ifis

Institut für Informationssysteme
Technische Universität Braunschweig

Information Retrieval and Web Search Engines

Wolf-Tilo Balke
Muhammad Usman

Institut für Informationssysteme
Technische Universität Braunschweig

Previous Lecture

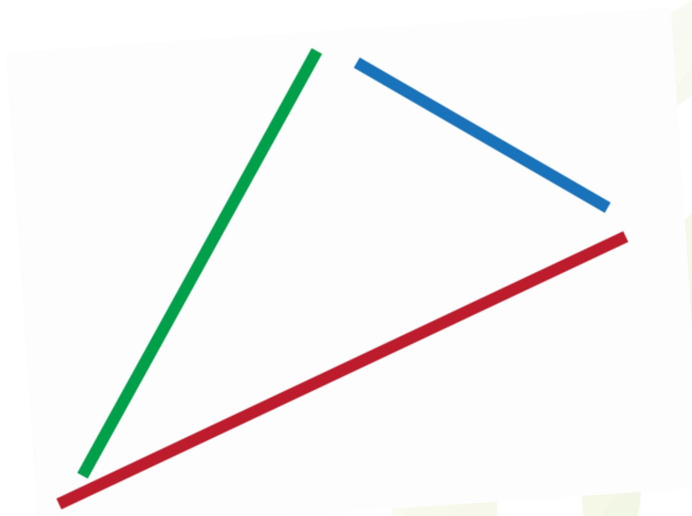
- Boolean retrieval:
  - **Documents:** Sets of words (index terms)
  - **Queries:** Propositional formulas
  - **Result:** The set of documents satisfying the query formula
  - **Example:**
    Document_1 = {step, mankind, man}
    Document_2 = {step, China, taikonaut}
    Document_3 = {step, China, mountaineer}

    Query = "step AND ((China AND taikonaut) OR man)"

    Result = {Document_1, Document_2}

Today's Lecture

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



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!
  - No binary set membership, but **graded membership!**
  - **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

The class "tall person"

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:**

$$\mu(A \wedge B) = \min\{\mu(A), \mu(B)\}$$

  - **Disjunction:**

$$\mu(A \vee B) = \max\{\mu(A), \mu(B)\}$$

  - **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:

$\text{Document}_1 = \{\text{step}/0.4, \text{China}/0.4\}$
$\text{Document}_2 = \{\text{step}/0.3, \text{China}/1\}$

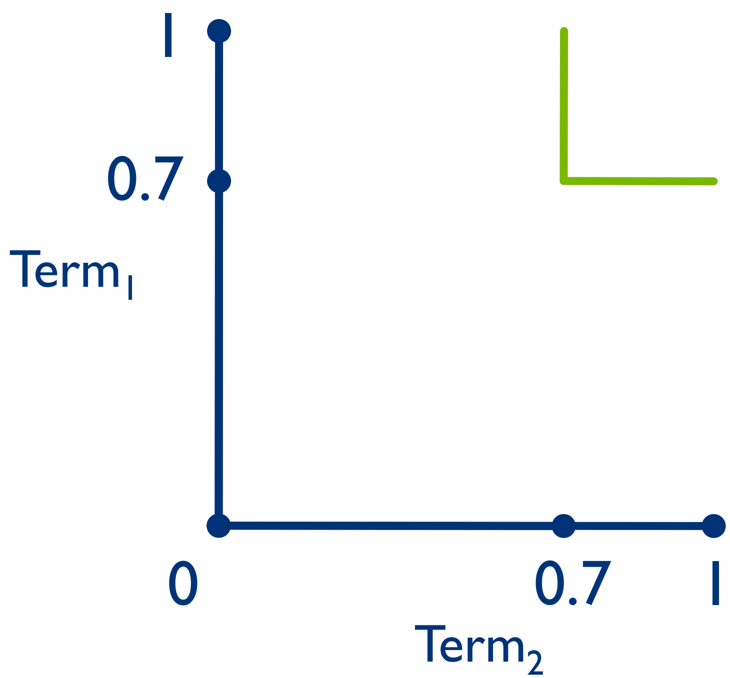
Query = “step AND China”

$\text{Result} = \{ \text{Document}_1/0.4, \text{Document}_2/0.3 \}$

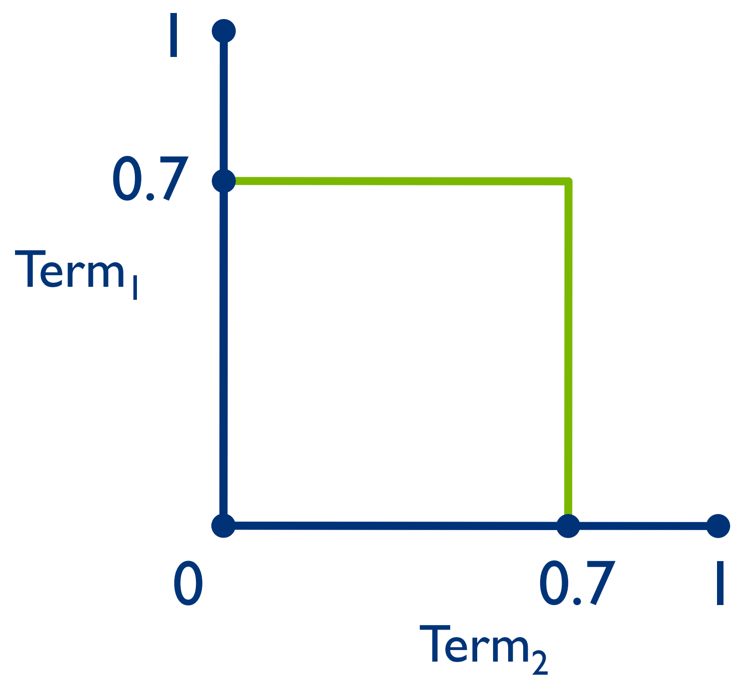
Intuitive?

- All documents lying on the green line
  are satisfying the query equally well (degree 0.7):

Query = "$Term_1$ AND $Term_2$"



Query = "$Term_1$ OR $Term_2$"



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**

Fuzzy Index Terms

- **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 ≤ 1)

{step, China, mountaineer} → {step/1, China/1, mountaineer/1, alpinist/0.8, Asia/0.4}

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

- **Jaccard index:**
  - Measures which terms **co-occur** in the document collection
  - The Jaccard index $c(t, u)$ of the term pair (t, u) is

$$\frac{\#\text{documents containing } \textbf{both} \text{ term } t \text{ and term } u}{\#\text{documents containing } \textbf{at least one} \text{ of term } t \text{ and term } u}$$

  - 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

Fuzzy Index Terms

- **Jaccard index:**
  - Document$_1$ = {step, man, mankind}
  - Document$_2$ = {step, man, China}
  - Document$_3$ = {step, mankind}

c(*t*, *u*)	step	man	mankind	China
step	1	0.67	0.67	0.33
man		1	0.33	0.5
mankind			1	0
China				1

$$\frac{\text{\#documents containing } \textbf{both} \text{ term } t \text{ and term } u}{\text{\#documents containing } \textbf{at least one} \text{ of term } t \text{ and term } u}$$

Fuzzy Index Terms

- Ogawa *et al.* (1991) compute the fuzzy index terms as follows:
  - The **fuzzy membership degree** of term t with respect to document D (represented as crisp set of terms) is

$$W(D, t) = 1 - \prod_{u \in D} (1 - c(t, u))$$

  - $1 - c(t, u)$ is the fraction of documents containing one of term t and term u but not both
  - **$t \in D$ implies $W(D, t) = 1$**
  - **Idea:** Give terms a high fuzzy membership degree that usually occur together with the other document terms; those terms will capture the document's topic best

Example

- Document_1 = {step, man, mankind}
- Document_2 = {step, man, China}
- Document_3 = {step, mankind}

$W(D, t)$	step	man	mankind	China
Document_1	1	1	1	0.67
Document_2	1	1	0.78	1
Document_3	1	0.78	1	0.33

$c(t, u)$	step	man	mankind	China
step	1	0.67	0.67	0.33
man		1	0.33	0.5
mankind			1	0
China				1

$$W(D, t) = 1 - \prod_{u \in D} (1 - c(t, u))$$

Fuzzy Retrieval Model

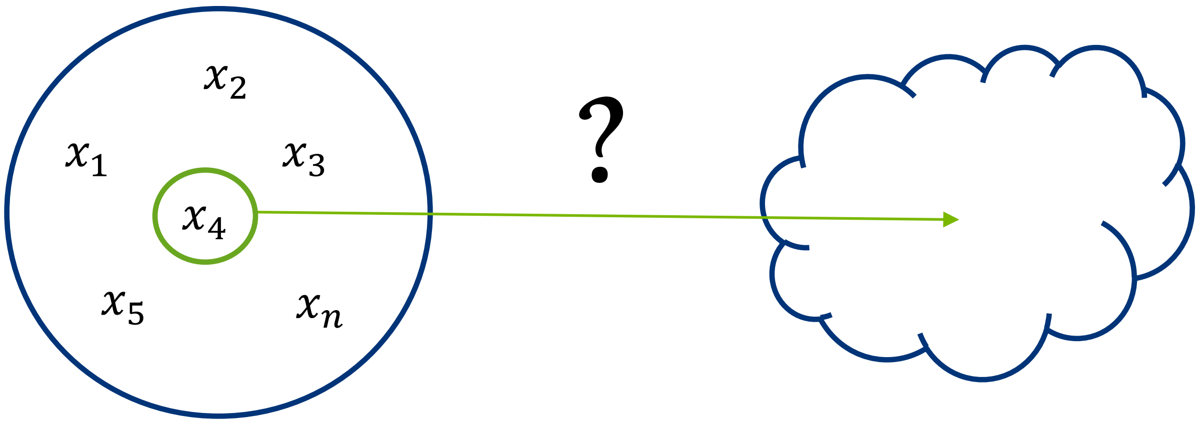
- Cons:
  - Computation of fuzzy membership weights usually is difficult
    - Main problem: All weights must be within [0, 1]
  - Lack of intuitive query processing
    - But: There are many other ways to define fuzzy conjunction and disjunction (using **t-norms** and **t-conorms**)

- 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

Fuzzy Logic's degrees of truth

Detour

- Fuzzy Logic is all about degrees of truth



Crisp set X= group of people

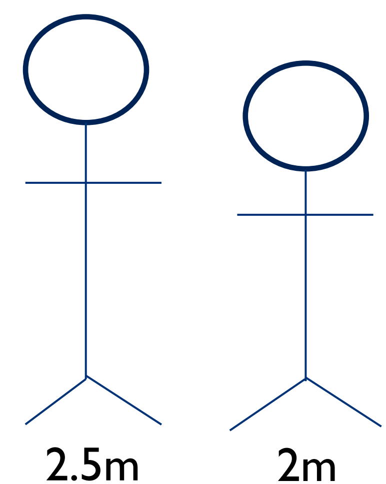
Fuzzy subset F = group of tall people
Where F ⊆ X

- Degree of truth is absolutely true (1), absolutely false(0), or some intermediate truth

Fuzzy logic vs Probability

Detour

- Lotfi Zadeh argues that fuzzy logic is different from probability theory
- Zadeh defines **Possibility theory**; Fuzzy alternative to Probability
- Fuzzy logic and probability refer to different kinds of uncertainty
  - Fuzzy logic: deals with imprecision of facts and produce fuzzy statements (e.g. rather tall)
  - Probability theory: deals with chances of something happening, but produces precise statements (e.g. tall, not tall)



Fuzzy logic vs Probability

Detour

- Possibility is different from probability!
- Zadeh's own example:

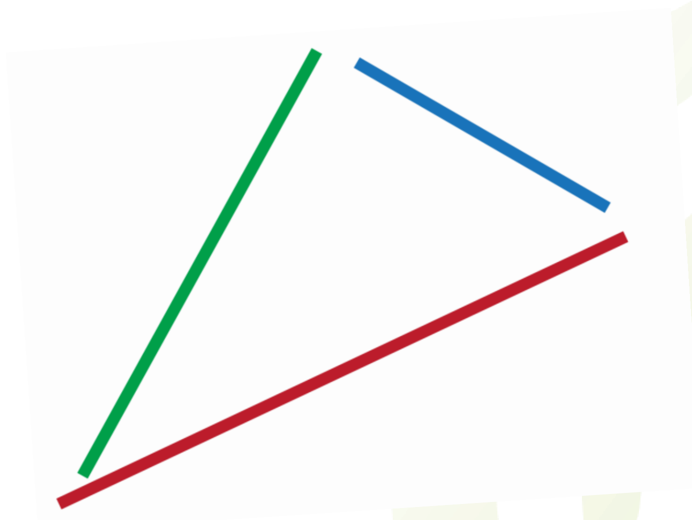
"Hans ate *X* eggs for breakfast"

X	1	2	3	4	5	6	7	8
Possibility	1	1	1	1	0.8	0.6	0.4	0.2
Probability	0.1	0.8	0.1	0	0	0	0	0

The possibility of an event doesn't mean its probablility.

Today's Lecture

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



Bag-of-Words Queries

- 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

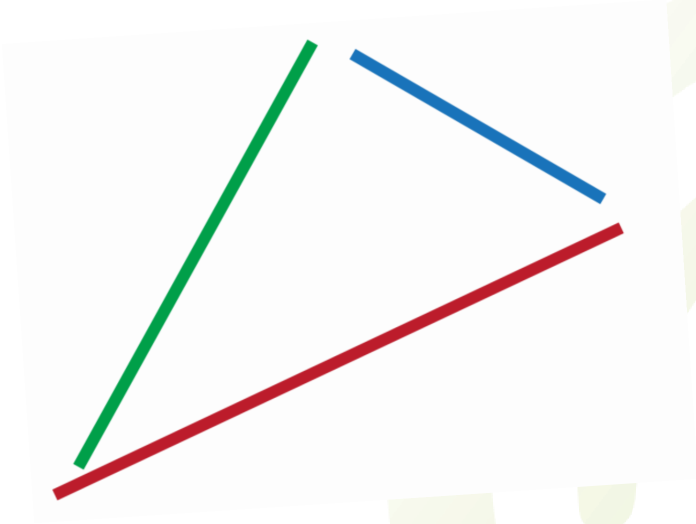
- **Coordination level matching (CLM)** is
  a straightforward approach to bag-of-words queries
  - **Idea:** Documents whose index records have n different terms in common with the query are more relevant than documents with $n - 1$ different terms held in common
- The **coordination level** (also called "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

- Document_1 = {step, man, mankind}
  Document_2 = {step, man, China}
  Document_3 = {step, mankind}
- Query_1 = {man, mankind}
  Result:
  1. Document_1 (2)
  2. Document_2, Document_3 (1)
- Query_2 = {China, man, mankind}
  Result:
  1. Document_1, Document_2 (2)
  2. Document_3 (1)

Today's Lecture

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




Information Spaces

- **Spatial structure of libraries:**
  Topically related books
  are standing side by side

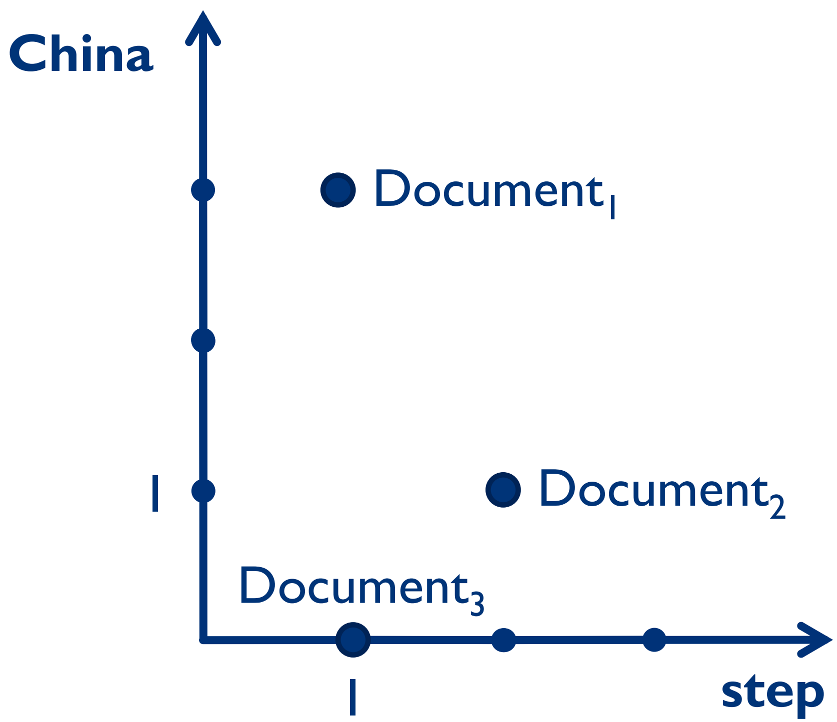
- Can we transfer this principle
  to information retrieval?
- **Idea:**
  Represent documents and queries
  as points in an **abstract semantic space**
  - Measure **similarity** by **proximity**

Vector Space Model

- The **vector space model** was proposed by Gerard Salton (Salton, 1975)
- Documents and queries are represented as point 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**

Example

- $Document_1 = \{step, China/3\}$
  $Document_2 = \{step/2, China\}$
  $Document_3 = \{step\}$



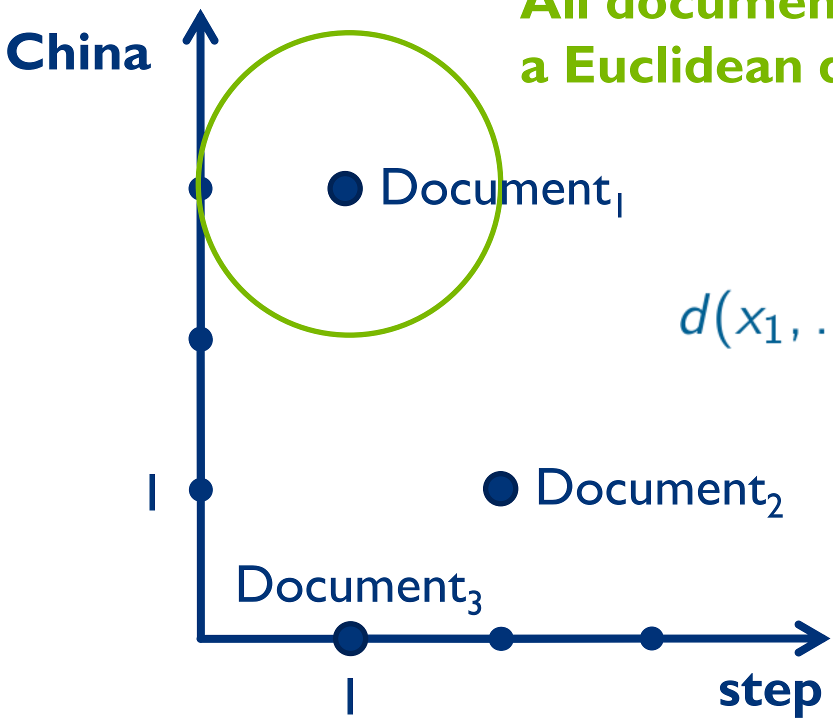
Distance and Similarity

- **How to define similarity/proximity?**
- A **metric** on a set X is a function $d : X \times X \to \mathbb{R}$ 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**

$$d(x_1, \ldots, x_n; y_1, \ldots, y_n) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

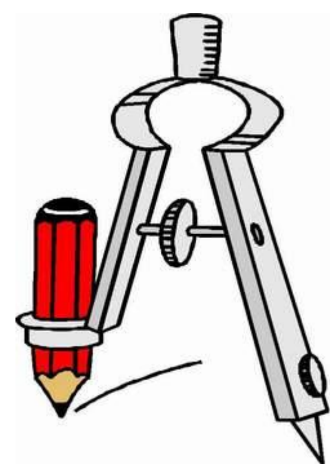
Euclidean Distance

- **Geometric meaning** of Euclidean distance:



All documents on the circle have a Euclidean distance of 1 from Document1

$$d(x_1, \ldots, x_n; y_1, \ldots, y_n) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$



Similarity

- A **similarity measure** on a set X is a function $s : X \times X \rightarrow [0, 1]$ 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

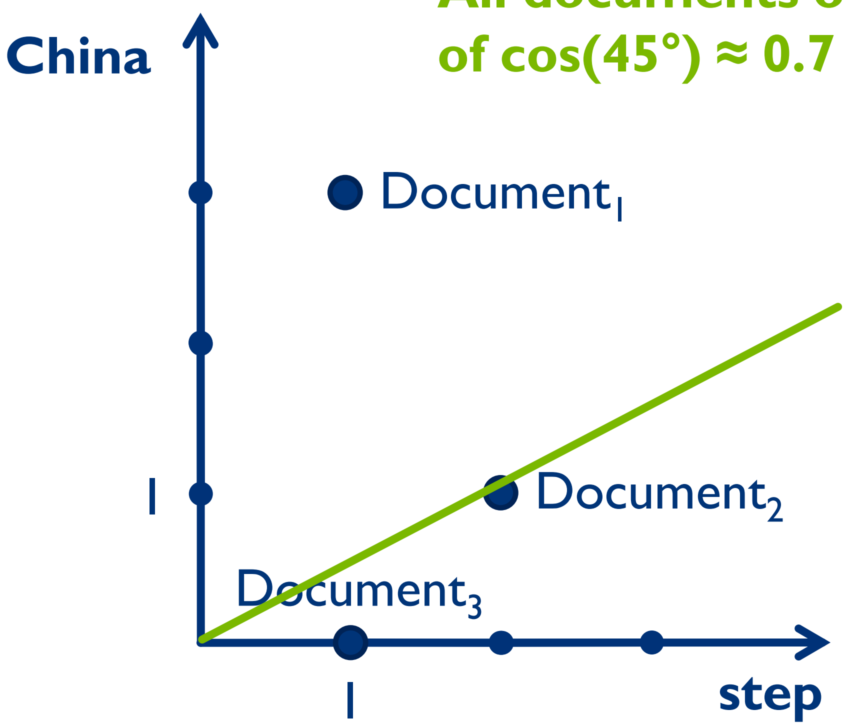
$$s(x, y) = \cos(\alpha)$$

  - α 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°) ≈ 0.71 to Document$_1$



$s(x, y) = \cos(\alpha)$

Cosine Similarity

- How to **compute the angle α** between two vectors?

$$\cos(\alpha) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

- "·" denotes the **dot product** (aka scalar product), i.e.

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

- "‖·‖" denotes the **Euclidean norm** (aka ℓ^2-norm), i.e.

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

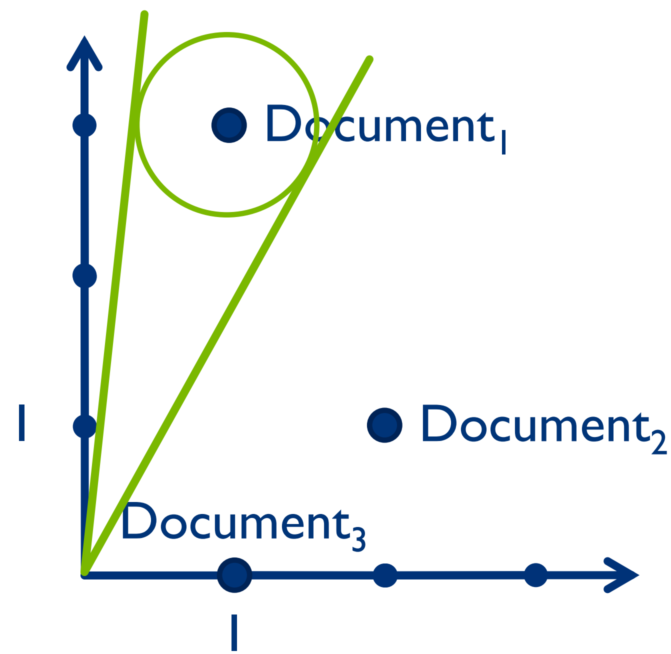
Recap: Coordination Level Matching

- 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*

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

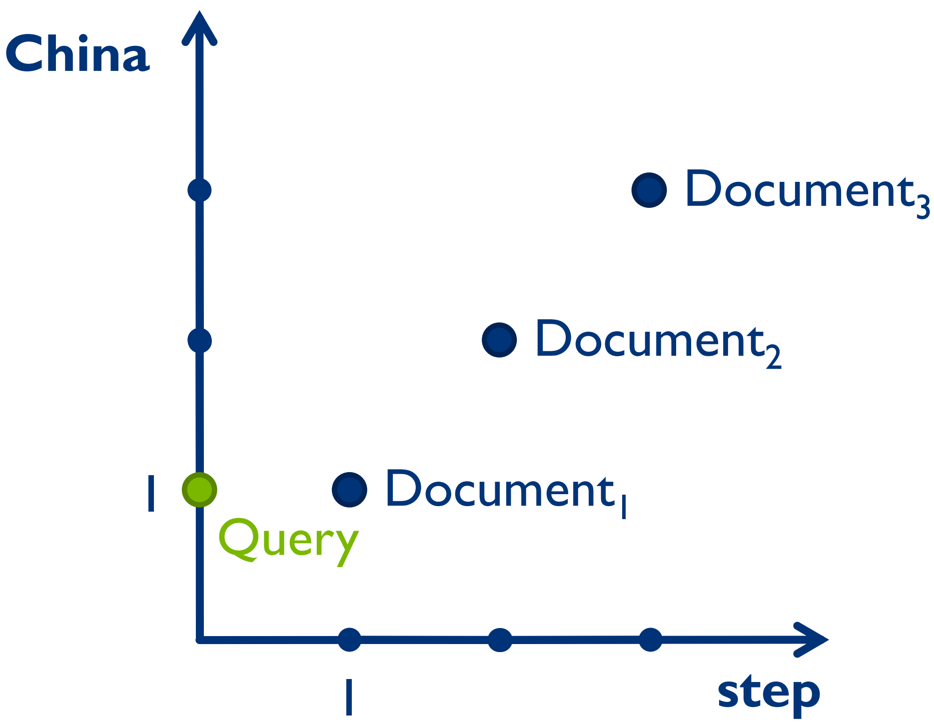
The "Right" Measure

- **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

- 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:

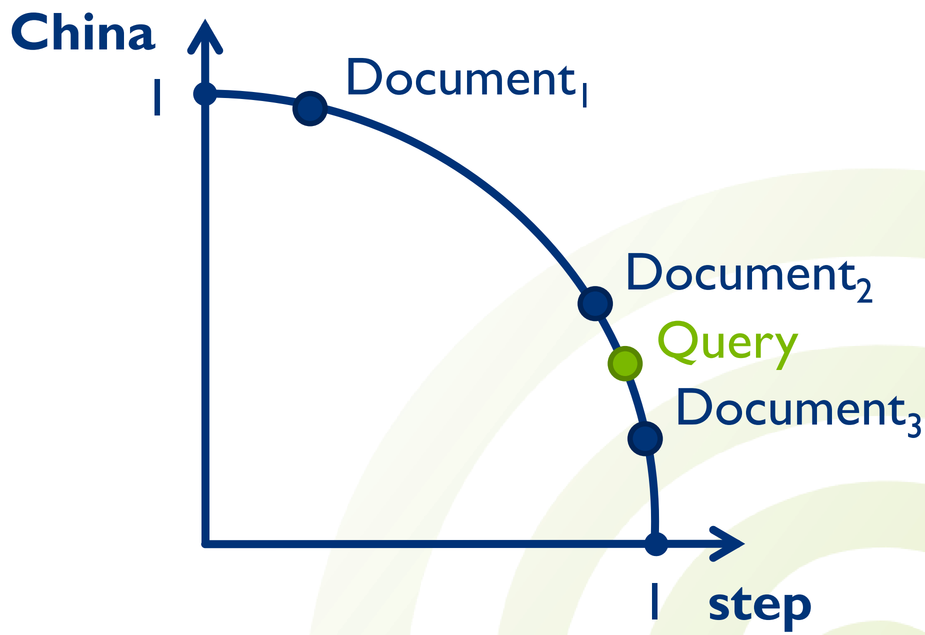
$$\frac{x}{\max_{i=1}^{n} x_i}$$

  - Divide each coordinate by the sum the vector's coordinates:

$$\frac{x}{\sum_{i=1}^{n} x_i}$$

Normalization

- 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

- 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

- **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 **disc(t)**
  - There are many ways to formalize discriminative power …
- General term weighting framework:
  - Higher term frequency $\Rightarrow$ Higher term weight
  - Higher discriminative power $\Rightarrow$ Higher term weight
- **Term weight should be proportional to**

$$\mathbf{tf}(d, t) \cdot \mathbf{disc}(t)$$

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

- 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:

$$\mathrm{tf}(d, t) \cdot \frac{1}{\mathrm{df}(t)}$$

  - "IDF" = **"inverse document frequency"**

TF-IDF

- 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):

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

  - *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

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:

$$s_{\text{avg}} = \frac{1}{N^2} \sum_{d, e \in C} s(d, e)$$

  - Define $s_{\text{avg}, t}$ 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

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

Term Discrimination

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

- **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

Collection	MEDLARS	ISI	INSPEC	CACM
#documents	1033	1460	12684	3204
#queries	30	35	77	52
Boolean	0.21	0.11	0.12	0.18
Fuzzy	0.24	0.10	0.13	0.16
Vector space	0.55	0.16	0.23	0.30

  – 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 lectures…)
  - 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

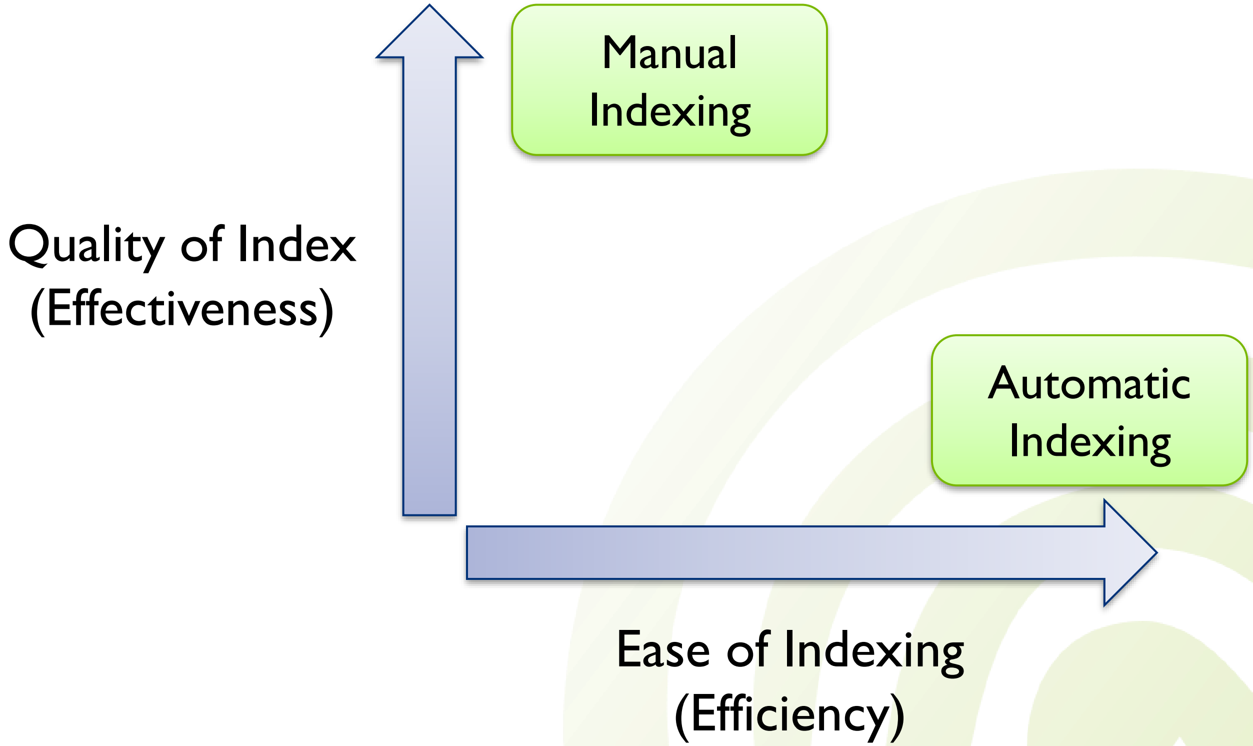
Detour

- 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

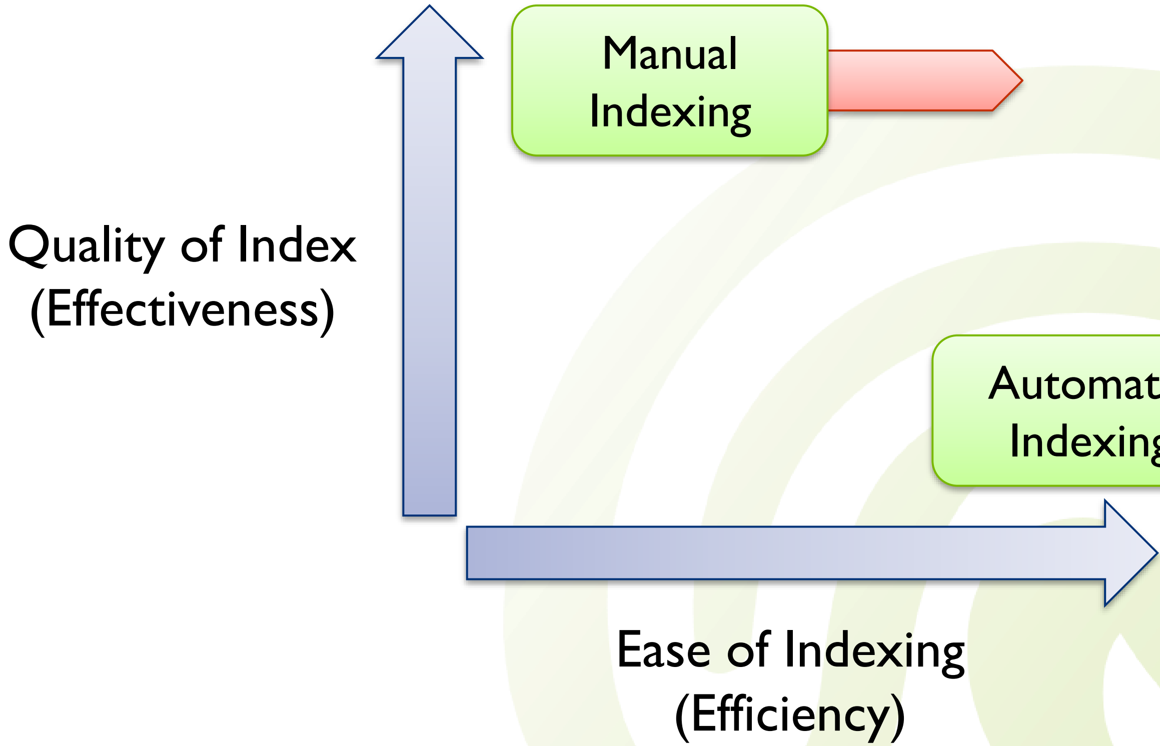
- The situation around 1960:



Manual vs. Automatic Indexing

Detour

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



Manual vs. Automatic Indexing

Detour

- 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**

Manual vs. Automatic Indexing

Detour

Cyril Cleverdon
(1914–1997)

"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

Detour

- **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

Detour

- Early hardware: **IBM 7094**

- “A basic machine operating cycle of 2 microseconds”

SMART

Detour

- 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
```

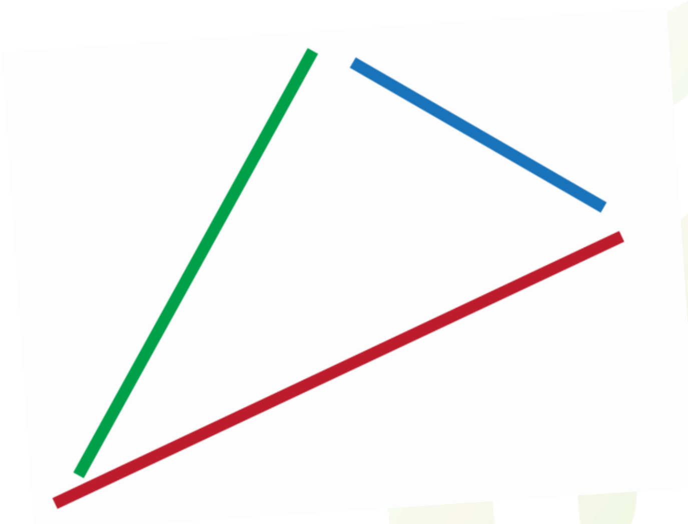
SMART

Detour

- Early versions of SMART have been evaluated on many **test collections:**
  - ADI: 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

Today's Lecture

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

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

Winning:
At least 9 in total
or second roll is 1

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Probability

- What's the probability of rolling at least 9 in total?

  Answer: 10/36 ≈ 0.28

- What's the probability of getting 1 in the second roll?

  Answer: 1/6 ≈ 0.17

- What's the probability of winning?

  Answer: 16/36 ≈ 0.44

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

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*

$$\Pr(A|B) = \frac{\Pr(A \text{ and } B)}{\Pr(B)}$$

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

- 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 / 16/36 = 3/16 ≈ 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(win | 4 in first roll) = 1/2
  - Pr(win) = 16/36
  - Pr(4 in first roll) = 1/6

  Answer: (1/6 / 16/36) · 1/2 = 3/16 ≈ 0.19

Bayes' Theorem

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

- $\Pr(A)$ 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

- Probabilistic retrieval models