

# Information Retrieval

## Lecture 1: Introduction to concepts and problems

Computer Science Tripos Part II



UNIVERSITY OF  
CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

### Information access in an ideal world

2

---

**Question:** “What was the historical development of Boolean algebra and set theory?”

**Answer:** “In 1854 George Boole published a seminal work *An investigation into the Laws of Thought, on Which are founded the Mathematical Theories of Logic and Probabilities.*”...

The user’s information need is ideally met:

- Right type of response for information need
- Expected amount of information
- In perfect English, natural interaction
- And the information is of course correct!

Instead of the ideal world scenario: Formulate question as a **query** containing **terms**, and review **documents** which the system returns:

Q<sub>1</sub>: set\* bool\* algebra

IR System manipulates terms:

Q<sub>1</sub>': set\* .467 bool\* .751 algebra .091

via

$$weight_{w,q,D} = tf_{w,q} \cdot idf_{w,D}$$

---

## Answers: documents, in order of estimated relevance

A':

[DOC 1 — 1.309]

... **Boole's algebra**, (that is in fact a set of **algebra** structures) was introduced in 1847 so as to propose an algebraic formulation of logical proposals. ...

[DOC2 — 1.286]

Answer your questions about **algebra** and other areas of mathematics by **setting** up your IR system – You can use **Boolean** queries.

[DOC3 — 1.211]

This was a breakthrough for mathematics and **Boole** was the first to prove that logic is part of mathematics and not of philosophy as was commonly accepted by scientists of this era.

- Precise information-seeking search → don't care where the information comes from, expect at least one document answering it ("when was Boole born?")
- Known-item search: Know that a certain item is there, want to refind it → want to find exactly that item (Boole's book)
- Open-ended search ("topic search"): → do not know if a document exists; potentially, many exist ("has anybody implemented a probabilistic version of Boolean algebra?")

## Two search problems

- **Information scarcity problem** (or needle-in-haystack problem): hard to find rare information
  - Information need since 2001: Lord Byron's first words? 3 years old? Long sentence to the nurse in perfect English?
  - Found in Oct 2007:
    - ... Lord Macaulay's first words had not come until he was four; a lady had spilt hot tea on him at a party, and he had said, drawing back a little from her solicitous caress:  

"Thank you, Madam, the agony is somewhat abated."
- **Information abundance problem** (for more clear-cut information needs): redundancy of obvious information
  - What is toxoplasmosis?

## “Boole’s book”

- If I know title, author, year (“An investigation of the laws of thought, on which are founded the mathematical theories of logic and probabilities, George Boole, 1854”)
  - Boole, G ... 1854 → Location: Betty & Gordon Moore Library Class-mark: QA9 .B65
- “Bool? Boole? on algebra??, 19th Century” → L paper catalogue as above
- “on algebra, 19th century, called ‘*Laws of Thought*’” → L subject index Algebra 347.9 → Shelves

## “Boole’s book”

Search	Query	Results
(full text)	“laws thought”	→ 10000 entries (truncated)
(title)	“laws thought”	→ 0 entries
(title)	“laws of thought”	→ 2 entries
(title)	“law of thought”	→ 0 entries
(title)	“algebra”	→ 623 entries
(full text)	“algebra”	→ 2474 entries
(Boolean)	“logic AND boole”	→ 9 entries

### Google

- “law” → Lawyers, legal services, law schools
- “laws” → Lists of laws, public and private laws, four spiritual laws  
→ law and laws are different search terms on Google!
- Boole does not show up in the first 10 pages

## More google – string search

- “law thought”
  - “Savannah NOW: Local News - Mother-in-law thought mechanic was a ...”

→ Unexpected other meanings of terms (here: multi-word term)

→ Grammatical function of term ignored (verb instead of noun)

→ Treatment of dash

Rest: divorce, law as thought control, law and social thought, Scottish thought...

- “thought law”
  - “And you thought law enforcement was boring?”

→ Order of terms matters in Google (but capitalisation does not)

# A better Google search for initial query: (develop|history|historical) set theory (boole|boolean) algebra 11

---

STEP II: [Develop](#) Course Objectives and Outcomes

STEP II: [Develop](#) Course Objectives and Outcomes COURSE ... partially ordered sets, lattices, [Boolean](#) algebras, semigroups ... and predicate logic 2. [Set theory](#) and its ...

06: Order, lattices, ordered algebraic structures

... [History](#). ... especially on infinite sets is the study of Ordinals in [Set Theory](#); ...

03E: [Set theory](#)

... do no better describing the [history](#) of [Set](#) ... propositions; 03E30: Axiomatics of classical [set theory](#) and its ... Other aspects of forcing and [Boolean](#)-valued models; ... Description: From Dave Rusin's "Known Math" collection.

Introduction to [Algebra: History](#)

... The next major development in the [history](#) of algebra ... [Boole](#)'s original notation is no longer used, and ... uses the symbols of either [set theory](#), or propositional ...

HiLight ... to a particular epoch in human [history](#), that of ... Every [historical](#) position to achieve is like a ... Algebra Classical propositional logic and [set theory](#) are often ...

Selected course [history](#)

... Rosen). Basic [set theory](#), discrete probability, combinatorics, [Boolean algebra](#), graph [theory](#). Fundamentals of Dynamical Systems. Elementary

MA003 MATHEMATICS FOR COMPUTING

... [develop](#) and use the concepts presented in the lectures. In particular emphasis will be put on parallel or similar systems such as [set theory](#) & [Boolean algebra](#) ...

[Boolean algebra](#)

... a new method of diagramming [Boole](#)'s notation; this was ... When used in [set theory](#), [Boolean](#) notation can demonstrate the ... indicating what is in each [set](#) alone, what ...

Graduate Courses

... Combinations, logic [set theory](#), [Boolean algebra](#), relations and functions, graph ... The [historical](#) evolution of non-Euclidean geometries ... [History](#) of Mathematics. ...

Barnes & Noble.com - Ones and Zeros: Understanding [Boolean](#) ...

... features include: a [history](#) of mathematical logic, an ... Electronic digital computers, [Set theory](#), Design. Logic, Symbolic and mathematical, Circuits, Algebra, [Boolean](#). ...

---

## Results... after some reading and searching 12

### Somewhere in document 4:

... Boolean algebra was formulated by the English mathematician George Boole in 1847 ...

### Somewhere in document 8:

Boolean algebra, an abstract mathematical system primarily used in computer science and in expressing the relationships between sets (groups of objects or concepts). The notational system was developed by the English mathematician George Boole c.1850 to permit an algebraic manipulation of logical statements.

→ Searching requires knowledge about underlying model to be effective

law\*

- law
- laws
- lawyers
- lawn
- Lawndale, CA

AltaVista also does stemming!

## Factors in searching I

---

- Syntax of query language:
  - Are my search terms stemmed?
  - Are there wildcards?
  - Can I use phrases? (“laws of thought”)
  - Does capitalisation matter?
  - If more than one search term is present, is there an implicit “AND”?
  - Which other boolean operators are available: disjunction? exclusion?
  - Can I make a search term compulsory?
  - If there are more than one search term, does the order matter?
  - Can I use proximity? (Altavista’s NEAR: within 10 words of each other)

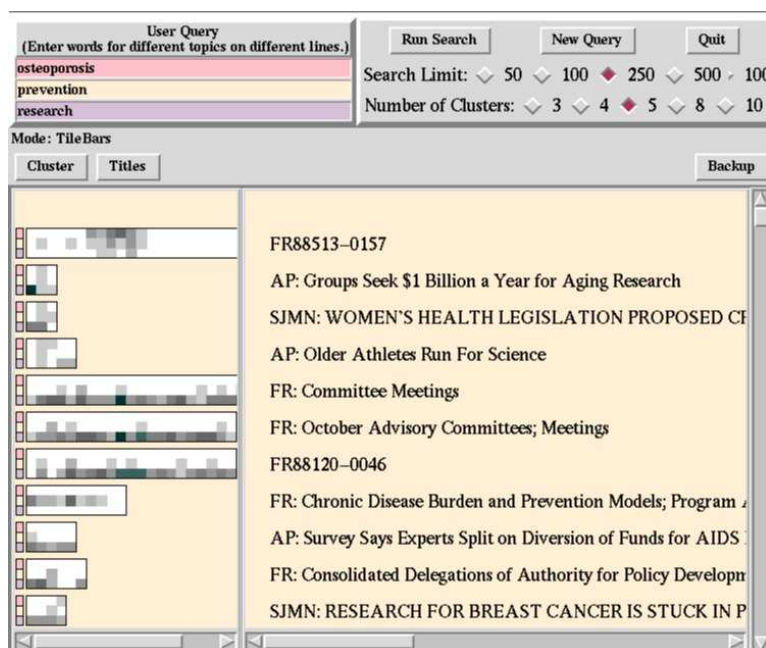
- Search Ground:
  - Which text representation is indexed? Abstract, title, full text?
  - Can I specify where I am searching?
- Interpreting the output:
  - Is the output ranked?
  - If more than one search term is used, is each term guaranteed to be there?

### How to choose search terms

- More specific terms are generally better
- Some terms mean other things too (think of different word senses, use exclusion)
- Other terms can mean the same thing (think of synonyms)
- Interference from WWW-specific key terms (“schema”, “domain”)

### Beyond IR: More fine-grained search (Hearst, 1993)

TileBars: Show distribution of query terms (e.g. “osteoporosis” “prevention” “research”) in segments of the document





“What did George Boole invent?”

“And when?”

Solution One (the shortcut):

Ask google for the following exact strings:

```
"Boole invented"  
"Boole developed"  
"Boole invents"  
"invented by Boole"  
"invented by George Boole"  
"developed by Boole"  
"Boole discovered"
```

- Near-synonyms
- Passive/Active Voice variation

---

## Getting simple answers

George Boole invented [a mathematical tool for future computer-builders](#)—an “algebra of logic” that was used nearly a a hundred years later to link the process of human reason to the operations of machines.

George Boole invented [a system for symbolic and logical reasoning, called Boolean Algebra](#), which became the basic design tool for computer design.

George Boole invented [a branch of mathematic colled Coolean Algebra](#) which has been applied to the development of logic and electrical relays.

1854 AD, Boole invented [Boolean algebra](#).

George Boole invented [Boolean logic](#), which birthed discrete mathematics, which enabled the development of the transistor and ultimately the home computer, so essentially this one man is the reason our country is so fat, detached, sequestered, posh, and lazy and the reason we are all headed for a sublime Hell on Earth not unlike the environment depicted in the video “Sober” by Tool.

Around the 1850s, the British mathematician George Boole invented [a new form of mathematics](#), in which he represented logical expressions in a mathematical form now known as Boolean Algebra.

Boole invented [“Boolean algebra”](#) (switching theory)

Boole invented [the truth table](#) to test the truth and validity of compound propositions.

Boole invented [the first practical system of logic in algebraic form](#)

While working here in UCC in 1854, Boole invented [Boolean Algebra](#) which has become the cornerstone of modern electronics and information technology.

George Boole invented [propositional logic](#) (1847).

In 1854, Boole invented [Boolean algebra](#). George Boole invented [the branch of mathematics known as Boolean algebra](#). In the 19th century George Boole invented [Boolean algebra](#) as a theoretical study.

Later in the year 1850, Charles Boole invented [binary codes](#), which uses only numbers 0 and 1.

In the 1850s, George Boole invented [a mathematical system](#) of symbolic logic that would later become the basis for modern computer design.

The English mathematician, George Boole, developed [an algebra of logic](#), which has become the basis for computer database searches.

Boole developed [an algebraic calculus](#) to interpret whether composite assertions were true or false

Initially a schoolteacher in Lincolnshire and Yorkshire, England, Boole developed [his ideas about symbolic logic](#) without the benefit of a formal university training.

Boole developed "[The Mathematical Analysis of Logic](#)."

Boole developed [the meaning of the logical operators](#).

By considering assertions to be true or false, Boole developed [an algebraic calculus](#) to interpret whether composite assertions were true or false in terms of how they composition was formed.

The first major advance came when George Boole developed [an algebra of logic](#).

[The Algebra of Logic](#) was originally developed by Boole

[Boolean algebra](#) was developed by Boole.

[The modern concept of abstract algebra](#) was developed by Boole In the 1930's Claude Shannon, an American mathematician who later worked for Bell Labs, noted that the algebra developed by Boole was the appropriate

[The idea of a completely formalistic logic](#), however, was developed by Boole in the 19th century.

[an internationally recognized system of logic](#) invented by Boole

[Algebra](#) invented by Boole allows easy manipulation of symbols

[Probabilistic logic](#), invented by Boole, is a technique for drawing inferences from uncertain propositions for which there are no independence assumptions.

This page defines 'boolean,' pertaining to the logical operations described in the [algebra](#) invented by George Boole

[That little bit of computer logic](#) was actually invented by George Boole, a British mathematician and logician.

[Boolean Logic](#), invented by George Boole, began the process for what are today's computational methods.

Shannon explained how [the algebra](#) invented by George Boole in the mid-1800s could be used to ...

[Boolean circuits are certain electric circuits](#) that were invented by George Boole.

So are the bit and bytes that obey complex [laws](#) first invented by George Boole a hundred or so years ago.

Search engines use Boolean logic, which is [a system of logic](#) invented by George Boole, a nineteenth century mathematician.

Explain how Boole applied his new algebra to logic, and explain (according to Davis) how it was that Boole discovered [the importance/validity of using the sets 1 and 0](#) by proceeding in accord with the Aristotelian "principle of contradiction" Bayes's work.

Boole discovered [analogy of algebraic symbols and those of logic](#).

In 1854 George Boole "discovered [pure mathematics](#)"(Bertrand Russell's expression) this equivalence

- ... the following are not answers:

- What Boole discovered in that meadow and worked out on paper two decades later was destined to become the mathematical linchpin that coupled the logical abstractions of software with the physical operations of electronic machines.
- Lots of Rules and EXECs, developed by Boole SSEs and customers.

- ... the following might be answers, but more work is needed:

- Boole invented **this method** back in the 18th century, so that human thought could be strictly analysed and evaluated.
- **It** was invented by George Boole, a British mathematician in the 1840's.
- Leibniz ... invented binary numbers in this case, but he didn't even invent propositional calculus; **that** was invented by Boole one hundred and fifty years later.

- ... the last sentence cancels our sought-for fact:

- Many colleagues in Algebra and Logic think that Boole developed either Boolean Algebra, or Boolean Rings. He did neither.

---

## Information extraction

We implicitly used information extraction in the QA-“shortcut” solution.

But the string-based approach does not generalise to similar mentions, eg. “Boole was the inventor of”

Components of the task of information extraction:

1. **Template Recognition:**

Find predefined relations in unrestricted text, for example inventors/invention patterns

2. **Named Entity Recognition:**

Find entities of a certain semantic type in unrestricted text

HERE: Find names, find dates, in all possible formulations, with robustness to typos and formatting

3. **Coreference Resolution:**

Decide which strings refer to the same entities

The INVENTOR relation/template has pre-determined slots and relationship between slots

- “To accomplish this, we’ll first learn about the concept of Boolean algebra - a system of logic designed by George Boole.”

INVENTOR: George Boole

INVENTED: Boolean algebra

- “Peirce developed what amounts to a semantics for three-valued logic.”

INVENTOR: Peirce

INVENTED: a semantics for three-valued logic

- Information need known beforehand → template
- Domain dependence (only talk about invention events)

## Question Answering – deeper approaches

Necessary if:

- information need cannot be expressed with templates a priori
- no simple meaning–surface mapping and/or not enough data.

QA:

- Need to understand something about the **question**:
  - Grammatical function – who does what to whom?
  - What is the expected answer type?
- Need to understand something about the **document**:
  - Locate the expected answer type in the text
  - Not all necessary information locally available (e.g. pronouns)
- Need to produce an **answer**:
  - Answers are phrases, sentences, paragraphs, 50Byte strings
  - “Information packaging”, reversal of non-local effects

- The holy grail of NLP
- Methods working today are either very simple or very complicated
- The simple ones are robust but have many other disadvantages
  - Textual problems for all effects above the sentence-level
  - No guarantee of truth preservation
- The complicated ones are not robust
- Two recent tasks:
  - Multi-document summarisation
  - Incremental-time-line summarisation
- Evaluation is a major problem

## Summarisation by sentence extraction

- Each sentence is represented by a set of 'importance indicators' (features)
- These are combined in such a way to rank the sentences
- The N highest-ranking sentences are extracted and constitute the summary (here: black sentences)

	Importance indicators
1 Algebra provides a generalization of arithmetic by using symbols, usually letters, to represent numbers.	0 1 3 1
2 For example, it is obviously true that $2 + 3 = 3 + 2$	0 0 2 1
...	
14 In about 1100, the Persian mathematician Omar Khayyam wrote a treatise on algebra based on Euclid's methods.	0 0 1 0
...	
26 Boolean algebra is the algebra of sets and of logic.	1 0 1 1
27 It uses symbols to represent logical statements instead of words.	1 0 1 1
28 Boolean algebra was formulated by the English mathematician George Boole in 1847.	1 1 2 1

29	Logic had previously been largely the province of philosophers, but in his book, <i>The Mathematical Analysis of Logic</i> , Boole reduced the whole of classical, Aristotelian logic to a set of algebraic equations.	0	0	0	1
30	Boole's original notation is no longer used, and modern Boolean algebra now uses the symbols of either set theory, or propositional calculus.	0	0	0	1
31	Boolean algebra is an uninterpreted system - it consists of rules for manipulating symbols, but does not specify how the symbols should be interpreted.	0	0	0	1
32	<b>The symbols can be taken to represent sets and their relationships, in which case we obtain a Boolean algebra of sets.</b>	0	1	3	0
33	Alternatively, the symbols can be interpreted in terms of logical propositions, or statements, their connectives, and their truth values.	0	0	0	1
34	This means that Boolean algebra has exactly the same structure as propositional calculus.	0	0	0	1
35	<b>The most important application of Boolean algebra is in digital computing.</b>	1	0	2	1
36	Computer chips are made up of transistors arranged in logic gates.	0	0	0	1
37	Each gate performs a simple logical operation.	0	0	0	1
38	For example, an AND gate produces a high voltage electrical pulse at the output $r$ if and only if a high voltage pulse is received at both inputs $p, q$ .	0	0	0	1
39	The computer processes the logical propositions in its program by processing electrical pulses - in the case of the AND gate, the proposition represented is $p \wedge q$ .	0	0	0	1
40	A high pulse is equivalent to a truth value of "true" or binary digit 1, while a low pulse is equivalent to a truth value of "false", or binary digit 0.	0	0	0	1
41	The design of a particular circuit or microchip is based on a set of logical statements.	0	0	0	1
42	These statements can be translated into the symbols of Boolean algebra.	0	0	0	1
43	<b>The algebraic statements can then be simplified according to the rules of the algebra, and translated into a simpler circuit design.</b>	0	3	1	1
44	An algebraic equation shows the relationship between two or more variables.	0	0	0	1
45	The equation below states that the area ( $a$ ) of a circle equals $\pi$ ( $\pi$ , a constant) multiplied by the radius squared ( $r^2$ ).	0	0	0	1

## Summarisation by re-generation

**Doc1**  
 Born in Lincoln, England on November 2, 1815, George Boole was the son of a poor shoemaker. As a child, Boole was educated at a National Society primary school. He received very little formal education, but was determined to become self-educated.

**Doc2**  
 George was born in 1815 at 34 Silver Street, which is now occupied by Langleys, the Solicitors. He was the eldest son of John Boole, a shoemaker.

**Doc3**  
 Born in the English industrial town of Lincoln, Boole was lucky enough to have a father who passed along his own love of math.

**Doc4**  
 Boole was born of humble parents in 1815, the same year as the battle of Waterloo. It is doubtful he received early schooling in mathematics beyond that required for the most basic commerce. Unsatisfied with mathematics texts of the time, he set about reading the great masters, Gauss, Laplace, Leibnitz, and others.

## Cluster 1:

Born in Lincoln, England on November 2, 1815, George Boole was the son of a poor shoemaker. George was born in 1815 at 34 Silver Street, which is now occupied by Langleys, the Solicitors.

→ S1: "George Boole was born in 1815".

## Cluster 2:

Born in Lincoln, England on November 2, 1815, George Boole was the son of a poor shoemaker. Born in the English industrial town of Lincoln, Boole was lucky enough to have a father who passed along his own love of math.

→ S1': "George Boole was born in 1815 in Lincoln, England.". (Aggregation (information packaging))

## Cluster 3:

Boole was born of humble parents in 1815, the same year as the battle of Waterloo.

He was the eldest son of John Boole, a shoemaker.

Born in Lincoln, England on November 2, 1815, George Boole was the son of a poor shoemaker.

→ S2: "He was the son of a shoemaker".

## Cluster 4:

As a child, Boole was educated at a National Society primary school.

He received very little formal education, but was determined to become self-educated.

→ S3: "He received little schooling as a child."

## Today: summary

- Why searching is difficult: mapping from natural language to the underlying search model
- Query constructs: Search terms and how to combine them
- Retrieval models: Finding the best answer document
- Beyond Information Retrieval: Information extraction
  - Type of required information is known beforehand
  - Information need can be expressed by templates
- Question answering: domain-independent
- Summarisation: additional problem of text production

- 
- IR {
    - Lecture 2: Information retrieval models
    - Lecture 3: IR evaluation methodology
    - Lecture 4: Search engines and linkage algorithms
  - IE {
    - Lecture 5: Information extraction
    - Lecture 6: Bootstrapping in information extraction
  - Lecture 7: Question answering
  - Lecture 8: Summarisation; outlook

## Information Retrieval

### Lecture 2: Retrieval models

Computer Science Tripos Part II



UNIVERSITY OF  
CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk



- Definition of the information retrieval problem
- Query languages and retrieval models
  - Boolean model
  - Vector space model
- Logical model of a document/a term
  - Term weighting
  - Term stemming

## Information Retrieval: the task

---

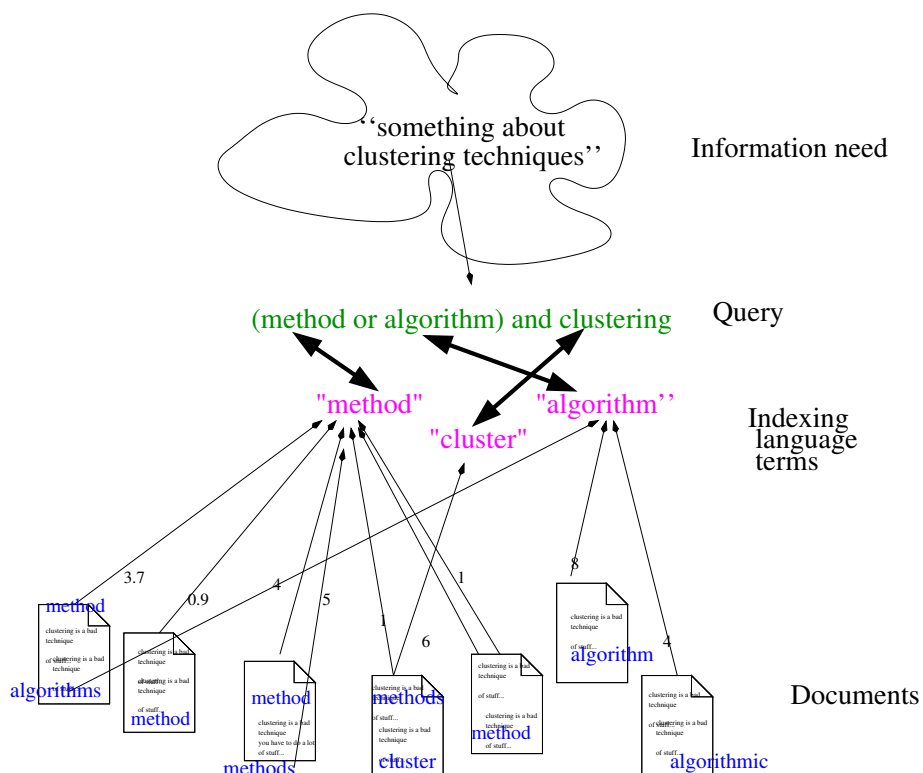
34

Problem: given a query, find documents that are “relevant” to the query

- Given: a large, static document collection
- Given: an information need (reformulated as a keyword-based query)
- Task: find all and only documents that are relevant to this query

### Issues in IR:

- How can I formulate the query? (Query type, query constructs)
- How does the system find the best-matching document? (Retrieval model)
- How are the results presented to me (unsorted list, ranked list, clusters)?



- Indexing: the task of finding terms that describe documents well
- Manual indexing by cataloguers, using fixed vocabularies ("thesauri")
  - labour and training intensive
- Automatic indexing
  - Term manipulation (certain words count as the same term)
  - Term weighting (certain terms are more important than others)
  - Index terms can only be those words or phrases that occur in the text

- Large vocabularies (several thousand items)
- Examples: ACM – subfields of CS; Library of Congress Subject Headings
- Problems:
  - High effort in training in order to achieve consistency
  - Subject matters emerge → schemes change constantly
- Advantages:
  - High precision searches
  - Works well for valuable, closed collections like books in a library

## Examples, fixed indexing languages

Medical Subject Headings (MeSH)	
...	
Eye Diseases	C11
Asthenopia	C11.93
Conjunctival Diseases	C11.187
Conjunctival Neoplasms	C11.187.169
Conjunctivitis	C11.187.183
Conjunctivitis, Allergic	C11.187.183.200
Conjunctivitis, Bacterial	C11.187.183.220
Conjunctivitis, Inclusion	C11.187.183.220.250
Ophthalmia Neonatorum	C11.187.183.220.538
Trachoma	C11.187.183.220.889
Conjunctivitis, Viral	C11.187.183.240
Conjunctivitis, Acute Hemorrhagic	C11.187.183.240.216
Keratoconjunctivitis	C11.187.183.394
Keratoconjunctivitis, Infectious	C11.187.183.394.520
Keratoconjunctivitis Sicca	C11.187.183.394.550
Reiter's Disease	C11.187.183.749
Pterygium	C11.187.781
Xerophthalmia	C11.187.810
...	

ACM Computing Classification System (1998)	
B	Hardware
B.3	Memory structures
B.3.0	General
B.3.1	Semiconductor Memories <a href="#">(NEW)</a> <a href="#">(was B.7.1)</a>
	Dynamic memory (DRAM) <a href="#">(NEW)</a>
	Read-only memory (ROM) <a href="#">(NEW)</a>
	Static memory (SRAM) <a href="#">(NEW)</a>
B.3.2	Design Styles <a href="#">(was D.4.2)</a>
	Associative memories
	Cache memories
	Interleaved memories
	Mass storage (e.g., magnetic, optical, RAID)
	Primary memory
	Sequential-access memory
	Shared memory
	Virtual memory
B.3.3	Performance Analysis and Design Aids
	Formal models
	Simulation
	Worst-case analysis
B.3.4	Reliability, Testing, and Fault-Tolerance
	Diagnostics
	Error-checking
	Redundant design
	Test generation
...	

- No predefined set of index terms
- Instead: use natural language as indexing language
- Mappings words → meanings is not 1:1
  - Synonymy (n words : 1 meaning) sofa – couch
  - Polysemy (1 word : n meanings) bank – bank
- Do the terms get manipulated?
  - De-capitalised? Turkey – turkey
  - Stemmed? advice – advised
  - Stemmed and POS-tagged? can – can
- Use important phrases, instead of single words  
cheque book (rather than cheque and book)

Implementation of indexes: inverted files

## Inverted files

Doc 1	
Except	Russia
and	Mexico
no	country
had	had
the	decency
to	come
to	the
rescue	of
the	government.

Doc 2	
It	was
a	dark
and	stormy
night	in
the	country
manor.	The
time	was
past	midnight.

Term	Doc no	Freq	Offset
a	2	1	2
and	1	1	2
and	2	1	4
come	1	1	11
country	1	1	5
country	2	1	9
dark	2	1	3
decency	1	1	9
except	1	1	0
government	1	1	17
had	1	2	6,7
in	2	1	7
it	2	1	0
manor	2	1	10
mexico	1	1	3
midnight	2	1	17
night	2	1	6
no	1	1	4
of	1	1	15
past	2	1	15
rescue	1	1	14
russia	1	1	1
stormy	2	1	5
the	1	2	8,13
the	2	2	8,12
time	2	1	14
to	1	2	10,12
was	2	2	16

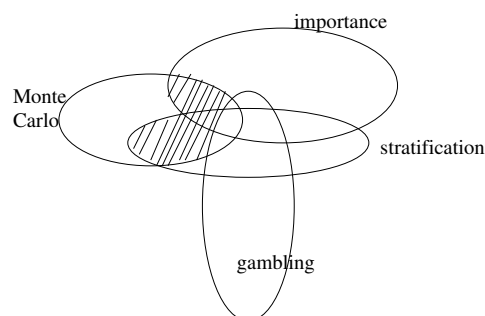
Information kept for each term:

- Document ID where this term occurs
- Frequency of occurrence of this term in each document
- Possibly: Offset of this term in document

- Boolean search
  - Binary decision: Document is relevant or not (no ranking)
  - Presence of term is necessary and sufficient for match
  - Boolean operators are set operations (AND, OR, NOT, BUT)
- Ranked algorithms
  - Ranking takes frequency of terms in document into account
  - Not all search terms necessarily present in document
  - Incarnations:
    - \* The vector space model (SMART, Salton et al, 1971)
    - \* The probabilistic model (OKAPI, Robertson/Spärck Jones, 1976)
    - \* Web search engines

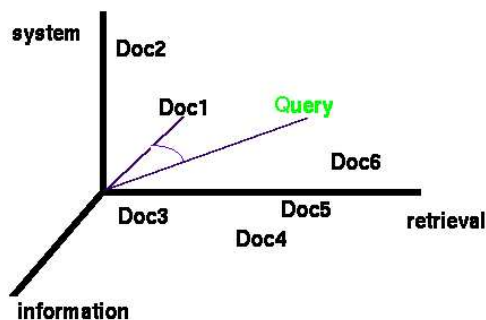
## The Boolean model

Monte Carlo AND (importance OR stratification) BUT gambling



- Set theoretic interpretation of connectors AND OR BUT
- Often in use for bibliographic search engines (library)
- Problem 1: Expert knowledge necessary to create high-precision queries
- Problem 2: Binary relevance definition → unranked result lists (frustrating, time consuming)

- A document is represented as a point in high-dimensional vector space
- Query is also represented in vector space
- Select document(s) with highest document–query similarity
- Document–query similarity is model for relevance → ranking



3-dimensional term vector space:

- Dimension 1: “information”
- Dimension 2: “retrieval”
- Dimension 3: “system”

## Documents and queries in term feature space

	Doc <sub>1</sub>	Doc <sub>2</sub>	Doc <sub>3</sub>	...	Doc <sub>n</sub>		Q
term <sub>1</sub>	14	6	1	...	0	↔	0
term <sub>2</sub>	0	1	3	...	1	↔	1
term <sub>3</sub>	0	1	0	...	2	↔	0
...	...	...	...	...	...	↔	...
term <sub>N</sub>	4	7	0	...	5	↔	1

Decisions to take:

1. Choose dimensionality of vector: what counts as a term?
2. Choose weights for each term/document mapping (cell)
  - presence or absence (binary)
  - term frequency in document
  - more complicated weight, eg. TF\*IDF (cf. later in lecture)
3. Choose a proximity measure

A **proximity measure** can be defined either by similarity or dissimilarity. Proximity measures are

- **Symmetric** ( $\forall i, j : d(j, i) = d(i, j)$ )
- **Maximal/minimal for identity:**
  - For similarity measures:  $\forall i : d(i, i) = \max_k d(i, k)$
  - For dissimilarity measures:  $\forall i : d(i, i) = 0$
- A **distance metric** is a dissimilarity metric that satisfies the triangle inequality
 
$$\forall i, j, k : d(i, j) + d(i, k) \geq d(j, k)$$
- **Distance metrics are non-negative:**  $\forall i, k : d(i, k) \geq 0$

## Similarity measures, binary

X is the set of all terms occurring in document  $D_X$ , Y is the set of all terms occurring in document  $D_Y$ .

- **Raw Overlap:**  $raw\_overlap(X, Y) = |X \cap Y|$
- **Dice's coefficient:** (normalisation by average size of the two original vectors)

$$dice(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}$$

- **Jaccard's coefficient:** (normalisation by size of combined vector – penalises small number of shared feature values)

$$jacc(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

- **Overlap coefficient:**

$$overlap\_coeff(X, Y) = \frac{|X \cap Y|}{\min(|X|, |Y|)}$$

- **Cosine:** (normalisation by vector lengths)

$$cosine(X, Y) = \frac{|X \cap Y|}{\sqrt{|X|} \cdot \sqrt{|Y|}}$$

Weighted versions of Dice's and Jaccard's coefficient exist, but are used rarely for IR:

- Vectors are extremely sparse
- Vectors are of very differing length

**Cosine** (or normalised inner product) is the measure of choice for IR

Document  $i$  is represented as a vectors of terms or lemmas ( $\vec{w}_i$ );  $t$  is the total number of index terms in system,  $w_{i,j}$  is the weight associated with  $j$  th term of vector  $\vec{w}_i$ .

Vector length normalisation by the two vectors  $|\vec{w}_i|$  and  $|\vec{w}_k|$ :

$$\cos(\vec{w}_i, \vec{w}_k) = \frac{\vec{w}_i \cdot \vec{w}_k}{|\vec{w}_i| \cdot |\vec{w}_k|} = \frac{\sum_{j=1}^d w_{i,j} \cdot w_{k,j}}{\sqrt{\sum_{j=1}^d w_{i,j}^2} \cdot \sqrt{\sum_{j=1}^d w_{k,j}^2}}$$

## Distance measures

- **Euclidean distance:** (how far apart in vector space)

$$euc(\vec{w}_i, \vec{w}_k) = \sqrt{\sum_{j=1}^d (w_{i,j} - w_{k,j})^2}$$

- **Manhattan distance:** (how far apart, measured in 'city blocks')

$$manh(\vec{w}_i, \vec{w}_k) = \sum_{j=1}^d |w_{i,j} - w_{k,j}|$$



Most frequent words in a large language sample, with frequencies:

Rank	English (BNC)	German	
1	the	61847	der
2	of	29391	die
3	and	26817	und
4	a	21626	in
5	in	18214	den
6	to	16284	von
7	it	10875	zu
8	is	9982	das
9	to	9343	mit
10	was	9236	sich
11	I	8875	des
12	for	8412	auf
13	that	7308	für
14	you	6954	ist
15	he	6810	im

Zipf's Law: The frequency rank of a word is reciprocally proportional to its frequency:

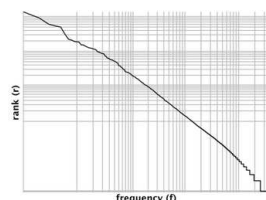
$$\text{freq}(\text{word}_i) = \frac{1}{i^\theta} \text{freq}(\text{word}_1)$$

( $\text{word}_i$  is the  $i$ th most frequent word of the language);  $1.5 < \theta < 2$  for most languages)

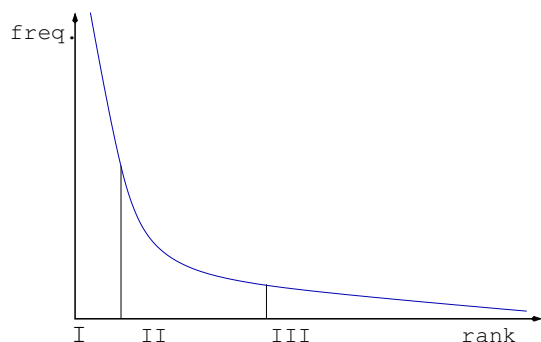
## Other collections (allegedly) obeying Zipf's law

50

- Sizes of settlements
- Frequency of access to web pages
- Income distributions amongst top earning 3% individuals
- Korean family names
- Size of earth quakes
- Word senses per word
- Notes in musical performances
- ...



Plotting a Zipfian distribution on a log-scale:



- **Zone I:** High frequency words tend to be function words. Top 135 vocabulary items account for 50% of words in the Brown corpus. These are not important for IR. Can be excluded by **stoplist** (e.g. SMART stoplist), containing words such as “the”, “is” and “this”.
- **Zone II:** Mid-frequency words are the best indicators of what the document is about
- **Zone III:** Low frequency words tend to be typos or overly specific words (not important, for a different reason) (“Uni7ed”, “super-noninteresting”, “87-year-old”, “0.07685”)

## Term Weighting: TF\*IDF

Not all terms describe a document equally well:

- Terms which are **frequent** in a document are better  $\rightarrow tf_{w,d} = freq_{w,d}$  should be high
- Terms that are **overall rare** in the document collection are better  $\rightarrow idf_{w,D} = \log \frac{|D|}{n_{w,D}}$  should be high  $\rightarrow$
- TF\*IDF formula:  $tf * idf_{w,d,D} = tf_{w,d} * idf_{w,D}$  should be high
- Improvement: **Normalise**  $tf_{w,d}$  by term frequency of most frequent term in document:  $tf_{norm,w,d} = \frac{freq_{w,d}}{\max_{l \in d} freq_{l,d}}$ 
  - Normalised TF\*IDF:  $tf * idf_{norm,w,d,D} = tf_{norm,w,d} * idf_{w,D}$

$tf_{w,d}$ :	Term frequency of word $w$ in document $d$
$n_{w,D}$ :	Number of documents in document collection $D$ which contain word $w$
$idf_{w,D}$ :	Inverse document frequency of word $w$ in document collection $D$
$tf * idf_{w,d,D}$ :	TF*IDF weight of word $w$ in document $d$ in document collection $D$
$tf * idf_{norm,w,d,D}$ :	Length-normalised TF*IDF weight of word $w$ in document $d$ in document collection $D$
$tf_{norm,w,d}$ :	Normalised term frequency of word $w$ in document $d$
$\max_{l \in d} freq_{l,d}$ :	Maximum term frequency of any word in document $d$

Document set: 30,000

Term	tf	$n_{w,D}$	TF*IDF
the	312	28,799	5.55
in	179	26,452	9.78
general	136	179	302.50
fact	131	231	276.87
explosives	63	98	156.61
nations	45	142	104.62
1	44	2,435	47.99
haven	37	227	78.48
2-year-old	1	4	3.88

$$\text{IDF}(\text{"the"}) = \log\left(\frac{30,000}{28,799}\right) = 0.0178$$

$$\text{TF*IDF}(\text{"the"}) = 312 \cdot 0.0178 = 5.55$$

Example: VSM (TF\*IDF; cosine)

Query: hunter gatherer Scandinavia

	Q	D <sub>7655</sub>	D <sub>454</sub>
hunter	19.2	56.4	112.2
gatherer	34.5	122.4	0
Scandinavia	13.9	0	30.9
30,000	0	457.2	0
years	0	12.4	0
BC	0	200.2	0
prehistoric	0	45.3	0
deer	0	0	23.6
rifle	0	0	452.2
Mesolithic	0	344.2	0
barber	0	0	25.2
household	0	204.7	0
...	...	...	...

(Normally there would be many more terms in D<sub>7655</sub> and D<sub>454</sub>)

$$\cos(Q, D_{7655}) = \frac{19.2 \cdot 56.4 + 34.5 \cdot 122.4 + 13.9 \cdot 0}{\sqrt{19.2^2 + 34.5^2 + 13.9^2} \cdot \sqrt{56.4^2 + 122.4^2 + 457.2^2 + 12.4^2 + 200.2^2 + 45.3^2 + 344.2^2 + 204.7^2 + \dots}} = .1933303426$$

$$\cos(Q, D_{454}) = \frac{19.2 \cdot 112.2 + 34.5 \cdot 0 + 13.9 \cdot 30.9}{\sqrt{19.2^2 + 34.5^2 + 13.9^2} \cdot \sqrt{112.2^2 + 30.9^2 + 23.6^2 + 452.2^2 + 25.2^2 + \dots}} = .1318349238$$

→ choose document D<sub>7655</sub>

- Build a document-term matrix for three (very!) short documents of your choice
- Weight by presence/absence (binary) and by TF\*IDF (with estimated IDF's)
- Write a suitable query
- Calculate document–query similarity, using
  - cosine
  - inner product (i.e. cosine without normalisation)
- What effect does normalisation have?

## Term Manipulation

- So far: each term is indexed and weighted only in string-equal form
- This misses many semantic similarities between morphologically related words (“whale” → “whaling”, “whales”)
- Automatic models of term identity
  - The same string between blanks or punctuation
  - The same prefix (eg. up to 6 characters)
  - The same stem (e.g. Porter stemmer)
  - The same linguistic lemma (sensitive to Parts-of-speech)
- Effect of term manipulation on retrieval result
  - changes the counts, reduces total number of terms
  - increases recall
  - might decrease precision, introduction of noise

M. Porter, “An algorithm for suffix stripping”,  
Program 14(3):130-137, 1980

CONNECT
CONNECTED
CONNECTING
CONNECTION
CONNECTIONS

- Removal of suffixes without a stem dictionary, only with a suffix dictionary
- Terms with a common stem have similar meanings:
- Deals with inflectional and derivational morphology
- Conflates relate — relativity — relationship
- Sees no difference between sand — sander and wand — wander (does not conflate either)
- Root changes (deceive/deception, resume/resumption) aren't dealt with, but these are rare

## Stemming: Representation of a word

$[C] (VC)\{m\}[V]$

C	one or more adjacent consonants
V	one or more adjacent vowels
[ ]	optionality
( )	group operator
{x}	repetition x times
m	the “measure” of a word

shoe	$[sh]_c[oe]_v$	m=0
Mississippi	$[M]_c([i]_v[ss]_c)([i]_v[ss]_c)([i]_v[pp]_c)[i]_v$	m=3
ears	$([ea]_v[rs]_c)$	m=1

Notation: m is calculated on the word excluding the suffix of the rule under consideration (eg. In m=1 for 'element' in rule “(m > 1) EMENT”, so this rule would not trigger.)

Rules in one block are run through in top-to-bottom order; when a condition is met, execute rule and jump to next block

Rules express criteria under which suffix may be removed from a word to leave a valid stem: (condition)  $S1 \rightarrow S2$

Possible conditions:

- constraining the measure:

$(m > 1) \text{ EMENT} \rightarrow \epsilon$  ( $\epsilon$  is the empty string)

$\text{REPLACEMENT} \rightarrow \text{REPLAC}$

- constraining the shape of the word piece:

- \*S – the stem ends with S

- \*v\* – the stem contains a vowel

- \*d – the stem ends with a double consonant (e.g. -TT, -SS).

- \*o – the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP)

- expressions with AND, OR and NOT:

- $(m > 1 \text{ AND } (*S \text{ OR } *T))$  – a stem with  $m > 1$  ending in S or T

## Porter stemmer: selected rules

<p>SSES <math>\rightarrow</math> SS          IES <math>\rightarrow</math> I          SS <math>\rightarrow</math> SS          S <math>\rightarrow</math></p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

caresses  $\rightarrow$  caress (Rule 1 fires  $\checkmark$ )

cares  $\rightarrow$  care (Rule 4 fires  $\checkmark$ )

<p><math>(m &gt; 0) \text{ EED} \rightarrow \text{EE}</math></p>
------------------------------------------------------------------

feed  $\rightarrow$  feed (Rule does not fire  $\checkmark$ )

agreed  $\rightarrow$  agree (Rule fires  $\checkmark$ )

BUT: freed (should fire but doesn't),  
 succeed (fires but shouldn't)

<p><math>(*v^*) \text{ ED} \rightarrow</math></p>
---------------------------------------------------

plastered  $\rightarrow$  plaster (Rule fires  $\checkmark$ )

bled  $\rightarrow$  bled (Rule doesn't fire  $\checkmark$ )

## Step 1: plurals and past participles

## Step 1a

SSES	→ SS	caresses	→ caress
IES	→ I	ponies	→ poni
		ties	→ ti
SS	→ SS	caress	→ caress
S	→ ε	cats	→ cat

## Step 1b

(m>0)	EED	→ EE	feed	→ feed
			agreed	→ agree
(*v*)	ED	→ ε	plastered	→ plaster
			bled	→ bled
(*v*)	ING	→ ε	motoring	→ motor
			sing	→ sing

If rule 2 or 3 in Step 1b applied, then clean up:

AT	→ ATE	conflat(ed/ing)	→ conflate
BL	→ BLE	troubl(ed/ing)	→ trouble
IZ	→ ISE	populariz(ed/ing)	→ popularise
IS	→ ISE	popularis(ed/ing)	→ popularise
(*d and not (*L or *S or *Z))	→ single letter	hopp(ed/ing)	→ hop
		hiss(ed/ing)	→ hiss
(m=1 and *o)	→ E	fil(ed/ing)	→ file
		fail(ed/ing)	→ fail

## Step 1c

(*v*)	Y	→ I	happy	→ happi
			sky	→ sky

## Step 2: derivational morphology

(m>0)	ATIONAL	→ ATE	relational	→ relate
(m>0)	TIONAL	→ TION	conditional	→ condition
			rational	→ rational
(m>0)	ENCI	→ ENCE	valenci	→ valence
(m>0)	ANCI	→ ANCE	hesitanci	→ hesitance
(m>0)	IZER	→ IZE	digitizer	→ digitize
(m>0)	ABLI	→ ABLE	conformabli	→ conformable
(m>0)	ALLI	→ AL	radicalli	→ radical
(m>0)	ENTLI	→ ENT	differentli	→ different
(m>0)	ELI	→ E	vileli	→ vile
(m>0)	OUSLI	→ OUS	analogousli	→ analogous
(m>0)	IZATION	→ ISE	vietnamization	→ vietnamise
(m>0)	ISATION	→ ISE	vietnamisation	→ vietnamise
(m>0)	ATION	→ ATE	predication	→ predicate
(m>0)	ATOR	→ ATE	operator	→ operate
(m>0)	ALISM	→ AL	feudalism	→ feudal
(m>0)	IVENESS	→ IVE	decisiveness	→ decisive
(m>0)	FULNESS	→ FUL	hopefulness	→ hopeful
(m>0)	OUSNESS	→ OUS	callousness	→ callous
(m>0)	ALITI	→ AL	formaliti	→ formal
(m>0)	IVITI	→ IVE	sensitiviti	→ sensitive
(m>0)	BILITI	→ BLE	sensibiliti	→ sensible

## Step 3: more derivational morphology

(m>0)	ICATE	→	IC	triplicate	→	triplic
(m>0)	ATIVE	→	ε	formative	→	form
(m>0)	ALIZE	→	AL	formalize	→	formal
(m>0)	ALISE	→	AL	formalise	→	formal
(m>0)	ICITI	→	IC	electriciti	→	electric
(m>0)	ICAL	→	IC	electrical	→	electric
(m>0)	FUL	→	ε	hopeful	→	hope
(m>0)	NESS	→	ε	goodness	→	good

#### Step 4: even more derivational morphology

(m>1)	AL	→	ε	revival	→	reviv
(m>1)	ANCE	→	ε	allowance	→	allow
(m>1)	ENCE	→	ε	inference	→	infer
(m>1)	ER	→	ε	airliner	→	airlin
(m>1)	IC	→	ε	gyroscopic	→	gyroscop
(m>1)	ABLE	→	ε	adjustable	→	adjust
(m>1)	IBLE	→	ε	defensible	→	defens
(m>1)	ANT	→	ε	irritant	→	irrit
(m>1)	EMENT	→	ε	replacement	→	replac
(m>1)	MENT	→	ε	adjustment	→	adjust
(m>1)	ENT	→	ε	dependent	→	depend
(m>1 and (*S or *T))	ION	→	ε	adoption	→	adopt
(m>1)	OU	→	ε	homologou	→	homolog
(m>1)	ISM	→	ε	communism	→	commun
(m>1)	ATE	→	ε	activate	→	activ
(m>1)	ITI	→	ε	angulariti	→	angular
(m>1)	OUS	→	ε	homologous	→	homolog
(m>1)	IVE	→	ε	effective	→	effect
(m>1)	ISE	→	ε	bowdlerise	→	bowdler
(m>1)	IZE	→	ε	bowdlerize	→	bowdler

#### Step 5: cleaning up

##### Step 5a

(m>1)	E	→	ε	probate	→	probat
				rate	→	rate
(m=1 and not *o)	E	→	ε	cease	→	ceas

##### Step 5b

(m > 1 and *d and *L)	→	single letter	controll	→	control
			roll	→	roll



1. Show which stems *rationalisations*, *rational*, *rationalizing* result in, and which rules they use.
2. Explain why *sander* and *sand* do not get conflated.
3. What would you have to change if you wanted to conflate them?
4. Find five different examples of incorrect stemmings.
5. Can you find a word that gets reduced in every single step (of the 5)?
6. Exemplify the effect that stemming (eg. with Porter) has on the Vector Space Model, using your example from before.

- Indexing languages
- Retrieval models
- Term weighting
- Term stemming

Textbook (Baeza-Yates and Ribeiro-Neto):

- 2.5.2 Boolean model
- 6.3.3 Zipf's law
- 2.5.3 Vector space model, TF\*IDF
- 7.2 Term manipulation, stemming

# Information Retrieval

## Lecture 3: Evaluation methodology

Computer Science Tripos Part II



UNIVERSITY OF  
CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

Today

68

- 
1. General concepts in IR evaluation
  2. The TREC competitions
  3. IR evaluation metrics

- 
- IR system
    - in: a query
    - out: relevant documents
  - Evaluation of IR systems
  - Goal: predict future from past experience
  - Reasons why IR evaluation is hard:
    - Large variation in human information needs and queries
    - The precise contributions of each component are hard to entangle:
      - \* Collection coverage
      - \* Document indexing
      - \* Query formulation
      - \* Matching algorithm

- 
- Test only “system parameters”
    - Index language devices for description and search
    - Methods of term choice for documents
    - Matching algorithm
    - Type of user interface
  - Ignore environment variables
    - Properties of documents → use many documents
    - Properties of users → use many queries

- Text REtrieval Conference since 1992
- Run by NIST (US National Institute of Standards and Technology)
- 60–80 commercial and research groups participate each year
- Marks a new phase in retrieval evaluation
  - common task and data set
  - many participants
  - continuity
- Large test collection: text, queries, relevance judgements
  - Queries devised and judged by information specialist (same person)
  - Relevance judgements done only for up to 1000 documents/query

## Today's test collections

---

A test collection consists of:

- Document set:
  - TREC-3: 742,000 documents
  - TREC Web-track: small snapshot of the web
- Queries/Topics
  - short description of information need
  - TREC “topics”: longer description detailing relevance criteria
  - “frozen” → reusable
- Relevance judgements
  - binary
  - done by same person who created the query

- capture user (writer) variation
  - support claims of statistical significance in results
  - demonstrate that systems scale up → commercial credibility
- 
- In 60s and 70s, very small test collections, arbitrarily different, one per project
    - in 60s: 35 queries on 82 documents
    - in 1990: still only 35 queries on 2000 documents

## Sample TREC query

<num> Number: 508

<title> hair loss is a symptom of what diseases

<desc> Description:

Find diseases for which hair loss is a symptom.

<narr> Narrative:

A document is relevant if it positively connects the loss of head hair in humans with a specific disease. In this context, "thinning hair" and "hair loss" are synonymous. Loss of body and/or facial hair is irrelevant, as is hair loss caused by drug therapy.

	Relevant	Non-relevant	Total
Retrieved	A	B	A+B
Not retrieved	C	D	C+D
Total	A+C	B+D	A+B+C+D

**Recall:** proportion of retrieved items amongst the relevant items ( $\frac{A}{A+C}$ )

**Precision:** proportion of relevant items amongst retrieved items ( $\frac{A}{A+B}$ )

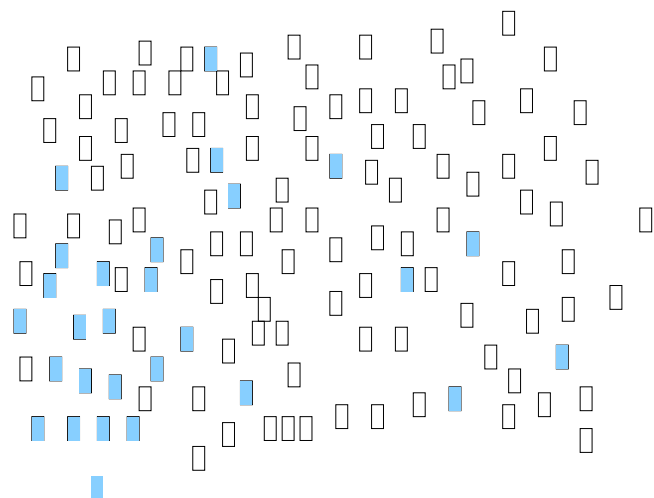
**Accuracy:** proportion of correctly classified items as relevant/irrelevant ( $\frac{A+D}{A+B+C+D}$ )

Recall: [0..1]; Precision: [0..1]; Accuracy: [0..1]

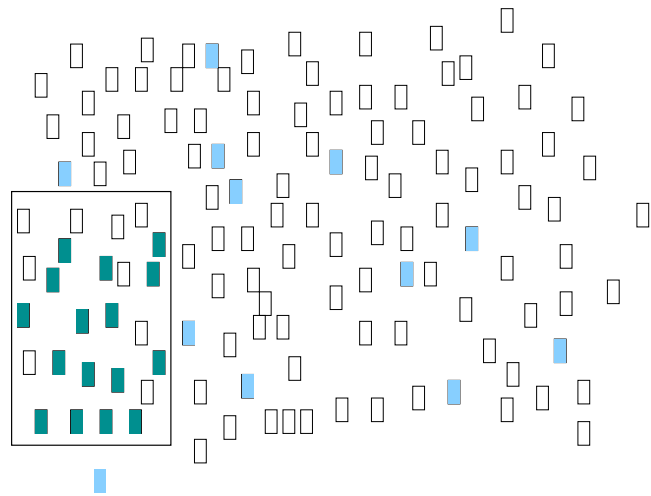
Accuracy is not a good measure for IR, as it conflates performance on relevant items (A) with performance on irrelevant (uninteresting) items (D)

## Recall and Precision

- All documents:  
A+B+C+D = 130
- Relevant documents  
for a given query:  
A+C = 28



- System 1 retrieves 25 items:  $(A+B)_1 = 25$
- Relevant and retrieved items:  $A_1 = 16$

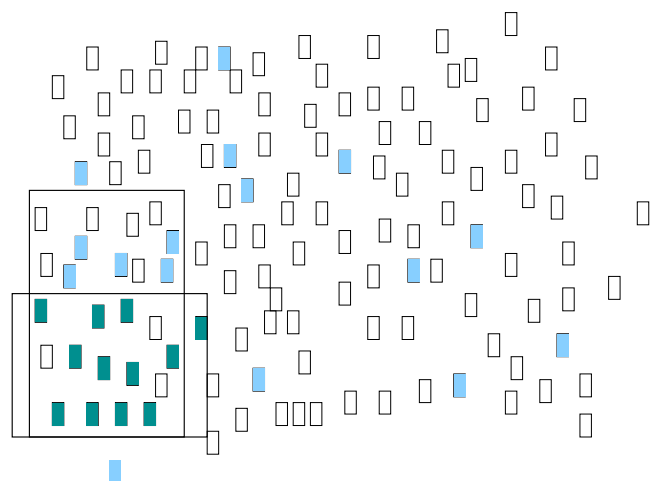


$$R_1 = \frac{A_1}{A+C} = \frac{16}{28} = .57$$

$$P_1 = \frac{A_1}{(A+B)_1} = \frac{16}{25} = .64$$

$$A_1 = \frac{A_1 + D_1}{A+B+C+D} = \frac{16+93}{130} = .84$$

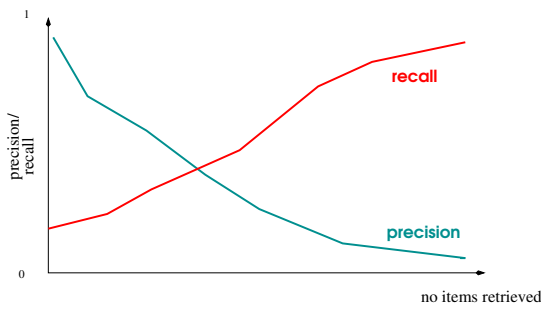
- System B retrieves set  $(A+B)_2 = 15$  items
- $A_2 = 12$



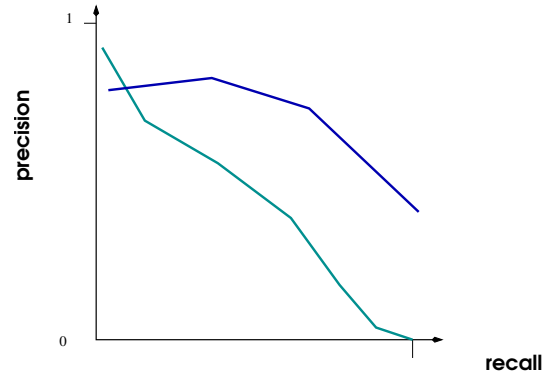
$$R_2 = \frac{12}{28} = .43$$

$$P_2 = \frac{12}{15} = .8$$

$$A_2 = \frac{12+99}{130} = .85$$



- Plotting precision and recall (versus no. of documents retrieved) shows inverse relationship between precision and recall
- Precision/recall cross-over can be used as combined evaluation measure



- Plotting precision versus recall gives recall-precision curve
- Area under normalised recall-precision curve can be used as evaluation measure

# Recall-criticality and precision-criticality

- Inverse relationship between precision and recall forces general systems to go for compromise between them
- But some tasks particularly need good precision whereas others need good recall:

Precision-critical task	Recall-critical task
Little time available	Time matters less
A small set of relevant documents answers the information need	One cannot afford to miss a single document
Potentially many documents might fill the information need (redundantly)	Need to see <i>each</i> relevant document
Example: web search for factual information	Example: patent search



- Users disagree **with each other** in what they find relevant
- One reason is that document can be relevant to a query in different ways:
  - Answer precise question precisely
  - Partially answer question
  - Give background information
  - Remind the user of other knowledge
  - Suggest a source for more information
  - Be objectively relevant, although user knows the document already
- But users also disagree with **themselves** (over time) about what is relevant.
- Problem: comparability of systems under unstable conditions

- Countermeasure A: Use guidelines
  - Relevance defined independently of novelty
  - Then, relevance decisions are independent of each other
- Countermeasure B: Use large samples of queries. This counteracts natural variation in users and information needs
- Then: Relative success measurements on systems stable across judges (but not necessarily absolute ones) (Voorhees, 2000)
- Stable **relative** measures are good enough for comparing systems



Humans decide which document–query pairs are relevant.

## F-measure

- Weighted harmonic mean of P and R (Rijsbergen 1979)

$$F_{\alpha} = \frac{PR}{(1 - \alpha)P + \alpha R}$$

- High  $\alpha$ : Precision is more important
- Low  $\alpha$ : Recall is more important

- Most commonly used with  $\alpha=0.5 \rightarrow$

$$F_{0.5} = \frac{2PR}{P + R}$$

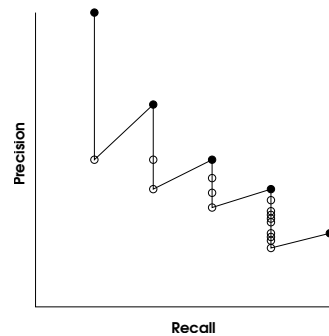
- Maximum value of  $F_{0.5}$ -measure (or F-measure for short) is a good indication of best P/R compromise
- F-measure is an approximation of cross-over point of precision and recall

- Recall problem: for a collection of non-trivial size, it becomes impossible to inspect each document
- It would take 6500 hours to judge 800,000 documents for **one** query (30 sec/document)
- Pooling addresses this problem

Pooling (Sparck Jones and van Rijsbergen, 1975)

- Pool is constructed by putting together top  $N$  retrieval results from a set of  $n$  systems (TREC:  $N = 100$ )
- Humans judge every document in this pool
- Documents outside the pool are automatically considered to be irrelevant
- There is overlap in returned documents: pool is smaller than theoretical maximum of  $N \cdot n$  systems (around  $\frac{1}{3}$  the maximum size)
- Pooling works best if the approaches used are very different
- Large increase in pool quality by manual runs which are recall-oriented, in order to supplement pools

- With ranked list of return documents there are many P/R data points
- Sensible P/R data points are those after each new relevant document has been seen (black points)



A retrieval table:

Rank	Relev.	R	P
1	X	0.20	1.00
2	"	"	0.50
3	X	0.40	0.67
4	"	"	0.50
5	"	"	0.40
6	X	0.60	0.50
7	"	"	0.43
8	"	"	0.38
9	"	"	0.33
10	X	0.80	0.40
11	"	"	0.36
12	"	"	0.33
13	"	"	0.31
14	"	"	0.29
15	"	"	0.27
16	"	"	0.25
17	"	"	0.24
18	"	"	0.22
19	"	"	0.21
20	X	1.00	0.25

## Summary IR measures

- Precision at a certain rank:  $P(100)$
- Precision at a certain recall value:  $P(R=.2)$
- Precision at last relevant document:  $P(\text{last\_relev})$
- Recall at a fixed rank:  $R(100)$
- Recall at a certain precision value:  $R(P=.1)$

- Want to average over queries
- Problem: queries have differing number of relevant documents
- Cannot use one single cut-off level for all queries
  - This would not allow systems to achieve the theoretically possible maximal values in all conditions
  - Example: if a query has 10 relevant documents
    - \* If cutoff  $> 10$ ,  $P < 1$  for all systems
    - \* If cutoff  $< 10$ ,  $R < 1$  for all systems
- Therefore, more complicated joint measures are required

## Mean Average Precision (MAP)

- Also called “average precision at seen relevant documents”
- Determine precision at each point when a new relevant document gets retrieved
- Use  $P=0$  for each relevant document that was not retrieved
- Determine average for each query, then average over queries

$$MAP = \frac{1}{N} \sum_{j=1}^N \frac{1}{Q_j} \sum_{i=1}^{Q_j} P(doc_i)$$

with:

- $Q_j$  number of relevant documents for query  $j$
- $N$  number of queries
- $P(doc_i)$  precision at  $i$ th relevant document

Query 1		
Rank	Relev.	$P(doc_i)$
1	X	1.00
2		
3	X	0.67
4		
5		
6	X	0.50
7		
8		
9		
10	X	0.40
11		
12		
13		
14		
15		
16		
17		
18		
19		
20	X	0.25
AVG:		0.564

Query 2		
Rank	Relev.	$P(doc_i)$
1	X	1.00
2		
3	X	0.67
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15	X	0.2
AVG:		0.623

- MAP favours systems which return relevant documents **fast**
- Precision-biased

$$MAP = \frac{0.564+0.623}{2} = 0.594$$

## 11 point average precision

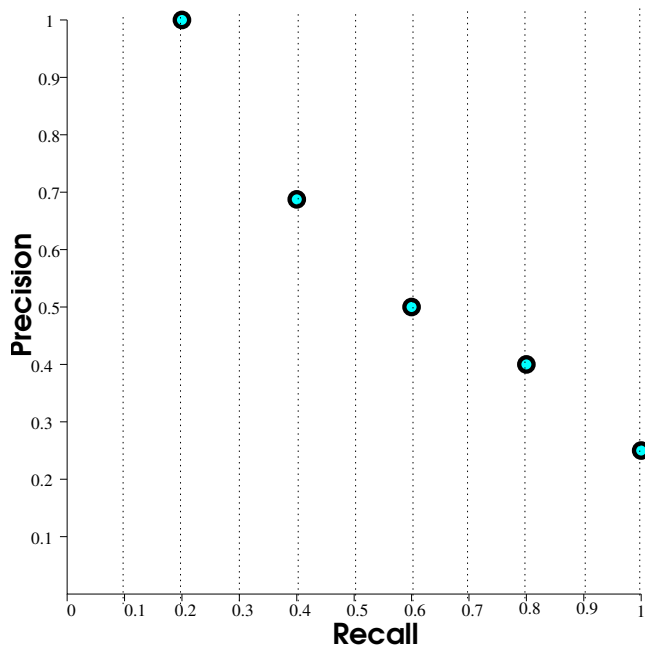
$$P_{11-pt} = \frac{1}{11} \sum_{j=0}^{10} \frac{1}{N} \sum_{i=1}^N \tilde{P}_i(r_j)$$

with  $\tilde{P}_i(r_j)$  the precision at the  $j$ th recall point in the  $i$ th query (out of  $N$  queries)

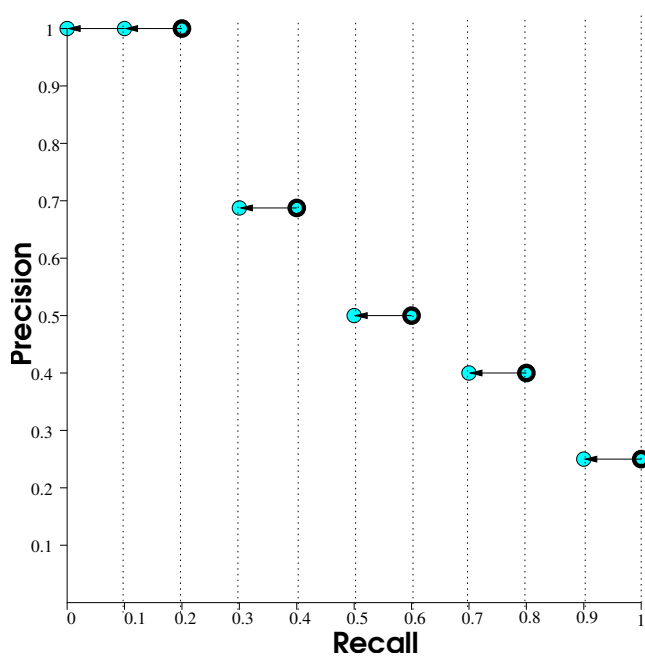
- Define 11 standard recall points  $r_j = \frac{j}{10}$ :  $r_0 = 0, r_1 = 0.1 \dots r_{10} = 1$
- We need  $\tilde{P}_i(r_j)$ ; i.e. the precision at our recall points
- $P_i(R = r)$  can be measured: the precision at each point when recall changes (because a new relevant document is retrieved)
- Problem: unless the number of relevant documents per query is divisible by 10,  $\tilde{P}_i(r_j)$  does not coincide with a measurable data point  $r$
- Solution: interpolation

$$\tilde{P}_i(r_j) = \begin{cases} \max(r_j \leq r < r_{j+1}) P_i(R = r) & \text{if } P_i(R = r) \text{ exists} \\ \tilde{P}_i(r_{j+1}) & \text{otherwise} \end{cases}$$

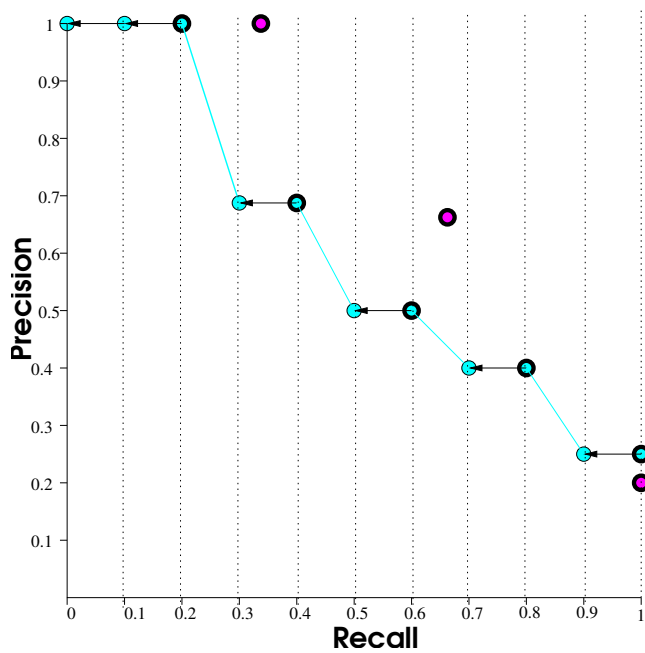
- Note that  $P_i(R = 1)$  can always be measured.



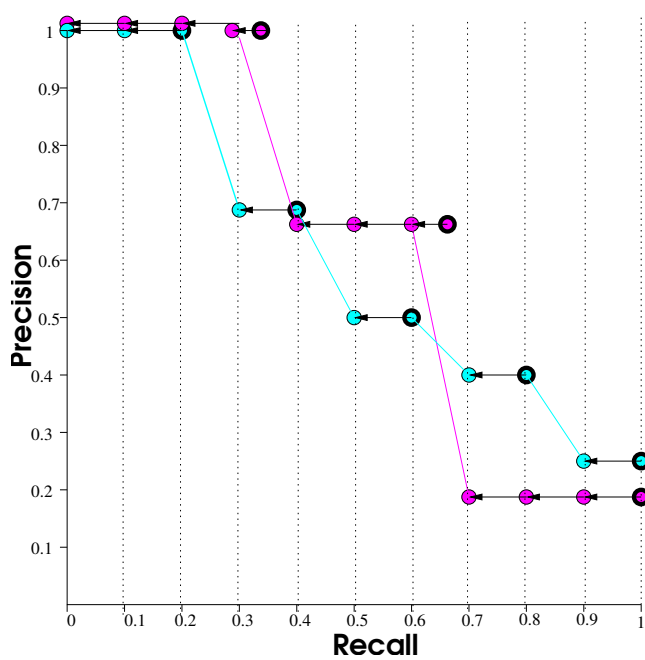
- Blue for Query 1
- Bold Circles measured
- Five  $r_j$ s coincide with data-point



- Blue for Query 1
- Bold Circles measured
- Thin circles interpolated

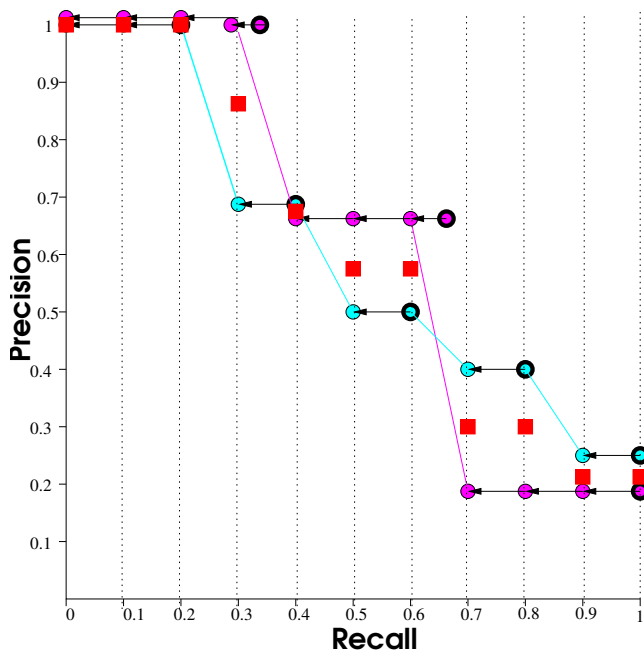


- Red for Query 2
- Bold Circles are measured
- Only  $r_{10}$  coincides with a data point



- Red for Query 2
- Bold Circles measured
- Thin circles interpolated





- Now average at each  $p_r$
- over N (number of queries)
- → 11 data points
- Numerical result is an approximation of area under recall/precision curve

Query 1			$P_1(r_i)$	$\sum_{j=1}^N P_j(r_i)$	$P_2(r_i)$	Query 2		
#	X	R				R	X	#
1	X	0.20	$\tilde{P}_1(r_0) = 1.00 \rightarrow$	1.00	$\leftarrow$	0.33	X	1
2			$\tilde{P}_1(r_1) = 1.00 \rightarrow$	1.00	$\leftarrow$			2
3	X	0.40	$\tilde{P}_1(r_2) = P_1(R = .2) = 1.00 \rightarrow$	1.00	$\leftarrow$	0.67	X	3
4			$\tilde{P}_1(r_3) = 0.67 \rightarrow$	0.84	$\leftarrow$			4
5			$\tilde{P}_1(r_4) = P_1(R = .4) = 0.67 \rightarrow$	0.67	$\leftarrow$			5
6	X	0.60	$\tilde{P}_1(r_5) = 0.50 \rightarrow$	0.59	$\leftarrow$			6
7			$\tilde{P}_1(r_6) = P_1(R = .6) = 0.50 \rightarrow$	0.59	$\leftarrow$			7
8			$\tilde{P}_1(r_7) = 0.40 \rightarrow$	0.30	$\leftarrow$			8
9			$\tilde{P}_1(r_8) = P_1(R = .8) = 0.40 \rightarrow$	0.30	$\leftarrow$			9
10	X	0.80	$\tilde{P}_1(r_9) = 0.25 \rightarrow$	0.23	$\leftarrow$			10
11								11
12								12
13								13
14								14
15								15
16								16
17				0.23	$\leftarrow$			17
18								18
19								19
20	X	1.00	$\tilde{P}_1(r_{10}) = P_1(R = 1.0) = 0.25 \nearrow$		$\leftarrow$		X	20
				$P_{11-pt} = 0.61$				

$\tilde{P}_i(r_j)$  is (interpolated) precision of  $i$ th query, at  $j$ th recall point.  $P_i(R = r_j)$  (black) are exactly measured precision values.

- TREC-7 and 8: P(30) between .40 and .45, using long queries and narratives (one team even for short queries); P(10) = .5 even with short queries, > .5 with medium length queries
- Systems must have improved since TREC-4, 5, and 6 → manual performance (sanity check) remained on a plateau of around .6
- The best TREC-8 ad-hoc systems not stat. significantly different → plateau reached? Ad hoc track discontinued after TREC-8.
- New tasks: filtering, web, QA, genomics, interactive, novelty, robust, video, cross-lingual,...
- 2006 is TREC-15. Latest tasks: spam, terabyte, blog, web, legal

## TREC tracks since 1992

TRACK	TREC													
	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
Ad Hoc	18	24	26	23	28	31	42	41						
Routing	16	25	25	15	16	21								
Interactive			3	11	2	9	8	7	6	6	6			
Spanish			4	10	7									
Confusion				4	5									
Database Merging				3	3									
Filtering				4	7	10	12	14	15	19	21			
Chinese					9	12								
NLP					4	2								
Speech						13	10	10	3					
Cross-Language						13	9	13	16	10	9			
High Precision						5	4							
Very Large Corpus							7	6						
Query							2	5	6					
Question Answering								20	28	36	34	33	28	33
Web								17	23	30	23	27	28	
Video										12	19			
Novelty Detection											13	14	14	
Genomic												29	33	41
HARD												14	16	16
Robust												16	14	17
Terabyte													17	23
Enterprise														19
Spam														13
	22	31	33	36	38	51	56	66	68	87	93	93	103	117

- IR evaluation as currently performed (TREC) only covers one small part of the spectrum:
  - System performance in batch mode
  - Laboratory conditions; not directly involving real users
  - Precision and recall measured from large, fixed test collections
- However, this evaluation methodology is very stable and mature
  - Host of elaborate performance metrics available, e.g. MAP
  - Relevance problem solvable (in principle) by query sampling, guidelines, relative system comparisons
  - Recall problem solvable (in practice) by pooling methods
  - Provable that these methods produce stable evaluation results

- Teufel (2007): Chapter *An Overview of evaluation methods in TREC Ad-hoc Information Retrieval and TREC Question Answering*. In: L. Dybkjaer, H. Hemsén, W. Minker (Eds.) *Evaluation of Text and Speech Systems*. Springer, Dordrecht, The Netherlands.

# Information Retrieval

## Lecture 4: Search engines and linkage algorithms

Computer Science Tripos Part II



UNIVERSITY OF  
CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

Today

104

- 
- Fixed document collections → World Wide Web:  
What are the differences?
  - Linkage-based algorithms
    - PageRank (Brin and Page, 1998)
    - HITS (Kleinberg, 1998)

- 
- Large-volume
    - Estimates of 80 billion pages for 2006 (1600 TB)  
(1TB = 1024 GB =  $2^{40}$ B)
    - Google indexed 8 billion pages in 2004; coverage 15-20% of web
    - Size of the web is doubling every half a year (Lawrence and Giles, "Searching the world wide web", Science, 1998)
  - Redundant (copied or dynamically generated)
  - Unstructured/differently structured documents
  - Heterogenous (length, quality, language, contents)
  - Volatile/dynamic
    - 1 M new pages per day; average page changes every 2-3 weeks
    - 2-9% of indexed pages are invalid
  - Hyperlinked

- 
- Different syntactic features in query languages
    - Ranked with proximity, phrase units, order relevant, with or without stemming
  - Different indexing ("web-crawling")
    - Heuristic enterprise; not all pages are indexed (est. 15-20% (2005); 28-55% (1999) of web covered)
  - Different heuristics used (in addition to standard IR measures)
    - Proximity and location of search terms (Google)
    - Length of URL (AltaVista)
    - Anchor text pointing to a page (Google)
    - Quality estimates based on link structure

- At search time, browsers do not access full text
- Index is built off-line; crawlers/spiders find web pages
  - Start with popular URLs and recursively follow links
  - Send new/updated pages to server for indexing
  - Search strategy: breadth-first, depth-first, backlink count, estimated popularity
- Parallel crawling
  - Avoid visiting the same page more than once
  - Partition the web and explore each partition exhaustively
- Agreement `robots.txt`: directories off-limits for crawlers
- In 1998, Google processed 4 M pages/day (50 pages, 500 links per second); fastest crawlers today: 10 M pages/day
- In 1998, AltaVista used 20 processors with 130G RAM and 500 GB disk each for indexing.

- Links contain valuable information: latent human judgement
- Idea: derive quality measure by counting links
- Cf. citation index in science: papers which are cited more are considered to be of higher quality
- Similarity to scientific citation network
  - Receiving a “backlink” is like being cited (practical caveat: on the web, there is no certainty about the number of backlinks)

---

**Suggestion:** of all pages containing the search string, return the pages with the most backlinks

- Generalisation problem
  - Many pages are not sufficiently self-descriptive
  - Example: the term “car manufacturer” does not occur anywhere on Honda homepage
  - No endogenous information (ie. information found in the page itself, rather than elsewhere) will help
- Page quality not considered at all, only raw backlink number
  - Overall popular page (Yahoo, Amazon) would be wrongly considered an expert on every string it contains
  - A page pointed to by an important page is also important (even if it has only that one single backlink)
  - Possible to manipulate this measure

---

## Additional problem: manipulatability

- Web links are **not** quite like scientific citations
  - Large variation in web pages: quality, purpose, number of links, length (scientific articles are more homogeneous)
    - \* No publishing/production costs associated with web sites
    - \* No quality check (cf. peer review in scientific articles)
    - \* No cost associated with links (cf. length restrictions in scientific articles)
  - Therefore, linking is gratuitous (replicable), whereas citing is not
  - Any quality evaluation strategy which counts replicable features of web pages is prone to manipulation
- Therefore, raw counting will work less well than it does in scientific area
- Must be more clever when using link structure: PageRank, HITS

- L. Page et al: “The PageRank Citation Ranking: Bringing order to the web”, Tech Report, Stanford Univ., 1998
- S. Brin, L. Page: “The anatomy of a large-scale Hypertextual Web Search Engine”, WWW7/Computer Networks 30(1-7):107-117, 1998
- Goal: estimate overall relative importance of web pages
- Simulation of a random surfer
  - Given a random page, follows links for a while (randomly), with probability  $q$  — assumption: never go back on already traversed links
  - Gets bored after a while and jumps to the next random page, with probability  $1 - q$
  - Surfs infinitely long
- PageRank is the number of visits to each page

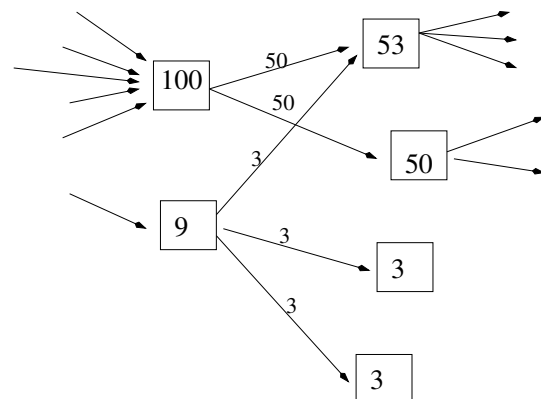
## PageRank formula (simple case)

$$R(u) = (1 - q) + q \sum_{v \in B_u} \frac{R(v)}{N_v}$$

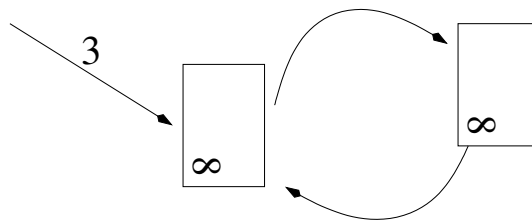
$u$  a web page  
 $F_u$  set of pages  $u$  points to (“Forward” set)  
 $B_u$  set of pages that point to  $u$  (“Back” set)  
 $N_u = |F_u|$  number of pages  $u$  points to  
 $q$  probability of following local link

This formula assumes that no PageRank gets lost in any iteration. In order for this to be the case, each page must have at least one outgoing link.

Simplified PageRank ( $q=1.0$ ):

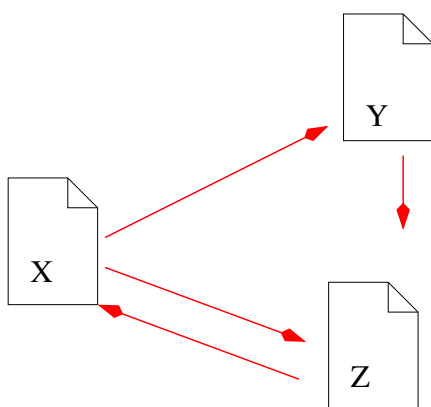






- The amount of pagerank in the web should be equal to  $N$  (so that the average page rank on the web is 1)
- Rank must stay constant in each step, but rank sinks lose infinitely much rank
- Rank also gets lost in each step for pages without onward links
- Solution: rank source  $\vec{e}$  counteracts rank sinks
- $\vec{e}$  is the vector of the probability of random jumps of random surfer to a random page

## An example: PageRank computation



$$R_{n+1}(u) = (1 - q) + q \sum_{v \in B_u} \frac{R_n(v)}{N_v}$$

$$\vec{r} = c(qA\vec{r} + (1 - q)m\vec{1})$$

$c$  is chosen such that  $\|\vec{r}\|_1 = N$ . ( $\|\vec{r}\|_1$  is the  $L_1$  norm of  $\vec{r}$ ).

$$\vec{r} = c(qA + \frac{1-q}{N}\mathbf{1})\vec{r} \text{ (with } \vec{e} = \frac{1-q}{N}\mathbf{1})$$

$A$  normalised link matrix of the web:

$$A_{uv} = \begin{cases} \frac{1}{N_v} & \text{if } \exists v \rightarrow u \\ 0 & \text{otherwise} \end{cases}$$

$\vec{r}$  PageRank vector (over all web pages), the desired result.

$\vec{1}$  a column vector consisting only of ones

$\mathbf{1}$  a matrix filled with all ones

$m$  average pagerank per page (e.g., 1).

We know from linear algebra that  $\vec{r} := A\vec{r}$ ; normalise ( $\vec{r}$ );  $\vec{r} := A\vec{r} \dots$  will make  $\vec{r}$  converge to the dominant eigenvector of  $A$  (independently of  $\vec{r}$ 's initial value), with eigenvalue  $c$ .

## Pagerank, matrix algorithm

1. Initialise  $\vec{r}$ ,  $A$

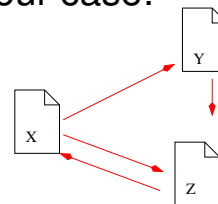
2. Loop:

- $r_{i+1}^{\vec{r}} = c_{i+1}(qA + \frac{1-q}{N}\mathbf{1})r_i^{\vec{r}}$
- $B = qA + \frac{1-q}{N}\mathbf{1}$
- iterate  $\{ r_n^{\vec{r}} = B r_{n-1}^{\vec{r}}; \text{ normalise } r_n^{\vec{r}} \}$
- Stop criterion:  $\|r_{i+1}^{\vec{r}} - r_i^{\vec{r}}\| < N\epsilon$  ( $\|r_{i+1}^{\vec{r}} - r_i^{\vec{r}}\|$  is page-wise “movement” in PageRank between two iterations)

3. This will result in a Page rank vector  $\vec{r}$  whose average PageRank per page is 1:

$$\|\vec{r}\|_1 = N$$

In our case:



$$A = \begin{bmatrix} 0 & 0 & 1 \\ .5 & 0 & 0 \\ .5 & 1 & 0 \end{bmatrix}$$

$$\vec{r}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}; q = .85$$

$$B = \begin{bmatrix} .050 & .050 & .900 \\ .475 & .050 & .050 \\ .475 & .900 & .050 \end{bmatrix}$$

$$B = \begin{bmatrix} .050 & .050 & .900 \\ .475 & .050 & .050 \\ .475 & .900 & .050 \end{bmatrix}$$

Iterate  $r_n^{\vec{}} = Br_{n-1}^{\vec{}}$ :

$$r_0^{\vec{}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}; r_1^{\vec{}} = \begin{bmatrix} 1.0000 \\ 0.5750 \\ 1.4250 \end{bmatrix}; r_2^{\vec{}} = \begin{bmatrix} 1.3613 \\ 0.5750 \\ 1.0637 \end{bmatrix}; r_3^{\vec{}} = \begin{bmatrix} 1.0542 \\ 0.7285 \\ 1.2173 \end{bmatrix}; r_4^{\vec{}} = \begin{bmatrix} 1.1847 \\ 0.5980 \\ 1.2173 \end{bmatrix}; r_5^{\vec{}} = \begin{bmatrix} 1.1847 \\ 0.6535 \\ 1.1618 \end{bmatrix};$$

$$r_6^{\vec{}} = \begin{bmatrix} 1.1375 \\ 0.6535 \\ 1.2090 \end{bmatrix}; r_7^{\vec{}} = \begin{bmatrix} 1.1776 \\ 0.6335 \\ 1.1889 \end{bmatrix}; r_8^{\vec{}} = \begin{bmatrix} 1.1606 \\ 0.6505 \\ 1.1889 \end{bmatrix}; r_9^{\vec{}} = \begin{bmatrix} 1.1606 \\ 0.6432 \\ 1.1962 \end{bmatrix}; r_{10}^{\vec{}} = \begin{bmatrix} 1.1667 \\ 0.6432 \\ 1.1900 \end{bmatrix} \dots$$

## PageRank computation (practicalities)

- Space
  - Example: 75 M unique links on 25 M pages
  - Then: memory for PageRank 300MB
  - Link structures is compact (8B/link compressed)
- Time
  - Each iteration takes 6 minutes (for the 75 M links)
  - Whole process: 5 hours
  - Convergence after 52 iter. (322M links), 48 iter. (161M links)
  - Scaling factor linear in  $\log n$
- Pages without children removed during iteration
- Raw data can be obtained during web crawl; cost of computing PageRank is insignificant compared to the cost of building a full index

- Difference between linking behaviour (public) and actual usage data (web page access numbers from NLNR)
  - PageRank uses only public information; thus fewer privacy implications than usage data (pages that are accessed but not linked to)
  - PageRank produces a finer resolution compared to small usage sample
  - But: not all web users create links
- Propagation simulates word-of-mouth effects in complex network (ahead of time):
  - PageRank can change fast (one link on Yahoo)
    - \* Good pages often have only a few important backlinks (at first)
    - \* Those pages would not be found by simply back-link counting
  - Net traffic can change fast (one mention on the radio)

## Summary PageRank

- Model of collaborative trust; users want information from “trusted” sources
- PageRank is immune to manipulation: it must convince an important site, or many unimportant ones, to point to it
  - Spamming PageRank costs real money – a good property for a search algorithm
  - Google’s business model: never sell PageRank (only advertising space)
- PageRank is a good predictor of optimal crawling order

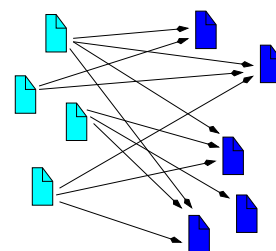
Download Netscape Software	11589.00
http://www.w3.org	10717.70
Welcome to Netscape	8673.51
Point: It's what you're searching for	7930.92
Web-Counter home page	7254.97
THE Blue Ribbon Campaign for Online Free Speech	7010.39
CERN Welcome	6562.49
Yahoo!	6561.80
Welcome to Netscape	6203.47
Wusage 4.1: A Usage Statistics System for Web Servers	5963.27
The World Wide Web consortium (W3C)	5672.21
Lycos, Inc. Home Page	4683.31
Starting Point	4501.98
Welcome to Magellan!	3866.62
Oracle Corporation	3587.63

Benefits for search with PageRank are greatest for underspecified queries

## Hypertext Induced Topic Search (HITS)

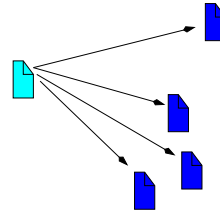
122

- J. Kleinberg, "Authoritative sources in a hyperlinked environment", ACM-SIAM 1998
- Goal: find authorities on a certain topic (relevance, popularity)
- Idea: There are **hubs** and **authorities** on the web, which exhibit a mutually reinforcing relationship
- **Hubs**: Recommendation pages with links to high-quality pages (authorities), e.g. compilations of favourite bookmarks, "useful links"
- **Authorities**: Pages that are recognised by others (particularly by hubs!) as experts on a certain topic
- Authorities are different from universally popular pages (high backlink count), which are not particular experts on that topic



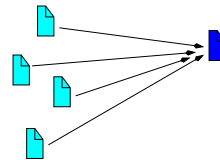
- Each page has two non-negative weights: an authority weight  $a$  and a hub weight  $h$
- At each iteration, update the weights:
  - If a page points at many good authorities, it is probably a good hub:

$$h_p = \sum_{q: \langle p, q \rangle \in A} a_q$$



- If a page is pointed to by many good hubs, it is probably a good authority:

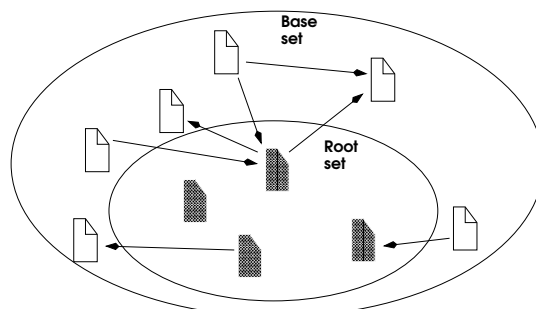
$$a_p = \sum_{q: \langle q, p \rangle \in A} h_q$$



- Normalise weights after each iteration

## HITS: Initialisation

- Start with the **root set**: set of web pages containing the query terms
- Create the **base set**: root set plus all pages pointing to the root set (cut-off if too many), and being pointed to by the root set
- The base set typically contains 1000-5000 documents



Given:

- a set  $D = \{D_1 \dots D_n\}$  of documents (base set)
- $A$ , the linking matrix: edge  $\langle i, j \rangle \in A$  iff  $D_i$  points to  $D_j$
- $k$ , the number of desired iterations

Initialise:  $\vec{a} = \{1, 1, \dots, 1\}$ ;  $\vec{h} = \{1, 1, \dots, 1\}$

Iterate: for  $c = 1 \dots k$

- for  $i = 1 \dots n$  :  $a_p = \sum_{q: \langle q, p \rangle \in A} h_q$
- for  $i = 1 \dots n$  :  $h_p = \sum_{q: \langle p, q \rangle \in A} a_q$

Normalise  $\vec{a}$  and  $\vec{h}$ :  $\sum_{i \in D_i} a_i = \sum_{i \in D_i} h_i = 1$

## HITS: Convergence

- Updates:

$$\vec{a} = A^T \vec{h} \qquad \vec{h} = A \vec{a}$$

- After the first iteration:

$$\vec{a}_1 = A^T A \vec{a}_0 = (A^T A) \vec{a}_0 \qquad \vec{h}_1 = A A^T \vec{h}_0 = (A A^T) \vec{h}_0$$

- After the second iteration:

$$\vec{a}_2 = (A^T A)^2 \vec{a}_0 \qquad \vec{h}_2 = (A A^T)^2 \vec{h}_0$$

- Convergence to

- $\vec{a} \leftarrow$  dominant eigenvector( $A^T A$ )
- $\vec{h} \leftarrow$  dominant eigenvector( $A A^T$ )

### Authorities on “java”

0.328	<a href="http://www.gamelan.com">http://www.gamelan.com</a>	Gamelan
0.251	<a href="http://java.sun.com">http://java.sun.com</a>	JavaSoft home page
0.190	<a href="http://www.digitalfocus.com/digital">http://www.digitalfocus.com/digital</a>	The Java Developer: How do I

### Authorities on “censorship”

0.376	<a href="http://www.eff.org">http://www.eff.org</a>	EFF – The Electronic Frontier Foundation
0.344	<a href="http://www.eff.org/blueribbon.html">http://www.eff.org/blueribbon.html</a>	The Blue Ribbon Campaign for Online Free Speech
0.238	<a href="http://www.cdt.org">http://www.cdt.org</a>	The Center for Democracy and Technology
0.235	<a href="http://www.vtw.org">http://www.vtw.org</a>	Voters Telecommunication Watch
0.218	<a href="http://www.aclu.org">http://www.aclu.org</a>	ACLU: American Civil Liberties Union

### Authorities on “search engine”

0.346	<a href="http://www.yahoo.com">http://www.yahoo.com</a>	Yahoo
0.291	<a href="http://www.excite.com">http://www.excite.com</a>	Excite
0.239	<a href="http://www.mckinley.com">http://www.mckinley.com</a>	Welcome to Magellan
0.231	<a href="http://www.lycos.com">http://www.lycos.com</a>	Lycos Home Page
0.231	<a href="http://www.altavista.digital.com">http://www.altavista.digital.com</a>	AltaVista: Main Page

## Summary

- Both HITS and PageRank infer quality/“expert-ness” from link structure of the web
- Link structure contains latent human judgement
- Use different models of type of web pages
- Iterative algorithms
- Use of these weights for search (in different ways)
- Other differences between closed-world assumption (IR) and world wide web: data, indexing, query constructs, search heuristics



# Information Retrieval

## Lecture 5: Information Extraction

Computer Science Tripos Part II



UNIVERSITY OF  
CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

### Information Extraction: the task

130

- 
- Identify instances of a particular class of events or relationships in a natural language text
  - Limited semantic range of events/relationships (domain-dependence)
  - Extract the relevant arguments of the event or relationship into pre-existing “templates” (tabular data structures)
  - MUC (Message Understanding Conference; NIST) 1986-97 competitive evaluation

- 
- 1980s and before: lexico-semantic patterns written by hand (FRUMP, satellite reports, patient discharge summaries...)
  - 1987 **First MUC** (Message Understanding Conference); domain: naval sightings
  - 1989 **Second MUC**; domain: naval sightings
  - 1991 **Third MUC**; domain: terrorist acts
    - Winner (SRI) used partial parsing
  - 1992 **Fourth MUC**; domain: terrorist acts
  - 1993 **Fifth MUC**; domain: joint ventures/electronic circuit fabrication
    - Performance of best systems ~ 40% R, 50% P (Humans in 60-80% range)
    - Lehnert et al.: first bootstrapping method

- 
- 1995 **Sixth MUC**; domain: labour unit contract negotiations/changes in corporate executive management personnel
    - Encourage more portability and deeper understanding
    - Separate tasks into
      - \* **NE**: Named Entity
      - \* **CO**: Coreference
      - \* **TE**: Template Element
      - \* **ST**: Scenario Templates
  - 1995: IE for summarisation (Radev and McKeown)
  - 1998: **Seventh MUC**; domain: satellite rocket launch events
    - Mikheev et al., hybrid methods for NE
  - 2003: **CoNLL** NE recognition task; similar training data to MUC

- Participants get a description of the scenario and a training corpus (a set of documents and the templates to be extracted from these)
- 1-6 months time to adapt systems to the new scenario
- NIST analysts manually fill templates of test corpus (“answer key”)
- Test corpus delivered; systems run at home
- Automatic comparison of system response with answer key
- Primary scores: precision and recall
- Participants present paper at conference in spring after competition
- Show system’s workings on predefined “walk through” example

## Template example (MUC-3)

0	MESSAGE ID	TST1-MUC3-0080
1	TEMPLATE ID	1
2	DATE OF INCIDENT	03 APR 90
3	TYPE OF INCIDENT	KIDNAPPING
4	CATEGORY OF INCIDENT	TERRORIST ACT
5	PERPETRATOR: ID OF INDIV(S)	“THREE HEAVILY ARMED MEN”
6	PERPETRATOR: ID OF ORG(S)	“THE EXTRADITABLES”
7	PERPETRATOR: CONFIDENCE	CLAIMED OR ADMITTED: “THE EXTRADITABLES”
8	PHYSICAL TARGET: ID(S)	*
9	PHYSICAL TARGET: TOTAL NUM	*
10	PHYSICAL TARGET: TYPE(S)	*
11	HUMAN TARGET: ID(S)	“FEDERICO ESTRADA VELEZ” (“LIBERAL SENATOR”)
12	HUMAN TARGET: TOTAL NUM	1
13	HUMAN TARGET: TYPE(S)	GOVERNMENT OFFICIAL: “FEDERICO ESTRADA VELEZ”
14	TARGET: FOREIGN NATION(S)	–
15	INSTRUMENT: TYPE(S)	*
16	LOCATION OF INCIDENT	COLOMBIA: MEDELLIN (CITY)
17	EFFECT ON PHYSICAL TARGETS	*
18	EFFECT ON HUMAN TARGETS	*

TST-1-MUC3-0080

BOGOTA, 3 APR 90 (INRAVISION TELEVISION CADENA 1) - [REPORT] [JORGE ALONSO SIERRA VALENCIA] [TEXT] LIBERAL SENATOR FEDERICO ESTRADA VELEZ WAS KIDNAPPED ON 3 APRIL AT THE CORNER OF 60TH AND 48TH STREETS IN WESTERN MEDELLIN, ONLY 100 METERS FROM A METROPOLITAN POLICE CAI [IMMEDIATE ATTENTION CENTER]. THE ANTIOQUIA DEPARTMENT LIBERAL PARTY LEADER HAD LEFT HIS HOUSE WITHOUT ANY BODYGUARDS ONLY MINUTES EARLIER. AS WE WAITED FOR THE TRAFFIC LIGHT TO CHANGE, THREE HEAVILY ARMED MEN FORCED HIM TO GET OUT OF HIS CAR AND INTO A BLUE RENAULT.

HOURS LATER, THROUGH ANONYMOUS TELEPHONE CALLS TO THE METROPOLITAN POLICE AND TO THE MEDIA, THE EXTRADITABLES CLAIMED RESPONSIBILITY FOR THE KIDNAPPING. IN THE CALLS, THEY ANNOUNCED THAT THEY WILL RELEASE THE SENATOR WITH A NEW MESSAGE FOR THE NATIONAL GOVERNMENT. LAST WEEK, FEDERICO ESTRADA HAD REJECTED TALKS BETWEEN THE GOVERNMENT AND THE DRUG TRAFFICKERS.

## Text Example (MUC-5)

<DOC>

<DOCNO> 0592 </DOCNO>

<DD> NOVEMBER 24, 1989, FRIDAY </DD>

<SO>Copyright (c) 1989 Jiji Press Ltd.;</SO>

<TXT>

BRIDGESTONE SPORTS CO. SAID FRIDAY IT HAS SET UP A JOINT VENTURE IN TAIWAN WITH A LOCAL CONCERN AND A JAPANESE TRADING HOUSE TO PRODUCE GOLF CLUBS TO BE SHIPPED TO JAPAN.

THE JOINT VENTURE, BRIDGESTONE SPORTS TAIWAN CO., CAPITALIZED AT 20 MILLION NEW TAIWAN DOLLARS, WILL START PRODUCTION IN JANUARY 1990 WITH PRODUCTION OF 20,000 IRON AND "METAL WOOD" CLUBS A MONTH. THE MONTHLY OUTPUT WILL BE LATER RAISED TO 55,000 UNITS, BRIDGESTON SPORTS OFFICIALS SAID.

THE NEW COMPANY, BASED IN KAOHSIUNG, SOUTHERN TAIWAN, IS OWNED 75 PCT BY BRIDGESTONE SPORTS, 15 PCT BY UNION PRECISION CASTING CO. OF TAIWAN AND THE REMAINDER BY TAGA CO., A COMPANY ACTIVE IN TRADING WITH TAIWAN, THE OFFICIALS SAID.

BRIDGESTONE SPORTS HAS SO FAR BEEN ENTRUSTING PRODUCTION OF GOLF CLUBS PARTS WITH UNION PRECISION CASTING AND OTHER TAIWAN COMPANIES.

WITH THE ESTABLISHMENT OF THE TAIWAN UNIT, THE JAPANESE SPORTS GOODS MAKER PLANS TO INCREASE PRODUCTION OF LUXURY CLUBS IN JAPAN.

</TXT>

</DOC>

```

<TEMPLATE-0592-1> :=
  DOC NR: 0592
  DOC DATE: 241189
  DOCUMENT SOURCE: "Jiji Press Ltd."
  CONTENT: <TIE_UP_RELATIONSHIP-0592-1>
<TIE_UP_RELATIONSHIP-0592-1> :=
  TIE-UP STATUS: EXISTING
  ENTITY:<ENTITY-0592-1>
    <ENTITY-0592-2>
    <ENTITY-0592-3>
  JOINT VENTURE CO:<ENTITY-0592-4>
  OWNERSHIP: <OWNERSHIP-0592-1>
  ACTIVITY:<ACTIVITY-0592-1>
<ENTITY-0592-1>:=
  NAME: BRIDGESTONE SPORTS CO
  ALIASES: "BRIDGESTONE SPORTS"
    "BRIDGESTON SPORTS"
  NATIONALITY: Japan (COUNTRY)
  TYPE: COMPANY
  ENTITY RELATIONSHIP:<ENTITY_RELATIONSHIP-0592-1>
<ENTITY-0592-2>:=
  NAME: UNION PRECISION CASTING CO
  ALIASES: "UNION PRECISION CASTING"
    "BRIDGESTON SPORTS"
  LOCATION: Taiwan (COUNTRY)
  NATIONALITY: Taiwan (COUNTRY)
  TYPE: COMPANY
  ENTITY RELATIONSHIP:<ENTITY_RELATIONSHIP-0592-1>
<ENTITY-0592-3>:=
  NAME: TAGA CO
  NATIONALITY: Japan (COUNTRY)
  TYPE: COMPANY
  ENTITY RELATIONSHIP:<ENTITY_RELATIONSHIP-0592-1>

<ENTITY-0592-4>:=
  NAME: BRIDGESTONE SPORTS TAIWAN CO
  ALIASES: "UNION PRECISION CASTING"
    "BRIDGESTON SPORTS"
  LOCATION: "KAOHSIUNG" (UNKNOWN) Taiwan (COUNTRY)
  TYPE: COMPANY
  ENTITY RELATIONSHIP:<ENTITY_RELATIONSHIP-0592-1>
<INDUSTRY-0592-1>:=
  INDUSTRY-TYPE: PRODUCTION
  PRODUCT/SERVICE: (CODE 39 "20,000 IRON AND 'METAL WOOD")
[CLUBS])
<ENTITY_RELATIONSHIP-0592-1>:=
  ENTITY1: <ENTITY-0592-1>
    <ENTITY-0592-2>
    <ENTITY-0592-3>
  ENTITY2: <ENTITY-0592-4>
  REL OF ENTITY2 To ENTITY1: CHILD
  STATUS: CURRENT
<ACTIVITY-0592-1>:=
  INDUSTRY: <INDUSTRY-0592-1>
  ACTIVITY-SITE: (Taiwan (COUNTRY) <ENTITY-0592-4>)
  START TIME: <TIME-0592-1>
<TIME-0592-1>:=
  DURING: 0190
<OWNERSHIP-0592-1>:=
  OWNED: <ENTITY-0592-4>
  TOTAL-CAPITALIZATION: 20000000 TWD
  OWNERSHIP-%: (<ENTITY-0592-3> 10)
    (<ENTITY-0592-2> 15)
    (<ENTITY-0592-1> 75)
    
```

## Example Task

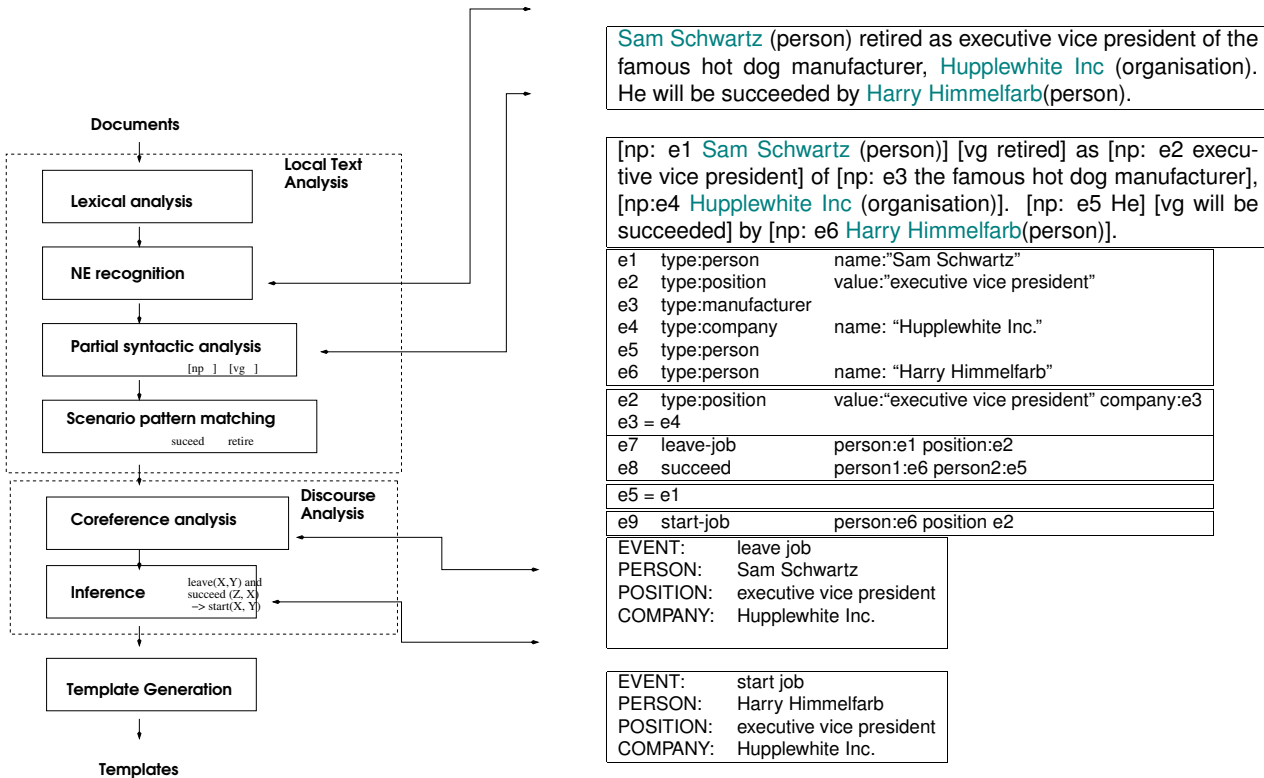
### Input Text:

Sam Schwartz retired as executive vice president of the famous hot dog manufacturer, Hupplewhite Inc. He will be succeeded by Harry Himmelfarb.

### Desired Output:

EVENT: leave job  
 PERSON: Sam Schwartz  
 POSITION: executive vice president  
 COMPANY: Hupplewhite Inc.

EVENT: start job  
 PERSON: Harry Himmelfarb  
 POSITION: executive vice president  
 COMPANY: Hupplewhite Inc.



[np: e1 Sam Schwartz (person)] [vg retired] as [np: e2 executive vice president] of [np: e3 the famous hot dog manufacturer], [np:e4 Hupplewhite Inc (organisation)]. [np: e5 He] [vg will be succeeded] by [np: e6 Harry Himmelfarb(person)].

System knows:

- Appositions (X, Y) → X = Y
- X of COMPANY → X is a position

Intermediate Knowledge:

e1	type:person	name:"Sam Schwartz"
e2	type:position	value:"executive vice president"
e3	type:manufacturer	
e2	type:position	value:"executive vice president" company:e3
e4	type:company	name: "Hupplewhite Inc."
e5	type:person	
e6	type:person	name: "Harry Himmelfarb"
e3 = e4		

Direct matches on lexico-semantic patterns.

- “X retires from Y as Z” →
- X is PERSON in LEAVE-JOB template; Y is COMPANY; Z is POSITION.

EVENT:	leave job
PERSON:	Sam Schwartz
POSITION:	executive vice president
COMPANY:	Hupplewhite Inc.

- e5=e1 from coreference module
- INFERENCE RULE: LEAVE-JOB(X,Y,P) & succeed(Y,Z) → START-JOB(Z,Y,P) with X,Z persons, Y company, P position.

EVENT:	start job
PERSON:	Harry Himmelfarb
POSITION:	executive vice president
COMPANY:	Hupplewhite Inc.

- NE types:
  - ENAMEX (type= person, organisation, location)
  - TIMEX (type= time, date)
  - NUMEX (type= money, percent)
- Allowed to use **gazetteers** (fixed list containing names of a certain type, e.g. countries, last names, titles, state names, rivers...)
- ENAMEX is harder, more context dependent than TIMEX and NUMEX:
  - Is **Granada** a COMPANY or a LOCATION?
  - Is **Washington** a PERSON or a LOCATION?
  - Is **Arthur Anderson** a PERSON or an ORGANISATION?

- NE markup with subtypes:  
<ENAMEX TYPE='PERSON'>Flavel Donne</ENAMEX> is an analyst with <ENAMEX TYPE='ORGANIZATION'>General Trends</ENAMEX>, which has been based in <ENAMEX TYPE='LOCATION'>Little Spring</ENAMEX> since <TIMEX>July 1998</TIMEX>.
- Most systems use manually written regular expressions
  - Rules about mid initials, postfixes, titles
  - Gazetteers of common first names
  - Acronyms: Hewlett Packard Inc. → HP

PATTERN: “president of <company>” matches

*executive vice president of Hupplewhite*



- Gazetteer of full names impossible and not useful, as both first and last names can occur on their own
- **Last name** gazetteer impractical
  - Almost infinite set of name patterns possible: last names are productive (1.5M surnames in US alone)
  - Overlap with common nouns/verbs/adjectives
    - \* First 2 pages of Cambridge phone book include 237 names
    - \* Of those, 6 (2.5%) are common nouns: Abbey, Abbot, Acres, Afford, Airs, Alabaster
- **First name** gazetteer less impractical, but still not foolproof
  - First names can be surprising, eg. MUC-7 walk-through example: “Llennel Evangelista”
  - First names are productive, eg. Moonunit Zappa, Apple Paltrow . . .

- – Overlap with common nouns:
  - \* River and Rain Phoenix, Moon Unit Zappa, Apple Paltrow
  - \* “Virtue names”: Grace (134), Joy (390), Charity (480), Chastity (983), Constance, Destiny
  - \* “Month names”: June, April, May
  - \* “Flower names”: Rose, Daisy, Lily, Erica, Iris . . .
  - \* From US Social Security Administration’s list of most popular girls’ names in 1990, with rank:  
Amber (16), Crystal (41), Jordan (59), Jade (224), Summer (291), Ruby (300), Diamond (450), Infant (455), Precious (472), Genesis (528), Paris (573), Princess (771), Heaven (902), Baby (924) . . .
- Additional problem: non-English names alliterated into English; variant spellings
- Complicated name patterns with titles: Sammy Davis Jr, HRH The Prince of Wales, Dr. John T. Maxwell III

- Ambiguity of name types: *Columbia* (Org.) vs. (British) *Columbia* (Location) vs. *Columbia* (Space shuttle)
- Company names often use common nouns (“Next”, “Boots”, “Thinking Machines”...) and can occur in variations (“Peter Stuyvesant”, “Stuyvesant”)
- Coordination problems/ left boundary problems:
  - One or two entities in *China International Trust and Investment Corp invests \$2m in... ?*
  - Unknown word at beginning of potential name: in or out?  
*Suspended Ceilings Inc* vs *Yesterday Ceilings Inc*  
*Mason, Daily and Partners* vs. *Unfortunately, Daily and Partners*
- Experiments show: simple gazetteers fine for locations (90%P/80%R) but not for person and organisations (80%P/50%R)

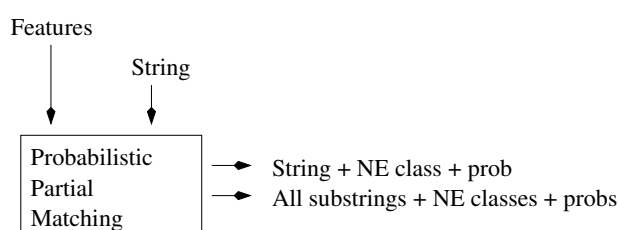
- Staged combination of rule-based system with probabilistic partial matching
- Use machine learning to decide type of NE
- Use internal phrase structure of name
- Make high-precision decisions first
- Keep off decision about unsure items until all evidence has been seen
- Assume: one name type per discourse (article)
  - unless signalled by writer with additional context information

Rule	Assign	Example
(Xx+)+ (is ,) a? JJ* PROF	PERS	Yuri Gromov, a former director
(Xx+)+ is? a? JJ* REL	PERS	John White is beloved brother
(Xx+)+ himself	PERS	White himself
(Xx+)+, DD+ ,	PERS	White, 33,
share in (Xx+)+	ORG	shares in Trinity Motors
(Xx+)+ Inc.	ORG	Hummingbird Inc.
PROF (of at with) (Xx+)+	ORG	director of Trinity Motors
(Xx+)+ (region area)	LOC	Lower Beribidjan area

## Mikheev et al. (1998): ML; External and internal evidence 150

### External:

- Position in sentence (sentence initial)
- Word exists in lexicon in lowercase
- Word seen in lowercase in document



### Internal:

- Contains any non-alpha characters
- Number of words it consists of
- Suffix, Prefix
- Adjectives ending in “an” or “ese” + whose root is in Gazetteer

1. **Apply Grammar Rule Set 1** (“Sure fire” rules)  
→ tag as definite NEs of given type
2. **Use ML for variants** (probabilistic partial match)
  - Generate all possible substrings of sure-fire tagged NEs:
    - *Adam Kluver Ltd* → *Adam Kluver, Adam Ltd, Kluver Ltd*
  - ME model gives probability for possible string and NE type
  - Tag all occurrences of NE in text (over prob. threshold) with type
3. **Apply Grammar Rule Set 2** (Relaxed rules)
  - Mark anything that looks like a PERSON (using name grammar)
  - Resolve coordination, genitives, sentence initial capitalized modifiers
    - Coordinated or possessive name parts, or rest of sentence initial coordinated name seen on their own? If not, assume one name (*Murdoch’s News Corp, Daily, Bridge and Mason*)
4. **Apply ML again** (for new variants)
  - X and Y are of same type → resolved typo ‘ ‘Un7ited States and Russia’ ’
5. **Apply specialised ME model to title** (capitalisation, different syntax).

### MURDOCH SATELLITE CRASH UNDER FBI INVESTIGATION

London and Tomsk. The crash of Rupert Murdoch Inc’s news satellite yesterday is now under investigation by Murdoch and by the Siberian state police. Clarity J. White, vice president of Hot Start, the company which produced the satellite’s ignition system, yesterday stated that her company considered human failure the most likely cause of the crash. Investigator Robin Black, 33, who investigates the crash for the FBI, recently arrived by train at the crash site in the Tomsk region. Neither White nor Black were available for comment today; Murdoch have announced a press conference for tomorrow.

LONDON and TOMSK	Org
Rupert Murdoch	Person
Murdoch	Person
Neither White	Person
Investigator Robin Black	Person

Additional problem: `Black` and `White` have last names which overlap with adjectives and first names which overlap with common nouns (`Robin` and `Clarity`), thus they cannot be in a gazetteer.

### MURDOCH SATELLITE CRASH UNDER FBI INVESTIGATION

London and Tomsk. The crash of [Rupert Murdoch Inc](#)(ORG)'s news satellite yesterday is now under investigation by Murdoch and by the Siberian state police. Clarity J. White, vice president of [Hot Start](#)(ORG), the company which produced the satellite's ignition system, yesterday stated that her company considered human failure the most likely cause of the crash. [Investigator Robin Black](#)(PERSON), 33, who investigates the crash for the FBI, recently arrived by train at the crash site in the [Tomsk](#)(LOC) region. Neither White nor Black were available for comment today; Murdoch have announced a press conference for tomorrow.

Underlined instances: newly suggested in this round

- Sure fire rules applied
- But exact extend of name not known yet: Investigator Robin Black? Black?

**MURDOCH SATELLITE CRASH UNDER INVESTIGATION**

London and [Tomsk](#)(LOC?). The crash of [Rupert Murdoch Inc](#)(ORG?)'s news satellite yesterday is now under investigation by [Murdoch](#)(ORG?) and by the Siberian state police. Clarity J. White, vice president of [Hot Start](#)(ORG?), the company which produced the satellite's ignition system, yesterday stated that her company considered human failure the most likely cause of the crash. [Investigator Robin Black](#)(PERSON?), 33, recently arrived by train at the crash site in the [Tomsk](#)(LOC?) region. Neither White nor [Black](#)(PERSON?) were available for comment today; [Murdoch](#)(ORG?) have announced a press conference for tomorrow.

Green instances: around from last round

- All instances from last round and their substrings are now hypothesized; they and their context are now subjected to ML

**MURDOCH SATELLITE CRASH UNDER INVESTIGATION**

London and [Tomsk](#)(LOC✓). The crash of [Rupert Murdoch Inc](#)(ORG✓)'s news satellite yesterday is now under investigation by [Murdoch](#)(ORG✓) and by the Siberian state police. Clarity J. White, vice president of [Hot Start](#)(ORG✓), the company which produced the satellite's ignition system, yesterday stated that her company considered human failure as the most likely cause of the crash. Investigator [Robin Black](#)(PERS✓), 33, recently arrived by train at the crash site in the [Tomsk](#)(LOC✓) region. Neither White nor [Black](#)(PERS✓) were available for comment today; [Murdoch](#)(ORG✓) have announced a press conference for tomorrow.

- ML has reconfirmed some instances (Robin Black) and discarded others (Investigator Robin Black)

**MURDOCH SATELLITE CRASH UNDER INVESTIGATION**

London and [Tomsk](#)(LOC). The crash of [Rupert Murdoch Inc](#)(ORG)'s news satellite yesterday is now under investigation by [Murdoch](#)(ORG) and by the Siberian state police. [Clarity J. White](#)(PERS?), vice president of [Hot Start](#)(ORG), the company which produced the satellite's ignition system, yesterday stated that her company considered human failure as the most likely cause of the crash. Investigator [Robin Black](#)(PERS), 33, recently arrived by train at the crash site in the [Tomsk](#)(LOC) region. [Neither White](#)(PERS?) nor [Black](#)(PERS) were available for comment today; [Murdoch](#)(ORG) have announced a press conference for tomorrow.

- Relaxed rules: Mark everything as a possibility which roughly follows Name shape (blue, underlined)
- (plus confirmed NEs from last round in green)

**MURDOCH SATELLITE CRASH UNDER INVESTIGATION**

[London](#)(LOC $\checkmark$ ) and [Tomsk](#)(LOC). The crash of [Rupert Murdoch Inc](#)(ORG)'s news satellite yesterday is now under investigation by [Murdoch](#)(ORG) and by the Siberian state police. [Clarity J. White](#)(PERS $\checkmark$ ), vice president of [Hot Start](#)(ORG), the company which produced the satellite's ignition system, yesterday stated that her company considered human failure as the most likely cause of the crash. Investigator [Robin Black](#)(PERS), 33, recently arrived by train at the crash site in the [Tomsk](#)(LOC) region. Neither [White](#)(PERS $\checkmark$ ) nor [Black](#)(PERS) were available for comment today; [Murdoch](#)(ORG) have announced a press conference for tomorrow.

- Some of these possibilities reconfirmed by ML, others discarded
- "London" found by X and Y rule.
- Missing step: different segmentation and ML for title; 'Murdoch' is found there.

93.39% combined P and R – best and statistically different from next contender

		ORG		PERSON		LOC	
		R	P	R	P	R	P
1	Sure fire rules	42	98	40	99	36	96
2	Partial Match 1	75	98	80	99	69	93
3	Relaxed Rules	83	96	90	98	86	93
4	Partial Match 2	85	96	93	97	88	93
5	Title Assignment	91	95	95	97	95	93

- System design: Keep precision high at all stages, raise recall if possible
- Gazetteers improve performance, but system can determine persons and organizations reasonably well even without any gazetteer (ORG: P86/R85; PERSON: P90/R95), but not locations (P46/R59)

## Summary of today

160

- IE consists of different tasks (as defined by MUC): NE, CO, TE, ST
- Today: NE
  - Principal problems with NE
  - NE with manual rules
  - Mikheev et al. (1998)
    - \* Use internal and external evidence
    - \* Cascaded design: commit in order of confidence/supportive evidence from text, not in text order!



- Mikheev, Moens and Grover (1998). Description of the LTG system. MUC-7 Proceedings.
- Mikheev, Moens and Grover (1999). Named Entity Recognition without Gazetteers. EACL'99
- R. Grishman (1997): Information Extraction: Techniques and challenges, in: Information Extraction, Springer Verlag, 1997.

## Information Retrieval

### Lecture 6: Information Extraction and Bootstrapping

Computer Science Tripos Part II



UNIVERSITY OF  
CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

- Range of problems that make named entity recognition (NE) hard
- Mikheev et al's (1998) cascading NE system
- NE is the simplest kind of IE task: no relations between entities must be determined
- NIST MUC conferences pose three kinds of harder IE tasks
- Today: more of the full task (scenario templates), and on learning

- “Flattened-out” semantic representations with lexemes directly hard-wired into them
- String-based matching with type of semantic category to be found directly expressed in lexical pattern
- Problem with all string-based mechanisms: generalisation to other strings with similar semantics, and to only those
- Do generalisation by hand...
  - <Perpetrator> (APPOSITION) {blows/blew/has blown} {himself/herself} up
  - <Perpetrator> detonates
  - {blown up/detonated} by <Perpetrator>
- Manual production of patterns is time-consuming, brittle, and not portable across domains

- UMASS participant system in MUC-4: AutoSlog
- Lexico-semantic patterns for MUC-3 took 1500 person hours to build → knowledge engineering bottleneck
- AutoSlog achieved 98% performance of manual system; AutoSlog dictionary took 5 person hours to build
- “Template mining:”
  - Use MUC training corpus (1500 texts + human answer keys; 50% non-relevant texts) to learn contexts
  - Have human check the resulting templates (30% - 70% retained)

- 389 Patterns (“concept nodes”) with enabling syntactic conditions, e.g. active or passive:
  - kidnap-passive: <VICTIM> expected to be subject
  - kidnap-active: <PERPETRATOR> expected to be subject
- use FIRST occurrence of answer string as hypothesis for template
- Abstract over actual string using a parser
- Find key word there which triggers the pattern: *kidnap*, *shot*,
- Hard and soft constraints for fillers of slots
- Semantic lexicon with 5436 entries (including semantic features) used

ID: DEV-MUC4-0657

Slot Filler: "public buildings"

Sentence: IN LA OROYA, JUNIN DEPARTMENT, IN THE CENTRAL PERUVIAN MOUNTAIN RANGE, PUBLIC BUILDINGS WERE BOMBED AND A CAR-BOMB WAS DETONATED.

Trigger: bomb  
Constraints: subject must be of type PHYS-TARGET  
Enabling Conditions: passive  
THEN  
Variable slots: subject is the TARGET  
Constant slots: type is BOMBING

ID: DEV-MUC4-0071

Slot Filler: "guerrillas"

Sentence: THE SALVADORAN GUERRILLAS ON MAR\_12\_89, TODAY, THREATENED TO MURDER INDIVIDUALS INVOLVED IN THE MAR\_19\_88 PRESIDENTIAL ELECTIONS IF THEY DO NOT RESIGN FROM THEIR POSTS.

Trigger: murder  
Enabling Conditions: active + trigger must be preceded by "threatened to"  
THEN  
Variable slots: subject is the PERPETRATOR

ID: DEV-MUC4-1192

Slot Filler: "gilberto molasco

Sentence: THEY TOOK 2-YEAR-OLD GILBERTO MOLASCO, SON OF PATRICIO RODRIGUEZ, AND 17-OLD ANDRES ALGUETA, SON OF EMIMESTO ARGUETA.

Trigger: take

Enabling Conditions: active

THEN

Variable slots: direct object is the VICTIM

Constant slots: TYPE is a KIDNAPPING

System/Test Set	Recall	Prec	F-measure
MUC-4/TST3	46	56	50.5
AutoSlog/TST3	43	56	48.7
MUC-4/TST4	44	40	41.9
AutoSlog/TST4	39	45	41.8

- 5 hours of sifting through AutoSlog's patterns
- Porting to new domain in less than 10 hours of human interaction
- But: creation of training corpus ignored in this calculation

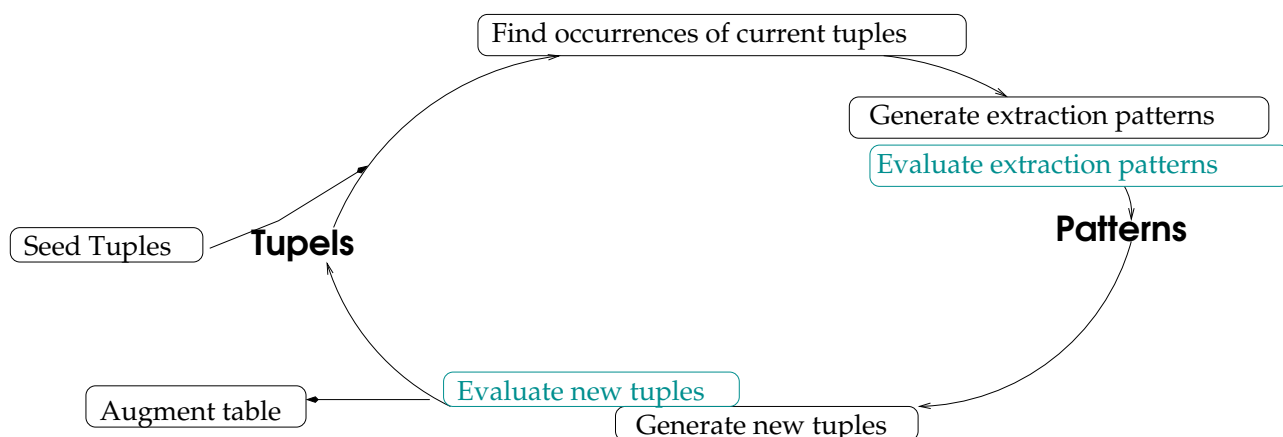
- Find locations of headquarters of a company and the corresponding company name ( $\langle o, l \rangle$  tuples)

Organisation	Location of Headquarters
Microsoft	Redmond
Exxon	Irving
IBM	Armonk
Boeing	Seattle
Intel	Santa Clara

“Computer servers at **Microsoft’s** headquarters in **Redmond**”

- Use minimal human interaction (handful of positive examples)
  - no manually crafted patterns
  - no large annotated corpus (IMass system at MUC-6)
- Automatically learn extraction patterns
- Less important to find **every** occurrence of patterns; only need to fill table with confidence

## Agichtein, Gravano (2000): Bootstrapping



- Start from table containing some  $\langle o, l \rangle$  tuples (which must exist in document collection)
- Perform NE (advantage over prior system DIPRE (Brin 98))
- System searches for occurrences of the example  $\langle o, l \rangle$  tuples in documents
- System learns extraction patterns from these example contexts, e.g.:

$\langle \text{ORGANIZATION} \rangle$  's headquarters in  $\langle \text{LOCATION} \rangle$   
 $\langle \text{LOCATION} \rangle$ -based  $\langle \text{ORGANIZATION} \rangle$

- Evaluate patterns; use best ones to find new  $\langle o, l \rangle$  tuples
- Evaluate new tuples, choose most reliable ones as new seed tuples
- Iteratively repeat the process

## Agichtein, Gravano (2000): Context generalisation and patterns

A SNOWBALL pattern is a 5-tuple  $\langle \text{left}, \text{tag1}, \text{middle}, \text{tag2}, \text{right} \rangle$

left	Tag1	middle	Tag2	right
The	Irving	-based	Exxon Corporation	
$\langle \{ \langle \text{the}, 0.2 \rangle \}$ ,	$\text{LOCATION}$ ,	$\{ \langle -, 0.5 \rangle \}$	$\langle \text{based}, 0.5 \rangle \}$ ,	$\text{ORGANIZATION}$ ,
				$\{ \} \rangle$

- Associate term weights as a function of frequency of term in context
- Normalize each vector so that norm is 1; then multiply with weights  $W_{left}, W_{right}, W_{mid}$ .
- Degree of match between two patterns  $t_p = \langle l_p, t_1, m_p, t_2, r_p \rangle$  and  $t_s = \langle l_s, t'_1, m_s, t'_2, r_s \rangle$ :

$$\text{match}(t_p, t_s) = l_p l_s + m_p m_s + r_p r_s \text{ (if tags match, 0 otherwise)}$$

- Similar contexts form a pattern
  - Cluster vectors using a clustering algorithm (minimum similarity threshold  $\tau_{sim}$ )
  - Vectors represented as cluster centroids  $\bar{l}_s, \bar{m}_s, \bar{r}_s$
- Generalised Snowball pattern defined via centroids:

$$\langle \bar{l}_s, tag_1, \bar{m}_s, tag_2, \bar{r}_s \rangle$$

- Remember for each Generalised Snowball pattern
  - All contexts it came from
  - The distances of contexts from centroid

- We want **productive** and **reliable** patterns
  - productive but not reliable:
    - $\langle \{\}, ORGANIZATION, \{<" , 1 >\}, LOCATION, \{\} \rangle$
    - “Intel, Santa Clara, announced that. . .”
    - “Invest in Microsoft, New York-based analyst Jane Smith said. . .”
  - reliable but not productive:
    - $\langle \{\}, ORGANIZATION, \{< whose, 0.1 >, < headquarter, 0.4 >, < is, 0.1 > < located, 0.3 >, < in, 0.09 >, < nearby, 0.01 >\}, LOCATION, \{\} \rangle$
    - “Exxon, whose headquarter is located in nearby Irving. . .”
- Eliminate patterns supported by less than  $\tau_{sup} \langle o, l \rangle$  tuples



- If  $P$  predicts tuple  $t = \langle o, l \rangle$  and there is already tuple  $t' = \langle o, l' \rangle$  with high confidence, then: if  $l = l' \rightarrow P.positive++$ , otherwise  $P.negative++$  (uniqueness constraints: organization is key).

- Pattern reliability:  $Conf(P) = \frac{P.positive}{P.positive + P.negative}$  (range [0..1])

- Example:

$P_{43} = \langle \{\}, ORGANIZATION, \{<"", 1 >\}, LOCATION, \{\} \rangle$  matches

1. [Exxon, Irving](#), said... (CORRECT: in table)
  2. [Intel, Santa Clara](#), cut prices (CORRECT: in table)
  3. invest in [Microsoft, New York](#)-based analyst (INCORRECT, contradicted by entry  $\langle$ Microsoft, Redmont $\rangle$ )
  4. found at [ASDA, Irving](#). (????, unknown, no contradiction  $\rightarrow$  disregard evidence)
- disregard unclear evidence such as 4.
  - Thus,  $Conf(P_{43}) = \frac{2}{2+1}$

- Consider productivity, not just reliability:

$$Conf_{RlogF}(P) = Conf(P) \log_2(P.positive)$$

- Normalized  $Conf_{RlogFNorm}(P)$ :

$$Conf_{RlogFNorm}(P) = \frac{Conf_{RlogF}(P)}{\max_{i \in \mathcal{P}} Conf(i)}$$

(this brings  $Conf_{RlogFNorm}(P)$  into range [0...1])

- $\max_{i \in \mathcal{P}} Conf(i)$  is the largest confidence value seen with any pattern
- $Conf_{RlogFNorm}(P)$  is a rough estimate of the probability of pattern  $P$  producing a valid tuple (called  $Conf(P)$  hereafter)

- Confidence of a tuple  $T$  is probability that at least one valid tuple is produced:

$$Conf(T) = 1 - \prod_{i=0}^{|P|} (1 - Conf(P_i) Match(C_i, P_i))$$

$P = \{P_i\}$  is the set of patterns that generated  $T$

$C_i$  is the context associated with an occurrence of  $T$

$Match(C_i, P_i)$  is goodness of match between  $P_i$  and  $C_i$

- Explanation: probability of every pattern matched incorrectly:

$$Prob(T \text{ is NOT valid}) = \prod_{i=0}^{|P|} (1 - P(i))$$

- Formula due to the assumption that for an extracted tuple  $T$  to be valid, it is sufficient that at least **one** pattern matched the “correct” text context of  $T$ .

- Then reset confidence of patterns:

$$Conf(P) = Conf_{new}(P)W_{updt} + Conf_{old}(P)(1 - W_{updt})$$

$W_{updt}$  controls learning rate: does system trust old or new occurrences more? Here:  $W_{updt} = 0.5$

- Throw away tuples with confidence  $< \tau_t$

Conf	middle	right
1	<based, .53>, <in, .53>	<”,”, .01>
.69	<””, .42>, <s, .42>, <headquarters, .42>, <in, .42>	
.61	<(, .93>	<), .12>

- Use training corpus to set parameters:  $\tau_{sim}, \tau_t, \tau_{sup}, I_{max}, W_{left}, W_{right}, W_{middle}$
- Only input: 5  $\langle o, l \rangle$  tuples
- Punctuation matters: performance decreases when punctuation is removed
- Recall b/w .78 and .87 ( $\tau_{sup} > 5$ ); precision .90 ( $\tau_{sup} > 4$ )
- High precision possible (.96 with  $\tau_t = .8$ ); remaining problems come from NE recognition
- Pattern evaluation step responsible for most improvement over DIPRE

## Summary: IE and template matching, learning

- Possible to learn simple relations from positive examples (Snowball)
- Possible to learn more diverse relations from annotated training corpus (Riloff)
- Even modest performance can be useful
  - Later manual verification
  - In circumstances where there would be no time to review source documents, so incomplete extracted information is better than none

Current methods perform well if

- Information to be extracted is expressed directly (no complex inference is required)
- Information is predominantly expressed in a relatively small number of forms
- Information is expressed locally within the text

Difference between IE and QA (next time):

- IE is domain dependent, open-domain QA is not

- Ellen Riloff, Automatically constructing a dictionary for information extraction tasks. In Proc. 11th Ann. Conference of Artificial Intelligence, p 811-816, 1993
- Eugene Agichtein, Luis Gravano: Snowball: Extracting Relations from Large Plain-Text Collections, Proceedings of the Fifth ACM International Conference on Digital Libraries, 2000

# Information Retrieval

## Lecture 7: Question Answering

Computer Science Tripos Part II



UNIVERSITY OF  
CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

### Question Answering: Task definition in TREC-QA

186

- 
- QA Track since TREC-1999: Open-domain factual textual QA
  - Task requirements (in comparison with IR):
    1. Input: NL questions, not keyword-based queries
    2. Output: answers, not documents
  - Rules:
    - All runs completely automatic
    - Frozen systems once questions received; answers back to TREC within one week
    - Answers may be extracted or automatically generated from material in document collection only
    - The use of external resources (dictionaries, ontologies, WWW) is allowed
    - Each returned answer is checked manually by TREC-QA (no comparison to gold standard)

---

TREC-8	How many calories are there in a Big Mac? Where is the Taj Mahal?
TREC-9	Who invented the paper clip? How much folic acid should an expectant mother take daily? Who is Colin Powell?
TREC-10	What is an atom? How much does the human adult female brain weigh? When did Hawaii become a state?

## Questions in TREC

- 
- **Type of question:** reason, definition, list of instances, context-sensitive to previous questions (TREC-10)
  - **Source of question:** invented for evaluation (TREC-8); since TREC-9 mined from logs (Encarta, Excite)
    - → strong impact on task: more realistic questions are harder on assessors and systems, but more representative for training
  - **Type of answer string:** 250 Bytes (TREC-8/9, since TREC-12); 50 Bytes (TREC-8–10); exact since TREC-11
  - **Guarantee of existence of answer:** no longer given since TREC-10

What river in the US is known as the Big Muddy?

System A:	the Mississippi
System B:	Known as Big Muddy, the Mississippi is the longest
System C:	as Big Muddy , the Mississippi is the longest
System D:	messed with . Known as Big Muddy , the Mississip
System E:	Mississippi is the longest river in the US
System F:	the Mississippi is the longest river in the US
System G:	the Mississippi is the longest river(Mississippi)
System H:	has brought the Mississippi to its lowest
System I:	ipes.In Life on the Mississippi,Mark Twain wrote t
System K:	Southeast;Mississippi;Mark Twain;officials began
System L:	Known; Mississippi; US,; Minnessota; Cult Mexico
System M:	Mud Island,; Mississippi; "The; history; Memphis

Decreasing quality of answers

## Manual checking of answers

- Systems return [docid, answer-string] pairs; mean answer pool per question judged: 309 pairs
- Answers judged in the context of the associated document
- "Objectively" wrong answers okay if document supports them
  - Taj Mahal
- Considerable disagreement in terms of absolute evaluation metrics
- But relative MRRs (rankings) across systems very stable

- Ambiguous answers are judged as “incorrect”:

What is the capital of the Kosovo?

250B answer:

protestors called for intervention to end the “Albanian uprising”. At [Vucitrn](#), 20 miles northwest of [Pristina](#), five demonstrators were reported injured, apparently in clashes with police. Violent clashes were also repo

- Answers need to be supported by the document context → the second answer is “unsupported”:

What is the name of the late Phillippine President Marco’s wife?

- Ferdinand Marcos and his wife Imelda. . . → [supported]
- Imelda Marcos really liked shoes. . . → [unsupported]

## List task (TREC-10, since TREC-12)

192

- 25 questions: retrieve a given target number of instances of something
- Goal: force systems to assemble an answer from multiple strings
  - Name 4 US cities that have a ‘‘Shubert’’ theater
  - What are 9 novels written by John Updike?
  - What are six names of navigational satellites?
  - Name 20 countries that produce coffee.
- List should not be easily located in reference work
- Instances are guaranteed to exist in collection
- Multiple documents needed to reach target, though single documents might have more than one instance
- Since TREC-12: target number no longer given; task is to find all



- Task is precision-oriented: only look at top 5 answers
- Score for individual question  $i$  is the reciprocal rank  $r_i$  where the first correct answer appeared (0 if no correct answer in top 5 returns).

$$RR_i = \frac{1}{r_i}$$

- Possible reciprocal ranks per question: [0, 0.2, 0.25, 0.33, 0.5, 1]
- Score of a run (MRR) is mean over  $n$  questions:

$$MRR = \frac{1}{n} \sum_{i=1}^n RR_i$$

## Example: Mean reciprocal rank

162: What is the capital of Kosovo?

---

1 18 April, 1995, UK GMT Kosovo capital  
 2 Albanians say no to peace talks in Pr  
 3 0 miles west of Pristina, five demon  
 4 Kosovo is located in south and south  
 5 The provincial capital of the Kosovo

---

$$\rightarrow RR_{162} = \frac{1}{3}$$

23: Who invented the paper clip?

---

1 embrace Johan Vaaler, as the true invento  
 2 seems puzzling that it was not invented e  
 3 paper clip. Nobel invented many useful th  
 4 modern-shaped paper clip was patented in A  
 5 g Johan Valerand, leaping over Norway, in

---

$$\rightarrow RR_{23} = 1$$

2: What was the monetary value of the Nobel Peace Prize in 1989?

---

1 The Nobel poll is temporarily disabled. 1994 poll  
 2 perience and scientific reality, and applied to socie  
 3 Curies were awarded the Nobel Prize together with Beq  
 4 the so-called beta-value. \$40,000 more than expected  
 5 that is much greater than the variation in mean value

---

$$\rightarrow RR_2 = 0$$

$$\rightarrow MRR = \frac{4}{3} = .444$$

- 
- Average accuracy since 2003: only one answer per question allowed; accuracy is  $\frac{\text{Answers correct}}{\text{Total Answers}}$
  - Confidence-weighted score: systems submit one answer per question and order them according to the confidence they have in the answer (with their best answer first in the file)

$$\frac{1}{Q} \sum_{i=1}^Q \frac{\# \text{correct in first } i}{i}$$

( $Q$  being the number of questions). This evaluation metric (which is similar to Mean Average Precision) was to reward systems for their confidence in their answers, as answers high up in the file participate in many calculations.

## Results

- 
- In TREC-8, 9, 10 best systems returned MMR of .65–.70 for 50B answers, answering around 70–80% of all questions
  - In 55% of the cases where answer was found in the first 5 answers, this answer was in rank 1
  - Accuracy of best system in TREC-10's list task had an accuracy of .75
  - The best confidence-weighted score in TREC-11 achieved was .856 (NIL-prec .578, NIL recall .804)
  - TREC-12 (exact task): Best performance was an accuracy of .700

- Overview of three QA systems:
- Cymphony system (TREC-8)
  - NE plus answer type detection
  - Shallow parsing to analyse structure of questions
- SMU (TREC-9)
  - Matching of logical form
  - Feedback loops
- Microsoft (TREC-10)
  - Answer redundancy and answer harvesting
  - Claim: “Large amounts of data make intelligent processing unnecessary.”

- Question Processing
  - Shallow parse
  - Determine expected answer type
  - Question expansion
- Document Processing
  - Tokenise, POS-tag, NE-index
- Text Matcher (= Answer production)
  - Intersect search engine results with NE
  - Rank answers

- 
- Over 80% of 200 TREC-8 questions ask for a named entity (NE)
  - NE employed by most successful systems in TREC (Verhees and Tice, 2000))
  - MUC NE types: person, organisation, location, time, date, money, per-cent
  - Textract covers additional types:
    - frequency, duration, age
    - number, fraction, decimal, ordinal, math equation
    - weight, length, temperature, angle, area, capacity, speed, rate
    - address, email, phone, fax, telex, www
    - name (default proper name)
  - Textract subclassifies known types:
    - organisation → company, government agency, school
    - person → military person, religious person

---

### Who won the 1998 Nobel Peace Prize?

Expected answer type: PERSON

Key words: won, 1998, Nobel, Peace, Prize

### Why did David Koresh ask the FBI for a word processor?

Expected answer type: REASON

Key words: David, Koresh, ask, FBI, word, processor

#### Question Expansion:

Expected answer type: [because | because of | due to | thanks to | since | in order to | to VP]

Key words: [ask|asks|asked|asking, David, Koresh, FBI, word, processor]

R1: Name NP(city | country | company) → CITY|COUNTRY|COMPANY

VG[name] NP[a country] that VG[is developing] NP[a magnetic levitation railway system]

R2: Name NP(person\_w) → PERSON

VG[Name] NP[the first private citizen] VG[to fly] PP[in space]

(“citizen” belongs to word class `person_w`).

R3: CATCH-ALL: proper noun

Name a film that has won the Golden Bear in the Berlin Film Festival.

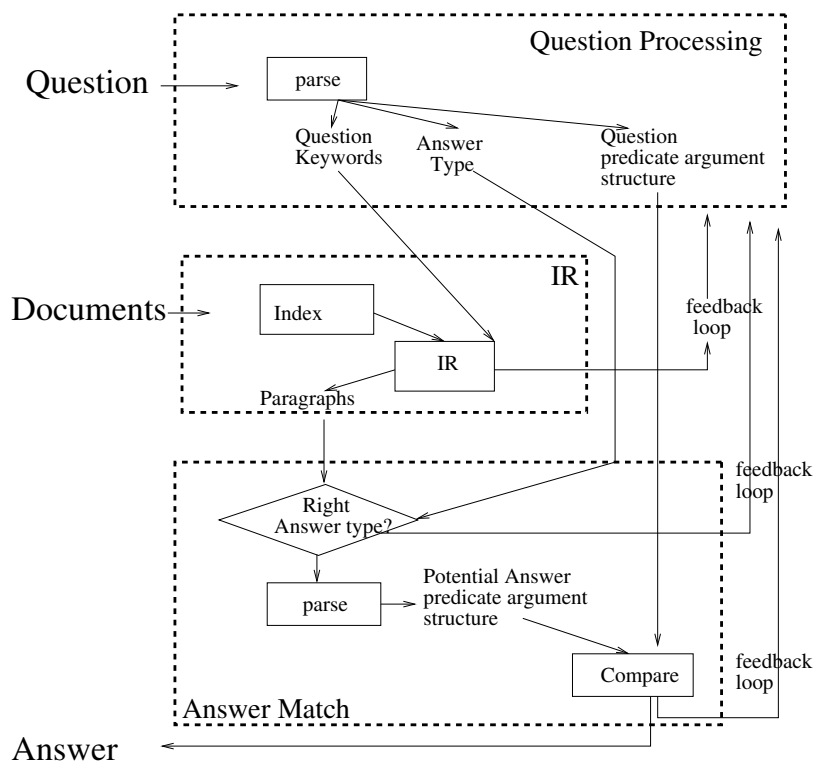
## Direct matching of question words

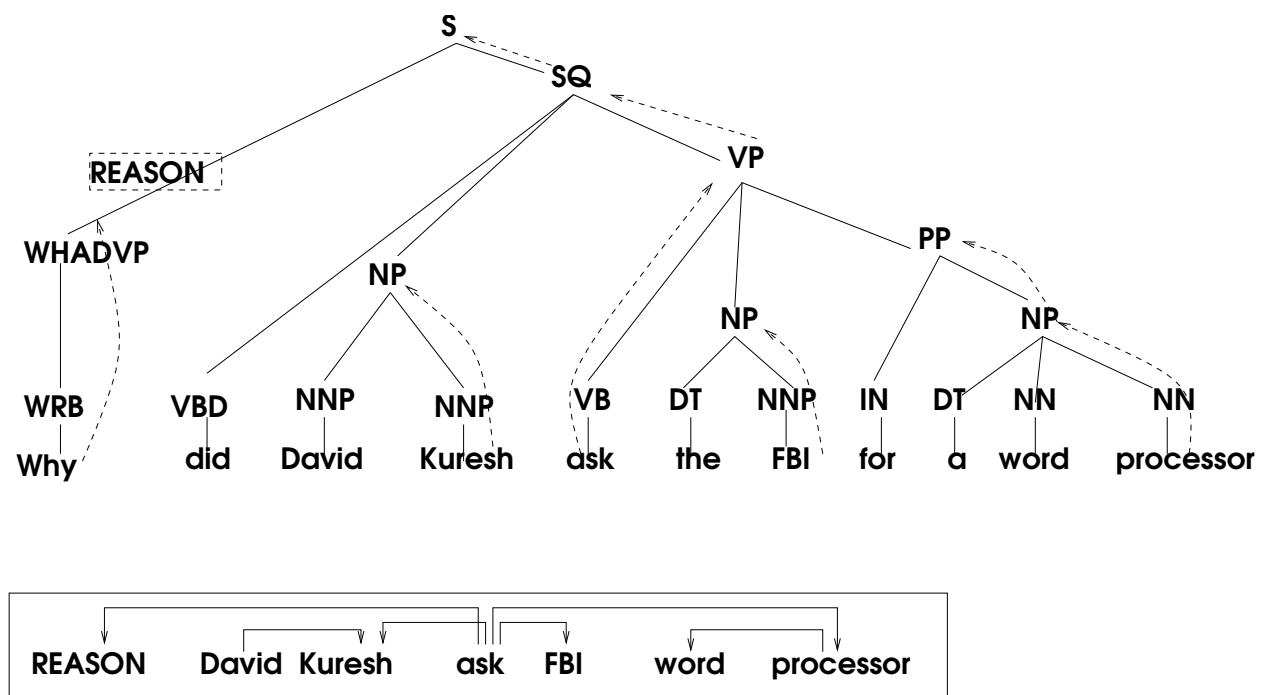
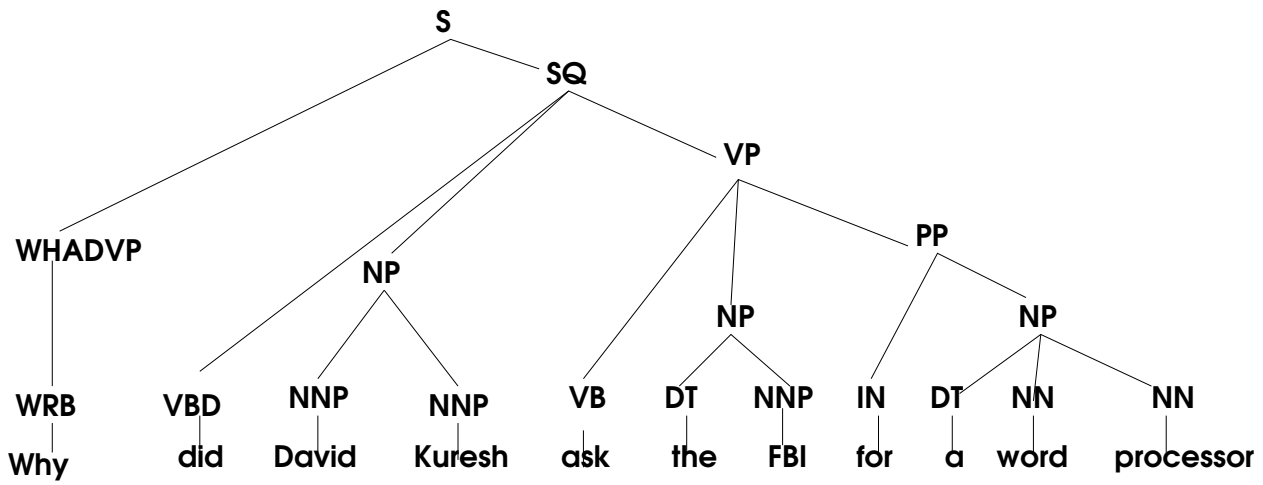
who/whom →	PERSON
when →	TIME/DATE
where/what place →	LOCATION
what time (of day) →	TIME
what day (of the week) →	DAY
what/which month →	MONTH
how often →	FREQUENCY
...	

This classification happens only if the previous rule-based classification did not return unambiguous results.

- Example of a deep processing system which has been extremely successful in TREC-QA (clear winner in most years)
- Machinery beyond answer type determination:
  1. **Variants/feedback loops**: morphological, lexical, syntactic, by reasoning
  2. Comparison between answer candidate and question on basis of **logical form**
- Deep processing serves to
  - capture semantics of open-domain questions
  - justify correctness of answers

## Overview of SMU system





- Morphological (+40%):
  - *Who invented the paper clip?* — Main verb “invent”, ANSWER-TYPE “who” (subject) → add keyword “inventor”
- Lexical (+52%; used in 129 questions):
  - *How far is the moon?* — “far” is an attribute of “distance”
  - *Who killed Martin Luther King?* — “killer” = “assassin”
- Semantic alternations and paraphrases, abductive reasoning (+8%; used in 175 questions)
  - *How hot does the inside of an active volcano get?*
  - Answer in “lava fragments belched out of the mountain were as hot as 300 degrees Fahrenheit”
  - Facts needed in abductive chain:
    - \* volcano IS-A mountain; lava PART-OF volcano
- Combination of loops increases results considerably (+76%)

- Circumvent difficult NLP problems by using more data
- The web has 2 billion indexed pages
- Claim: deep reasoning is only necessary if search ground is restricted
- The larger the search ground, the greater the chance of finding answers with a simple relationship between question string and answer string:

### Who killed Abraham Lincoln?

DOC 1	<a href="#">John Wilkes Booth</a> is perhaps America’s most infamous assassin. He is best known for having fired the bullet that ended Abraham Lincoln’s life.	TREC
DOC 2	<a href="#">John Wilkes Booth</a> killed Abraham Lincoln.	web



1. Question processing is minimal: reordering of words, removal of question words, morphological variations
2. Matching done by Web query (google):
  - Extract potential answer strings from top 100 summaries returned
3. Answer generation is simplistic:
  - Weight answer strings (frequency, fit of match) – learned from TREC-9
  - Shuffle together answer strings
  - Back-projection into TREC corpus: keywords + answers to traditional IR engine
4. Improvement: Expected answer type filter (24% improvement)
  - No full-fledged named entity recognition

## Query string generation

Rewrite module outputs a set of 3-tuples:

- Search string
- Position in text where answer is expected with respect to query string : LEFT|RIGHT|NULL
- Confidence score (quality of template)

Who is the world's richest man married to?

[ +is the world's richest man married to	LEFT	5 ]
[ the +is world's richest man married to	LEFT	5 ]
[ the world's +is richest man married to	RIGHT	5 ]
[ the world's richest +is man married to	RIGHT	5 ]
[ the world's richest man married +is to	RIGHT	5 ]
[ the world's richest man married to +is	RIGHT	5 ]
[ world's richest man married	NULL	2 ]
[ world's AND richest AND married	NULL	1 ]

- Obtain 1-grams, 2-grams, 3-grams from google short summaries
- Score each n-gram  $n$  according to the weight  $r_q$  of query  $q$  that retrieved it
- Sum weights across all summaries containing the ngram  $n$  (this set is called  $S_n$ )

$$w_n = \sum_{n \in S_n} r_q$$

$w_n$ : weight of ngram  $n$

$S_n$ : set of all retrieved summaries which contain  $n$

$r_q$ : rewrite weight of query  $q$

---

## Answer string generation

- Merge similar answers (ABC + BCD  $\rightarrow$  ABCD)
  - Assemble longer answers from answer fragments
  - Weight of new n-gram is maximum of constituent weights
  - Greedy algorithm, starting from top-scoring candidate
  - Stop when no further ngram tiles can be detected
  - But: cannot cluster “redwoods” and “redwood trees”
- Back-projection of answer
  - Send keywords + answers to traditional IR engine indexed over TREC documents
  - Report matching documents back as “support”
- Always return NIL on 5th position

- Time sensitivity of questions:  
Q1202: Who is the Governor of Alaska? → system returns governor in 2001, but TREC expects governor in 1989.
- Success stories:

Question	Answer	TREC document
What is the birth-stone for June?	Pearl	for two weeks during June (the pearl is the birth-stone for those born in that month)
What is the rainiest place on Earth?	Mount Wailaleale	and even Pago Pago, noted for its prodigious showers, gets only about 196 inches annually (The titleholder, according to the National Geographic Society, is Mount Wailaleale in Hawaii, where about 460 inches of rain falls each year).

## Microsoft system: Discussion

- Results: mid-range (.347 MRR, 49% no answer)
- Development time of less than a month
- Produced “exact strings” before TREC-11 demanded it: average returned length 14.6 bytes
- Does this system undermine of QA as a gauge for NL understanding?
  - If TREC wants to measure straight performance on factual question task, less NLP might be needed than previously thought
  - But if TREC wants to use QA as test bed for text understanding, it might now be forced to ask “harder” questions
- And still: the really good systems are still the ones that do deep NLP processing!

- 
- Open domain, factual question answering
  - TREC: Source of questions matters (web logs v. introspection)
  - **Mean reciprocal rank** main evaluation measure
  - MRR of best systems 0.68 - 0.58
  - Best systems answer about 75% of questions in the first 5 guesses, and get the correct answer at position 1.5 on avg ( $\frac{1}{.66}$ )
  - System technology
    - NE plus answer type detection (Cymphony)
    - Matching of logical form, Feedback loops (SMU)
    - Answer redundancy and answer harvesting (Microsoft)

- 
- Teufel (2007): Chapter *An Overview of evaluation methods in TREC Ad-hoc Information Retrieval and TREC Question Answering*. In: L. Dybkjaer, H. Hensen, W. Minker (Eds.) *Evaluation of Text and Speech Systems*. Springer, Dordrecht, The Netherlands.
  - Ellen Voorhees (1999): *The TREC-8 Question Answering Track Report*, Proceedings of TREC
  - R. Srihari and W. Li (1999): "Information-extraction supported question answering", TREC-8 Proceedings
  - S. Harabagiu et al (2001), "The role of lexico-semantic feedback in open-domain textual question-answering", ACL-2001
  - E. Brill et al (2001), "Data intensive question answering", TREC-10 Proceedings

# Information Retrieval

## Lecture 8: Automatic Summarisation

Computer Science Tripos Part II



UNIVERSITY OF  
CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

---

### Summarisation – an impossible task?

218

- Summarisation is intelligent and linguistically viable information compression
- Part of human activity in many different genres
  - TV guide: movie plot summaries
  - Blurb on back of book
  - Newsflashes
  - Subtitles
- Why do research in automatic summarisation?
  - Practical reasons: information compression needed in today's information world
  - Scientific reasons: summarisation is a test bed for current document understanding capabilities

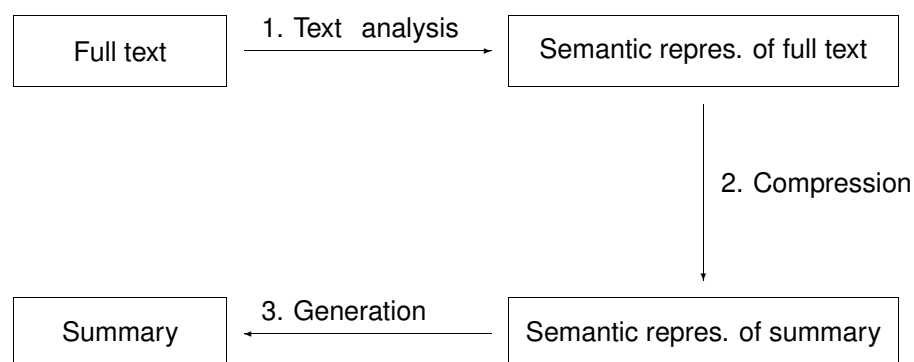
- Compress the “most important” points of a text, express these main points in textual form
- Information reduction
- Different types of summaries
  - informative/indicative
    - \* informative: summary replaces full document v.
    - \* indicative: decision aid for question “should I read the full document?”
  - abstract/extract
    - \* abstract (generated text) v.
    - \* extract (verbatim text snippets)

## Properties of a good summary

- Considerably shorter than the input text
- Covers main points of input text
- Truth-preserving
- A good text in its own right (coherence...)
- Additional goals: flexibility (with respect to length, user, task)

- Abstractors are employed at indexing/abstracting companies which produce abstract journals
- Need expert knowledge about summarising and about domain
- Several studies of human abstractors (Cremmins 1996, Endress-Niggemeyer 1995, Liddy 1991)
- Studies show that human abstractors
  - extract textual material, rework it (Cremmins, E-N)
  - only create new material from scratch when they have to, by generalisation and inference (Cremmins, E-N)
  - have a consistent building plan of a summary in their minds, but agree more on type of information to be put into summary than on the actual sentences (Liddy)
- But: Instructions for abstractors too abstract to be used for actual algorithms

## Text summarisation: the deep model



Steps of the deep model:

1. Analysis of text into semantic representation
2. Manipulation (compression) of semantic representation
3. Text generation from semantic representation

- Compression methods exist (step 2)
  - Summarisation model by Kintsch and van Dijk (1979), based on propositions and human memory restrictions
  - Reasoning theories, e.g. by Lehnert (1982)
- Natural and flexible text generation exists (step 3), working from semantic representation
  - McKeown et al.: Generation from basketball game statistics, weather reports
  - Moore and DiEugenio: Generation of tutor's explanations
- Bottleneck: text analysis (step 1)

## Summarisation by fact extraction (Radev and McKeown 1998, CL)

224

Compress several descriptions about the same event from multiple news stories

MESSAGE: ID	<b>TST-REU-0001</b>
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 3, 1996 11:30
PRMSOURCE: SOURCE	
INCIDENT: DATE	March 3, 1996
INCIDENT: LOCATION	Jerusalem
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	"killed: 18" "wounded: 10"
PERP: ORGANIZATION ID	

MESSAGE: ID	<b>TST-REU-0002</b>
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 4, 1996 07:20
PRMSOURCE: SOURCE	Israel Radio
INCIDENT: DATE	March 4, 1996
INCIDENT: LOCATION	Tel Aviv
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	"killed: at least 10" "wounded: 30"
PERP: ORGANIZATION ID	

MESSAGE: ID	<b>TST-REU-0003</b>
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 4, 1996 14:20
PRMSOURCE: SOURCE	
INCIDENT: DATE	March 4, 1996
INCIDENT: LOCATION	Tel Aviv
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	"killed: at least 13" "wounded: more than 100"
PERP: ORGANIZATION ID	"Hamas"

MESSAGE: ID	<b>TST-REU-0004</b>
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 4, 1996 14:30
PRMSOURCE: SOURCE	
INCIDENT: DATE	March 4, 1996
INCIDENT: LOCATION	Tel Aviv
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	"killed: at least 12" "wounded: 105"
PERP: ORGANIZATION ID	"Hamas"



- Reason over templates
- New templates are generated by combining other templates
- The most important template, as determined by heuristics, is chosen for generation
- Rules:
  - **Change of perspective:** If the same source reports conflicting information over time, report both pieces of information
  - **Contradiction:** If two or more sources report conflicting information, choose the one that is reported by **independent** sources
  - **Addition:** If additional information is reported in a **subsequent** article, include the additional information
  
  - **Refinement:** Prefer more specific information over more general one (name of a terrorist group rather than the fact that it is Palestinian)
  - **Agreement:** Agreement between two sources is reported as it will heighten the reader's confidence in the reported fact
  - **Superset/Generalization:** If the same event is reported from different sources and all of them have incomplete information, report the combination of these pieces of information
  - **Trend:** If two or more messages reflect similar patterns over time, these can be reported in one statement (e.g. three consecutive bombings at the same location)
  - **No Information:** Report the lack of information from a certain source when this would be expected
- Output summary, deep-generated:

Reuters reported that 18 people were killed in a Jerusalem bombing Sunday. The next day, a bomb in Tel Aviv killed at least 10 people and wounded 30 according to Israel Radio. Reuters reported that the radical Muslim group Hamas had claimed responsibility for the act.
- Problem: domain-specificity built into the templates

- 
- Split text in units (paragraphs or sentences or text tiles)
  - Assign each unit a score of importance/“extractworthiness”, using sentential and/or relational features
    - Sentential features of a unit can be calculated in isolation, e.g. number of TF/IDF words or location
    - Relational features of a unit are calculated in context of other units, e.g. unit with highest amount of shared terms
  - Extract sentences with highest score verbatim as extract

- 
- Text is globally structured (rhetorical sections, anecdotal/summary beginning in journalistic writing) – location feature
  - Text is locally structured (paragraph structure; headlines and sub-headlines) – paragraph structure feature
  - Important concepts/terms mark important prepositions – tf/idf feature
  - Certain typographic regions are good places to find important concepts: captions, title, headlines – title feature
  - Sentence length is important, but the experts argue; probably genre-dependent
  - Phrases mark important sections (“in this paper”, “most important”) and less important sections (hedging by auxiliaries, adverbs) – cue phrase feature

---

1. Concept feature (Luhn, 1958)

- Find concepts using  $tf$  (nowadays:  $tf*idf$ ), sentence score = no of frequency concepts in sentence

2. Header feature (Baxendale, 1959)

- Find concepts in title (variation: title and headlines), sentence score = no of title concepts in sentence

3. Location feature (Edmundson, 1969)

- Divide text into  $n$  equal sections
- sentences in section  $1 \leq i \leq n$  get sentence score =  $\frac{1}{i}$
- Always used in combination

---

## Sentential features, II

4. Paragraph feature

- First sentence in paragraph gets a higher score than last one, and higher than sentences in the middle
- Always used in combination

5. Cue phrases (Paice, 1991)

6. First-sentence-in-section feature

7. Sentence length

8. Occurrence of bonus or malus word (ADAM system, Zomora (1972))

9. Occurrence of a named entity (Kupiec et al., 1995)

- Combinations of features are more robust than single features
- Manual feature combination (Edmundson):

$$Score(S) = \alpha A + \beta B + \dots \omega O$$

A, B,..O: feature scores

$\alpha, \beta, \omega$ : manual weights

- Kupiec, Pedersen, Chen: A trainable document summariser, SIGIR 1995
- Create examples of sentences that are abstract-worthy, calculate their features, using 5 well-known features ( $F_1 \dots F_5$ )
- Use Naive Bayesian classifier:

$$P(s \in S | F_1, \dots, F_k) = \frac{P(F_1, \dots, F_k | s \in S) P(s \in S)}{P(F_1, \dots, F_k)} \approx \frac{P(s \in S) \prod_{j=1}^k P(F_j | s \in S)}{\prod_{j=1}^k P(F_j)}$$

$P(s \in S | F_1, \dots, F_k)$ : Probability that sentence  $s$  from the source text is included in summary  $S$ , given its feature values;

$P(s \in S)$ : Probability that a sentence  $s$  in the source text is included in summary  $S$  unconditionally; compression rate of the task (constant);

$P(F_j | s \in S)$ : probability of feature-value pair occurring in a sentence which is in the summary;

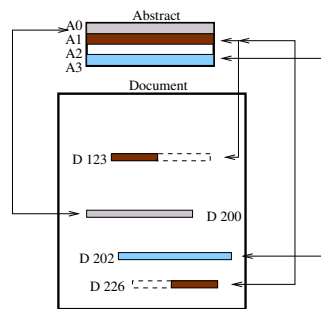
$P(F_j)$ : probability that the feature-value pair  $F_j$  ( $j$  th feature-value pair out of  $k$  feature-value pairs) occurs unconditionally;

Subjective measures:

- Humans subjects select sentences (system developers?)

Looking for more objective measures:

- Earl: indexible sentences
- Kupiec et al: sentences with similarity to abstract sentences



## Kupiec et al: gold standard

- Find best match for each abstract sentence by automatic similarity measure
- One example for a similarity measure is based on the longest common substring:

$$lcs(X, Y) = \frac{length(X) + length(Y) - edit_{i,d}(X, Y)}{2}$$

(where  $edit_{i,d}$  is the minimum number of deletions and insertions needed to transform X into Y).

- Possible similarity measures are the ratio of longest common substring to the maximum length of the two sentences, or the average.
- Reject sentences with similarity  $< .5$ ; accept sentences with similarity  $> 0.8$ , hand-judge sentences with medium similarity  $.5 \leq X \leq .8$

- Corpus of 85 articles in 21 journals
- Extract as many sentences as there are gold standards in the document  
→ precision = recall
- Very high compression makes this task harder
- Results:

Feature	Individual	Cumulative
Cue Phrases	33%	33%
Location	29%	42%
Sentence Length	24%	44%
<i>tf*idf</i>	20%	42%
Capitalization + <i>tf*idf</i>	20%	42%
Baseline		24%

## Example of an extract (Microsoft's AutoSummarize)

Distributional Clustering of English Sentences
<b>Distributional Similarity</b> To cluster nouns $n$ according to their conditional verb distributions $p_n$ , we need a measure of similarity between distributions.
We will take (1) as our basic clustering model.
In particular, the model we use in our experiments has noun clusters with cluster memberships determined by $p(n c)$ and centroid distributions determined by $p(v c)$ .
Given any similarity measure $d(n;c)$ between nouns and cluster centroids, the average cluster distortion is
If we maximize the cluster membership entropy
<b>Clustering Examples</b>
Figure 1 shows the five words most similar to the each [sic] cluster centroid for the four clusters resulting from the first two cluster splits.
<b>Model Evaluation</b>
1990. Statistical mechanics and phrase transitions in clustering.

Source: "Distributional Clustering of English Sentences" by Pereira, Tishby and Lee, ACL 1993

- Extraction is the basis of all robust and reliable summarisation technology widely deployed nowadays
- It can give readers a rough idea of what this text is about
- Information analysts work successfully with them
- Task-based evaluation results:
  - Tombros et al. (1998) show slight improvement in precision and recall and larger improvement in time for a human search task
  - Mani et al. (1999) slight loss in accuracy and large advantage in time saving (50% of the time needed) for a relevance decision task

- Unclear to reader why particular sentence was chosen
- Coherence (syntactic, local problems)
  - Dangling anaphora
  - Unconnected discourse markers
- Cohesion (semantic discontinuities, global)
  - Concepts and agents are not introduced
  - Succession of events does not seem coherent

- E.g. dangling anaphora:
  - resolve anaphora
  - recognize anaphoric use (as opposed to expletive use (“it”, Paice and Husk 1987), then either
    - \* exclude sentences with dangling anaphora
    - \* include previous sentence if it contains the referent (Johnson et al. 1993; also for definite NPs) – But: length!
- There are no fixes for cohesion

1. Subjective judgements:  
How much do subjects like this summary? How coherent, well-written, etc do they find it?
2. Comparison to “gold standard” (predefined right answer):  
In how far does this summary resemble the “right answer”?
3. Task-based evaluation:  
How well can humans perform a task if they are given this summary?
4. Usability evaluation (extrinsic):  
Does the recipient of the summary have to change it? How much?



---

### 1. Subjective judgements

- Subjects can be biased
- How to make sure they understand the same thing under "informativeness", for instance

### 2. Comparison to "gold standard"

- by sentence co-selection, surface string similarity or "information overlap"
- Problematic: humans do not agree on what a good summary is
- Doubt about existence of a "gold standard"

### 3. Task-based evaluation

- Probably the best evaluation around
- Hard to define the task/set up the experiment
- Time-consuming and expensive to do experiment
- For final, end-of-project evaluation, not for day-to-day evaluation

---

## Summary

- 
- Summarisation by deep methods and problems
  - Summarisation by text extraction
    - Importance features
    - Kupiec et al.'s (1995) method and training material
    - Lexical chains
  - Summarisation evaluation and its problems