Week 1: Retrieval Models

Ingemar J. Cox
COMP0084
University College London

Retrieval Models

- abstract away from the real world
- are mathematical representations of the essential aspects of a retrieval system
- aim at computing relevance and retrieving relevant documents
- thus, either explicitly or implicitly, they define relevance

How do we define relevance?

- There are many factors, such as
  - Topicality
  - Novelty
  - Freshness
  - Authority
- Focus on topicality, i.e.
  - Is the document on the same topic as the query?

Retrieval models

- Exact matching
- Best matching (ranked retrieval)

Exact matching

- Query specifies ***precise*** retrieval criteria
- Every document either matches or fails to match query
- relevant or non-relevant

Ranked retrieval

- Query describes retrieval criteria for desired documents
- Every document matches a query to *some degree*
- Result is a ranked list of documents on the basis of either probability of relevance or graded relevance
  - Web search is ranked retrieval

Retrieval models

- Boolean
  - Basic Boolean
  - Extended Boolean model
- Vector space model — SMART
- Probabilistic models
  - Basic probabilistic model
  - Two Poisson model – BM25 — Okapi
  - Bayesian inference networks — Indri
  - Statistical language models — Lemur
  - Portfolio retrieval
- Citation analysis models
  - Hubs & authorities — CLEVER by IBM
  - PageRank — Google
- Learning to rank

BASIC BOOLEAN RETRIEVAL

Binary full text representation of collection

	a	Aachen	abandon	abate	…	zygote
Doc_1	1	1	0	0	…	1
Doc_2	1	0	1	1	…	0
Doc_3	1	0	0	1		0
					…	
Doc_N	1	1	1	0	…	0

Example query

Abandon Aachen

Binary full text representation of collection

	a	Aachen	abandon	abate	…	zygote
Doc_1	1	1	0	0	…	1
Doc_2	1	0	1	1	…	0
Doc_3	1	0	0	1		0
					…	
Doc_N	1	1	1	0	…	0

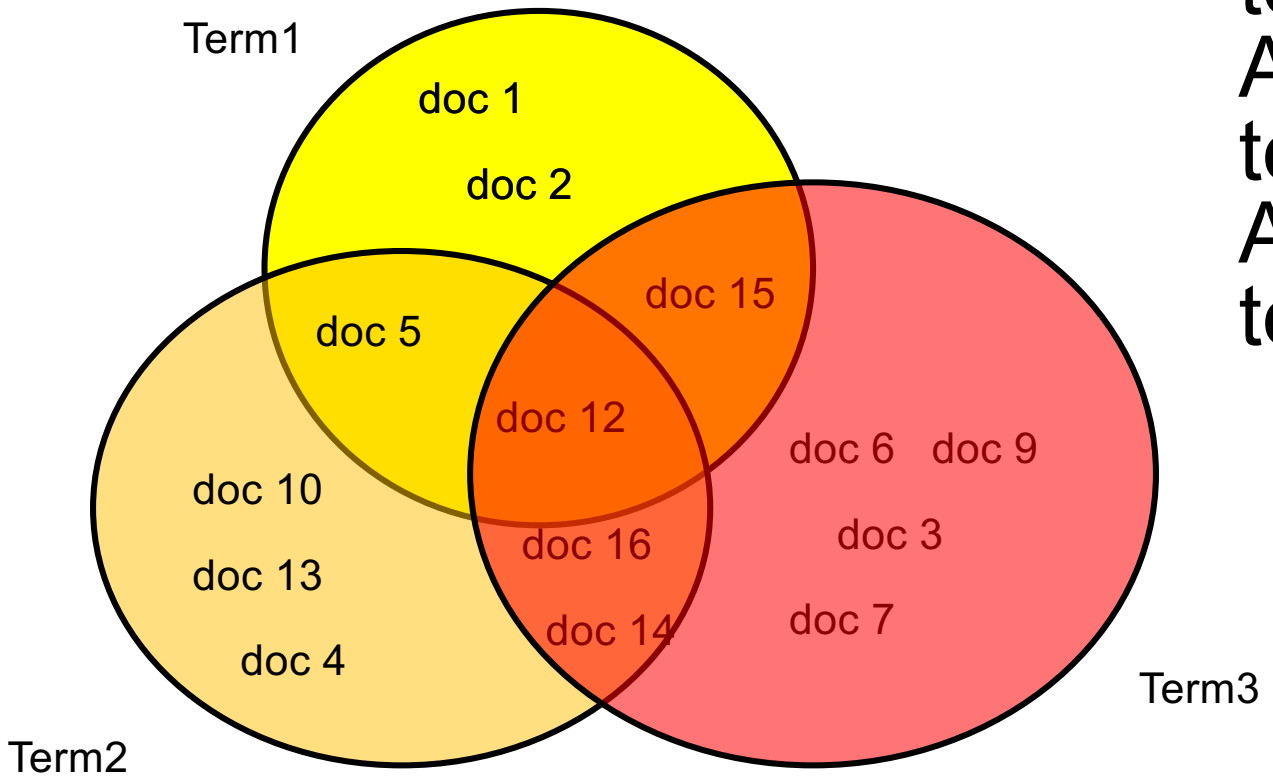
Query	0	1	1	0	…	0

How can we match the query to the documents?

The basic Boolean retrieval

- **The simplest Exact Match model**
  - Retrieve documents iff they satisfy a Boolean expression
  - Query specifies precise relevance criteria
  - Documents returned in no particular order
- **Document:** A bag of words
- **Query:** A Boolean expression
- **Operators:**
  - Logical operators: AND, OR, AND NOT
  - Proximity operators: number of intervening words between two query terms, etc
  - String matching operators: Wild-card

The basic Boolean retrieval



Query:
term1
AND
term2
AND NOT
term3

Boolean retrieval: application

- Largest commercial legal search service (www.westlaw.com)
- Over half a million subscribers performing millions of searches a day over tens of terabytes of text data
- In operation since 1975.
- In 2005, Boolean search was still the default, and used by a large percentage of users
- Although ranked retrieval has been available since 1992.

(Manning IIR 2008)

Boolean retrieval: Advantages

- Works well if you know exactly what you want
- Structured queries
- Simple to program
- Complete expressiveness
- Computationally efficient

Boolean retrieval: Disadvantages

- Artificial language
  - IN/smith AND APD//1/1/1790->12/31/2001
  - unintuitive, misunderstood
- Either too many or too few documents
  - Difficult to balance precision and recall
- Unordered output
  - must examine all of the results

Extensions to Boolean retrieval

- proximity operators
  - impose constraints on relative position of query-terms
- field restriction
  - impose constraints on location of query-terms, e.g. Title, Abstract
- wild-card operators
  - impose constraints on matching query-terms with index-terms

PROXIMITY OPERATORS

Ordered window

A OW/N B

Term A must appear no more than N terms before Term B

Ordered window: Example

Paris OW/2 Climate

Paris climate change accord

Paris hosts climate change

Climate of Paris

Paris climate in summer

Ordered window: Example

Paris OW/2 Climate

Paris climate change accord

Paris hosts climate change

Climate of Paris

Paris climate in summer

Ordered window: Example

Paris OW/2 Climate

Paris climate change accord

Paris hosts climate change

Climate of Paris

Paris climate in summer

Ordered window: Example

Paris OW/2 Climate

Paris climate change accord

Paris hosts climate change

Climate of Paris

Paris climate in summer

Ordered window: Example

Paris OW/2 Climate

Paris climate change accord

Paris hosts climate change

Climate of Paris

Paris climate in summer

Unordered window

A UW/N B

Term A must appear no more than N terms from Term B

Unordered window: Example

Paris UW/2 Climate

Paris climate change accord

Paris hosts climate change

Climate of Paris

Paris climate in summer

Unordered window: Example

Paris UW/2 Climate

Paris climate change accord

Paris hosts climate change

Climate of Paris

Paris climate in summer

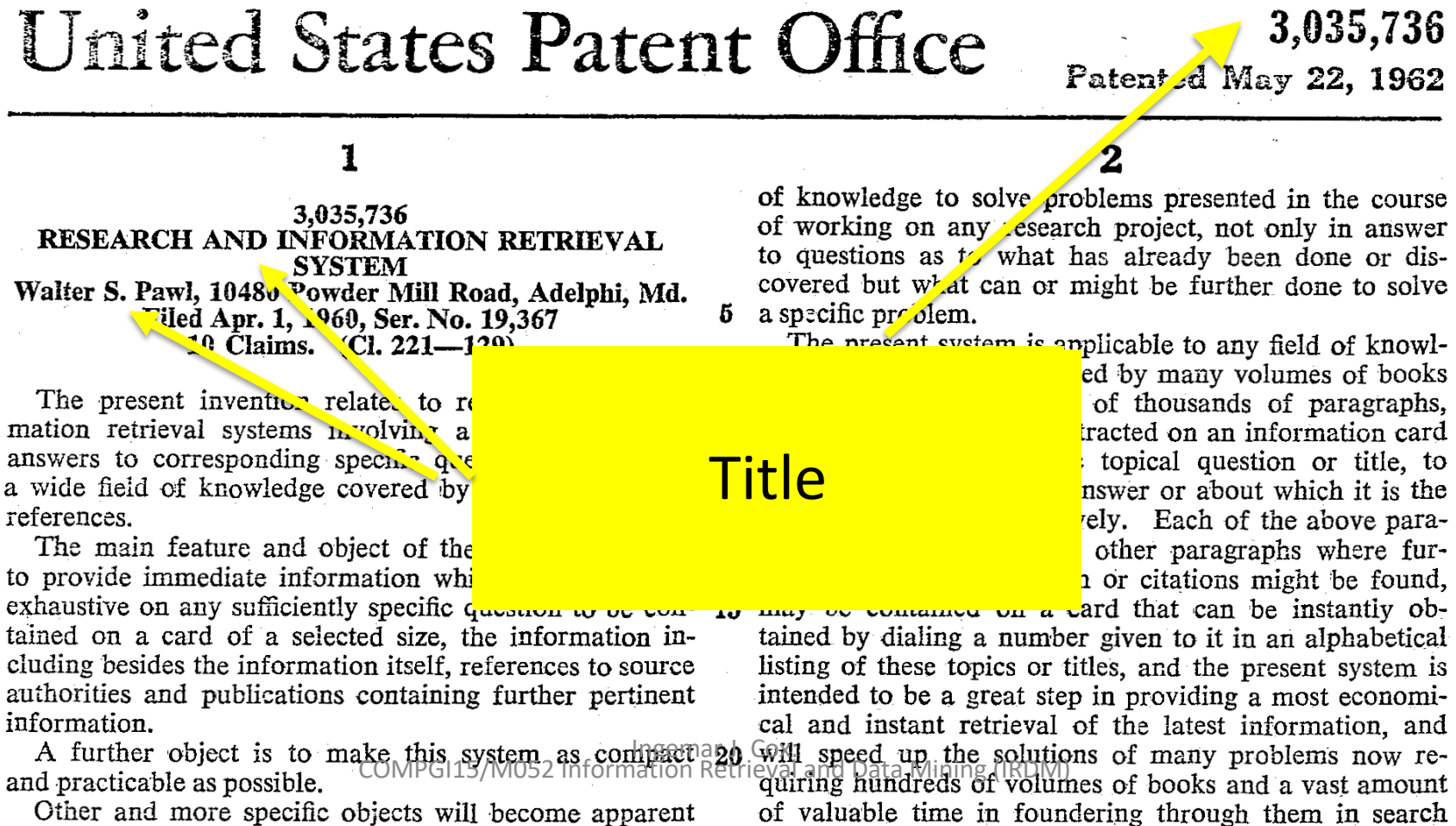
Phrase

“A B”

Term A must appear immediately before Term B

FIELD RESTRICTIONS

Field restrictions



United States Patent Office

3,035,736

Patented May 22, 1962

1

3,035,736

RESEARCH AND INFORMATION RETRIEVAL SYSTEM

Walter S. Pawl, 10480 Powder Mill Road, Adelphi, Md.

Filed Apr. 1, 1960, Ser. No. 19,367

10 Claims. (Cl. 221—129)

The present invention relates to re
mation retrieval systems involving a
answers to corresponding specific qu
a wide field of knowledge covered by
references.

The main feature and object of the
to provide immediate information whi
exhaustive on any sufficiently specific question to be con
tained on a card of a selected size, the information in-
cluding besides the information itself, references to source
authorities and publications containing further pertinent
information.

A further object is to make this system as compact
and practicable as possible.

Other and more specific objects will become apparent

2

of knowledge to solve problems presented in the course
of working on any research project, not only in answer
to questions as to what has already been done or dis-
covered but what can or might be further done to solve
a specific problem.

The present system is applicable to any field of knowl-
edge covered by many volumes of books
containing hundreds of thousands of paragraphs,
which are extracted on an information card
under a topical question or title, to
which the answer or about which it is the
most relevant. Each of the above para-
graphs may refer to other paragraphs where fur-
ther information or citations might be found,
may be contained on a card that can be instantly ob-
tained by dialing a number given to it in an alphabetical
listing of these topics or titles, and the present system is
intended to be a great step in providing a most economi-
cal and instant retrieval of the latest information, and
will speed up the solutions of many problems now re-
quiring hundreds of volumes of books and a vast amount
of valuable time in foundering through them in search

Title

Field restrictions: Example

freepatentsonline.com

expert search

IN/smith AND APD//1/1/1790->12/31/2001

WILDCARDS

CAN WE MODIFY BOOLEAN SEARCH TO RANKED SEARCH?

Ranked Boolean retrieval

- Ranked Boolean is another common Exact Match retrieval model
- Model and operators are the same as for Boolean
- The only difference is that matched documents are ranked by frequency of query terms
  - Document term weights: how often a term occurs in a document – may be normalized
  - AND weight: Minimum of argument weights
  - OR weight: Maximum of argument weights
  - and, sum of all argument weights

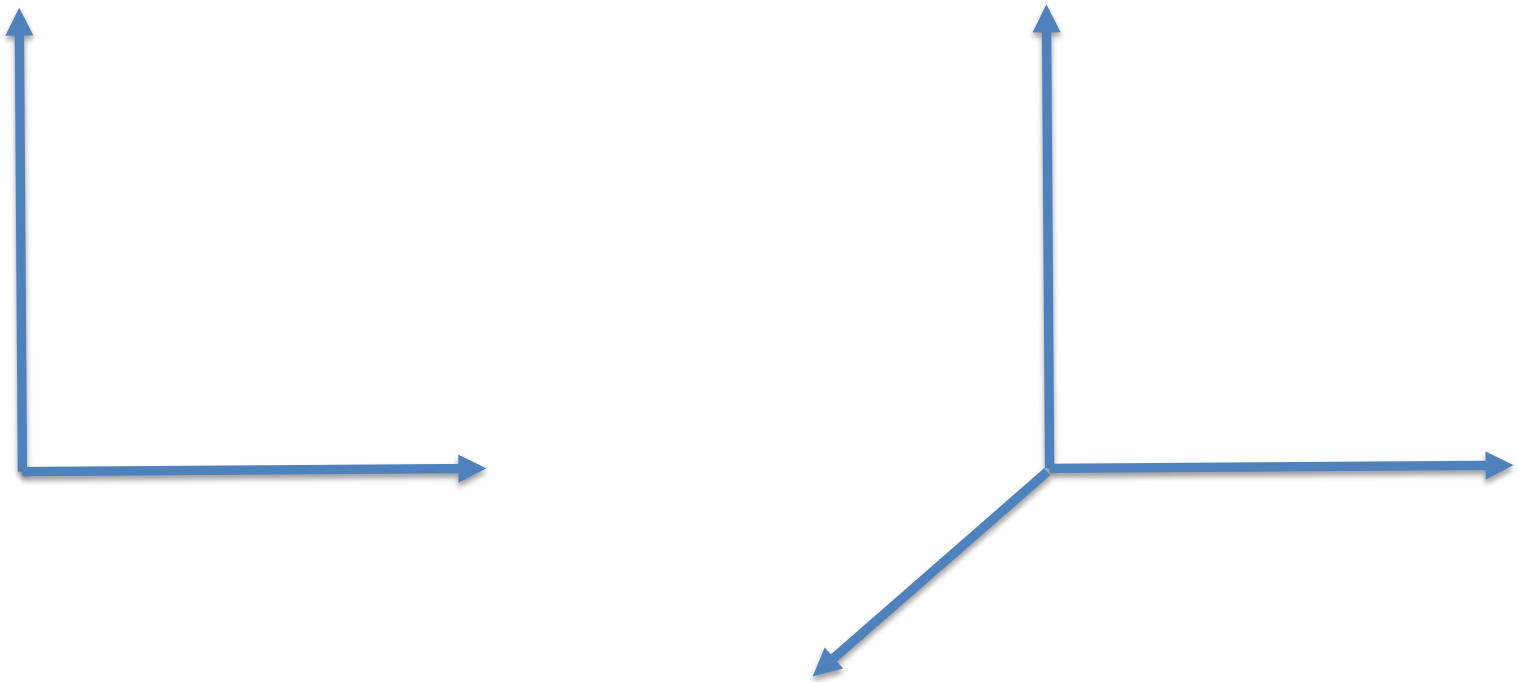
Ranked Boolean retrieval

- Query is “brown” AND “cat”
- Document1 contains 3 occurrences of “brown” and 5 of “cat”
  - Score = min(3,5) = 3
- Document2 contains 4 occurrences of “brown” and 5 of “cat”
  - Score = min(4,5) = 4
- Document2 is more relevant

VECTOR SPACE

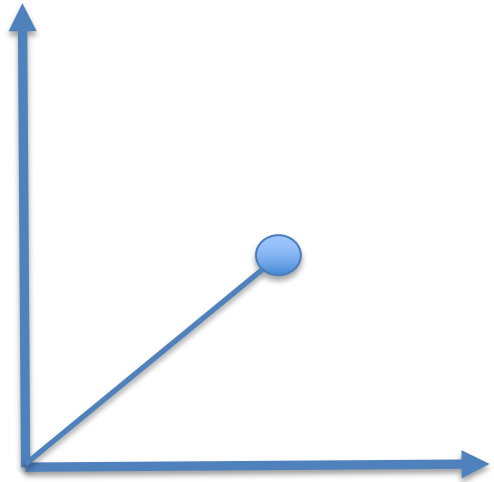
Vector space

Defined by a set of linearly independent basis vectors



Vector

A point in this space, [x, y]



Binary full text representation of collection

	a	Aachen	abandon	abate	…	zygote
Doc_1	1	1	0	0	…	1
Doc_2	1	0	1	1	…	0
Doc_3	1	0	0	1		0
					…	
Doc_N	1	1	1	0	…	0

Query	0	1	1	0	…	0

Binary representation of words

"The brown fox jumped over the brown dog and the black dog"

Word	Vector representation
and	000
brown	001
dog	010
fox	011
jumped	100
over	101
the	110

Binary representation of words

"The brown fox jumped over the brown dog and the black dog"

Word	Vector representation
and	0000001
brown	0000010
dog	0000100
fox	0001000
jumped	0010000
over	0100000
the	1000000

Binary representation of words

Aggregation: What if we add the two vectors “brown” and “dog”?

Word	Vector representation
and	000
brown	001
dog	010
fox	011
jumped	100
over	101
the	110

Binary representation of words

Aggregation: What if we add the two vectors “brown” and “dog”?

Word	Vector representation
and	0000001
brown	0000010
dog	0000100
fox	0001000
jumped	0010000
over	0100000
the	1000000

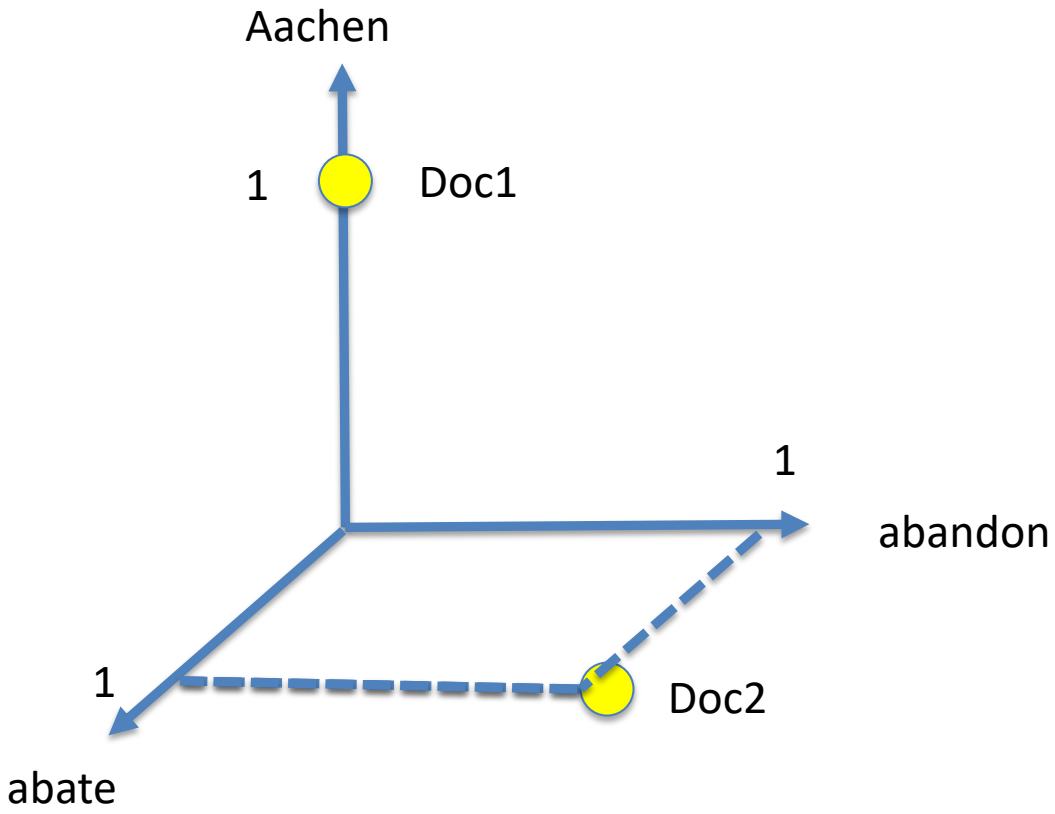
Binary representation of words

What is the distance between two words?

Word	Vector representation
and	0000001
brown	0000010
dog	0000100
fox	0001000
jumped	0010000
over	0100000
the	1000000

Vector space representation

Binary weights



Vector space representation

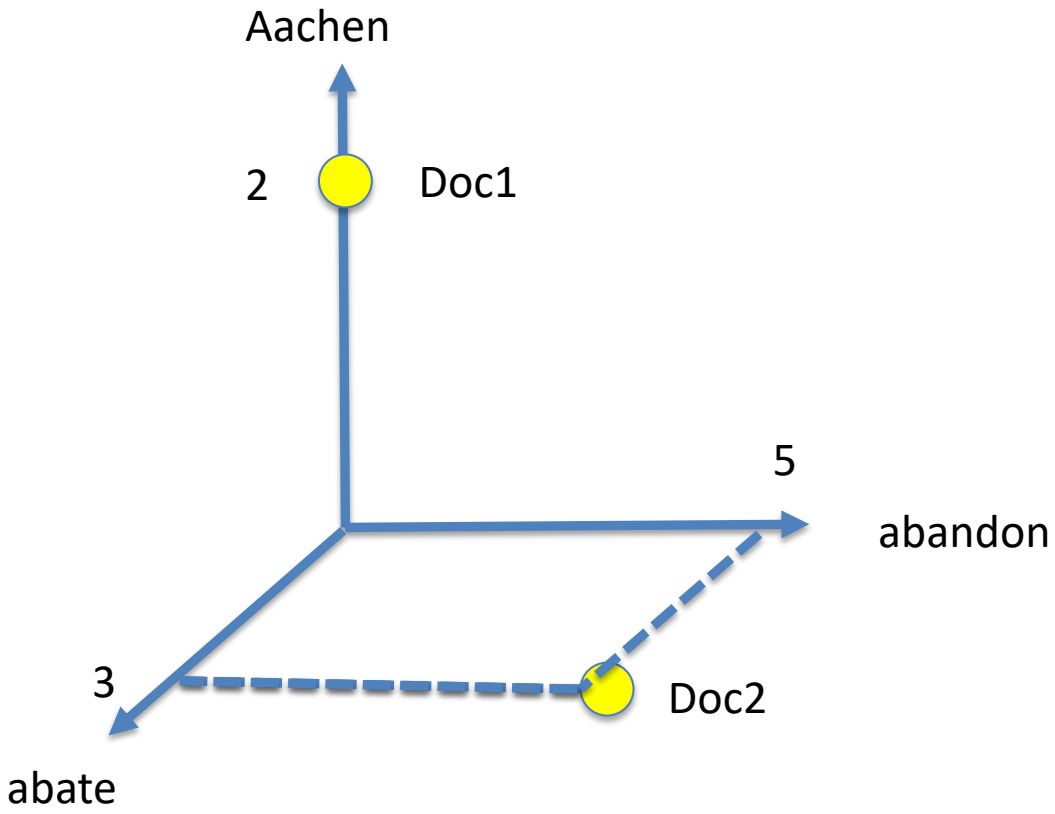
Query can also be treated as a vector

Vector space representation

Relevance is based on how close a document (in vector space) is to a query (in vector space)

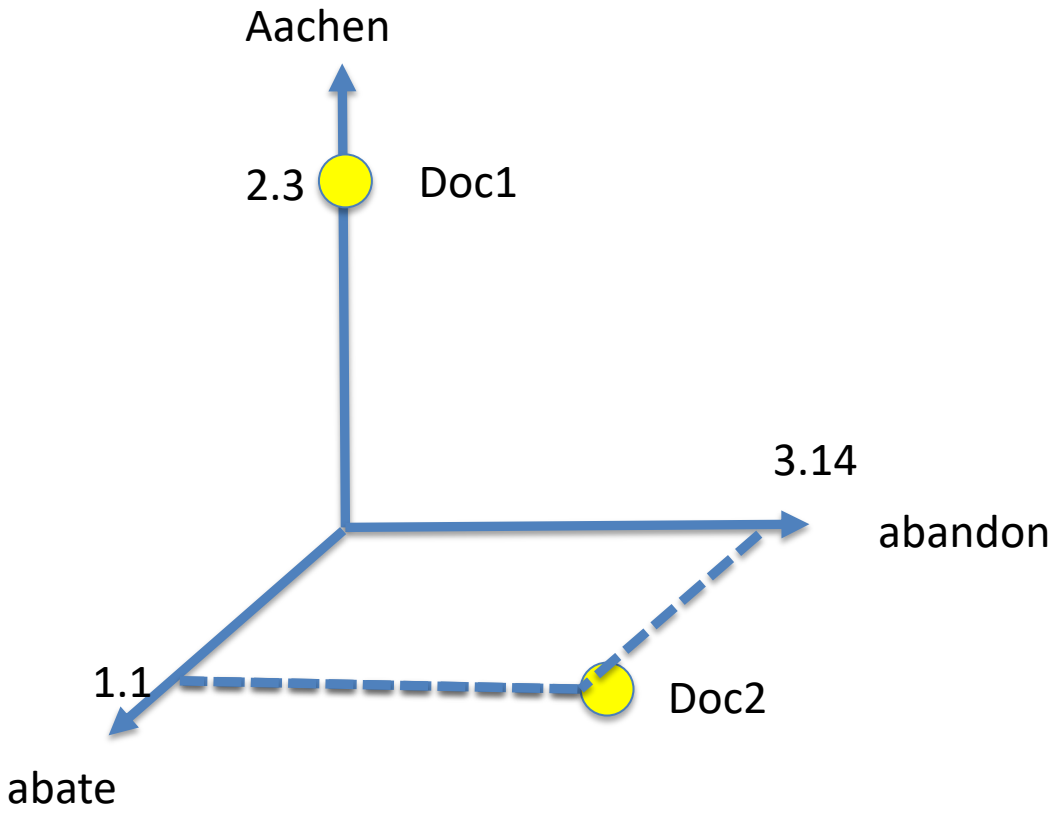
Vector space representation

Non-binary weights



Vector space representation

Non-binary weights



Documents as vectors

- So we have a |V|-dimensional vector space, where |V| is the vocabulary size
- Terms are axes of the space
- Documents are points or vectors in this space
- Very high-dimensional: tens of millions of dimensions when you apply this to a web search engine
- These are very sparse vectors - most entries are zero (inverted index exploits this)

Queries as vectors

- Do the same for queries: represent them as vectors in the space
- Rank documents according to their proximity to the query in this space
  - proximity = similarity of vectors
  - proximity ≈ inverse of distance
- Recall: We do this because we want to get away from the "either-in-or-out" Boolean model.
- The intent is to rank more relevant documents higher than less relevant documents

Formalizing vector space proximity

- We need to come up with a distance between two points
- But first, let's consider the binary vector representation
  - And then extend it

Query-document matching scores

We need a way of assigning a score (distance) to a query/document pair

Query-document matching scores

- Let's start with the following simple score:

$$\text{Score}(q,d) = |g \cap d| = \sum_{t \in (g \cap d)} 1$$

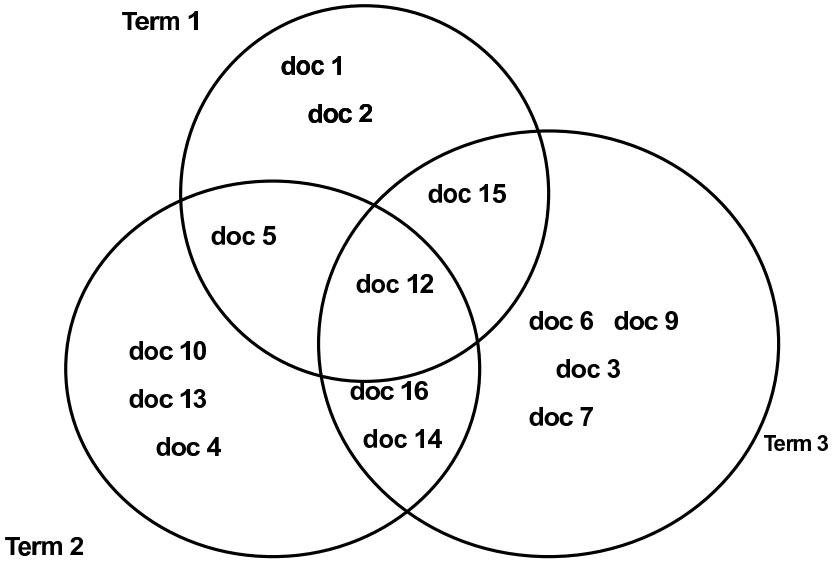
  - Counts the number of terms in common between a query and a document
  - permits AND logic
  - This number has become known as the co-ordination level

Cyril W. Cleverdon. Aslib cranfield research project: report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. *Cranfield Library*, 1962.

Co-ordination Level Matching

- Suppose we query “term#1 term#2 term#3”

Co-ordination level	*docs*
3	doc 12
2	doc 15 doc15 doc16 doc 14
1	the rest



Query-document matching scores

The score:

$$\text{Score}(q, d) = |g \cap d| = \sum_{t \in (g \cap d)} 1$$

is equivalent to

$$\sum_{i=1}^{V} d_i \times q_i$$

if d_i is binary

Inner product of vectors

$$\sum_{i=1}^{V} x_i \times y_i$$

Inner product of vectors

Multiply corresponding components and then sum the products.

When using 0’s and 1’s, this is just the number of terms in common between the query and the document

Inner product of binary vectors

Hamming distance measures number of positions that differ between two vectors

Inner product is equivalent to

V-(Hamming distance)

Inner product of vectors

What’s wrong with using the inner product to measure closeness?

Problems with inner product of vectors

- Which document is more relevant?
  - a 50-word document which contains all the query terms once, or
  - a 50-word document which contains all the query terms twice, or
  - A 500-word document which contains all the query terms thrice (3 times)?

Co-ordination level

- Assumes the more terms in common between a document and a query, the more likely it is that the document is relevant to the query
- However, it does not consider the frequency of a query term in a document
  - What if we assume that the more often a term appears in a document the more relevant it is?

TERM FREQUENCY
INVERSE DOCUMENT FREQUENCY

Term frequency

Previously elements of a vector were binary.

Indicated whether the term was present or absent

However the frequency of the term in the document is lost

Term frequency

Term frequency, tf, is the number of times the term occurs in a document.

Term frequency

"The brown fox jumped over the brown dog and the black dog"

Word	Term frequency
and	1
black	1
brown	2
dog	2
fox	1
jumped	1
over	1
the	3

Term Weighting

- Co-ordination Level Matching:
  - Assumption: the more terms in common between a document and a query, the more likely it is that the document is relevant to the query
  - However, it does not consider frequency of occurrence
- Could use the frequency of occurrence as a weight

$$\text{Score} = \sum_{t \in |q \cap d|} TF_{t,d}$$

- $TF_{t,d}$: the frequency of occurrence of a term *t* in document *d*

Gerard. Salton. *Automatic Information Organization and Retrieval.* McGraw Hill Text, 1968.

Term weighting

$$\sum_{t \in q \cap d} TF_{t,d}$$

$$\sum_{i=1}^{V} d_i \times q_i$$

Term weighting

$$\sum_{t \in |q \cap d|} TF_{t,d} \equiv \sum_{t=1}^{V} d_t \times q_t$$

If d_t is the frequency of term t, i.e. no longer binary.

Term frequency distribution

- *Zipf's law*: a commonly-used model of the distribution of terms in a collection
- It states that the ith most frequent term has frequency proportional to 1/i:
  $cf_i \propto \frac{1}{i}$, cf_i is the number of occurrences of the term in the collection.
- about half of all vocabulary terms occur only once in the collection.
- *Zipf's law* is an example of a power law distribution.

What's wrong with term weighting?

- All terms are treated equally
- Common words therefore have more weight
- What do we mean by "common"?

Fudgel

Fudgel:

‘Pretending to work when you’re not actually doing anything at all’.

Query:

‘What does fudgel mean”

Document ranking?

Inverse Document Frequency (IDF)

- In 1972, Spärck Jones introduced a measure of term specificity (discriminative power) called Inverse Document Frequency (IDF)
  - Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. 1972

Inverse Document Frequency

- Query terms are different in their ability to discriminate documents.
- A query term is not a good discriminator if it occurs in many documents.
- We should give it less weight than the one occurring in few documents.

TF·IDF Term Weighting

$$IDF_t = \log_{10}(N/n_t)$$

N = number of documents in collection

n_t = number of documents in which term t appears

IDF example, suppose N = 1 million

term	n_t	IDF_t
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

$$IDF_t = \log_{10}(N/n_t)$$

There is one IDF value for each term t in a collection.

Effect of IDF on ranking

- IDF has no effect on ranking one term queries, like
  - iPhone
- However, IDF affects the ranking of documents for queries with at least two terms
  - For the query “essex council”, IDF weighting makes occurrences of “essex” count for much more in the final document ranking than occurrences of “council”.

tf.idf weighting

$$tf_t \times idf_t$$

greater when the term is **frequent** in in the document

greater when the term is **rare** in the collection (does not appear in many documents)

TF·IDF weighting has many variants

Term frequency	Document frequency	Normalisation
n (natural):$\text{tf}_{t,d}$	n (no): 1	n (none):1
l (log.):$1+\log(\text{tf}_{t,d})$	t (idf):$\log \frac{N}{\text{df}_t}$	c (cosine): $\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (aug.):$0.5+\frac{0.5\times\text{tf}_{t,d}}{\max_t(\text{tf}_{t,d})}$	p (prob idf) : $\max\{0, \log\frac{N-df_t}{df_t}\}$	u (pivoted unique) : $1/u$
b (bool.) : $\begin{cases}1 & \text{if } tf_{t,d}>0\\ 0 & \text{otherwise}\end{cases}$		b (byte size): $(1/CharLength^{\alpha}, \alpha<1)$
L (log ave) : $\frac{1+\log(tf_{t,d})}{1+\log(ave_{t\in d}(tf_{t,d}))}$		

Similarity score

$$\text{Score}(\text{q}, \text{d}) = \sum_{t \in (q \cap d)} tf_t \times idf_t = \sum_{t=1}^{V} d_t \times q_t$$

$$d_t = tf_{d,t}$$

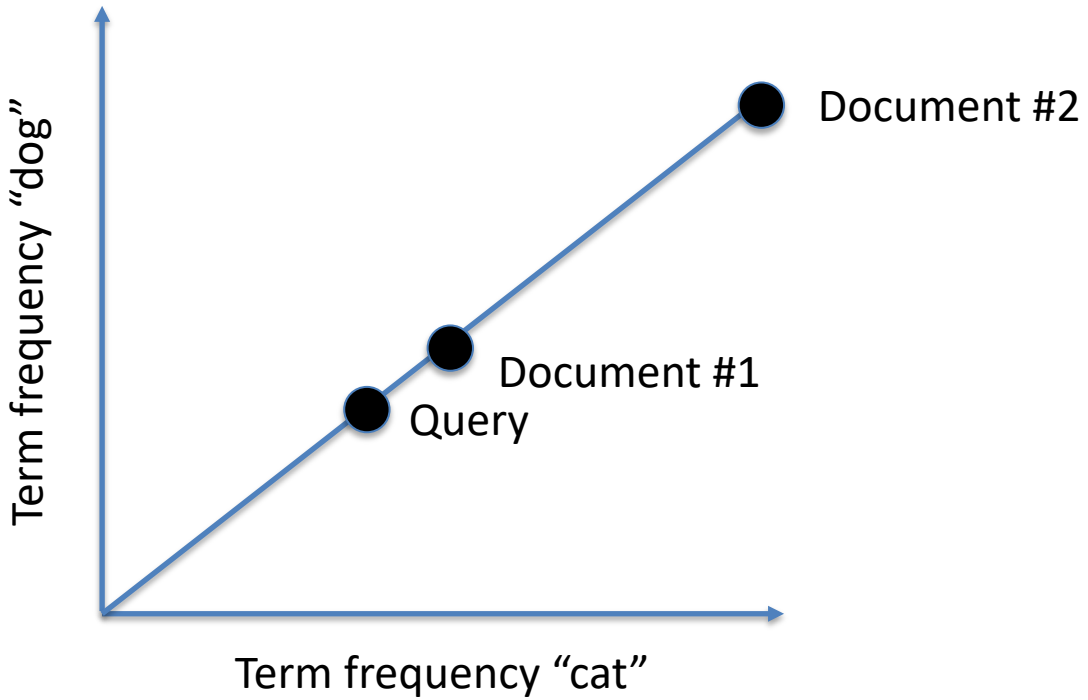
$$q_t = idf_t$$

Problems with inner product of (non-binary) vectors

- Take a document *d* and append it to itself. Call this document *d′*.
- "Semantically" d and d′ have the same content
- However, the distance between the two documents can be quite large
- Biased – longer documents are likely to score higher

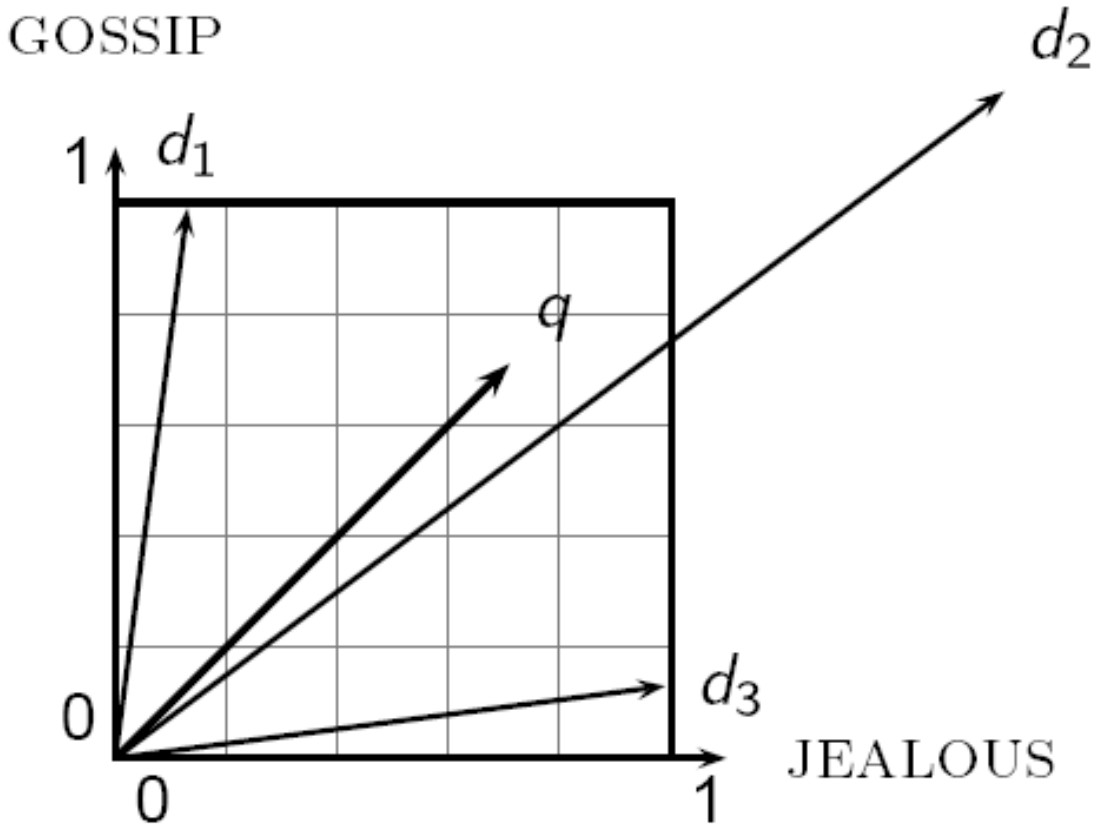
Formalizing vector space proximity

- We need to come up with a distance between two points
- Euclidean distance?
- Euclidean distance is a bad idea . . .
- . . . because Euclidean distance is large for vectors of different lengths.


Term frequency “dog”
Document #2
Document #1
Query
Term frequency “cat”

Why distance is a bad idea

The Euclidean distance between q and d_2 is large even though the distribution of terms in the query q and the distribution of terms in the document d_2 are very similar.



Cosine similarity

The numerator is the inner product

The denominator is the product of the two vector-lengths

Ranges from 0 to 1 (equals 1 if the vectors are identical)

Cosine similarity

Inner product

$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\sqrt{\sum_{i=1}^{V} x_i^2} \times \sqrt{\sum_{i=1}^{V} y_i^2}}$$

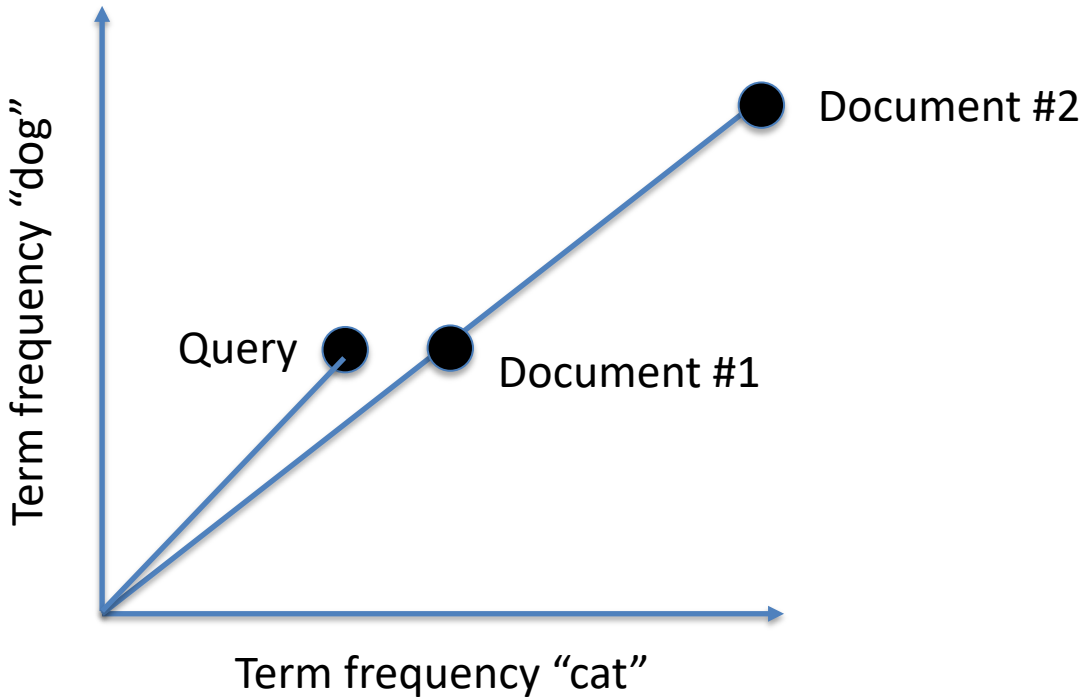
Length of x

Length of y

Cosine similarity

So called because it measures the cosine of the angle between two vectors.

Cosine similarity



Cosine similarity the same between Query and Doc#1, and Query and Doc#2

Cosine similarity

$$\frac{q.d}{|q||d|} = \frac{\sum_{t \in (q \cap d)} q_t \times d_t}{\sqrt{\sum_{t=1}^{V} q_t^2} \sqrt{\sum_{t=1}^{V} d_t^2}}$$

References and Further Readings

- Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. SIGIR '96
- Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. 1972
- Cyril W. Cleverdon. Aslib cranfield research project: report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. *Cranfield Library*, 1962.
- Gerard Salton, Edward A. Fox, Harry Wu Extended Boolean information retrieval Communications of