Information Retrieval and Web Search Engines

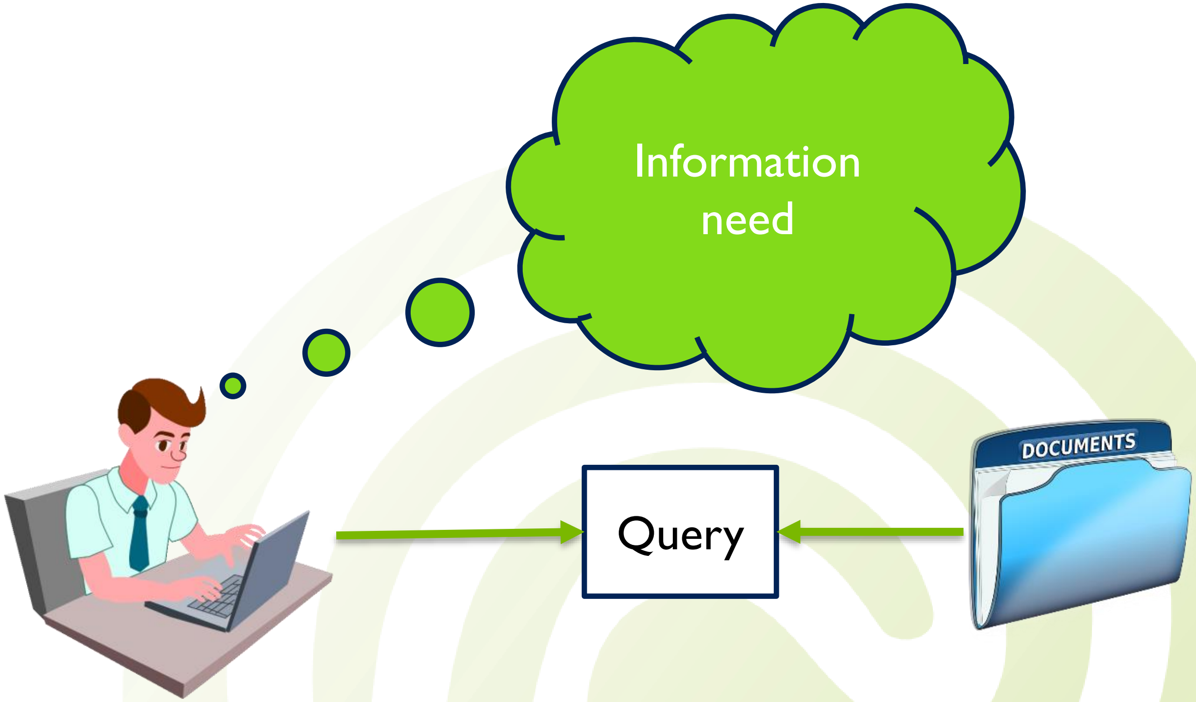
Lecture 6: Language Models and Evaluation

Wolf-Tilo Balke
Muhammad Usman

Institut für Informationssysteme
Technische Universität Braunschweig

Language Models and Evaluation

1. **Language models**
2. Evaluation of retrieval quality



Topics and Languages

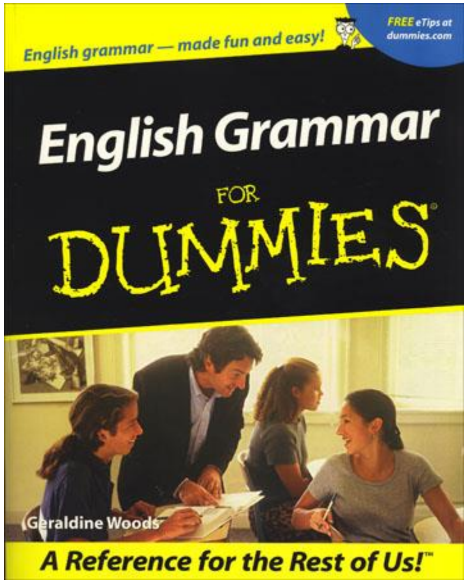
- **Observation:** There are many **different styles of writing,** especially depending on topics
  - For example, political news articles use a completely **different vocabulary** than personal blog entries
- **Idea in IR:**
  - Equate **“languages”** and fine-grained(!) **topics**
    - Each topic corresponds to a specific language
  - Represent each document by its corresponding language model (different parameters)
  - **Querying** then becomes:
    To which document’s language model the query fits best?
- There are **models** available to describe such “languages”… How to define a Model’s Language?

1. Formal Grammars

- How to describe “language” within a formal model?
  - Well-known from theoretical computer science: **Formal grammars**
  - A way to describe correct syntax
  - Example:
    - **sentence** → **noun_phrase verb_phrase**
    - **verb_phrase** → **verb noun_phrase**
    - **verb** → took
    - **noun_phrase** → the man
    - **noun_phrase** → the book

1. Formal Grammars

- Why formal grammars will not help us:
  - Grammars capture **syntactical correctness** but not **style**
  - **Natural language** does not strictly obey grammar rules
  - The **writing style** or **topic** of a document largely depends on how **typical** words, phrases, or sentences look like
  - Formal grammars fail to capture **statistical properties** of text, they just describe the set of "correct" documents


English grammar — made fun and easy!
FREE eTips at dummies.com
English Grammar
FOR
DUMMIES
Geraldine Woods
A Reference for the Rest of Us!

2. Statistical Language Models

- A different approach to modeling language are **statistical language models:**
  - **Ignore syntactical rules and grammar**
  - Focus on **statistical regularities** in the generation of language
- A **generative model** is used here:
  - Assumption:
    Every document is the result of a **random process**
  - Central quantity: **Pr(w_1, …, w_n),** the probability of generating a document containing the words w_1, …, w_n (in this order)

2. Statistical Language Models

- A statistical language model consists of **probability distributions:**
  - For any given n, there is a probability distribution such that every document $w_1, \ldots, w_n$ of length n (word count) gets assigned its **probability of generation** $\Pr(w_1, \ldots, w_n)$
- **Example:**
  - Assume that only the words "cat" and "dog" are generated

***n* = 0:**

doc	Pr(doc)
()	1

***n* = 1:**

doc	Pr(doc)
(cat)	0.3
(dog)	0.7

***n* = 2:**

doc	Pr(doc)
(cat, cat)	0.1
(cat, dog)	0
(dog, cat)	0.7
(dog, dog)	0.2

…

Building probabilities over term seq.

- Usually, some **structure** is assumed
- **Unigram** model (assume independence, ignore context):

$$\Pr(w_1, \ldots, w_n) = \Pr(w_1) \cdot \Pr(w_2) \cdot \cdots \cdot \Pr(w_n)$$

- **Bigram** model (assume dependence on the previous word only):

$$\begin{aligned}&\Pr(w_1, \ldots, w_n)\\ &= \Pr(w_1) \cdot \Pr(w_2 | w_1) \cdot \Pr(w_3 | w_2) \cdot \cdots \cdot \Pr(w_n | w_{n-1})\end{aligned}$$

- **Trigram** model (assume dependence on the previous two words):

$$\begin{aligned}&\Pr(w_1, \ldots, w_n)\\ &= \Pr(w_1) \cdot \Pr(w_2 | w_1) \cdot \Pr(w_3 | w_1, w_2) \cdot \Pr(w_4 | w_2, w_3) \cdot \cdots \cdot \Pr(w_n | w_{n-2}, w_{n-1})\end{aligned}$$

Bigram Model

Example of a three-word bigram model:

word	Pr(*word*)
cat	0.4
dog	0.5
mouse	0.1

Pr(*row* | *column*)	cat	dog	mouse
cat	0	0.1	1
dog	0.2	0.4	0
mouse	0.8	0.5	0

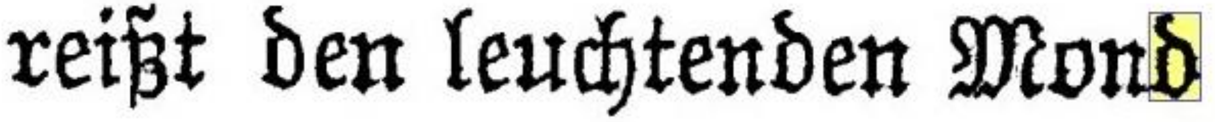
Some **randomly generated** 6-word sentences:

- dog mouse cat mouse cat mouse
- dog dog dog mouse cat mouse
- dog mouse cat mouse cat mouse
- cat mouse cat dog mouse cat
- cat mouse cat mouse cat mouse

Statistical Language Models

- **Observation:** Generative models can be used to…
  - **generate** documents, or
  - **recognize** documents

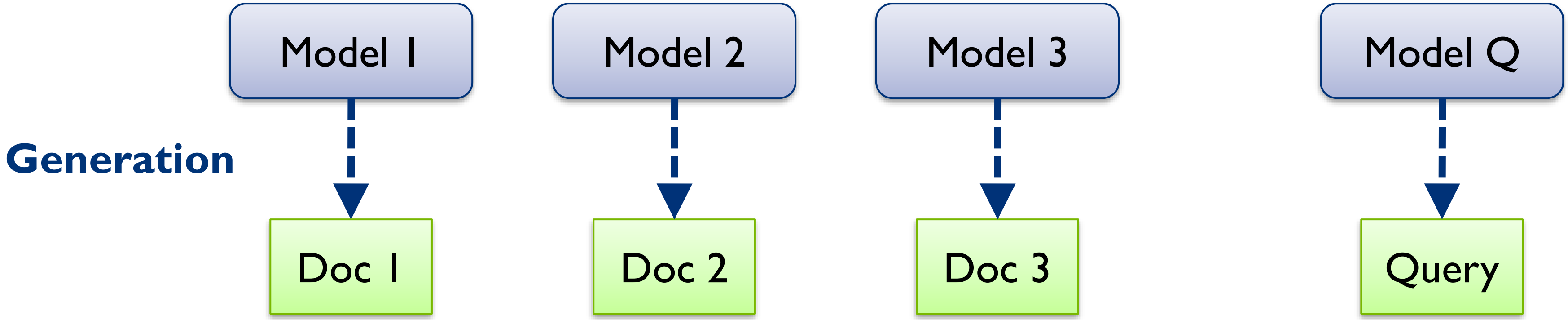
- Document **recognition:**
  - "Which document fits a given model best?"
  - Usually based on **probabilities of generation**
  - Popular applications: OCR, speech recognition, …



reißt den leuchtenden Mond

Language Models in IR

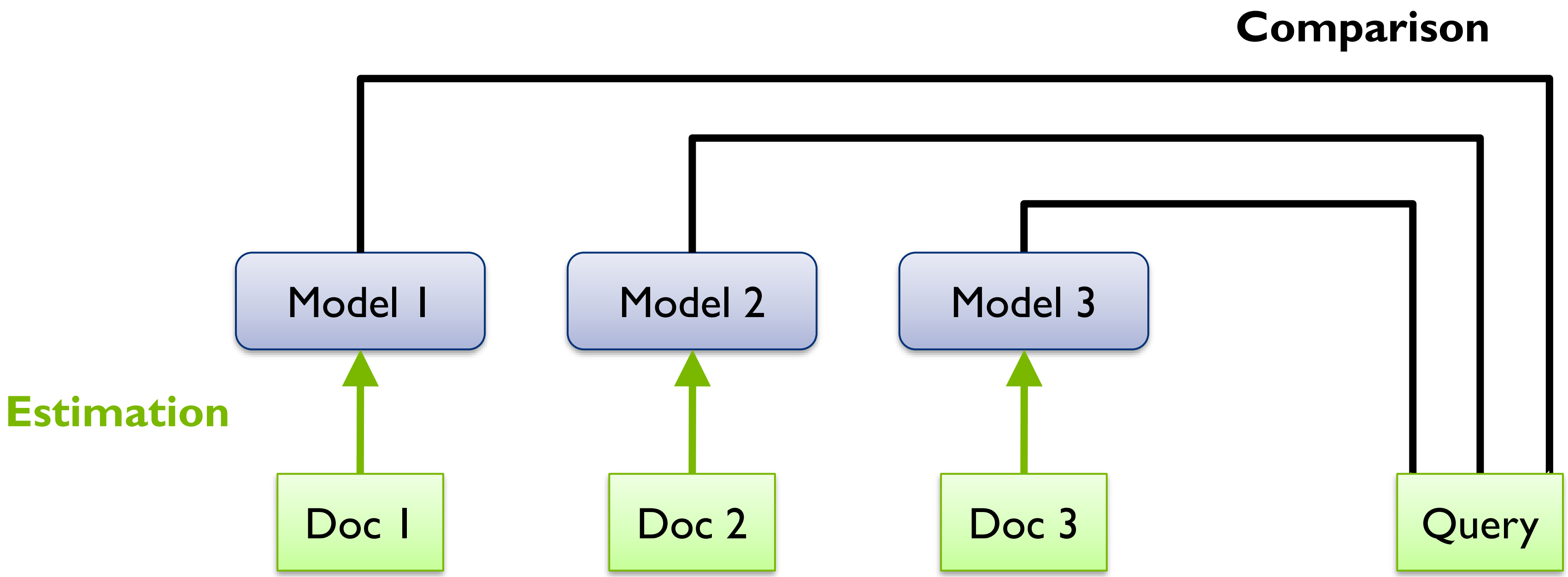
- How to apply language models in information retrieval?
- **Assumptions:**
  - For each document,
    there is a **"true"** (but unknown) statistical **document model**
  - Each document was generated from its corresponding model by a **random generation process,** i.e., it is a random **sample**
  - The **query** also is a **sample** or a description of an underlying language model describing the user's information need



Language Models in IR

Typical application of language models in IR:

1. **Estimate a model** for each document
2. For each estimated model,
   compute the **probability of generating the query**
3. **Rank** documents by these probabilities



Language Models – Open problems

Problem 1: Which language model should we use (unigram, bigram, …)?

- For **practical reasons,** unigram models are used in IR (sometimes bigram models)
- "Practical reasons" refers to:
  - Reduced **computational complexity**
  - Problem of **sparse data:**
    Documents usually are short and its size and content are fixed
  - **Losses from data sparseness** (i.e., bad estimations) tend to outweigh any gains from richer models

Language Models - Open problems

Problem 2: How to **estimate** the “true” language models from the observations (= documents) we have?

- Straightforward approach:
  - **Given:** Document $d = (w_1, \ldots, w_n)$
  - Estimate $\Pr(w_i)$ by $\frac{\mathrm{tf}(d, w_i)}{n}$
  - This is the **maximum likelihood estimator (MLE)**
- **Example:**
  - d = (the, big, dog, jumps, over, the, small, dog)
  - Estimate Pr(dog) by 2 / 8 = 0.25
  - Estimate Pr(cat) by 0

MLE Approach drawback

- **Document size often is too small**
  - Many terms would be missing in a doc, which implies a zero probability estimate
  - Probability of terms occurring once in the document normally is overestimated, because this occurrence was partly by chance

- **Solution:** Smoothing
  - Allocate some probability mass to missing terms
  - Pull all estimates in the direction of the collection mean
  - There are many ways to do this

Smoothing – First Approach

1. Simple smoothing (as used in TF-IDF):

- Add some small number α (e.g. 1 or 0.5) to all observed counts
- Renormalize to give a probability distribution

- Example (use α = 1):
  - d = (the, big, dog, jumps, over, the, small, dog)

word	initial estimate
the	3 / 8
big	2 / 8
dog	3 / 8
jumps	2 / 8
over	2 / 8
small	2 / 8
cat	1 / 8

normalize
(divide by 15/8)

word	final estimate
the	3 / 15
big	2 / 15
dog	3 / 15
jumps	2 / 15
over	2 / 15
small	2 / 15
cat	1 / 15

Smoothing – Second Approach

2. Linear smoothing:

- Estimate $\Pr(w_i)$ by

$$\lambda \frac{\mathrm{tf}(d, w_i)}{n} + (1 - \lambda) \frac{\mathrm{cf}(w_i)}{N}$$

- n: document size
- $\mathrm{cf}(w_i)$: collection frequency of w_i,
  i.e., the number of occurrences of w_i in the whole collection
- N: collection size, i.e., number of words in the whole collection
- λ: some parameter between 0 and 1

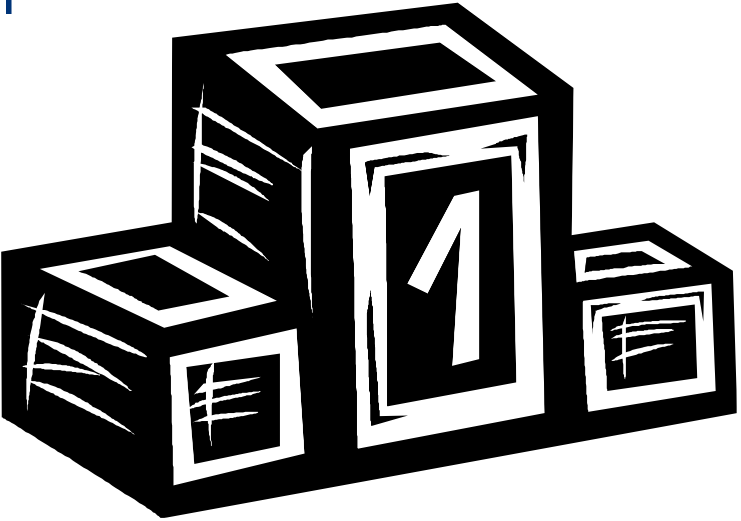
Smoothing – Third Approach

3. Approach by Ponte and Croft (1998):
   - Use corpus data to stabilize document model
   - If a term is missing in the document:
     Estimate its probability by its corpus probability
     (i.e., use an MLE over the whole collection)
   - If a term appears in the document:
     Smooth MLE using average MLE
     (over all documents containing the term)

- There are many more advanced smoothing methods…

Ranking

- How to compare a document model to the query?
  - Compute the query's generation probability with respect to the model
  - Given: Query $q = (q_1, \ldots, q_k)$
  - The score of a document then is our estimation of $\Pr(q_1, \ldots, q_k) = \Pr(q_1) \cdot \cdots \cdot \Pr(q_k)$ with respect to the document's language model



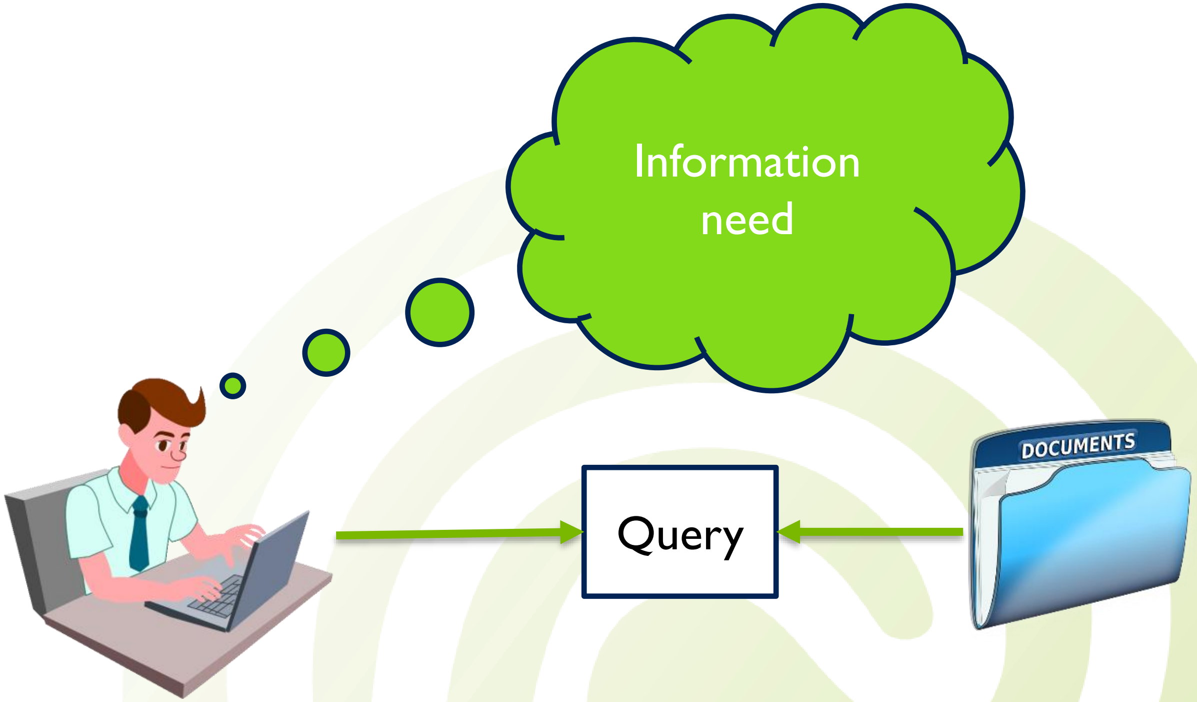
Pros and Cons of Language Models

- **Pros:**
  - Clear statistical foundation, no ad hoc weightings
  - Collection statistics are an integral part of the model, rather than being used heuristically
  - Works good, comparable to the vector space model

- **Cons:**
  - Independence assumption in unigram model
  - No explicit notion of relevance, integration of user feedback is difficult

Language Models and Evaluation

1. Language models
2. **Evaluation of retrieval quality**



What to Evaluate?

What should be evaluated in IR?

1. Efficiency
   - Use of system resources
   - Scalability

2. Effectiveness
   - Result quality
   - Usability

Efficiency

- Efficiency:
  - Storage space
  - CPU time
  - Number of I/O operations
  - Response time
  - …
- Depends on hardware and software
- Goal in IR: **“be efficient enough”**
- Efficiency usually is **easy** to evaluate,
  therefore it will not be discussed here any further

Effectiveness

- **Effectiveness:** How to measure **result quality?**
- Key concept is **relevance**
- There is **no fully satisfactory definition** of relevance
  - The same problem as with "information" and "intelligence"…
- What we will do next?
  - Point out some **important aspects** of relevance
  - Give a **pragmatic approach** from the **system builder's point of view**
- Fortunately, often we don't need a precise definition (think of probabilistic retrieval)

Relevance is Multidimensional

- Saracevic (2007) identifies
  five manifestations of relevance:
  - **System or algorithmic relevance**
  - **Topical or subject relevance**
  - **Cognitive relevance or pertinence**
  - **Situational relevance or utility**
  - **Affective relevance**

**objective
(or system-based)**

**subjective
(or user-based)**

System or Algorithmic Relevance

- **System or algorithmic relevance:**
  - Relevance as a **static and objective** concept
  - Not influenced by users
  - The most common and clearest definition of relevance
    - "How close is the fit between the retrieved set of documents and the user's query?"
  - In a narrow sense, system relevance is always perfect
  - **Criteria for measuring relevance**: effectiveness in inferring relevance

Topical or Subjective Relevance

- **Topical or subject relevance:**
  - Relevance as a **subjective** or **user-based** concept
  - Still a **static** concept
  - The concept of **topic** is understood as **aboutness,** not contents, i.e., an **intellectual assessment** of how a document corresponds to the topical area required and described by the query
    - "How close is the **semantic fit** between the query and the topics of the document retrieved?"
  - Consequently, based on **judgments**
  - **Criteria for measuring relevance**: aboutness independent of the query

Cognitive Relevance or Pertinence

- **Cognitive relevance or pertinence:**
  - Again, **subjective**
  - Relevance as relation between documents and the **cognitive state of knowledge** and **information need** of a user
  - “What is the user’s judgment about the **applicability** of the retrieved documents to the matter at hand?”
  - Relevance may be **dynamic,** i.e., change over session time

Situational Relevance or Utility

- **Situational relevance or utility:**
  - Again, **subjective** and **dynamic**
  - Relevance as the relation between the **situation, task, or problem at hand,** and documents
  - "Do the retrieved items allow the user to complete the task at hand?"
  - Involves **serendipity:**
    Information may be useful although you did not expect this in advance

Affective Relevance

- **Affective relevance:**
  - Again, **subjective** and **dynamic**
  - Relevance as the relation between documents and the **intents, goals, emotions, and motivations** of a user
  - Represents the human **drive for information**

Manifestations of Relevance

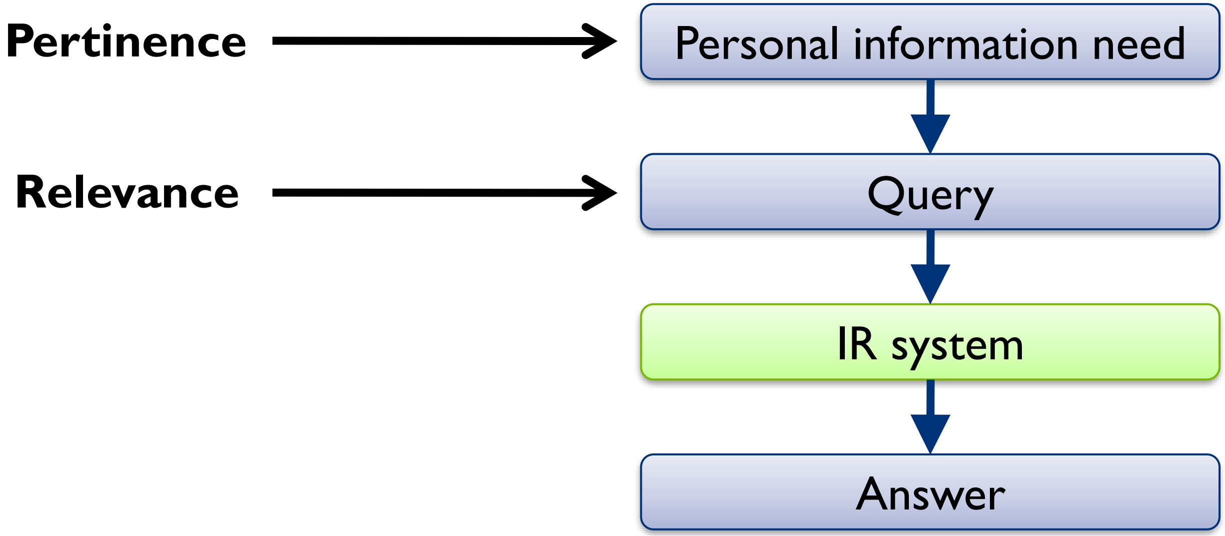
Type of Relevance	Criteria for measurment
System or algorithmic relevance	Rules for comparative judgments
Topical or subject relevance	Aboutness
Cognitive relevance or pertinence	Informativeness, novelty, information quality, …
Situational relevance or utility	Usefulness in decision making, appropriateness of information in resolution of a problem, reduction of uncertainty, …
Affective relevance	Satisfaction, success, accomplishment, …

What’s Our Notion of Relevance?

- Our notion of relevance: **Topical or subject relevance**
- **Current goal of IR:**
  - Build an algorithm resembling topical relevance for “most” users
- **Future goals (current research):**
  - Address the other subjective manifestations of relevance

Queries and Information Needs

- **Relevance vs. pertinence:**



- **Be careful:**
  – Often "relevant to a **query**" means "relevant to a **'typical' information need** that fits the query"

Evaluating Relevance

- Back to our initial question:
  How to **evaluate** a system's **result quality?**
- Traditional approach: Evaluation **benchmarks**
  - A benchmark document **collection**
  - A benchmark suite of **information needs,** expressible as **queries**
  - An **assessment** of the relevance of each query–document pair, called "gold standard" or "ground truth"
    - Usually, relevance is assessed in binary fashion
- Example of an information need:
  - "What are the prospects of the Quebec separatists achieving independence from the rest of Canada?"

Evaluating Relevance

- How to completely assess very large collections?
- The **pooling method** is widely used:
  - Run each query on a set of very different IR systems
  - “Pool” their results to form a set of documents, which have at least this recommendation of potential relevance
(usually, take top *k* results from each system)
  - The union of these retrieved sets is presented to human judges for relevance assessment
  - **Assumption: Unassessed documents are irrelevant!**

Test Collections

- Evaluate algorithmic relevance against topic relevance
- Underlying assumptions:
  - Laboratory retrieval resembles real retrieval
  - Intersubject reliability:
    There is at least some consistency between this user's opinion and those of others
  - Binary relevance
  - Independence of interdocument relevance assessments:
    The relevance of a document can be assessed independently of assessments of other documents

Cranfield Collection

Detour

- The **Cranfield collection:**
  - It was the Pioneering test collection
  - Cranfield University (UK)
  - Collected in1960s
  - Total size: 1.6 Mbytes
  - 1400 abstracts of aerodynamics (aircraft design) journal articles
  - 225 queries generated by some of the documents' authors
  - Exhaustive relevance judgments for all query–document pairs (done by students and "experts")

Cranfield Collection

Detour

- **Rating scale used for relevance judgments:**
  1. References which are a **complete answer** to the question
  2. References of a **high degree of relevance,** the lack of which either would have made the research impracticable or would have resulted in a considerable amount of extra work
  3. References which were **useful,** either as general background to the work or as suggesting methods of tackling certain aspects of the work
  4. References of **minimum interest,** for example, those that have been included from an historical viewpoint
  5. References of **no interest**

Cranfield Collection

Detour

- **Example document:**
  - “viscous flow along a flat plate moving at high speeds. by the distortion of coordinates, it is shown that, in the case of supersonic viscous flow past a flat plate, the boundary-layer and simple wave theories can be combined to give a complete representation of the velocity and pressure fields. […]”

- **Example query:**
  - “why does the compressibility transformation fail to correlate the high speed data for helium and air”

TREC

Detour

- **TREC**
  - Annual Text Retrieval Conference, beginning in 1992
  - Sponsored by the U.S. National Institute of Standards and Technology as well as the U.S. Department of Defense
  - Today: many different tracks, e.g. blogs, genomics, spam http://trec.nist.gov/tracks.html
  - Provides data sets and test problems
  - **Research competitions**

TREC

Detour

- **TREC collections:**
  - Best known:
    Test collections used for the TREC Ad Hoc track during the first eight TREC evaluations between 1992 and 1999
  - In total the test collection comprises:
    - 1.89 million documents (mainly newswire articles)
    - 450 information needs (specified in detailed text passages)
  - **Binary** relevance judgments (used the **pooling method**)

TREC

Detour

- **Example information need:**
  - **Title:**
    Endangered Species (Mammals)
  - **Description:**
    Compile a list of mammals that are considered to be endangered, identify their habitat and, if possible, specify what threatens them.
  - **Narrative:**
    Any document identifying a mammal as endangered is relevant. Statements of authorities disputing the endangered status would also be relevant. A document containing information on habitat and populations of a mammal identified elsewhere as endangered would also be relevant even if the document at hand did not identify the species as endangered. Generalized statements about endangered species without reference to specific mammals would not be relevant.

More Collections

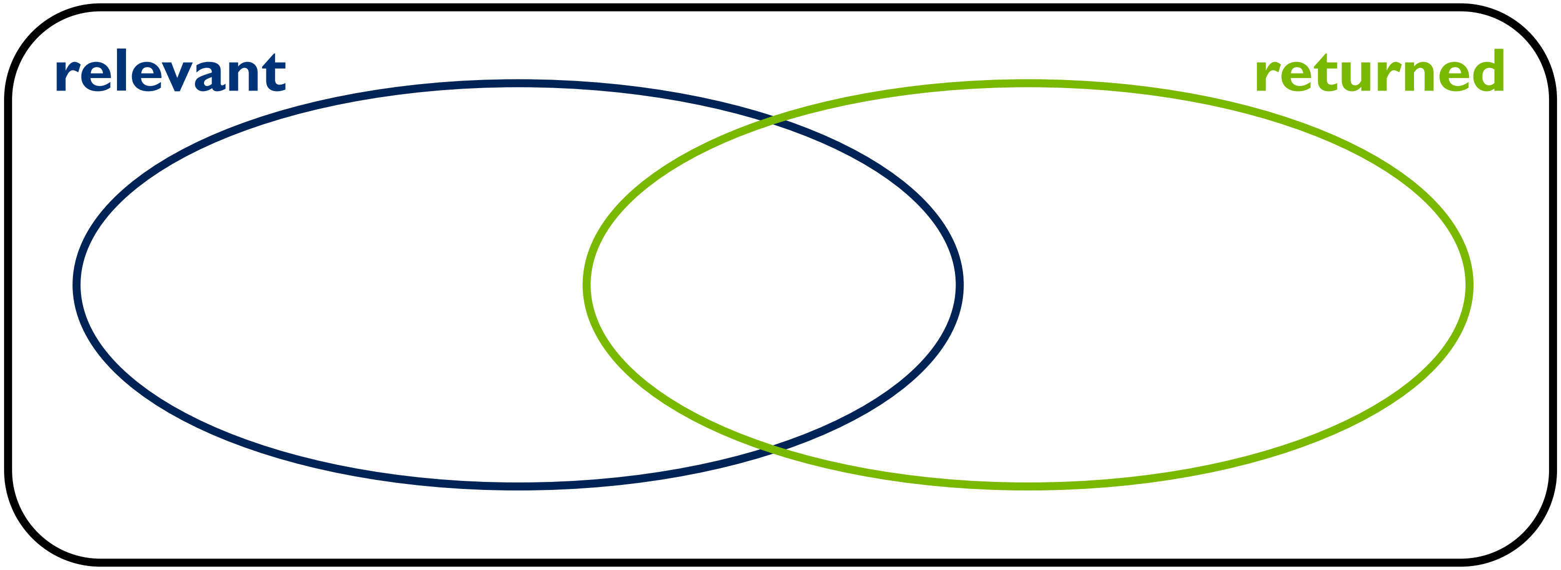
Detour

- **Some more collections:**
  - CACM
    - 3,204 titles and abstracts from the journal Communications of the ACM
  - Reuters-RCV1
    - Reuters Corpus Volume 1
    - 806,791 news stories in English
    - 2.5 Gbytes
  - ClueWeb09
    - Over 1.04 billion web pages, in 10 languages
    - Total Size: 25 Tbytes
    - The dataset is used by several tracks of the TREC conference.
    - http://lemurproject.org/clueweb09/

Evaluation of Answer Sets

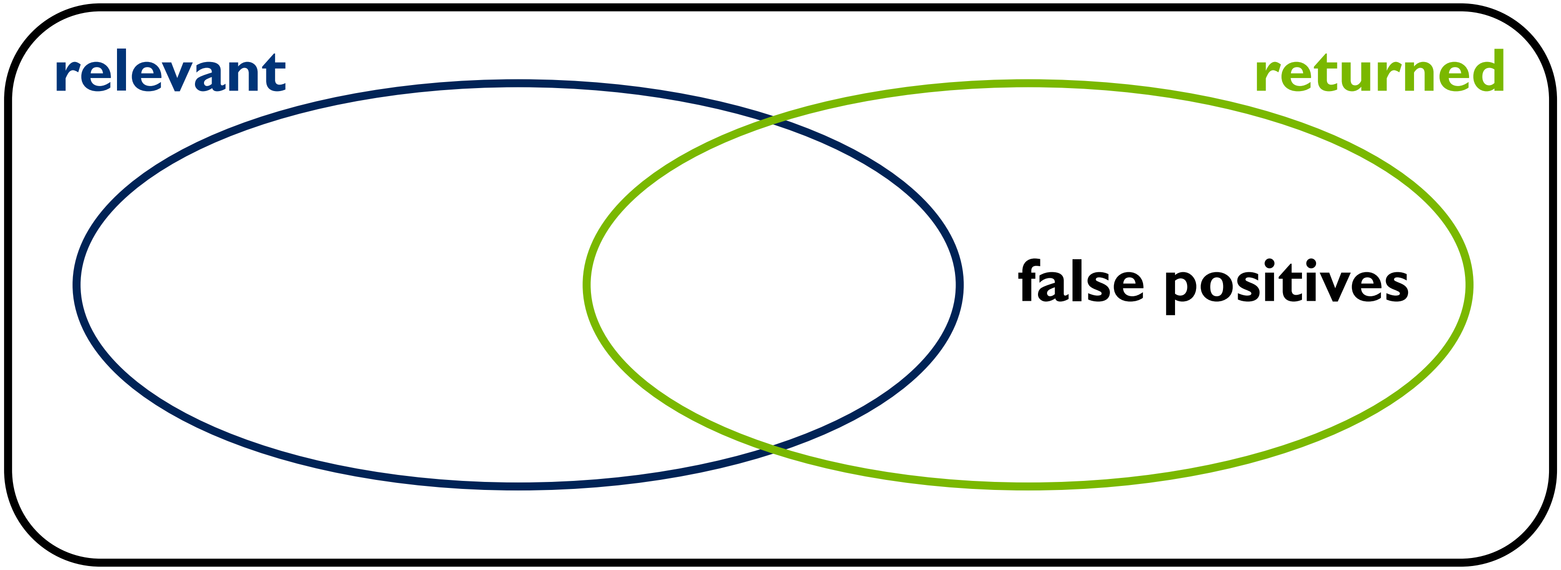
- First, we deal with the evaluation of IR systems that return **result sets,** i.e., they do not provide any ranking
- **Idea:** Compare result set with ground truth result set

- What sets are involved here?



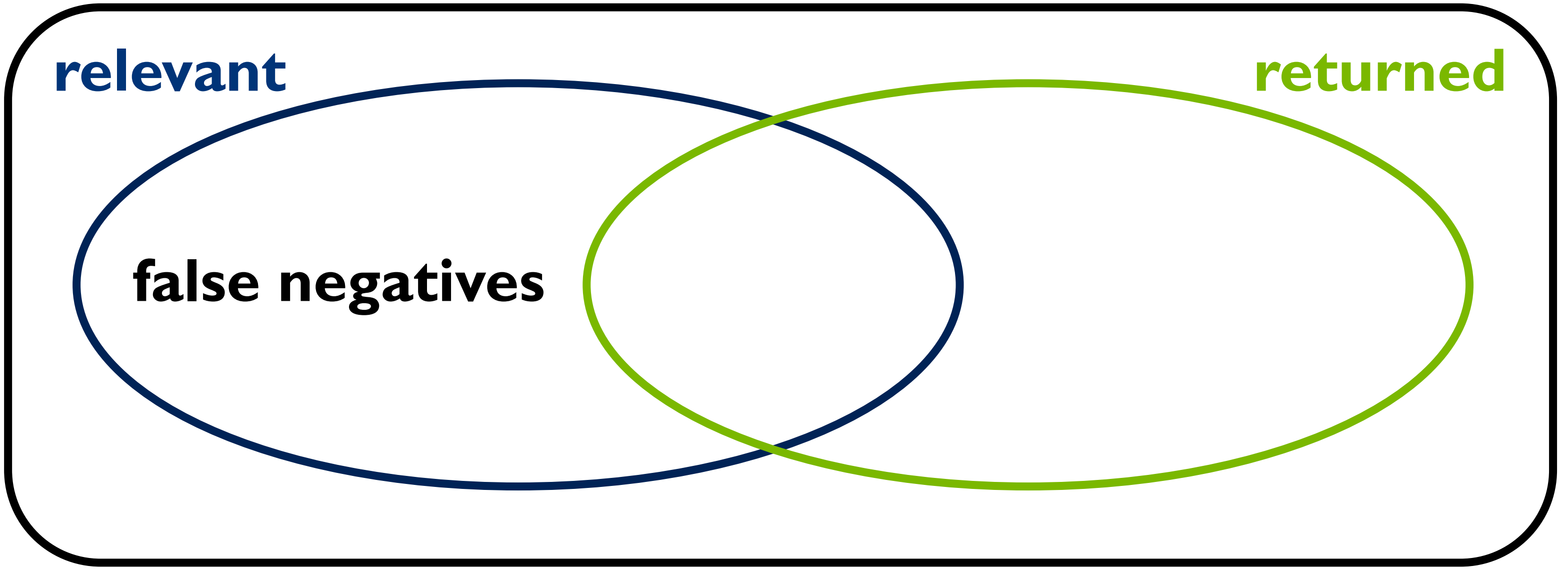
False Positives

- **False positives:**
  - Irrelevant documents returned by the system
  - Extend the result set unnecessarily
  - Often inevitable
  - Usually can be filtered out by the user quite easily



False Negatives

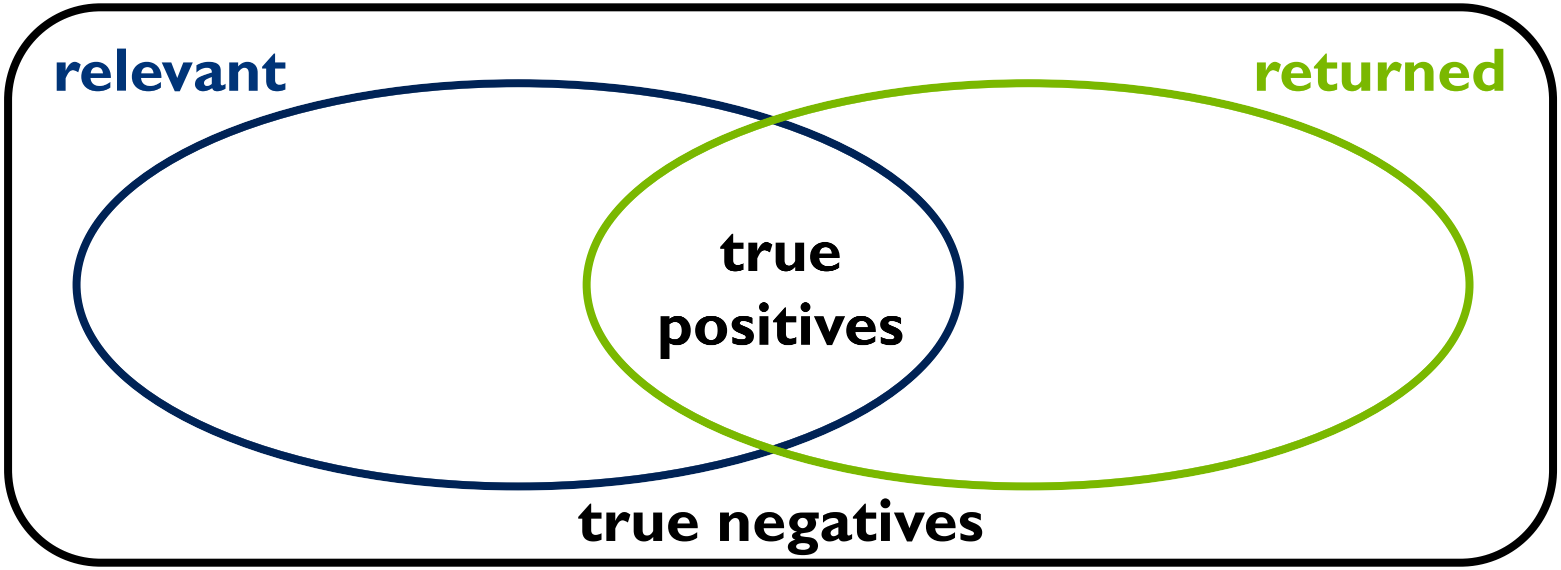
- **False negatives:**
  - Relevant documents not returned by the system
  - Problematic, since the user usually is not aware of them
    - Are there any “better” documents?
  - Often worse than false positives



True Positives and True Negatives

- Remaining sets: **True positives** and **true negatives**

	relevant	**nonrelevant**
returned	true positives	false positives
not returned	false negatives	true negatives



Precision

- **Precision, recall, and fallout** are important measures of (unranked) answer sets
- **Precision:**
  - Uses the number of **true positives** as measure of result quality
  - How many of the returned documents are relevant?
  - Definition:

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{items retrieved})}$$

  - Value in [0, 1], where 1 is best
  - High precision usually is important in **Web search** (result set = first page of results)

Recall

- **Recall:**
  - Also uses the number of **true positives** as measure of quality
  - How many of all relevant documents have been returned?
  - Definition:

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})}$$

  - Value in [0, 1], where 1 is best
  - High recall usually is important for **professional searchers** such as paralegals and intelligence analysts; it is also important for **desktop search**

Fallout

- **Fallout:**
  - Uses **false positives** to measure retrieval quality
  - How many returned documents have been nonrelevant?
  - Definition:

$$\text{Fallout} = \frac{\#(\text{nonrelevant items retrieved})}{\#(\text{nonrelevant items})}$$

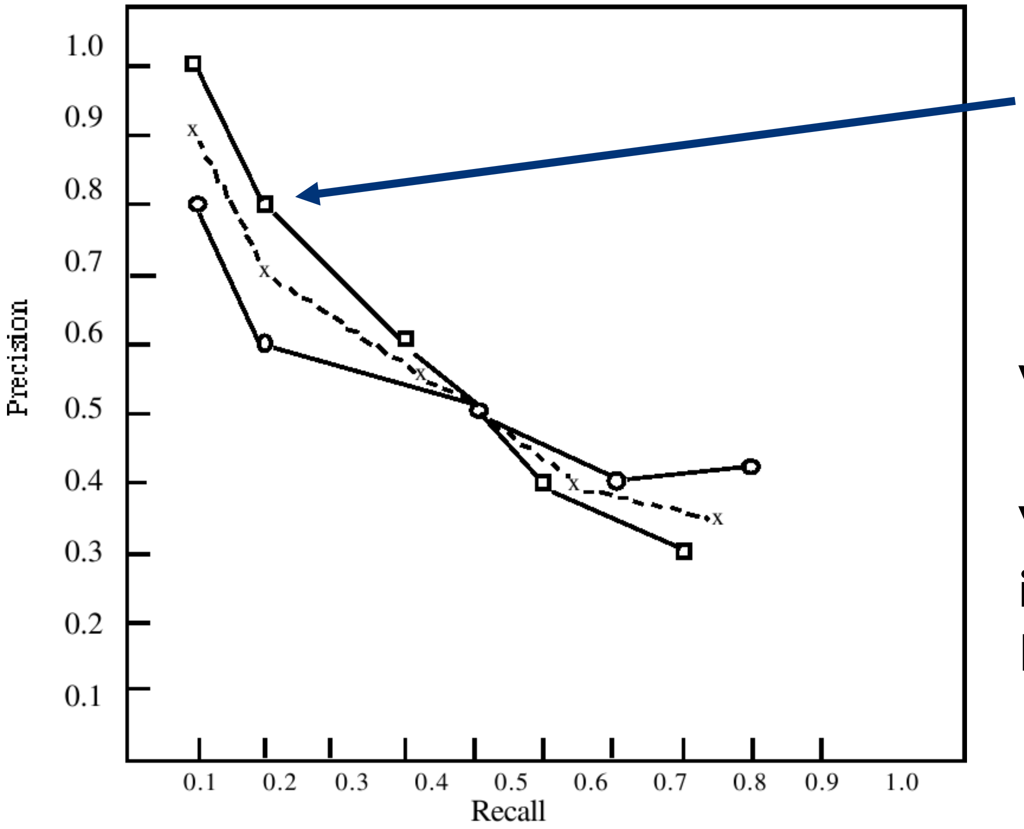
  - Value in [0, 1], where 0 is best
  - Zero fallout can be achieved by returning empty result sets
  - **Fallout usually only makes sense for large result sets**
    - For typical queries, most documents in the collection are nonrelevant

Precision versus Recall

- Precision and recall clearly trade off against one another:
  - Achieve **perfect recall** (but awful precision) by always returning all documents in the collection
  - Achieve **very good precision** (but awful recall) by always returning only the single result that seems to match best
- Normally, this leads to tradeoffs in system tuning
  - Small result sets usually lead to better precision but worse recall
- What about measurement?
  - **Precision is easy to measure**
  - **Measuring recall is at least very difficult, and often impossible**

The Precision–Recall Curve

Example: Comparison of three retrieval systems



Average precision of system 3 at recall level 0.2

Which system is best?

What's more important: Precision or recall?

The F Measure

- The **F measure** combines precision and recall
  - It’s a weighted **harmonic mean** of precision and recall
  - Definition:

$$F = \frac{1}{\alpha \cdot \frac{1}{\text{precision}} + (1 - \alpha) \cdot \frac{1}{\text{recall}}}$$

  - Parameterized by weighting factor $\alpha \in [0, 1]$
  - **Balanced F measure:** $\alpha = 1/2$
  - Value in [0, 1], where 1 is best
  - **Why do we use the harmonic mean?**
    With the arithmetic mean, an F measure of 0.5 could easily by achieved e.g. by returning all documents

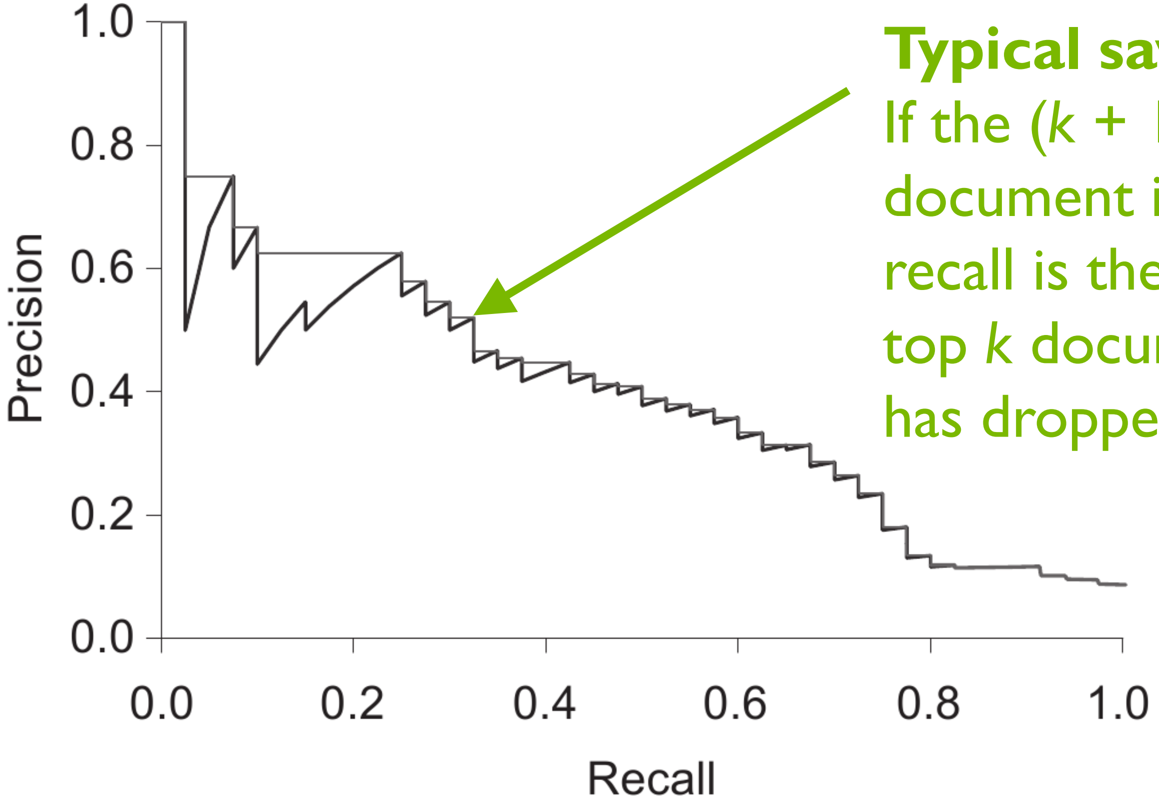
Ordered Result Lists

- Now, how to evaluate ordered result lists?
  - **Idea:** Compute precision and recall for the set of the **top *k*** retrieved documents; repeat this for many different *k*
  - We then get the **precision at *k*** and the **recall at *k***
  - **Example result list** (assume there are 5 relevant docs):

k	Relevant?	#Relevant	Recall at k	Precision at k
1	Yes	1	1/5 = 0.2	1
2	Yes	2	2/5 = 0.4	1
3	No	2	2/5 = 0.4	2/3 ≈ 0.67
4	Yes	3	3/5 = 0.6	3/4 = 0.75
5	No	3	3/5 = 0.6	3/5 = 0.6
6	No	3	3/5 = 0.6	3/6 = 0.5
7	No	3	3/5 = 0.6	3/7 ≈ 0.43

Ordered Result Lists

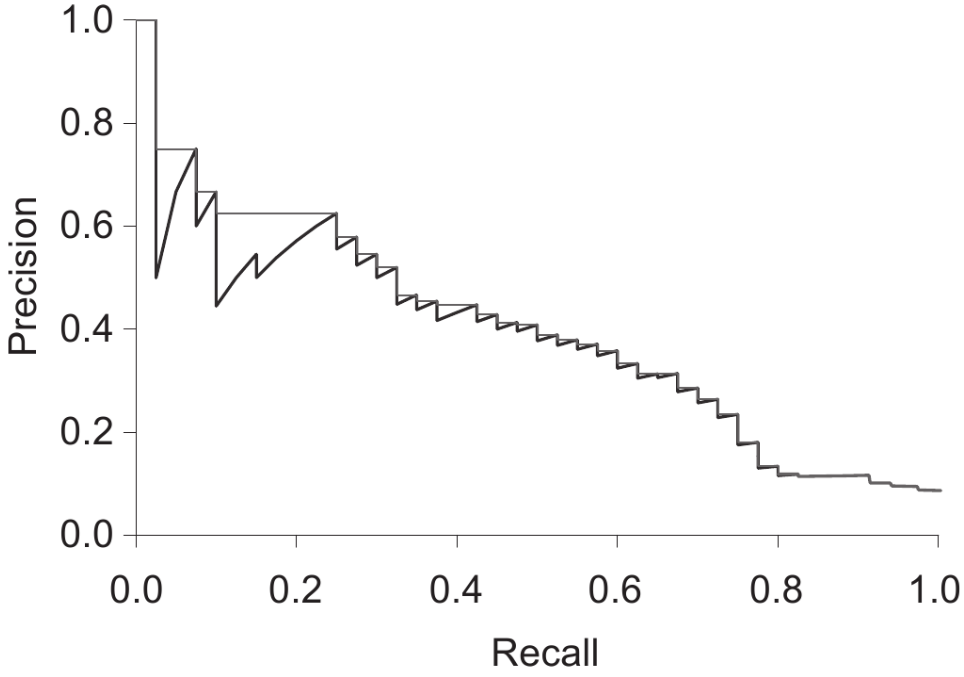
- Plotting precision at *k* and recall at *k*, for many *k*, again gives us a **precision–recall curve**
- Example from (Manning *et al*., 2008):



Typical sawtooth shape:
If the (*k* + 1)-th retrieved document is nonrelevant, then recall is the same as for the top *k* documents, but precision has dropped

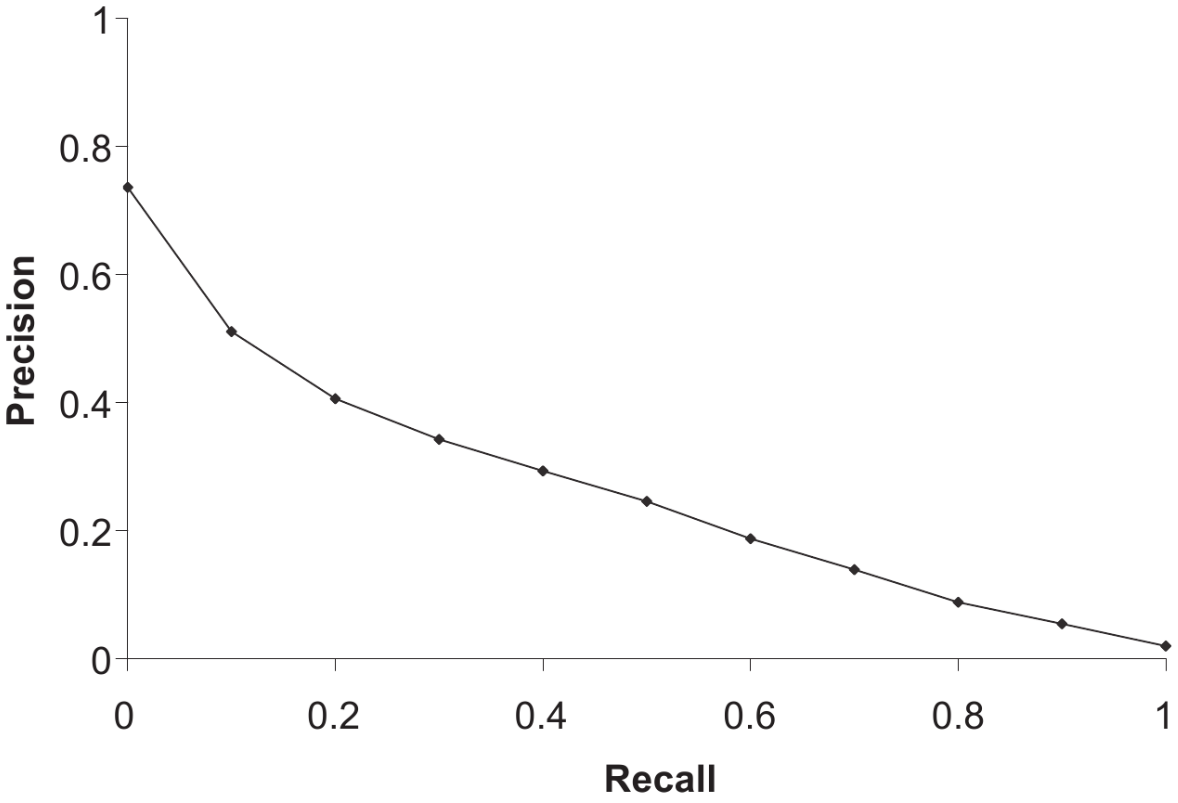
Ordered Result Lists

- To get rid of the sawtooth shape, we can use the **interpolated precision** at a certain recall level instead
- **Definition:** The *interpolated precision* at recall level *r* is the highest precision found for any recall level *r*’ > *r*



Ordered Result Lists

- TREC uses eleven-point interpolated average precision:
  - Recall levels used are 0.0, 0.1, …, 1.0
  - Precision values are averaged over many different queries
- Averaged eleven-point interpolated precision/recall; example from (Manning *et al.*, 2008):

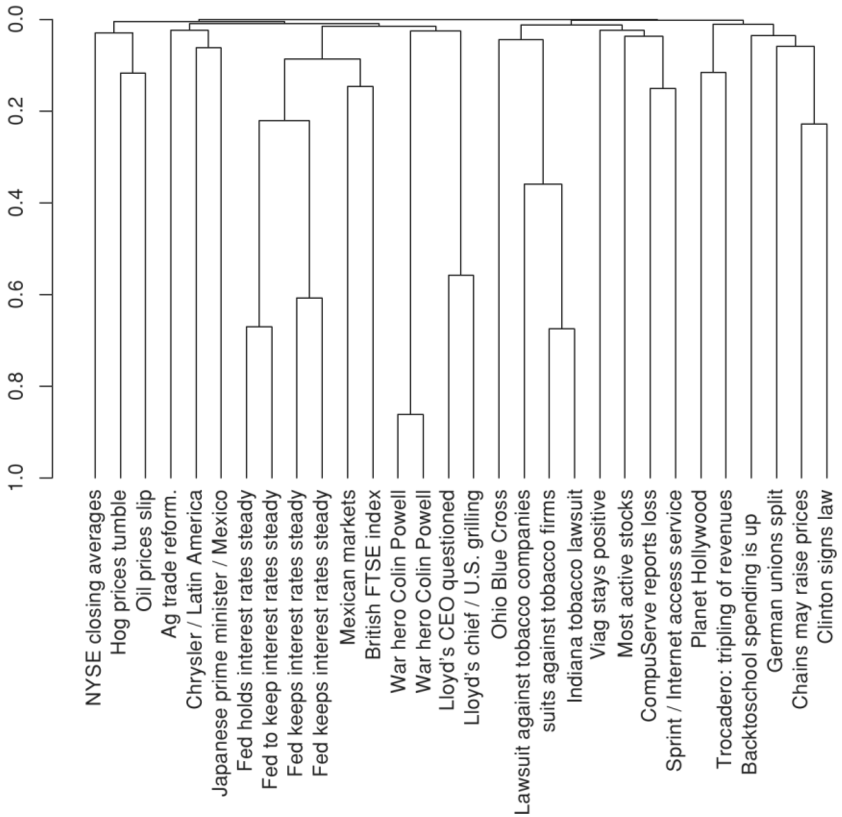


Ordered Result Lists

- Some people like single aggregate values instead of curves
- A popular one is the **mean average precision** (MAP)
  - MAP has been shown to have particularly **good discrimination** and **stability**
- **Definition:**
  1. Compute precision at *k*, for any *k* such that there is a relevant document at position *k* in the result list
  2. Then compute the arithmetic mean of all these precision values
  3. Compute the mean over many different queries; this value is the mean average precision of the IR system
- Broadly spoken: MAP is the **average area** under the precision–recall curve for a set of queries

Next Lecture

- Clustering


NYSE closing averages
Hog prices tumble
Oil prices slip
Ag trade reform.
Chrysler / Latin America
Japanese prime minister / Mexico
Fed holds interest rates steady
Fed to keep interest rates steady
Fed keeps interest rates steady
Fed keeps interest rates steady
Mexican markets
British FTSE index
War hero Colin Powell
War hero Colin Powell
Lloyd's CEO questioned
Lloyd's chief / U.S. grilling
Ohio Blue Cross
Lawsuit against tobacco companies
suits against tobacco firms
Indiana tobacco lawsuit
Viag stays positive
Most active stocks
CompuServe reports loss
Sprint / Internet access service
Planet Hollywood
Trocadero: tripling of revenues
Backtoschool spending is up
German unions split
Chains may raise prices
Clinton signs law