practice, over the decades, numerous attempts at a theoretical justification have been made. A comprehensive overview has been compiled by [Robertson 2004].

- For example, interpreting the term $\frac{|D|}{\mathit{df}(t,D)}$ as inverse of the probability $P_{\mathit{df}}(t) = \frac{\mathit{df}(t,D)}{|D|}$ of t occurring in a random document in D yields $\mathit{idf}(t,D) = \log\frac{|D|}{\mathit{df}(t,D)} = -\log P_{\mathit{df}}(t)$. Logarithms fit relevance functions ρ since both are additive, yielding the interpretation: “The less likely (on a random basis) it is that a given combination of terms occurs, the more likely it is that a document containing this combination is relevant to the question.” [Robertson 1972]

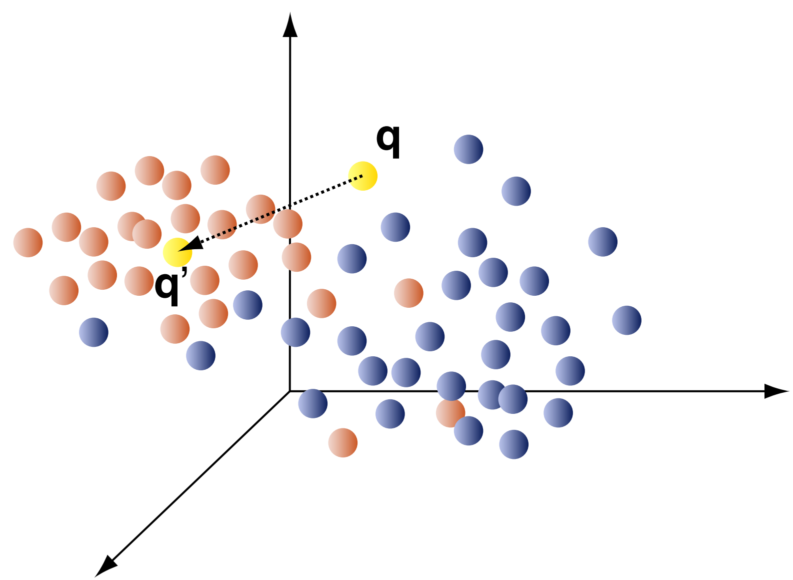
Vector Space Model

Query Refinement: Relevance Feedback

Given a result set R for a query q, and subsets $R^+ \subseteq R$ and $R^- \subseteq R$ of relevant and irrelevant documents, where $R^+ \cap R^- = \emptyset$, the query representation $\mathbf{q}$ can be refined using Rocchio's update formula:

$$\mathbf{q}' = \quad \alpha \cdot \mathbf{q} \quad + \quad \beta \cdot \frac{1}{|R^+|} \sum_{d^+ \in R^+} \mathbf{d}^+ \quad - \quad \gamma \cdot \frac{1}{|R^-|} \sum_{d^- \in R^-} \mathbf{d}^-,$$

where α, β, and γ adjust the impact of original query and (ir)relevant documents.



Vector Space Model

Query Refinement: Relevance Feedback

Given a result set R for a query q, and subsets $R^+ \subseteq R$ and $R^- \subseteq R$ of relevant and irrelevant documents, where $R^+ \cap R^- = \emptyset$, the query representation $\mathbf{q}$ can be refined using Rocchio's update formula:

$$\mathbf{q}' = \quad \alpha \cdot \mathbf{q} \quad + \quad \beta \cdot \frac{1}{|R^+|} \sum_{d^+ \in R^+} \mathbf{d}^+ \quad - \quad \gamma \cdot \frac{1}{|R^-|} \sum_{d^- \in R^-} \mathbf{d}^-,$$

where α, β, and γ adjust the impact of original query and (ir)relevant documents.

Observations:

- Terms not in query q may get added; often a limit is imposed (say, 50).
- Terms may accrue negative weight; such weights are set to 0.
- Moves the query vector closer to the centroid of relevant documents.
- Works well if relevant documents cluster; less suited for multi-faceted topics.

Relevance feedback can be obtained directly from the user, indirectly through user interaction, or automatically assuming the top-retrieved documents as relevant.

Vector Space Model

Discussion

Advantages:

- Severely improved retrieval performance compared to Boolean retrieval
- Partial query matching: not all query terms need to be present in a document for it to be retrieved
- The relevance function ρ defines a ranking among retrieved documents with respect to their computed similarity to the query

Disadvantages:

- Index terms are assumed to occur independent of one another