Processing of large document collections

Part 11 (Question answering systems; Closing of the course)

Helena Ahonen-Myka

Spring 2006

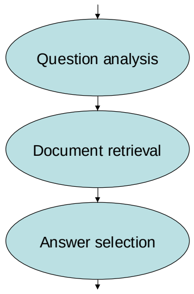
6. Question answering (QA) systems

- Moldovan, Harabagiu, et al: The structure and performance of an open-domain question answering system, 2000
- Cooper, Rüger: A simple question answering system, 2000
- Aunimo, Makkonen, Kuuskoski: Cross-language question answering for Finnish, 2004

Problem setting

- a user gives a natural language question
  - ”When did test pilot Chuck Yeager break the sonic barrier?”
- the question answering system returns (from a document collection) an answer that can be
  - an exact answer (word, phrase…)
  - a snippet of text, in which the answer can be found

Architecture of a Q/A system



Question analysis

Document retrieval

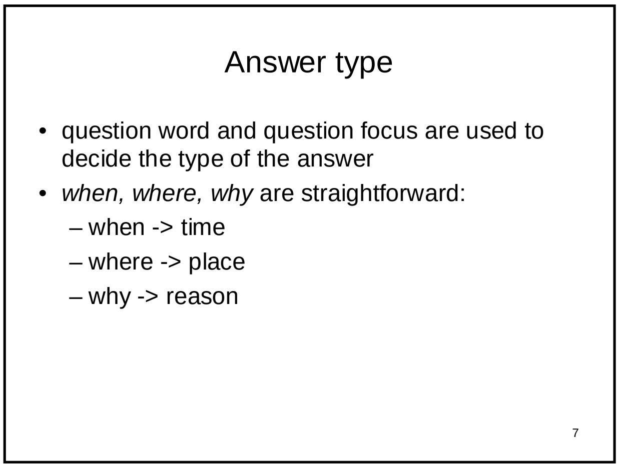
Answer selection

Question analysis

- question classification
  - predefined question classes, e.g. when, who, where, whom, why, description
- answer type
  - what kind of answer are we looking for?
- keyword extraction
  - which keywords should be given to the search engine?

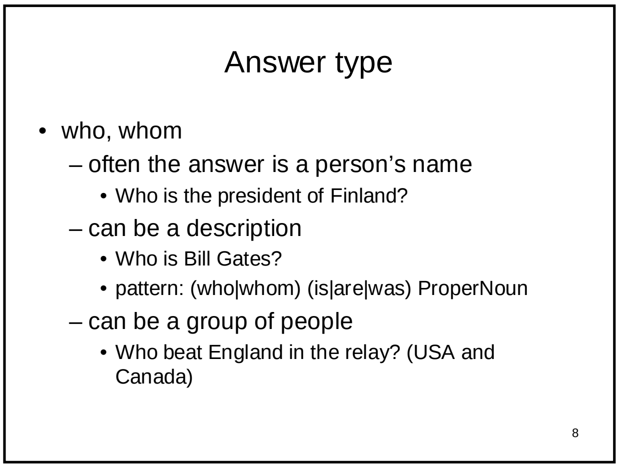
Question classification

- often the question word tells the question class
  - who question -> who class
- what, which, how, and name are less clear
  - What **time** is the train arriving?
  - What **city** is the train arriving at?
  - What is the name of the **driver** of the train?
- question focus is found
  - a phrase in the question that disambiguates it and emphasizes the type of answer being expected
    - the 1st noun group that is not the word ’name’
  - time -> when; city -> where; driver -> who



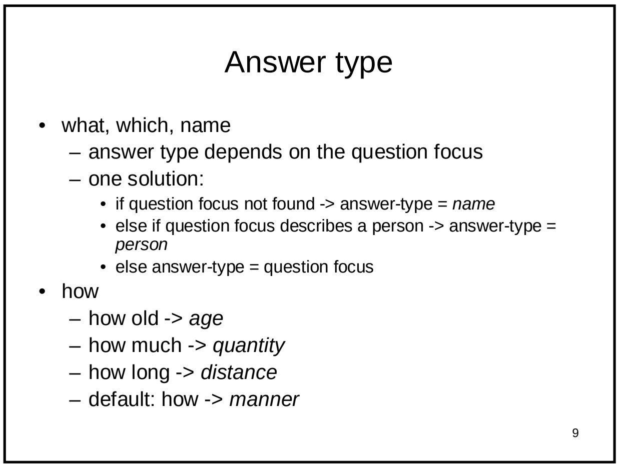
Answer type

- question word and question focus are used to decide the type of the answer
- *when, where, why* are straightforward:
  – when -> time
  – where -> place
  – why -> reason



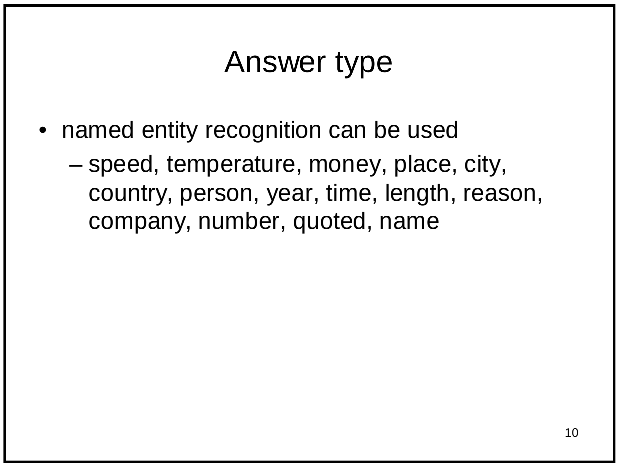
Answer type

- who, whom
  – often the answer is a person's name
    - Who is the president of Finland?
  – can be a description
    - Who is Bill Gates?
    - pattern: (who|whom) (is|are|was) ProperNoun
  – can be a group of people
    - Who beat England in the relay? (USA and Canada)



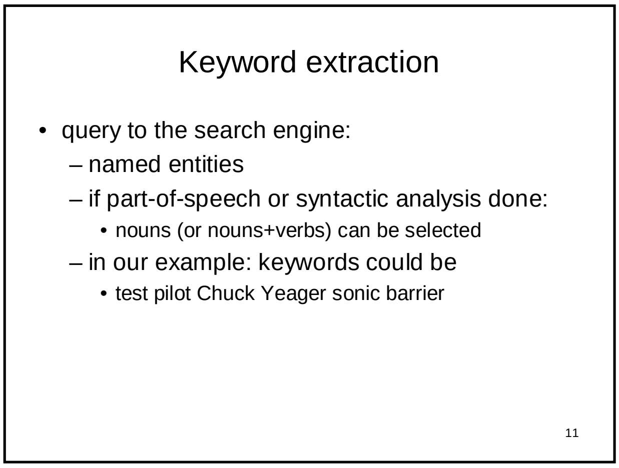
Answer type

- what, which, name
  – answer type depends on the question focus
  – one solution:
    - if question focus not found -> answer-type = *name*
    - else if question focus describes a person -> answer-type = *person*
    - else answer-type = question focus
- how
  – how old -> *age*
  – how much -> *quantity*
  – how long -> *distance*
  – default: how -> *manner*



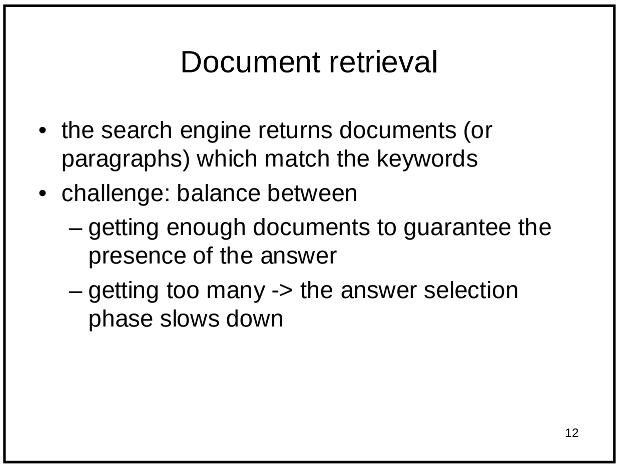
Answer type

- named entity recognition can be used
  – speed, temperature, money, place, city, country, person, year, time, length, reason, company, number, quoted, name



Keyword extraction

- query to the search engine:
  – named entities
  – if part-of-speech or syntactic analysis done:
    - nouns (or nouns+verbs) can be selected
  – in our example: keywords could be
    - test pilot Chuck Yeager sonic barrier



Document retrieval

- the search engine returns documents (or paragraphs) which match the keywords
- challenge: balance between
  – getting enough documents to guarantee the presence of the answer
  – getting too many -> the answer selection phase slows down

Document retrieval

- first, a rigid query can be given
  - test AND pilot AND Chuck AND Yeager AND sonic AND barrier
- if the query does not return enough results, it is relaxed (keywords are dropped)
- additional conditions can be stated
  - the keywords have to occur within a paragraph (or within n paragraphs)

Document retrieval

- keywords:
  - test pilot Chuck Yeager sonic barrier
- document fragment is found:
  - “For many, seeing **Chuck Yeager** – who made his historic super**sonic** flight **Oct. 14, 1947** – was the highlight of this year’s show, in which…”

Answer selection

- candidate answer extraction
  - mark up of regions that could be answers
- candidate scoring
  - heuristics are used to evaluate how likely a candidate is a correct answer
- candidate weighting
  - scores are combined into one final score

Candidate answer extraction

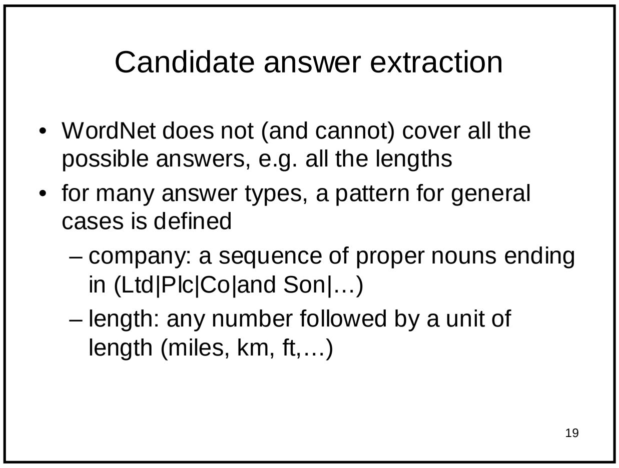
- retrieved text fragments (documents, a set of paragraphs) are split to sentences
  - (some of the) keywords occur in these fragments
- question class, answer type, and keywords guide candidate answer extraction
- named entity recognition (and other linguistic analysis) can be done first

Candidate answer extraction

- semantic knowledge can be used:
  - the question’s answer type is looked up in WordNet and all of its hyponyms are found
    - example: answer type = ”city”
    - WordNet: Helsinki (Tampere, Luanda…) is a kind of city
  - a regular expression is then built by taking a disjunction of those hyponyms
    - (Helsinki|Tampere|Luanda|…)
  - any region of text that matches the regular expression is marked up as a candidate answer

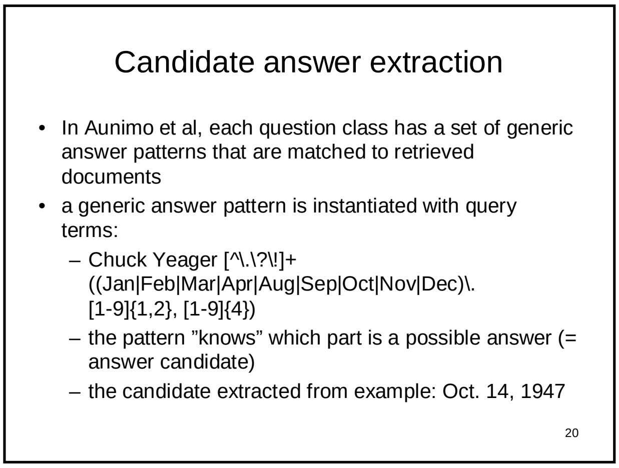
Candidate answer extraction

- exceptions: person, description, general cases
- if the answer concept is a person:
  - regular expression that matches proper names is used
  - WordNet has 300 hyponyms for person (e.g. consumer, coward, defender, guardian…)
- if the answer concept is a description
  - descriptions are hard to define in terms of what words make them up
  - when an entity is first introduced in a text, it is often followed by a comma and then a description
    - ”Bill Gates, Head of Microsoft, said today…”



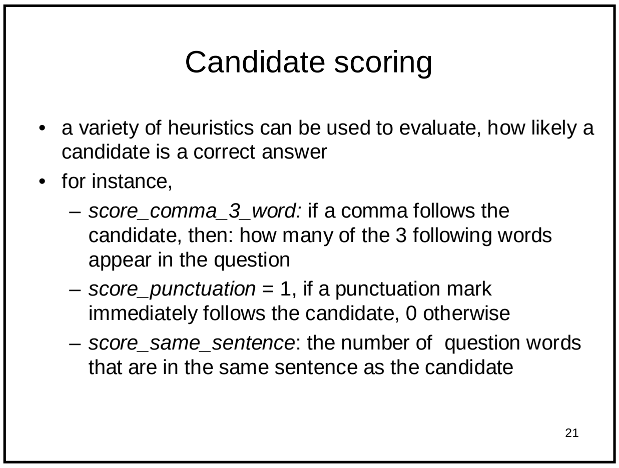
Candidate answer extraction

- WordNet does not (and cannot) cover all the possible answers, e.g. all the lengths
- for many answer types, a pattern for general cases is defined
  - company: a sequence of proper nouns ending in (Ltd|Plc|Co|and Son|…)
  - length: any number followed by a unit of length (miles, km, ft,…)



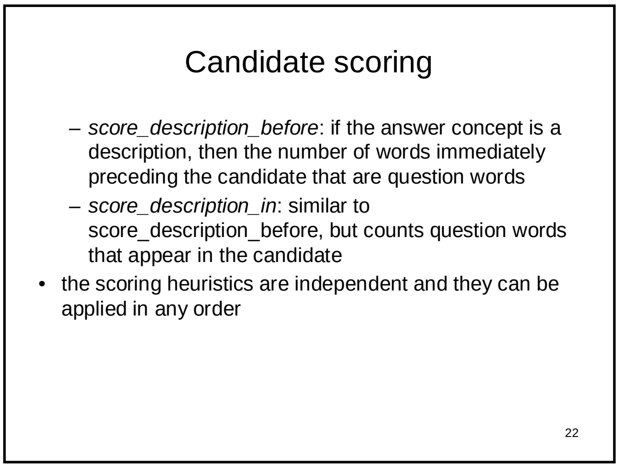
Candidate answer extraction

- In Aunimo et al, each question class has a set of generic answer patterns that are matched to retrieved documents
- a generic answer pattern is instantiated with query terms:
  - Chuck Yeager [^\.\?\!]+ ((Jan|Feb|Mar|Apr|Aug|Sep|Oct|Nov|Dec)\. [1-9]{1,2}, [1-9]{4})
  - the pattern ”knows” which part is a possible answer (= answer candidate)
  - the candidate extracted from example: Oct. 14, 1947



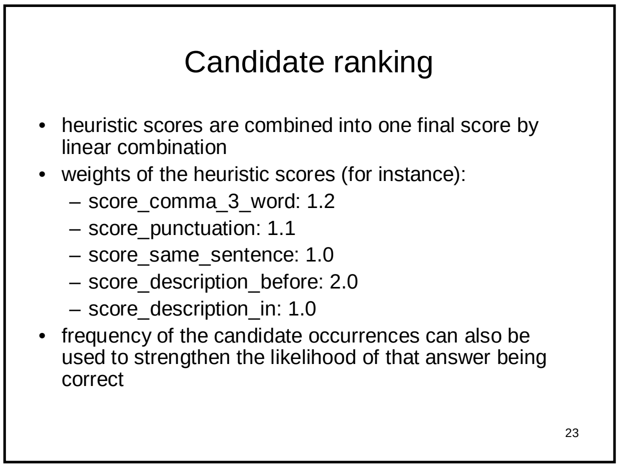
Candidate scoring

- a variety of heuristics can be used to evaluate, how likely a candidate is a correct answer
- for instance,
  - *score_comma_3_word:* if a comma follows the candidate, then: how many of the 3 following words appear in the question
  - *score_punctuation* = 1, if a punctuation mark immediately follows the candidate, 0 otherwise
  - *score_same_sentence*: the number of question words that are in the same sentence as the candidate



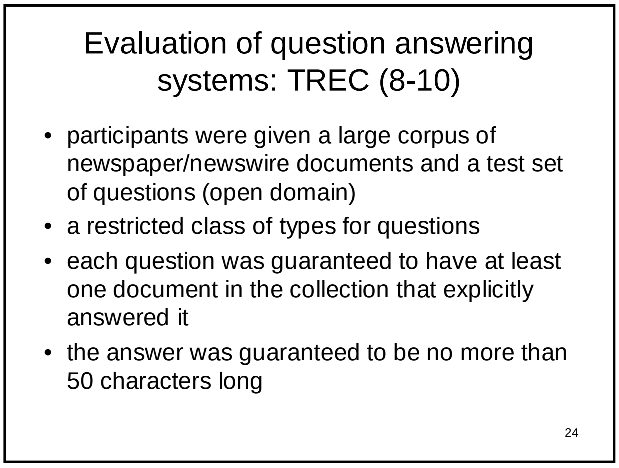
Candidate scoring

  - *score_description_before*: if the answer concept is a description, then the number of words immediately preceding the candidate that are question words
  - *score_description_in*: similar to score_description_before, but counts question words that appear in the candidate
- the scoring heuristics are independent and they can be applied in any order



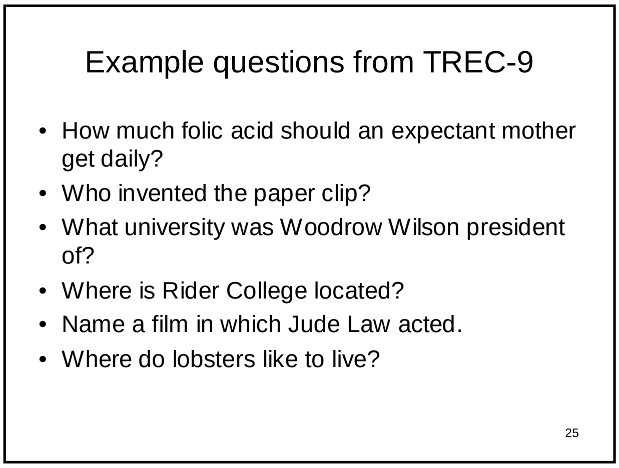
Candidate ranking

- heuristic scores are combined into one final score by linear combination
- weights of the heuristic scores (for instance):
  - score_comma_3_word: 1.2
  - score_punctuation: 1.1
  - score_same_sentence: 1.0
  - score_description_before: 2.0
  - score_description_in: 1.0
- frequency of the candidate occurrences can also be used to strengthen the likelihood of that answer being correct



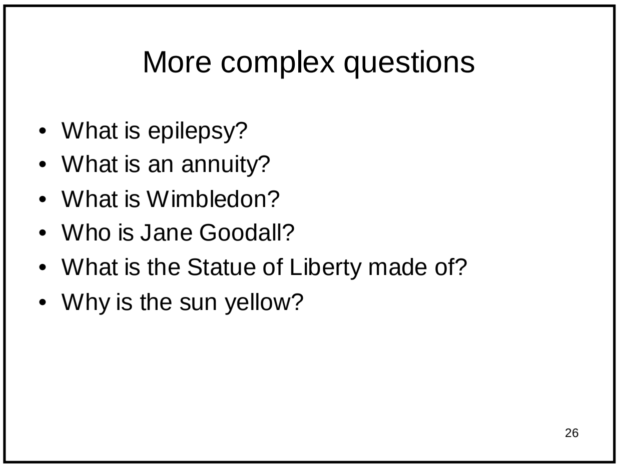
Evaluation of question answering systems: TREC (8-10)

- participants were given a large corpus of newspaper/newswire documents and a test set of questions (open domain)
- a restricted class of types for questions
- each question was guaranteed to have at least one document in the collection that explicitly answered it
- the answer was guaranteed to be no more than 50 characters long



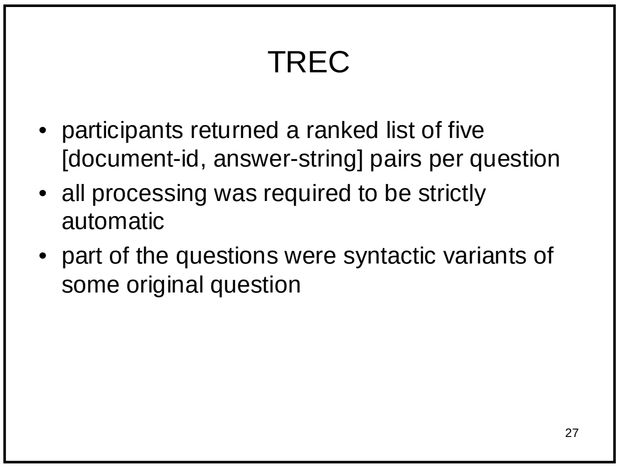
Example questions from TREC-9

- How much folic acid should an expectant mother get daily?
- Who invented the paper clip?
- What university was Woodrow Wilson president of?
- Where is Rider College located?
- Name a film in which Jude Law acted.
- Where do lobsters like to live?



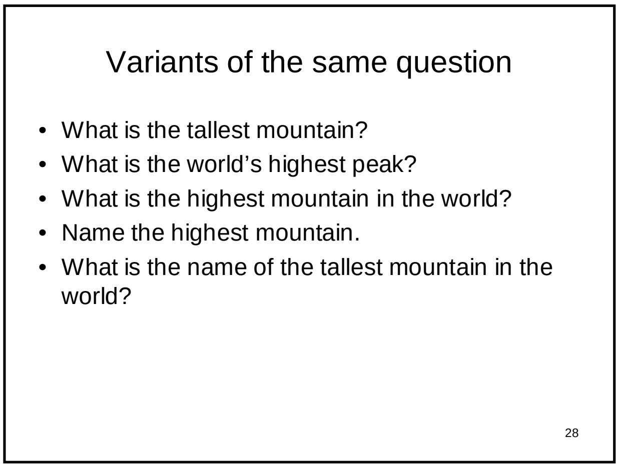
More complex questions

- What is epilepsy?
- What is an annuity?
- What is Wimbledon?
- Who is Jane Goodall?
- What is the Statue of Liberty made of?
- Why is the sun yellow?



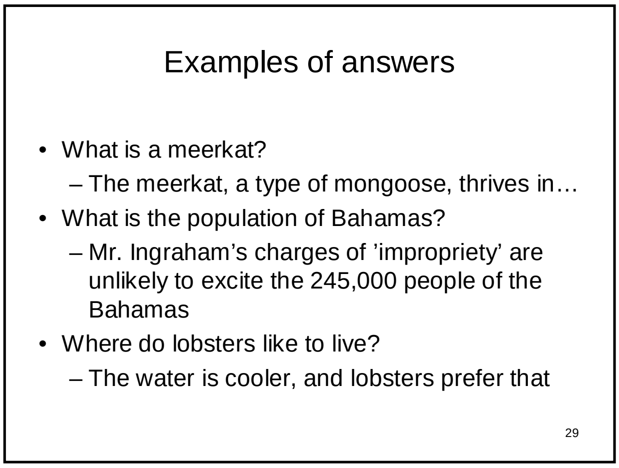
TREC

- participants returned a ranked list of five [document-id, answer-string] pairs per question
- all processing was required to be strictly automatic
- part of the questions were syntactic variants of some original question



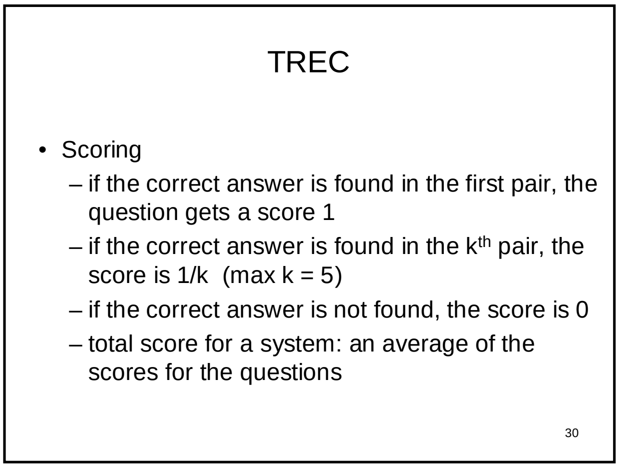
Variants of the same question

- What is the tallest mountain?
- What is the world’s highest peak?
- What is the highest mountain in the world?
- Name the highest mountain.
- What is the name of the tallest mountain in the world?



Examples of answers

- What is a meerkat?
  - The meerkat, a type of mongoose, thrives in…
- What is the population of Bahamas?
  - Mr. Ingraham’s charges of ’impropriety’ are unlikely to excite the 245,000 people of the Bahamas
- Where do lobsters like to live?
  - The water is cooler, and lobsters prefer that



TREC

- Scoring
  - if the correct answer is found in the first pair, the question gets a score 1
  - if the correct answer is found in the kth pair, the score is 1/k (max k = 5)
  - if the correct answer is not found, the score is 0
  - total score for a system: an average of the scores for the questions

Performance of Moldovan, Harabagiu et al (TREC)

	Percentage of questions in top 5	Score
Short answer	68,1%	55,5%
Long answer	77,7%	64,5%

Cross-language QA

- Example: Finnish questions & English answers
- question processing
  - – question in Finnish
    - ”Milloin koelentäjä Chuck Yeager rikkoi äänivallin?”
  - – translation into English word by word -> query
- candidates are retrieved from a collection of documents in English
- answer is extracted from candidates: ”Oct. 14, 1947”

Cross-language QA

- translation problems with Finnish
  - – compound words
    - tuliaselaki: tuliase + laki
    - kulttuuripääkaupunki: kultuuri + pääkaupunki
  - – vocabulary
    - immigrate (en) = tulla siirtolaisena
    - compare: immigrer (fr), immigrare (it), imigrar (pt), immigreren (du), immigrar (sp)

Cross-language QA

- – ambiguity
  - words can be ambiguous
    - – kerros: hiekkakerros, asuinkerros
  - translation adds ambiguity
    - – layer (kerros, kerrostuma, peite, taivukas)
    - – floor (lattia, istuntosali, kerros, tanssilattia, pohja)

Disambiguation

- solution:
  - – select keywords from the question, e.g. all nouns and verbs
  - – take for each keyword the first translation (in a dictionary) for each sense
    - e.g. koelentäjä: 1 translation; rikkoa: 20 translations; äänivalli: 2 translations
  - – build all combinations and make a query with all of them (40 queries)
    - use combinations which return documents (13)
    - number of translations after disambiguation: koelentäjä: 1 translation; rikkoa: 6 translations; äänivalli: 2 translations

Performance of Aunimo et al. (CLEF04)

- Cross-Language Evaluation Forum (CLEF), QA Track
  - – answer is not guaranteed to be found
  - – the system has to tell how confident it is that the answer is correct
  - – one answer only can be returned
  - – the answer is a word or a phrase
- Aunimo et al has two systems: the performance varies 22-29%

More complex QA tasks

- each question may require information from more than one document
  - Name 10 countries that banned beef imports from Britain in the 1990s.
- follow-up questions
  - Which museum in Florence was damaged by a major bomb explosion in 1993?
  - On what day did this happen?

Question-answering in a closed domain

- above, the types of questions belonged to some closed class, but the topics did not belong to any specific domain (open-domain topics)
- in practice, a question-answering system may be particularly helpful in some closed well-known domain, like within some company

Question-answering in a closed domain

- special features in real-life systems
  - the questions can have any type, and they may have errors and spoken-language expressions
  - the same questions (variants) probably occur regularly -> extensive use of old questions
  - closed domain: extensive use of domain-knowledge feasible
    - ontologies, thesauri, inference rules

QA vs IE

- open domain, closed domain?
  - questions, task definitions
- task definition
  - IE: static
  - QA: question defines the task dynamically
- answer
  - IE: structured template
  - QA: text snippet and/or exact answer (~one slot value of a template?)
- similar components can be used
  - language analysis, named entity recognition (WordNet, word lists,…), pattern matching

5. Closing of the course

TC vs TS vs IE vs QA?

- text categorization (TC)
- text summarization (TS)
- information extraction (IE)
- question answering (QA)

General issues

- performance requirements
  - – building vs. use
    - offline vs. online processing
  - – effectiveness vs. efficiency
  - – observed performance in some user task
  - – evaluation
    - simplified research settings vs. real-life environments

General issues

- portability, scalability
  - – amount and type of manual processing
  - – domain/language dependency
  - – are some components available off-the-shelf?
    - but do not use heavy processing for simple tasks
    - e.g. linguistic analysis vs. pattern matching
  - – static vs. dynamic system/component?

General issues

- What is our goal?
  - – automatic text undestanding?
    - and automatic processing based on that?
  - – probably tools for specific tasks for specific users are more reasonable than very generic, open-domain tools

Our learning goals were…

- learn to recognize components of applications/processes
- learn to recognize which (kind of) methods could be used in each component
- learn to implement some methods
- (meta)learn to control learning processes (What do I know? What should I know to solve this problem?)

Exam & exercise points

- exam: Thu 4.5. at 16-19, B123
- points: exam 50 pts, exercises 10 pts
  - – required: ~30 pts (= 1)
- exercise points:
  - – 3 exercises -> 1 point
  - – 5 -> 2
  - – 7 -> 3
  - – 8 -> 4
  - – 9 -> 5
  - – 10 -> 6
  - – 11 -> 7
  - – 12 -> 8
  - – 13 -> 9
  - – 14 -> 10

Kiitos!!!

- … ja hyvää kesää!!