

Chapter IR:III

III. Retrieval Models

- ❑ Overview of Retrieval Models
- ❑ Boolean Retrieval
- ❑ Vector Space Model
- ❑ Binary Independence Model
- ❑ Okapi BM25
- ❑ Divergence From Randomness
- ❑ Latent Semantic Indexing
- ❑ Explicit Semantic Analysis
- ❑ Language Models
- ❑ Combining Evidence
- ❑ Learning to Rank

Boolean Retrieval

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

Document representations \mathbf{D} .

- $T = \{t_1, \dots, t_m\}$ is the set of m index terms (lemmatized or stemmed words).
- T are the atoms of a logical formula for d with operators \wedge, \vee, \neg , and brackets.
- $\mathbf{d} = (\bigwedge_{t \in d} t) \wedge \neg(\bigwedge_{t \notin d} t)$, where $\mathcal{I}_d(t) = 1$ if t occurs in d , and $\mathcal{I}_d(t) = 0$ otherwise.

Query representations \mathbf{Q} .

- \mathbf{q} is a logical formula over T .

Relevance function ρ .

- $\rho(d, q) = \mathcal{I}(\mathbf{d} \rightarrow \mathbf{q})$, where \rightarrow is the logical implication.
- $\rho(d, q) = 1$ indicates relevance of d to q , and $\rho(d, q) = 0$ otherwise.
- $R_q \subseteq D$ is the set of documents $d \in D$ relevant to q , i.e., with $\rho(d, q) = 1$.
- $\rho'(d, q) = P(\mathcal{I}(\mathbf{d} \rightarrow \mathbf{q}) = 1) = P(\mathbf{d} \rightarrow \mathbf{q}) = P(q \mid d)$ relaxes relevance scoring.

Boolean Retrieval

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

Document representations \mathbf{D} .

- $T = \{t_1, \dots, t_m\}$ is the set of m index terms (lemmatized or stemmed words).
- T are the atoms of a logical formula for d with operators \wedge, \vee, \neg , and brackets.
- $\mathbf{d} = (\bigwedge_{t \in d} t) \wedge \neg(\bigwedge_{t \notin d} t)$, where $\mathcal{I}_{\mathbf{d}}(t) = 1$ if t occurs in d , and $\mathcal{I}_{\mathbf{d}}(t) = 0$ otherwise.

Query representations \mathbf{Q} .

- \mathbf{q} is a logical formula over T .

Relevance function ρ .

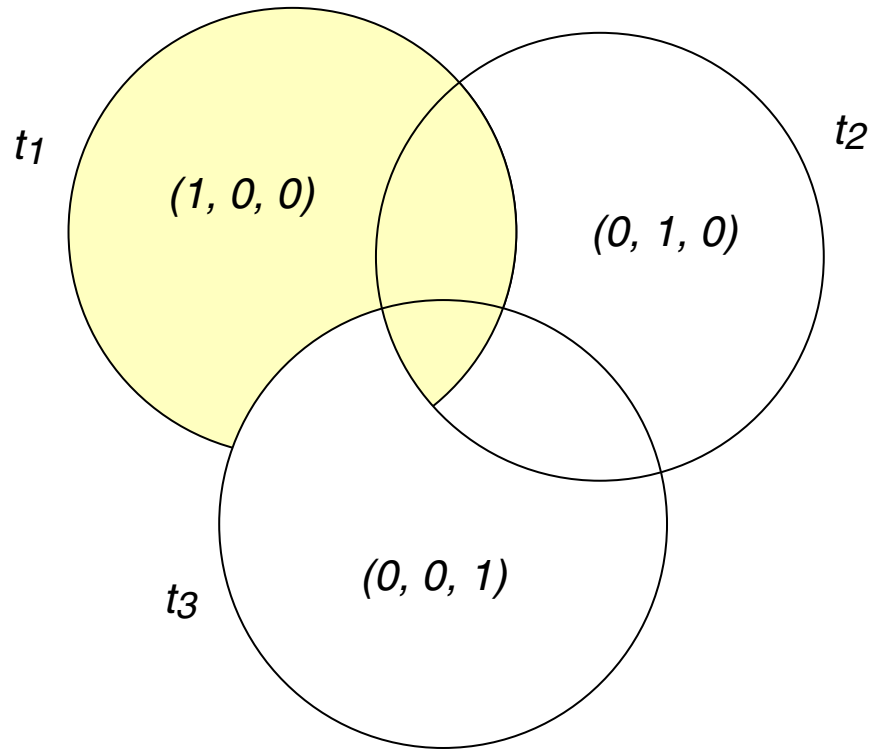
- $\rho(d, q) = \mathcal{I}(\mathbf{d} \rightarrow \mathbf{q})$, where \rightarrow is the logical implication.
- $\rho(d, q) = 1$ indicates relevance of d to q , and $\rho(d, q) = 0$ otherwise.
- $R_q \subseteq D$ is the set of documents $d \in D$ relevant to q , i.e., with $\rho(d, q) = 1$.
- $\rho'(d, q) = P(\mathcal{I}(\mathbf{d} \rightarrow \mathbf{q}) = 1) = P(\mathbf{d} \rightarrow \mathbf{q}) = P(q \mid d)$ relaxes relevance scoring.

Remarks:

- $\mathcal{I} : T \rightarrow \{0, 1\}$ and $\mathcal{I} : \{\alpha \mid \alpha \text{ is a logical formula over } T\} \rightarrow \{0, 1\}$ is the evaluation or interpretation function that assigns truth values to the atoms T as well as to propositional formulas over them.

Boolean Retrieval

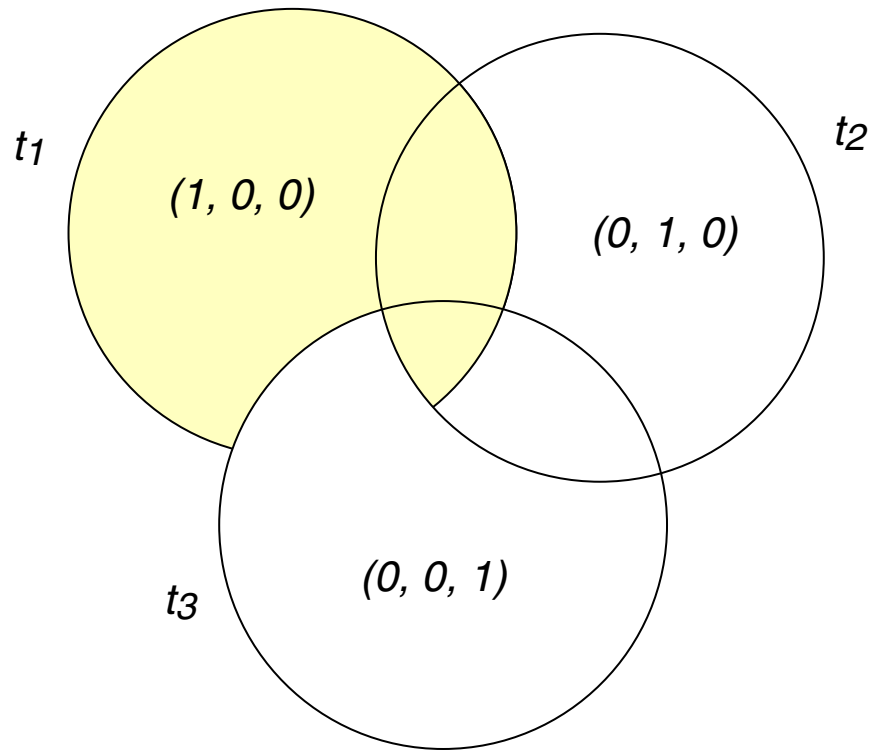
Relevance Function ρ



What query is illustrated?

Boolean Retrieval

Relevance Function ρ



What query is 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:

$$\begin{aligned}\mathbf{d} = & \text{chrysler} \wedge \text{deal} \wedge \text{usa} \\ & \wedge \text{china} \wedge \neg \text{cat} \wedge \text{sales} \\ & \wedge \neg \text{dog} \wedge \dots\end{aligned}$$

Query representation:

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

Relevance function:

$$\rho(d, q) = \mathcal{I}(\mathbf{d} \rightarrow \mathbf{q}) = 1, \text{ since } \mathcal{I}_d(\text{usa}) = 1, \mathcal{I}_d(\text{dog}) = 0, \text{ and } \mathcal{I}_d(\text{cat}) = 0.$$

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`

Results: many pages about cars, places, people

2. `president \wedge lincoln`

A 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`

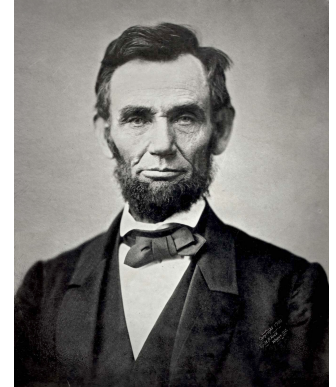
Not a result: “President Lincoln’s body departs Washington in a nine-car funeral train.”

4. `president \wedge lincoln \wedge \neg automobile \wedge biography \wedge life \wedge birthplace \wedge gettysburg`

Results: \emptyset

5. `president \wedge lincoln \wedge \neg automobile \wedge (biography \vee life \vee birthplace \vee gettysburg)`

A result: “President’s Day – Holiday activities – crafts, mazes, word searches, ...’The Life of Washington’ Read 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: medical retrieval, patent retrieval, eDiscovery (law))
- ❑ the size of the result set is difficult to be controlled

Vector Space Model

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

Document representations \mathbf{D} .

- $T = \{t_1, \dots, t_m\}$ is the set of m index terms (word stems, without stop words).
- T is interpreted as set of dimensions of an m -dimensional vector space.
- $\omega : \mathbf{D} \times T \rightarrow \mathbf{R}$ is a term weighting function, quantifying term importance.
- $\mathbf{d} = (w_1, \dots, w_m)^T$, where $w_i = \omega(\mathbf{d}, t_i)$ is the term weight of the i -th term in T .

Query representations \mathbf{Q} .

- $\mathbf{q} = (w_1, \dots, w_m)^T$, where $w_i = \omega(\mathbf{q}, t_i)$ is the term weight of the i -th term in T .

Relevance function ρ .

- Distance and similarity functions φ serve as relevance functions.
- $\rho(d, q) = \varphi(\mathbf{d}, \mathbf{q}) = \mathbf{d}^T \mathbf{q}$, the scalar product of vectors \mathbf{d} and \mathbf{q} .
- Normalizing \mathbf{d} and \mathbf{q} calculates cosine similarity, else Euclidean distance.

Vector Space Model

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

Document representations \mathbf{D} .

- $T = \{t_1, \dots, t_m\}$ is the set of m index terms (word stems, without stop words).
- T is interpreted as set of dimensions of an m -dimensional vector space.
- $\omega : \mathbf{D} \times T \rightarrow \mathbf{R}$ is a term weighting function, quantifying term importance.
- $\mathbf{d} = (w_1, \dots, w_m)^T$, where $w_i = \omega(\mathbf{d}, t_i)$ is the term weight of the i -th term in T .

Query representations \mathbf{Q} .

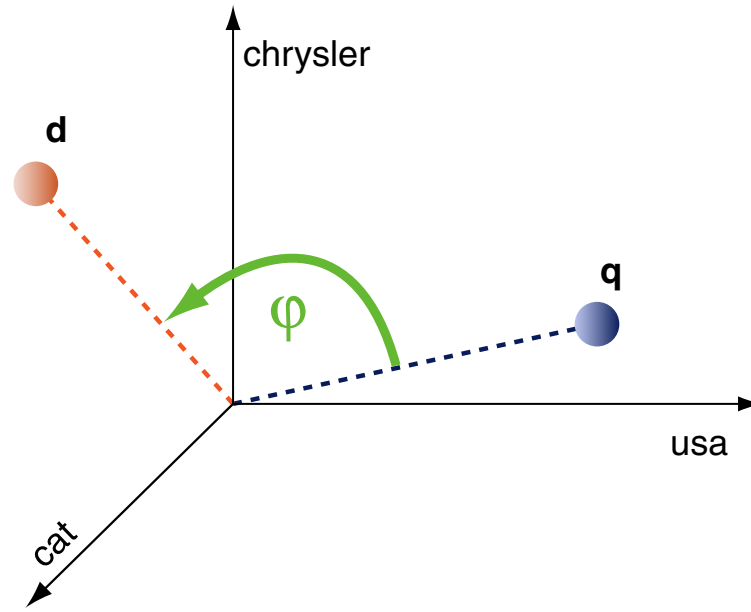
- $\mathbf{q} = (w_1, \dots, w_m)^T$, where $w_i = \omega(\mathbf{q}, t_i)$ is the term weight of the i -th term in T .

Relevance function ρ .

- Distance and similarity functions φ serve as relevance functions.
- $\rho(d, q) = \varphi(\mathbf{d}, \mathbf{q}) = \mathbf{d}^T \mathbf{q}$, the scalar product of vectors \mathbf{d} and \mathbf{q} .
- Normalizing \mathbf{d} and \mathbf{q} calculates cosine similarity, else Euclidean distance.

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:

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

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

Vector Space Model

Example

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

$$\mathbf{d}' = \begin{pmatrix} \text{chrysler} & 0.05 \\ \text{usa} & 0.2 \\ \text{cat} & 0.15 \\ \text{dog} & 0.35 \\ \text{mouse} & 0.25 \end{pmatrix}, \quad \mathbf{q}' = \begin{pmatrix} \text{chrysler} & 0.2 \\ \text{usa} & 0.2 \\ \text{cat} & 0.2 \\ \text{dog} & 0.2 \\ \text{elephant} & 0.2 \end{pmatrix}$$

Vector Space Model

Example

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

$$\mathbf{d}' = \begin{pmatrix} \text{chrysler} & 0.05 \\ \text{usa} & 0.2 \\ \text{cat} & 0.15 \\ \text{dog} & 0.35 \\ \text{mouse} & 0.25 \end{pmatrix}, \quad \mathbf{q}' = \begin{pmatrix} \text{chrysler} & 0.2 \\ \text{usa} & 0.2 \\ \text{cat} & 0.2 \\ \text{dog} & 0.2 \\ \text{elephant} & 0.2 \end{pmatrix}$$

Vector Space Model

Example

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

$$\mathbf{d}' = \begin{pmatrix} \text{chrysler} & 0.05 \\ \text{usa} & 0.2 \\ \text{cat} & 0.15 \\ \text{dog} & 0.35 \\ \text{mouse} & 0.25 \\ \text{elephant} & 0.0 \end{pmatrix}, \quad \mathbf{q}' = \begin{pmatrix} \text{chrysler} & 0.2 \\ \text{usa} & 0.2 \\ \text{cat} & 0.2 \\ \text{dog} & 0.2 \\ \text{mouse} & 0.0 \\ \text{elephant} & 0.2 \end{pmatrix}$$

The angle φ between \mathbf{d}' and \mathbf{q}' is about 48° , $\cos(\varphi) \approx 0.67$.

The weights in \mathbf{d}' and \mathbf{q}' denote the relative term frequency $w'_i = \frac{w_i}{\sum_{j=1}^5 w_j}$. Dimensions are aligned with zero padding. The product $\mathbf{d}'^T \mathbf{q}' = 0.15$, the norms $\|\mathbf{d}'\| = 0.5$ and $\|\mathbf{q}'\| = 0.447$.

Vector Space Model

Term Weighting: $tf \cdot idf$ [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 $tf \cdot idf$:

- $tf(t, d)$ 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*:

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

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

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

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

Vector Space Model

Term Weighting: $tf \cdot idf$ [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 $tf \cdot idf$:

- $tf(t, d)$ 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*:

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

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

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

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

Vector Space Model

Term Weighting: $tf \cdot idf$ [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 $tf \cdot idf$:

- $tf(t, d)$ 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*:

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

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

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

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

Vector Space Model

Term Weighting: $tf \cdot idf$ [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 $tf \cdot idf$:

- $tf(t, d)$ 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*:

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

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

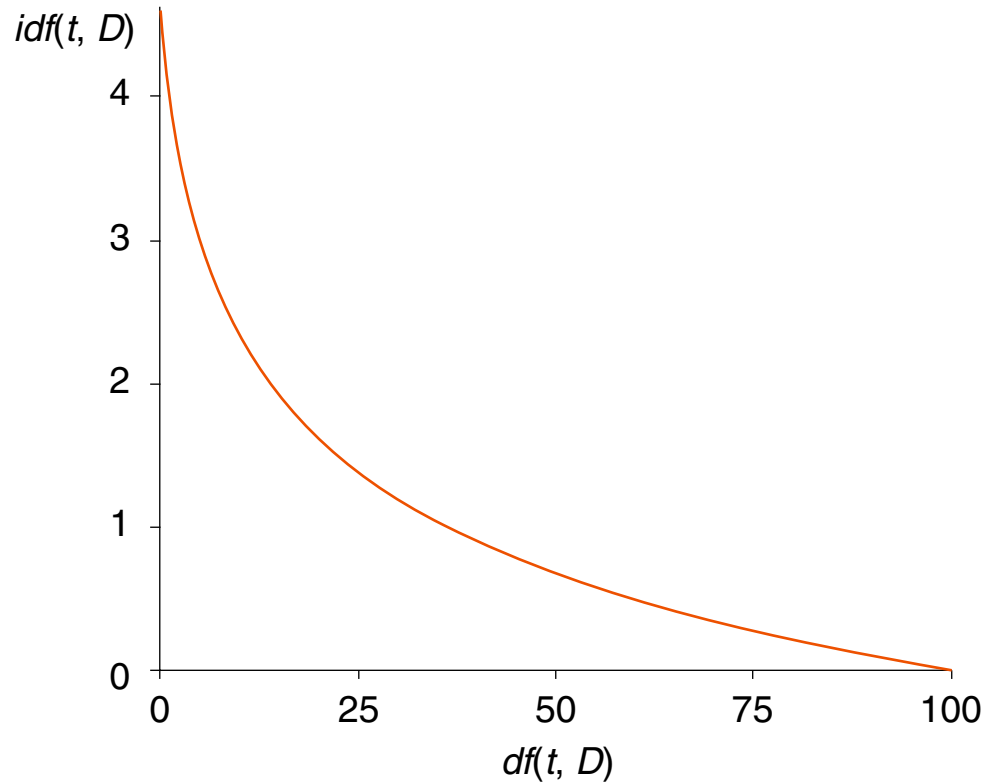
A term weight ω for term t in document $d \in D$ is computed as follows:

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

Vector Space Model

Term Weighting: $tf \cdot idf$

Plot of the function $idf(t, D) = \log \frac{|D|}{df(t, D)}$ 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 [\[Wikipedia\]](#):
 - $tf(t, d)/|d|$
 - $1 + \log(tf(t, d))$ for $tf(t, d) > 0$
 - $k + (1 - k) \frac{tf(t, d)}{\max_{t' \in d}(tf(t', d))}$, 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 $\frac{|D|}{df(t, D)}$ as inverse of the probability $P_{df}(t) = \frac{df(t, D)}{|D|}$ of t occurring in a random document in D yields $idf(t, D) = \log \frac{|D|}{df(t, D)} = -\log P_{df}(t)$. 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.” [\[Robertson 1972\]](#)

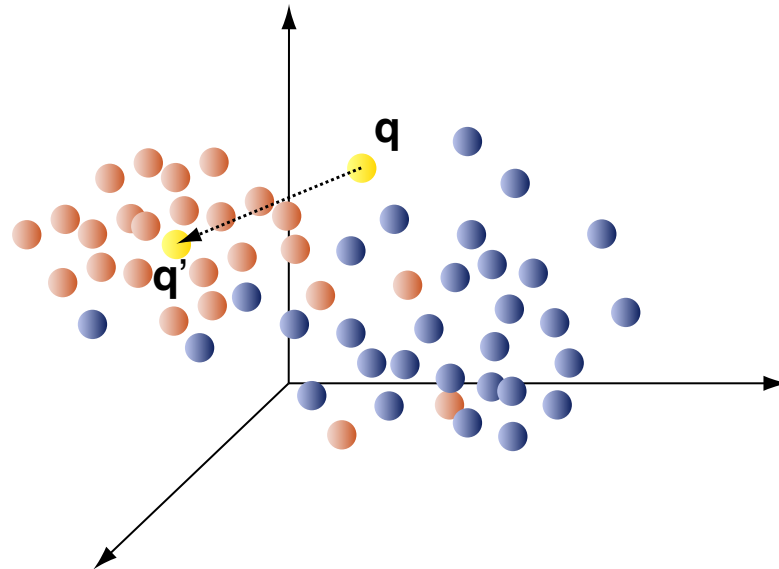
Vector Space Model

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 non-relevant documents, where $R^+ \cap R^- = \emptyset$, the query representation \mathbf{q} can be refined with the document representations \mathbf{R} using Rocchio's update formula:

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

where α , β , and γ adjust the impact of original query and (non-)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 non-relevant documents, where $R^+ \cap R^- = \emptyset$, the query representation \mathbf{q} can be refined with the document representations \mathbf{R} using Rocchio's update formula:

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

where α , β , and γ adjust the impact of original query and (non-)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:

- ❑ 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 the retrieved documents with respect to their computed similarity to the query

Disadvantages:

- ❑ Index terms are assumed to occur independent of one another