ALTSS Course

CENTRE FOR LANGUAGE TECHNOLOGY
MACQUARIE UNIVERSITY ~ SYDNEY

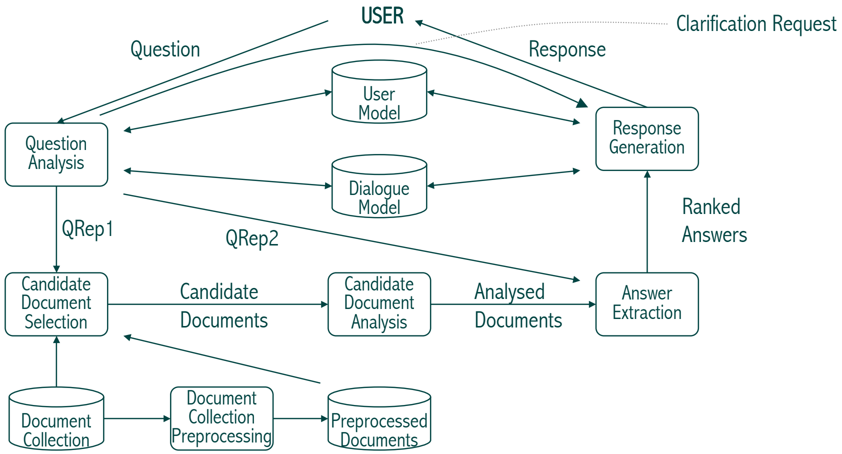
Information Extraction and Question Answering

Lectures 3 & 4

Question Answering

Diego Mollá
diego@ics.mq.edu.au

Architecture of a QA System



USER
Question
Response
Clarification Request
User Model
Question Analysis
Response Generation
Dialogue Model
QRep1
QRep2
Ranked Answers
Candidate Document Selection
Candidate Documents
Candidate Document Analysis
Analysed Documents
Answer Extraction
Document Collection
Document Collection Preprocessing
Preprocessed Documents

Question Analysis

- INPUT: A Natural Language Question
  - Perhaps using a controlled language?
    - form/menu driven interfaces
  - Need to keep track of previous interactions?
    - Dialogue systems, user modeling

Question Analysis

- OUTPUT: One or more representations of the question
- Steps:
  1. Identify semantic type of the entity sought by the question
     - Build hierarchies of question types
  2. Determine additional semantic constraints
     - Keywords
     - Syntactic, semantic relations

Document Collection Preprocessing

- GOAL: Produce an image of the documents that will be used by the other modules
  - Document indexing (like in IR systems)
  - Shallow linguistic analysis
    - tagging, NE recognition, chunking
  - Logical form

Candidate Answer Document Selection

- GOAL: Find the list of documents that most likely contain the answer
- Typical approach: Use a standard IR engine
  - off the shelf
  - customised for the task
  - specifically designed for the task

Candidate Answer Document Analysis

- GOAL: Prepare the documents for the answer extraction process
- (sometimes not needed if there was a document preprocessing stage)
- Possible tasks:
  - named entity recognition
  - Sentence splitting
  - PoS Tagging
  - Chunk parsing (identify NPs, VPs, PPs, etc.)
  - Perhaps semantic interpretation (to find the underlying logical form)

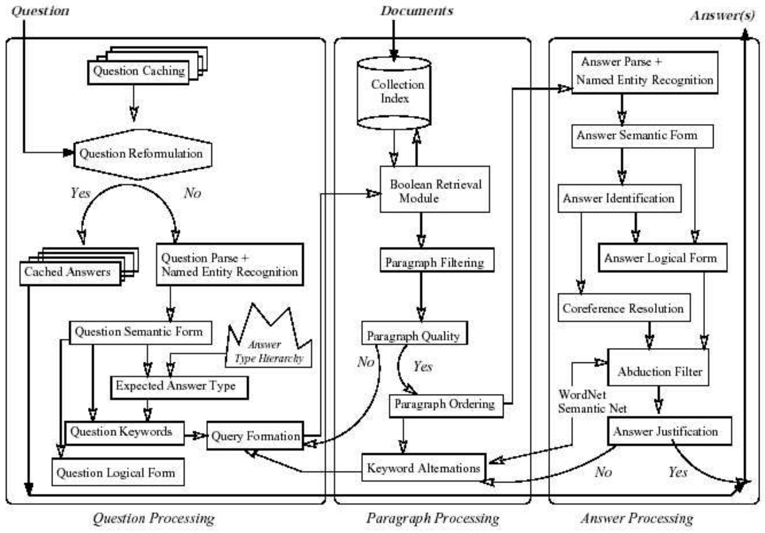
Answer Extraction

- GOAL: Find the answer candidates
- Typical process:
  1. Select text (or a sentence) that contains a string with semantic type compatible to that of the expected answer
     - Perhaps using hyponymy relations – WordNet
  2. Apply other constraints
     - Scoring
     - logical compatibility (though this may be too strict)

Response Generation

- GOAL: Produce an answer
  - Remove extraneous information
  - Add links to the source documents
  - Merge answers or remove duplicate answers

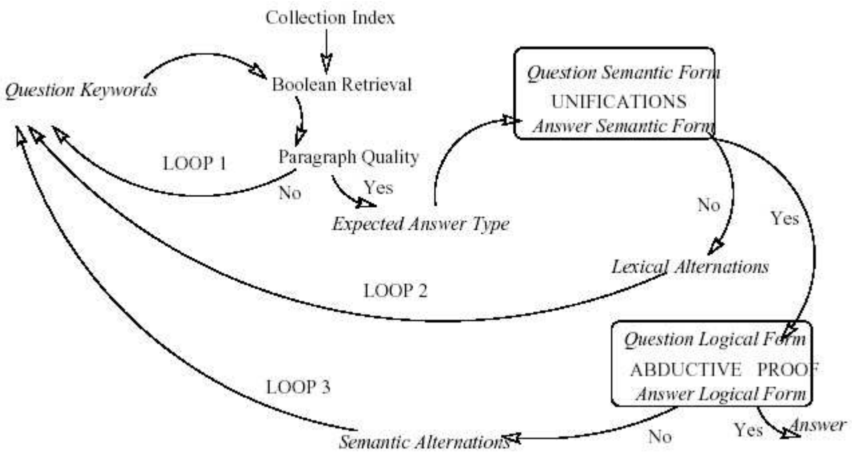
Falcon – System Architecture



Feedback Loops

- Paragraph quality:
  - Add or remove keywords until the number of paragraphs is reasonable for the answer type
- Semantic compatibility:
  - Try alternations of the question keywords until the question and answer semantic forms unify
- Logical justification:
  - Search WordNet for related concepts until there is a logical justification of the answer correctness

Feedback Loops



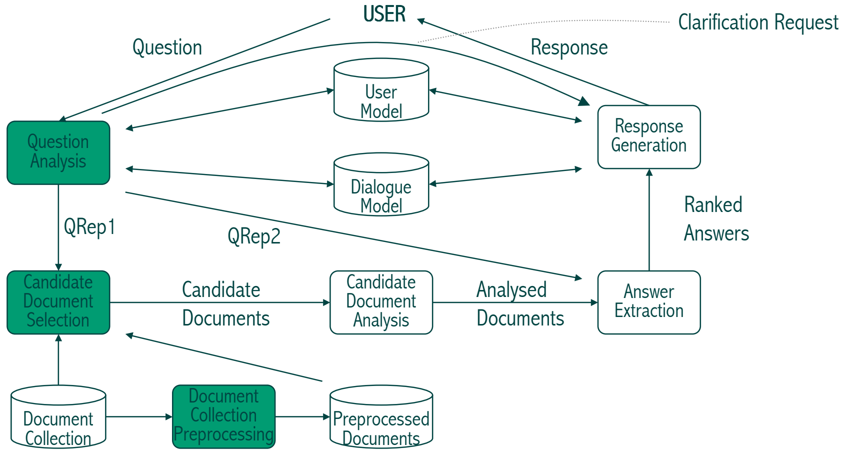
Today's Topics

- Finding Candidate Documents
- Question Analysis
- Answer Extraction
- Using the Web for QA
- ExtrAns
- The Future of QA

QA of large document sets

- The original document set can be very large
- How do we find the needle in the haystack?
- Basic Approach
  1. Get a set of documents that can possibly contain the answer
  2. Do more sophisticated question answering techniques on the preselected documents

Architecture of a QA System



Finding the Candidate Documents

- Query Analysis
  - Shallow approach: extract content words
- Document Preprocessing
  - Index the documents
- Candidate Document Selection
  - Apply standard IR techniques

The Impact of Document Preselection

- If the document preselection stage fails to preselect the right documents, the other modules wouldn't find the answers
- Data: TREC 2002

Preselected documents	Found a document with the answer
1	31.1%
5	53.8%
10	62%
20	68.2%
50	74%
100	76.4%
1000	83%

Document Retrieval as Question Answering

- Document Retrieval techniques can be used to retrieve
  - Documents
  - Paragraphs
- Can't we just use DR techniques to retrieve sentences?
  - Sentence retrieval as question answering
- Use of traditional techniques in TREC8-QA:
  - 250-byte run: 👍
  - 50-byte run: 👎

Information Retrieval

- Retrieving information from document repositories
- Query-based IR
  - Document (ad-hoc) retrieval ← Our focus
  - Passage retrieval
  - Answer extraction
  - Information extraction
  - Question answering

Information Retrieval on the Web

- IR Demos: well, any web search engine!
  - Altavista (http://www.altavista.com)
  - Google (http://www.google.com)
- IR Resources
  - http://www.acm.org/sigir/resources.html
- IR Research:
  - ACM SIGIR (http://www.acm.org/sigir/)
  - TREC (http://trec.nist.gov/)
- On-line Course on IR
  - http://rayuela.ieec.uned.es/~ircourse/

Two Stages in IR

1. Indexing
   - Off-line stage
   - Reduce the document to a description of it: the indices
   - Optimise the representation
2. Retrieval
   - Use the document indices to find the relevant documents
   - Indices can be seen as the indication of what the documents are about (the keywords)

Indexing

- What information to keep?
- Try to keep the words that discriminate the documents best
  - Remove stop words
  - Typically, the most important words are those that are not too frequent or too infrequent
- How to keep the information?
  - Inverted index: For every possible word, list the documents that contain the word

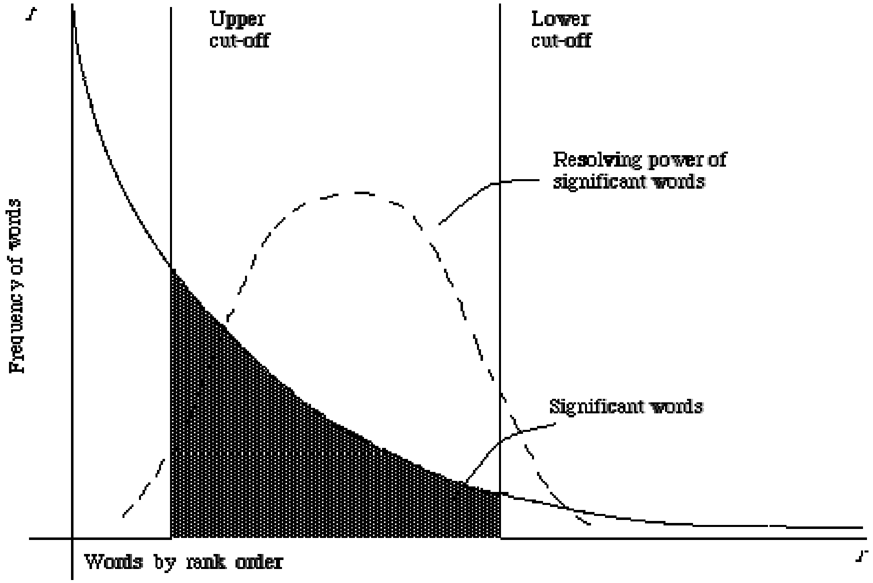
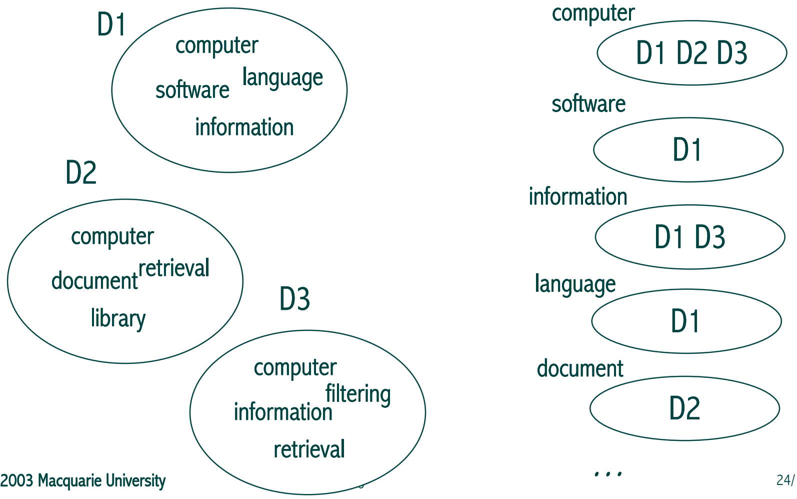


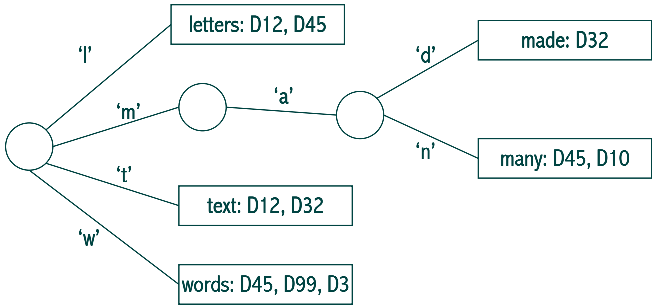
Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adapted from Schultz[44] page 120)

Indexing Inverted Indexing



Building an Inverted Index

- Find an efficient way to store and find the words
  - e.g. a vocabulary trie



‘l’

letters: D12, D45

‘m’

‘a’

‘d’

made: D32

‘n’

many: D45, D10

‘t’

text: D12, D32

‘w’

words: D45, D99, D3

Boolean Retrieval

- Boolean query: Combine the search elements with logical operators
  - x AND y: Documents that contain both x and y
  - x OR y: Documents that contain x, y, or both
  - NOT x: Documents that do not contain x
- Return the documents that satisfy the boolean query
- If using inverted index, the usual set operations can be used:
  - AND → set intersection
  - OR → set union
  - NOT → set complementation

Boolean Retrieval – Example

- Document keywords:

  D1: {computer, software, information, language}
  D2: {computer, document, retrieval, library}
  D3: {computer, information, filtering, retrieval}
- Inverted index:

  computer → {D1, D2, D3}, software → {D1}, information → {D1,D3}, language → {D1}, document → {D2}, retrieval → {D2, D3}, library → {D2}, filtering → {D3}
- Boolean query:

  (information OR document) AND retrieval
- Result:

  $(\{D1,D3\} \cup \{D2\}) \cap \{D2,D3\} = \{D2,D3\}$

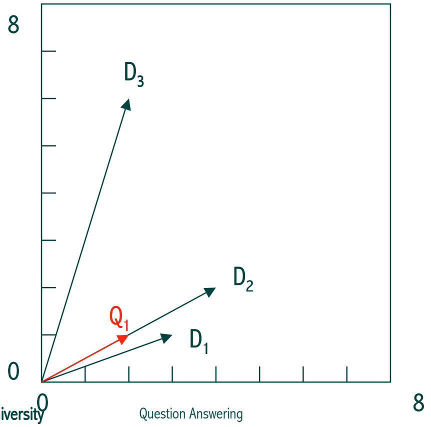
Vector Space Retrieval

- Represent a document as a vector:
  - Each element in the vector indicates the presence (or not) of a specific keyword

    D = (1,0,0,0,1,0,0,3,0,0,2,0,0) (weights are possible)
- Relevance of a document with respect to the query:
  - use a vector similarity function between the document and the query
    - Example with two dimensions:
      $D_1 = (3,1)$, $D_2 = (4,2)$, $D_3 = (2,6)$ $Q_1 = (2,1)$
      What document is most similar to the query?

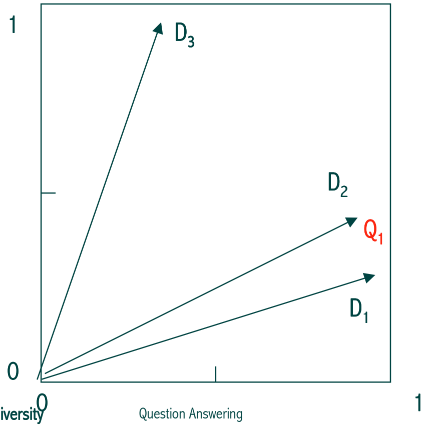
Vector Space Retrieval

- Example with 2 dimensions:



Vector Space Retrieval

- Normalised to the vector length:



Vector Space Retrieval

- Similarity on the basis of the cosine:

$$\mathrm{sim}(D_j, Q_k) = \cos(D_j, Q_k) = \frac{\sum_{i=1}^{N} D_{j,i} Q_{k,i}}{\sqrt{\sum_{i=1}^{N} D_{j,i}^2}\sqrt{\sum_{i=1}^{N} Q_{k,i}^2}}$$

- If the vectors are normalised the formula can be simplified:

$$\mathrm{sim}(D_j, Q_k) = \cos(D_j, Q_k) = \sum_{i=1}^{N} D_{j,i} Q_{k,i}$$

- Normalisation is convenient when the documents are stable:
  - They are normalised once only

Vector Space Retrieval – Term Weighting

- A term that is frequent in the document but it does not appear in many other documents in the collection should get a higher weight
  - Inverse document frequency of a term i:

    $$IDF_i = \log(N/DF_i)$$

    - DF$_i$: number of documents containing term i
    - N: total number of documents
  - Weights of the document vector j:

    $$D_{j,i} = TF_{ij} IDF_i$$

    - TF$_{ij}$: frequency of the term i in the document j
  - Weights of the query k:

    $$Q_{k,i} = (0.5 + \frac{0.5TF_{ik}}{\max_j TF_{jk}}) IDF_i$$

Boolean Retrieval vs. Vector Space Retrieval

preferred by developers of QA systems

- Boolean Retrieval
  - The documents are not ranked
    - You find the document or you don't find it
    - You need to use other means to rank documents (e.g. google ranking system)
  - The user may feel uncomfortable with boolean queries
- Vector Space Retrieval
  - The documents can be ranked
  - Best results when the query contains many words
    - but in real life queries are very short!
    - Solution: Use a document as a "query" (e.g. the option "similar documents" in many web search engines)

Problems of Traditional IR

- Traditional IR:
  - Focus on fast and robust domain-independent approaches
  - Treatment of large amounts of data
  - Limited linguistic analysis
  - Acceptable results for essay-writing scenarios, specially if the documents are relatively large and uniform
- Problems of traditional IR:
  - Recall: Different structures have similar meanings
  - Precision: Similar structures have different meanings
- We need to add linguistic information to IR systems

Different Structures with Similar Meanings

- Lexical meaning
  - Synonyms, spelling variations:
    - "dog" vs. "canine"
    - "file printing" vs. "document printing" (domain-dependent)
  - Hyponyms and hyperonyms:
    - "Accommodation" vs. "camping"
  - 25.7% of the errors of an Open-Domain QA system were due to missing related keywords (Moldovan et al. 2003)
- Passives, ditransitives
  - "plays written by Shakespeare" vs. "plays that Shakespeare wrote"
- Different types of modifiers
  - "books with a red cover" vs. "red-covered books"

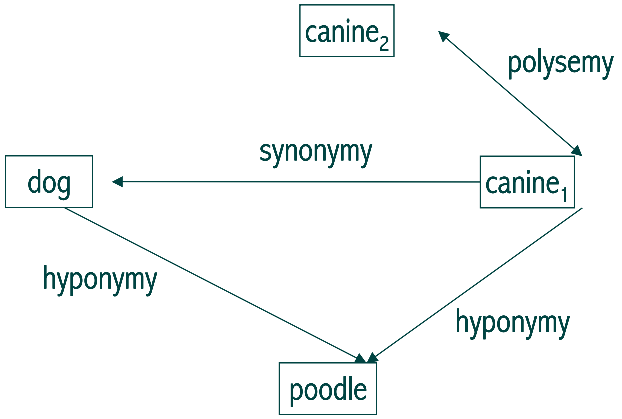
Similar Expressions with Different Meanings

- Homonymy, polysemy
  - "chips" vs. "chips"
- The following differences are lost if a stop list is used:
  - "exports from Australia" vs. "exports to Australia"
  - "dogs that chased cats" vs. "dogs chased by cats"
  - "design computers" vs. "design with computers"
  - "absence of evidence" vs. "evidence of absence"

Lexical Meaning

- Synonymy:
  - Two words have similar meaning
    - "dog" and "canine" are synonyms
- Hyponymy/hyperonymy:
  - One word refers to a subset/superset of another word's meaning
    - "dog" is a hyperonym of "poodle" (superset)
    - "poodle" is a hyponym of "dog" (subset)
- Homonymy and polysemy:
  - Homonymy: two different words have the same spelling
    - "bank": river bank or financial institution
  - Polysemy: one word has different senses
    - "canine": dog or tooth

Lexical Meaning



canine$_2$

polysemy

synonymy

dog

canine$_1$

hyponymy

hyponymy

poodle

Lexical Relations – Different Parts of Speech

- Nominalisations
  - *to walk in space – to do a spacewalk*
- Many actions can be expressed as nouns
  - *a play, a performance, a run*
- The thematic relation of an action may be indicated
  - *editor* – that who edits
  - *employee* – that who is employed

Expressions using Nominalisations

- One word
  - *walk in space – do a spacewalk*
- Compound nouns
  - *vi edits files – vi is a file editor*

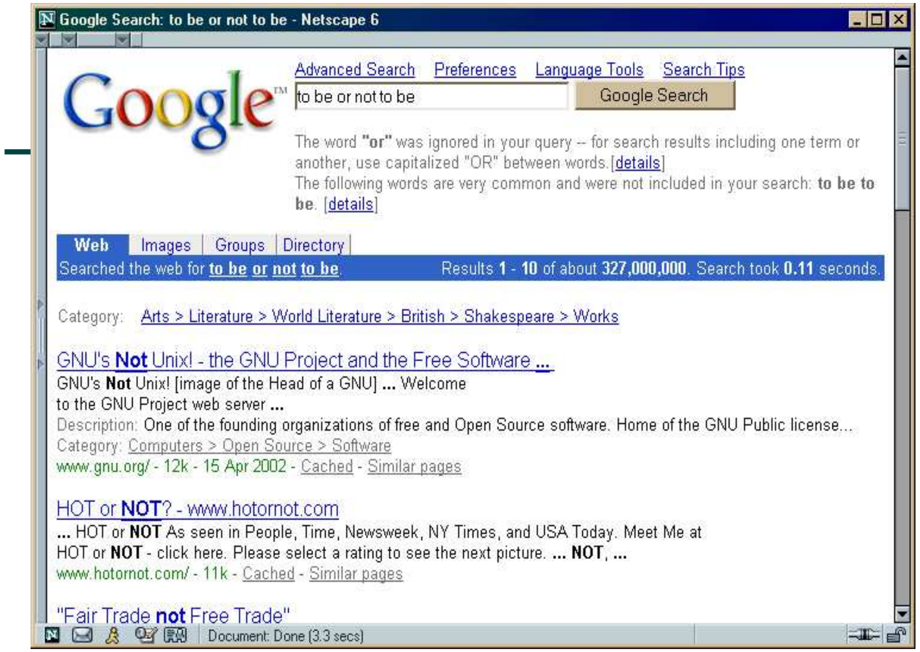
Effect of Lexical Meaning in IR

- Polysemy, homonymy
  - A query may retrieve a non-relevant document:
    - Precision decreases
    - Recall may decrease if there is a (small) fixed cut-off
- Hyponymy, hyperonymy, synonymy, nominalisations
  - A query may not retrieve a relevant document
    - Recall decreases
    - Precision may decrease if there is a (large) fixed cut-off

Term Selection and Creation

How to produce more accurate indexing terms?

- Stemming
  - Group several similar words into the same indexing term
  - Increase recall at the expense of precision
- Stop list
  - Discard closed-class words and other very frequent “noise” words
  - Increase recall at the expense of precision



Google Search: to be or not to be - Netscape 6

to be or not to be

The word **"or"** was ignored in your query -- for search results including one term or another, use capitalized "OR" between words. [details]
The following words are very common and were not included in your search: **to be to be**. [details]

Searched the web for **to be or not to be**. Results **1 - 10** of about **327,000,000**. Search took **0.11** seconds.

Category: Arts > Literature > World Literature > British > Shakespeare > Works

GNU's **Not** Unix! - the GNU Project and the Free Software ...
GNU's **Not** Unix! [image of the Head of a GNU] **...** Welcome
to the GNU Project web server **...**
Description: One of the founding organizations of free and Open Source software. Home of the GNU Public license...
Category: Computers > Open Source > Software
www.gnu.org/ - 12k - 15 Apr 2002 - Cached - Similar pages

HOT or **NOT**? - www.hotornot.com
... HOT or **NOT** As seen in People, Time, Newsweek, NY Times, and USA Today. Meet Me at
HOT or **NOT** - click here. Please select a rating to see the next picture. **... NOT, ...**
www.hotornot.com/ - 11k - Cached - Similar pages

"Fair Trade **not** Free Trade"

Document: Done (3.3 secs)

Word Sense Disambiguation

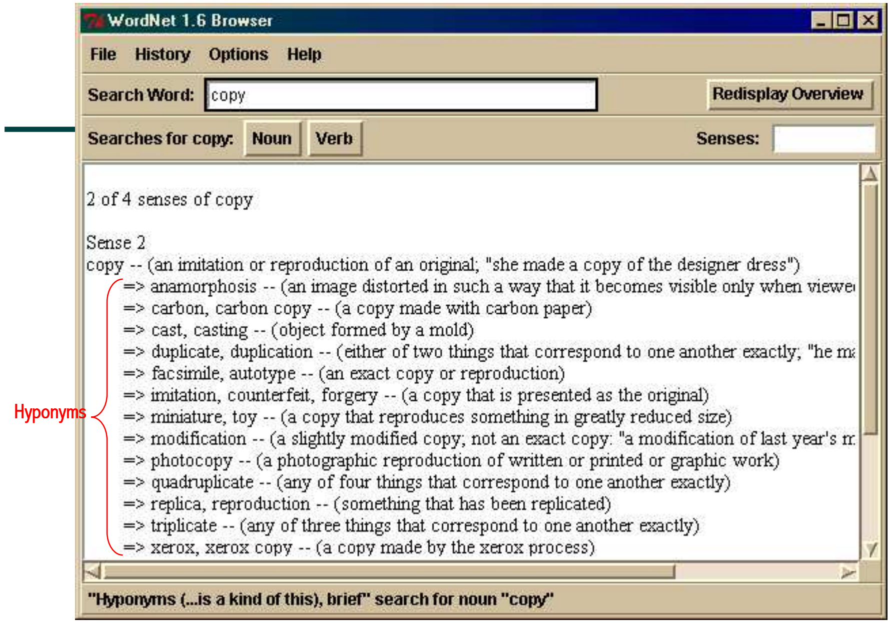
- Word sense disambiguation tries to solve the polysemy and homonymy problems
  - Try to increase precision without damaging recall
- Idea: Determine the right sense of the words in the query and the documents
  - “the financial advisor went to the bank” ← {financial, advisor, bank1}
- Mixed Results:
  - Precision does not improve much
    - Other terms in the query help to disambiguate the sense
  - Disambiguation errors can degrade performance
    - WSD accuracy of at least 90% is necessary

Query Expansion

- Morphological analysis
  - Add inflections and derivations
  - A more popular solution: stemming
- Lexical relations
  - Use a thesaurus
    - synonymy, hyponymy, etc.
  - Add related words
    - e.g. WordNet's gloss
- Term clustering: an alternative to query expansion
  - Group related words into one term
    - according to thesaurus
    - according to relations found statistically (e.g. Latent Semantic Indexing)
  - Stemming can be seen as a type of term clustering

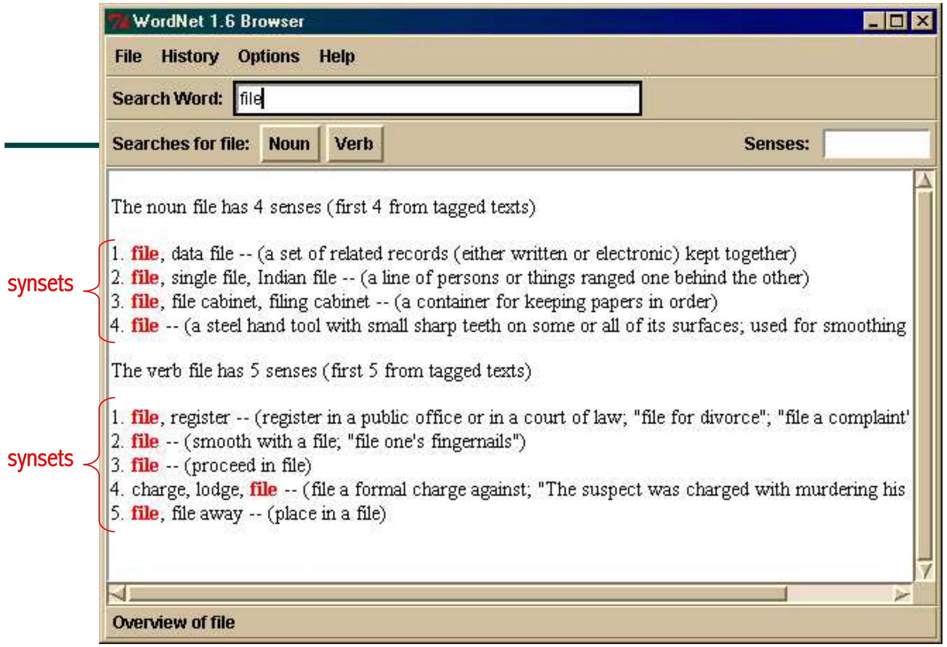
WordNet

- http://www.cogsci.princeton.edu/~wn/
- Developed by the Cognitive Science Laboratory at Princeton University
- Only open-class words are used
  - nouns, verbs, adjectives, adverbs
- Only lexical relations between words in the same PoS
  - Main lexical relations:
    - synonymy, hyponymy, meronymy, entailment
- Some QA systems use the gloss to derive loose lexical relations



```
2 of 4 senses of copy

Sense 2
copy -- (an imitation or reproduction of an original; "she made a copy of the designer dress")
    => anamorphosis -- (an image distorted in such a way that it becomes visible only when viewe
    => carbon, carbon copy -- (a copy made with carbon paper)
    => cast, casting -- (object formed by a mold)
    => duplicate, duplication -- (either of two things that correspond to one another exactly; "he m
    => facsimile, autotype -- (an exact copy or reproduction)
    => imitation, counterfeit, forgery -- (a copy that is presented as the original)
    => miniature, toy -- (a copy that reproduces something in greatly reduced size)
    => modification -- (a slightly modified copy; not an exact copy: "a modification of last year's m
    => photocopy -- (a photographic reproduction of written or printed or graphic work)
    => quadruplicate -- (any of four things that correspond to one another exactly)
    => replica, reproduction -- (something that has been replicated)
    => triplicate -- (any of three things that correspond to one another exactly)
    => xerox, xerox copy -- (a copy made by the xerox process)
```

"Hyponyms (...is a kind of this), brief" search for noun "copy"



```
The noun file has 4 senses (first 4 from tagged texts)

1. file, data file -- (a set of related records (either written or electronic) kept together)
2. file, single file, Indian file -- (a line of persons or things ranged one behind the other)
3. file, file cabinet, filing cabinet -- (a container for keeping papers in order)
4. file -- (a steel hand tool with small sharp teeth on some or all of its surfaces; used for smoothing

The verb file has 5 senses (first 5 from tagged texts)

1. file, register -- (register in a public office or in a court of law; "file for divorce"; "file a complaint'
2. file -- (smooth with a file; "file one's fingernails")
3. file -- (proceed in file)
4. charge, lodge, file -- (file a formal charge against; "The suspect was charged with murdering his
5. file, file away -- (place in a file)
```

Overview of file

Lexical Chainer – Example1

To determine if two words are related, FALCON computes the chain of lexical relations between the words

- Q1403: *When was the internal combustion engine invented?*
  - Answer: The first internal – combustion engine was built in 1867
  - Lexical chains:
    (1) invent:v#1 → HYPERNYM → create_by_mental_act:v#1 → HYPERNYM → create:v#1 → HYPONYM → build:v#1

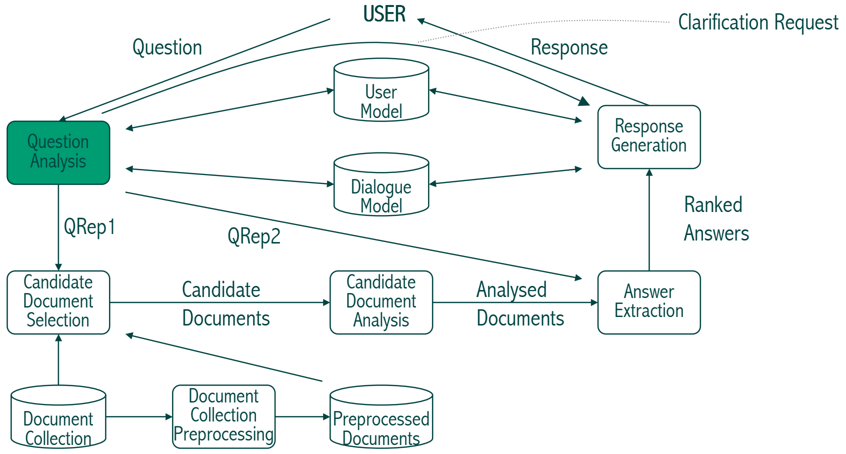
Lexical Chainer – Example2

- Q1518: *What year did Marco Polo travel to Asia?*
  - Answer: Marco Polo divulged the truth after returning in 1292 from his travels, which included several months in Sumatra
  - Lexical chains:
    (1) travel_to:v#1 → GLOSS → travel:v#1 → RGLOSS → travel:n#1
    (2) travel_to#1 → GLOSS → travel:v#1 → HYPONYM → return:v#1
    (3) Sumatra:n#1 → ISPART → Indonesia:n#1 → ISPART → Southeast_Asia:n#1 → ISPART → Asia:n#1

Today's Topics

- Finding Candidate Documents
- Question Analysis
- Answer Extraction
- Using the Web for QA
- ExtrAns
- The Future of QA

Architecture of a QA System



A Broad Classification of Questions

- *Yes/No questions*
  - The easiest type to process and evaluate
  - The least commonly occurring type of question
- *Factoid questions* ← Our focus
  - Who? what? when?
  - Easy to evaluate
  - The type of named entity is generally known
- *List questions*
  - Questions that ask to retrieve lists of facts

A Broad Classification of Questions

- *Definition questions*
  - The detail of the definition depends on the type of user
  - Harder to evaluate
- *Open-ended questions*
  - How? why? etc.
  - The answer may be scattered among several documents
  - Complex inferences required
  - The hardest type of questions to process and to analyse

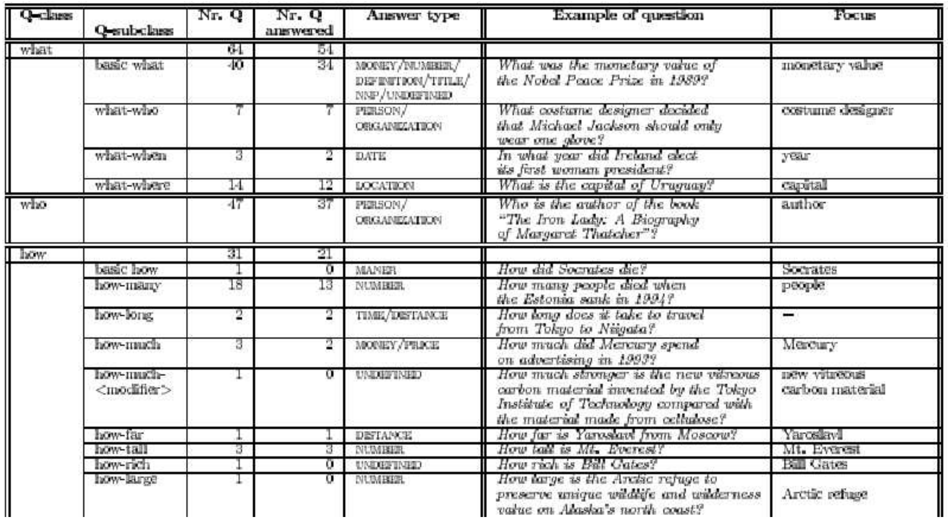
Analysing Factoid Questions

- For the Document Preselection module:
  - Relevant words (keywords) or phrases
- For the answer extraction module:
  - What *type of question*
  - What *type of answer* is expected
  - Additional information
    - The *question focus*
    - Syntactic or semantic information

The Focus

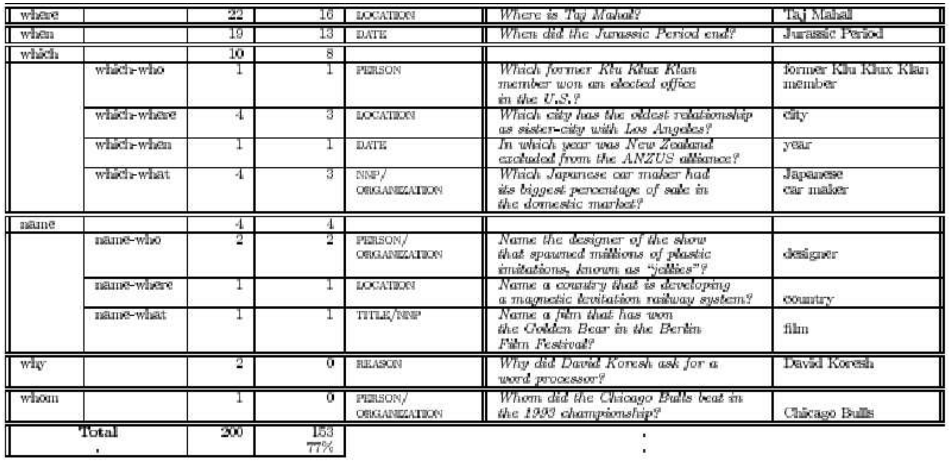
- A focus disambiguates the question
  - What is the question looking for?
  - What is the question about?
- Terms in the focus:
  - Help to determine the answer type
  - Typically don’t appear in the answer
    - Must remove them from the list of keywords
- Example
  - *What is the largest city in Germany?*
- The focus is typically the head of the noun phrase next to the question word

Question Types in Lasso



Q-class	Q-subclass	Nr. Q	Nr. Q answered	Answer type	Example of question	Focus
what		64	54			
	basic what	40	34	MONEY/NUMBER/ DEFINITION/TITLE/ NNP/UNDEFINED	*What was the monetary value of the Nobel Peace Prize in 1989?*	monetary value
	what-who	7	7	PERSON/ ORGANIZATION	*What costume designer decided that Michael Jackson should only wear one glove?*	costume designer
	what-when	3	2	DATE	*In what year did Ireland elect its first woman president?*	year
	what-where	14	12	LOCATION	*What is the capital of Uruguay?*	capital
who		47	37	PERSON/ ORGANIZATION	*Who is the author of the book "The Iron Lady: A Biography of Margaret Thatcher"?*	author
how		31	21			
	basic how	1	0	MANER	*How did Socrates die?*	Socrates
	how-many	18	13	NUMBER	*How many people died when the Estonia sank in 1994?*	people
	how-long	2	2	TIME/DISTANCE	*How long does it take to travel from Tokyo to Niigata?*	–
	how-much	3	2	MONEY/PRICE	*How much did Mercury spend on advertising in 1993?*	Mercury
	how-much-<modifier>	1	0	UNDEFINED	*How much stronger is the new vitreous carbon material invented by the Tokyo Institute of Technology compared with the material made from cellulose?*	new vitreous carbon material
	how-far	1	1	DISTANCE	*How far is Yaroslavl from Moscow?*	Yaroslavl
	how-tall	3	3	NUMBER	*How tall is Mt. Everest?*	Mt. Everest
	how-rich	1	0	UNDEFINED	*How rich is Bill Gates?*	Bill Gates
	how-large	1	0	NUMBER	*How large is the Arctic refuge to preserve unique wildlife and wilderness value on Alaska's north coast?*	Arctic refuge

Question Types in Lasso



where		22	16	LOCATION	*Where is Taj Mahal?*	Taj Mahal
when		19	13	DATE	*When did the Jurassic Period end?*	Jurassic Period
which		10	8			
	which-who	1	1	PERSON	*Which former Klu Klux Klan member won an elected office in the U.S.?*	former Klu Klux Klan member
	which-where	4	3	LOCATION	*Which city has the oldest relationship as sister-city with Los Angeles?*	city
	which-when	1	1	DATE	*In which year was New Zealand excluded from the ANZUS alliance?*	year
	which-what	4	3	NNP/ ORGANIZATION	*Which Japanese car maker had its biggest percentage of sale in the domestic market?*	Japanese car maker
name		4	4			
	name-who	2	2	PERSON/ ORGANIZATION	*Name the designer of the show that spawned millions of plastic imitations, known as "jellies"?*	designer
	name-where	1	1	LOCATION	*Name a country that is developing a magnetic levitation railway system?*	country
	name-what	1	1	TITLE/NNP	*Name a film that has won the Golden Bear in the Berlin Film Festival?*	film
why		2	0	REASON	*Why did David Koresh ask for a word processor?*	David Koresh
whom		1	0	PERSON/ ORGANIZATION	*Whom did the Chicago Bulls beat in the 1993 championship?*	Chicago Bulls
Total		200	153 77%			

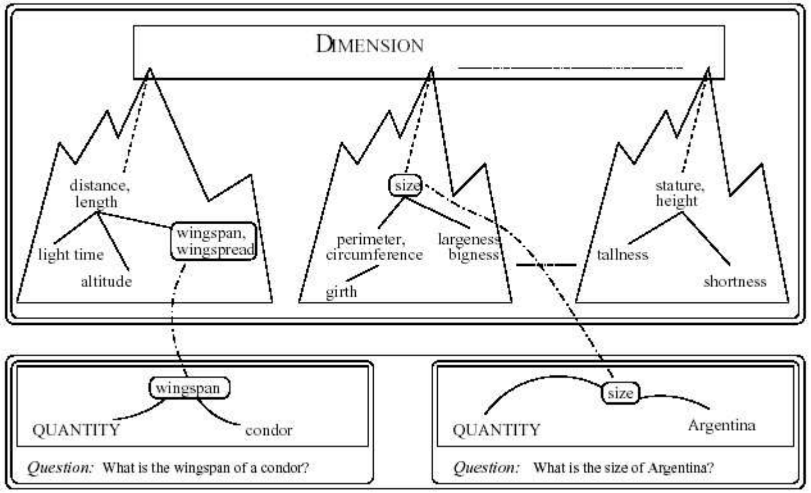
Approaches to Question Classification

- Rule-based Approaches
  - Simple patterns account for most of the questions
    - wh- words provide the strongest clue
    - Use regular expressions
- Statistical Approaches
  - Decision Lists
  - Maximum Entropy
  - . . .

Finding the Answer Type in Falcon

- The answer type is indicated by the question phrase most connected to other concepts
  1. Check the syntactic dependencies between words
     - The phrase with the most connections indicates the answer type
  2. Traverse WordNet to find a recognised answer type
- Example: *What do penguins eat?*
  - *food* is the most widely used concept in the glosses of the subhierarchy of the noun synset {*eating*,*feeding*}

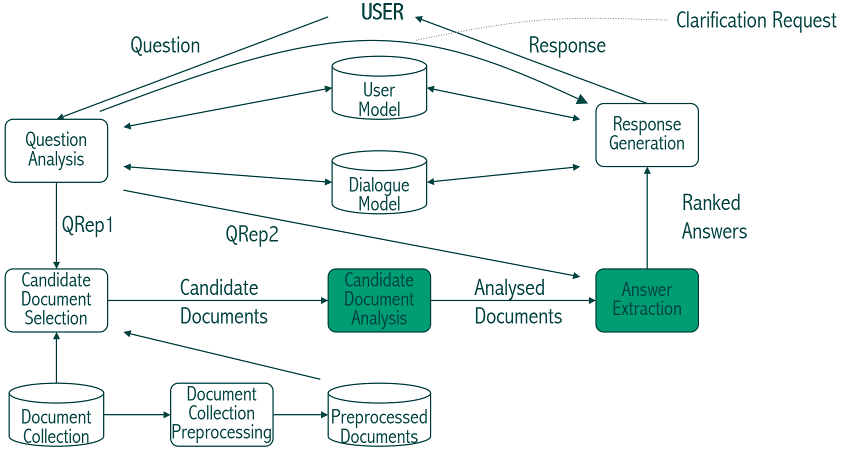
WordNet and the Answer Type



Today’s Topics

- Finding Candidate Documents
- Question Analysis
- Answer Extraction
- Using the Web for QA
- ExtrAns
- The Future of QA

Architecture of a QA System



USER

Clarification Request

Question

Response

User Model

Response Generation

Question Analysis

Dialogue Model

Ranked Answers

QRep1

QRep2

Candidate Document Selection

Candidate Documents

Candidate Document Analysis

Analysed Documents

Answer Extraction

Document Collection

Document Collection Preprocessing

Preprocessed Documents

Answer Extraction

- Now we know what documents (or paragraphs) may contain the answer
- How do we extract the answer?
- Typical approach
  1. Preselect answer candidate strings
     - check for type compatibility with the expected answer
  2. Return the answer candidate that is most likely to contain the answer
     - score the candidates
     - prove that the string in fact answers the question

Scoring Candidates in Lasso

- The score of an *answer window* is a combination of:

1. Same keywords in same sequence as in the question
2. The answer candidate is followed by punctuation
3. Number of words from the question following the answer candidate when the latter is succeeded by a comma
4. Number of words from the question found in the same parse sub-tree as the answer candidate
5. Number of words from the question found in the same sentence
6. Number of keywords in the answer-window
7. Sum of distances (in words) between the answer candidate and the other question words in the same window

Theorem-proving by Deduction

- The knowledge base consists of a set of axioms
- The questions are theorems to be proved
- Proving a theorem:
  - Deduce the answer to the question
- Example
  - *George is at home* introduces the axiom **AT(George,home)**
  - The question *Where is George?* is presented as the conjectured theorem **AT(George,x)**
  - The theorem can be proven if **x=home**

A More Complex Example

- Statement: *Smith is a man*
  MAN(Smith)
- Statement: *Man is an animal*
  $\forall x\ \mathtt{MAN(x)} \Rightarrow \mathtt{ANIMAL(x)}$
- Question: *Who is an animal?*
  $\exists y\ \mathtt{ANIMAL(y)}$
- Answer
  YES, y = Smith
  - **YES** means that the conjecture $\exists y\ \mathtt{ANIMAL(y)}$ has been proved
  - **y = Smith** indicates that “Smith” is an instance of y satisfying ANIMAL(y) – i.e., ANIMAL(Smith) is a theorem

Proving a Negation by Counterexample

- Statement: *A robot is a machine*
  $\forall x\ \mathtt{ROBOT(x)} \Rightarrow \mathtt{MACHINE(x)}$
- Statement: *Rob is a robot*
  ROBOT(Rob)
- Statement: *No machine is an animal*
  $\forall x\ \mathtt{MACHINE(x)} \Rightarrow \neg\mathtt{ANIMAL(x)}$
- Question: *Is everything an animal?*
  $\forall x\ \mathtt{ANIMAL(x)}$
- Answer
  NO, x = Rob

Introducing a Disjunction

- Statement: Either Smith is at work or Jones is at work
  $\texttt{AT(Smith,work)} \lor \texttt{AT(Jones,work)}$
- Question: Is anyone at work?
  $\exists\texttt{x AT(x,work)}$
- Answer
  $\texttt{YES, x = Smith or x = Jones}$

The Resolution Method

- Do some of the previous examples remind you of Prolog?
- Prolog’s answer procedure is based on resolution
- Method:
  - Insert the question clause negated
  - If a contradiction is found, then the question is answered
    - This is Prolog’s “yes” answer
  - If no contradiction is found, then there is insufficient information
    - This is Prolog’s “no” answer

Resolution Method – Example

- Statement: *Smith is a man*
  $\texttt{MAN(Smith)}$
- Statement: *Man is an animal*
  $\forall\texttt{x MAN(x)} \Rightarrow \texttt{ANIMAL(x)}$
- Question: *Who is an animal?*
  $\exists\texttt{y ANIMAL(y)}$
  translates as
  $\forall\texttt{y (}\neg\texttt{ANIMAL(y))}$

Resolution Method – Example (cont.)

We want to prove:

$\texttt{1. MAN(Smith)}$

$\texttt{2. }\forall\texttt{x MAN(x)} \Rightarrow \texttt{ANIMAL(x)}$

$\texttt{3. }\forall\texttt{y (}\neg\texttt{ANIMAL(y))}$

From 1. and 2. we can deduce:

$\texttt{4. ANIMAL(Smith)}$

From 3. we can deduce:

$\texttt{5. }\neg\texttt{ANIMAL(Smith)}$

There is a contradiction between 4. and 5. when y = Smith

Deduction – Recap

- Deduction process
  - We have a set of axioms
  - We have a theorem
  - We prove the theorem by consulting the axioms
- However
  - We may not have the complete information
  - e.g. Prolog's "no" does not mean that the question is wrong
    - Rather, there are no data in the database that can support the theorem
    - "Negation by failure"

Abduction

- A more realistic scenario:
  - Assume new axioms that will support the proof
  - Only direct contradictions disprove a theorem
- In other words
  - Increase recall of a question
  - Tolerant to incomplete knowledge bases

Abduction at Work (FALCON QA system)

- Question: *Who was the first Russian astronaut to walk in space?*
  first(x) ∧ astronaut(x) ∧ Russian(x) ∧ space(z) ∧ walk(y z x) ∧ HUMAN(x)
- Answer candidate: *The broad-shouldered but paunchy Leonov, who in 1965 became the first man to walk in space, signed autographs*
  paunchy(y) ∧ shouldered(e1 y x) ∧ broad(x) ∧ Leonov(x) ∧ first(z) ∧ man(z) ∧ space(t) ∧ walk(e2 t z) ∧ became(e3 z u x) … ∧ HUMAN(x)
- Assumption: *Leonov is a Russian astronaut*
  Leonov(x) ∧ Russian(x) ∧ astronaut(x)

What Information can We Abduce?

Axioms that can be abduced in FALCON:

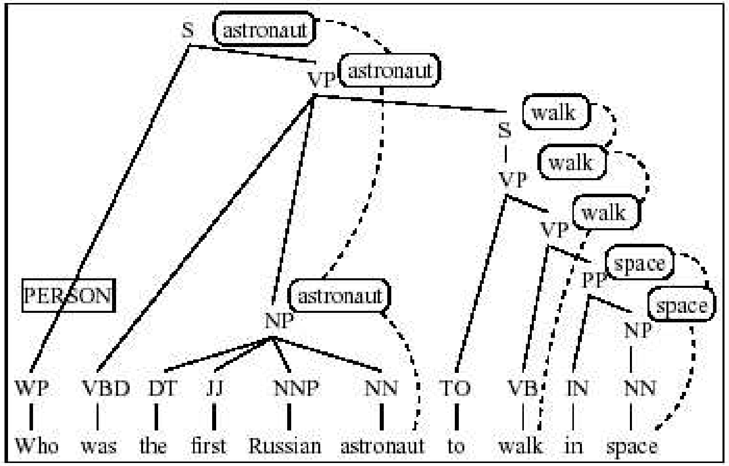
1. Axioms derived from the facts stated in the textual answer
2. Axioms representing world knowledge
   - WordNet
   - other external axioms
3. Axioms determined by coreference resolution in the answer text

Finding the Semantic Knowledge

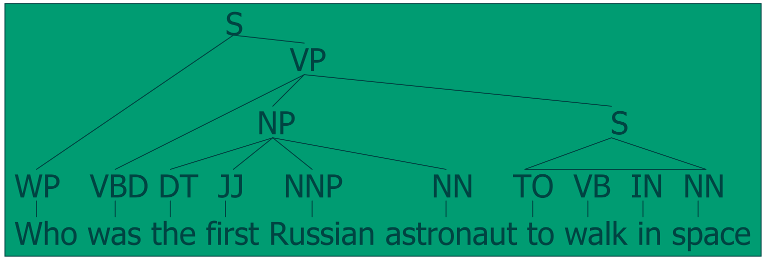
- FALCON: The semantics of the question can be approximated by deriving the dependencies between words

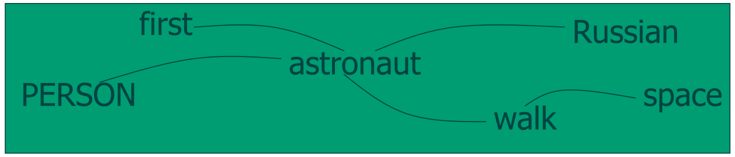
1. Find the syntactic parse
2. Extract the dependencies between the phrase heads by traversing the parse tree
3. Convert the dependencies into logical forms

Finding the Main Dependencies



Semantic Knowledge





first(x) ^ astronaut(x) ^ Russian(x) ^ space(z) ^ walk(y,z,x) ^ HUMAN(x)

Today’s Topics

- Finding Candidate Documents
- Question Analysis
- Answer Extraction
- Using the Web for QA
- ExtrAns
- The Future of QA

Web Based Question Answering

- What is Web Based QA?
  1. Search the Web to find the answer to a user question
  2. Use the Web as one more resource in a QA system

Is the Web Just Another Corpus?

- The Web is larger than any other corpus, and increasing in size
  - You may find on the Web almost anything you want to know about
- The links between documents can be exploited…
  - … to find related documents
  - … to determine the authority of a document
- The Web is written by scores of independent people with independent interests
  - Outdated information
  - Contradictory information
  - Unreliable information
  - Extensive information

Approaches to Web Based Question Answering

- Federated Approach
  - Portions of the Web are treated as if they were databases
  - Use of techniques for managing semistructured data
- Distributed Approach
  - Web data is viewed as unstructured text
  - Use knowledge mining techniques

Federated vs. Distributed Approaches

Different types of questions are best handled with different approaches:

- *What is the population of x?*
  - This question translates naturally into a database query
  - This is best handled with a federated approach
- *What format was VHS's main competition?*
  - Questions like this one are unique
  - The distributed approach provides a general purpose solution for handling such questions

The Federated Approach

- Pockets of structured knowledge exist on the Web as valuable resources for question answering
  - CIA World Factbook: political, geographic, and economic information about every country in the world
  - Biography.com: contains profiles of over twenty-thousand people
  - Internet Movie Database: information about hundreds of thousands of movies

Unifying Databases

- Different databases have different formats
- Many of these databases are “invisible” to search engine crawlers
- Portions of the Web can be viewed as a virtual, semistructured database
- How to handle these?
  - Slurp: Extract the information and populate a database
  - Wrap: Provide a programmatic interface to the resources

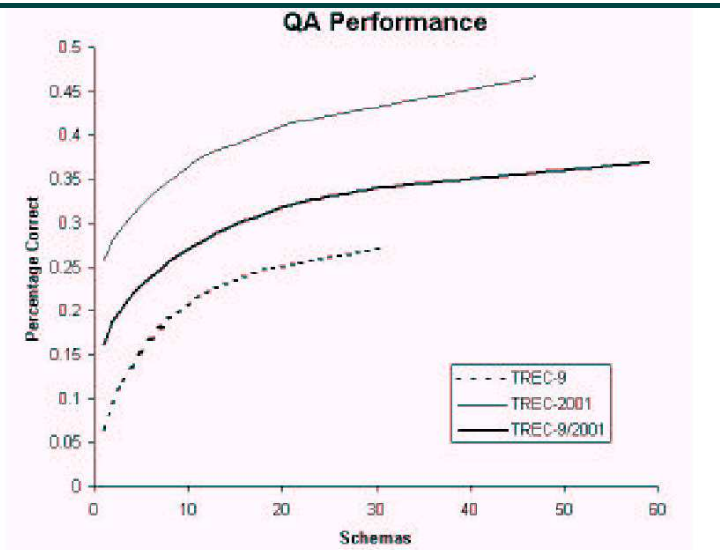
Slurping

- How do we extract the information?
  - Methodically query the Web database until all options are exhausted
    - Time-consuming
    - May be impossible (the method may generate circular queries)
  - Download the remote database and restructure it into our local database
    - The data may not be available
    - Permission to download the data may be difficult to obtain

Wrapping the Web

- What common interface can we use?
  - SQL – Why not?
  - The challenge: To convert a NL query into SQL
  - Approach: use schemas
- Zipf’s Law of QA:
  - A few question templates account for a large portion of all question instances

Zipf's Law for QA Performance



NL Questions as SQL Queries

- The schemas may be based on regular expressions or on logical form fragments
  - *What is the population of Taiwan?*
```
population(x) →
  SELECT population
  FROM CIA.countries
  WHERE country = x
```
  - *What is platinum?*
```
definition(x) →
  SELECT def
  FROM dictionary.defs
  WHERE word = x
```

Wrapping versus Slurping

Wrapping

👍

- The information is up-to-date
- Dynamic information is easy to access

👎

- Dependence on the Web database
  - limited functionality
  - reliability
  - wrapper maintenance

Slurping

👎

- The information is not up-to-date
- Resource limitations: can we store *all* the data?
- Practical issues: are we authorised to download the data in bulk?

👍

- High reliability

Challenges and Issues with Federation

- Challenges
  - Lack of explicit and uniform database schemas
  - Limited coverage of the system
- Other Issues
  - Possibility of creation of *virtual* databases

Lack of Database Schemas

- Typical approach: use site-specific wrappers
  - translate the local data into a form digestible by the integration system
- These wrappers can be time-consuming to produce
- Possible Solutions:
  - Use a well-designed authoring tool
    - Can reduce the amount of time required to integrate a knowledge source
  - Incorporate machine learning techniques to automate the wrapper generation process
    - wrapper induction
    - automatic wrapper adjustment (when the source changes its format)
  - Take advantage of the Semantic Web (if it ever happens!)

Coverage Issues

- Structured knowledge exists for domain-specific sites
- Broad coverage is more difficult to obtain

On the other hand:

- There is high certainty that coverage within a particular domain is complete

Virtual Databases

- The Web as a virtual database:
  - The knowledge is distributed around the Web and retrieved at query-time
- A federated system acts as a “knowledge broker”
  - like a librarian at the reference desk of a large research library

The Distributed Approach

- Most of the Web is composed of unstructured, textual documents, and most likely will remain so
- “Traditional” QA process:
  - Reduce the corpus to a smaller set of relevant documents
  - Attempt to “pinpoint” the exact location of the answer
- But:
  - The Web has a large degree of data redundancy
  - The quality of individual documents is rather low

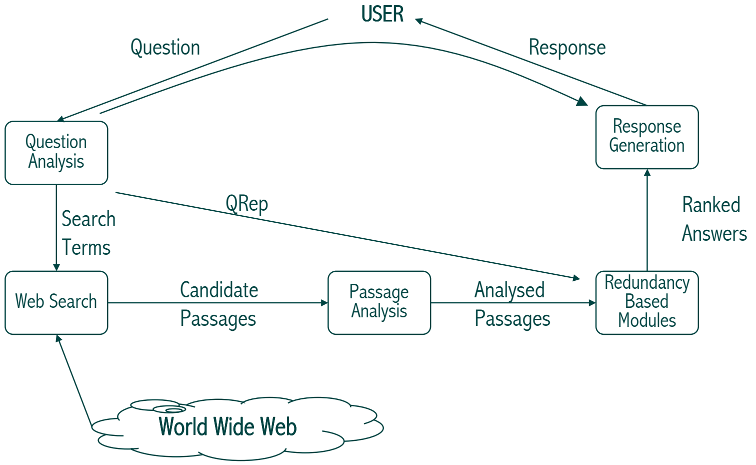
Capitalising on Data Redundancy

- As the size of the target document grows, the more likely it is that question answering systems can find statements that answer the question in an obvious way
- Simple pattern-matching techniques can replace the need to understand both the structure and meaning of language
- *Who killed Lincoln?*
  - *John Wilkes Booth killed Lincoln*
  - *John Wilkes Booth is perhaps America's most infamous assassin. He is best known for firing the bulled that ended Abraham Lincoln's life*

Redundancy as Answer Quality

- The average quality of individual documents is low
- Text extracted from a single document cannot be trusted as the correct answer
- BUT multiple occurrences of the same answer in different documents lends credibility to the proposed answer
  - Voting mechanisms

Distributed Approach – Generic Architecture



Redundancy Based Approaches

- Match answers using surface patterns
  - Use regular expressions
- Leverage statistics and multiple answer occurrences
  - Generate n-grams
  - Use voting mechanisms
  - May need to integrate document "authority"
    - The most frequent answer is not always the best answer
- Apply information extraction technology
  - Named entity recognition on the question and candidate passages

Challenges and Issues with Distribution

- Advantages
  - The generality of the solution
- Issues
  - How to index the entire Web?
    - Sol: Use search engines
  - The approach is not suitable to certain types of questions
    - Federated approach may work better in specific types of questions
  - It is very difficult to control answer quality tailored to different users

Answer Quality and Users

- The correctness of an answer depends on the type of user who asks the question
- e.g.: a definition
  - hypernym? dictionary entry? encyclopedia entry?
- A federated approach offers a nice solution:
  - use one of the available knowledge sources
  - but what knowledge source to choose if there are several available?

Challenges and Issues with Distribution

- Issues
  - How do we determine the patterns to match the answer of a question?
  - True natural language processing techniques may be required to achieve high levels of answer quality
    - How far can simple techniques be pushed?
  - Projecting the answer found in the Web on a separate corpus

How do We Determine the Patterns?

- Try all possible permutations?
  - e.g: Microsoft's approach in TREC 2001 QA

    What is relative humidity?
    - ["+is relative humidity", LEFT, 5]
    - ["relative +is humidity", RIGHT, 5]
    - ["relative humidity +is", RIGHT, 5]
    - ["relative humidity", NULL, 2]
    - ["relative" AND "humidity", NULL, 1]
- Integrate query expansion?
- Use machine learning techniques?

Answer Projection

- TREC required to return a document from a separate corpus that supports the answer
- People prefer paragraph-sized answers
  - the answers found in the Web may come with irrelevant context
- An Answer Projection Mechanism
  - Use document-retrieval or passage-retrieval algorithms
    - Including the answer in the search query

Web-based QA – Summary

- Federated approach:
  - Good for specific types of questions
- Distributed approach:
  - Good for other types of questions, and as a general mechanism
- These two approaches complement each other
  - Use the federated approach for the questions for which it works best, use the distributed approach for the other questions

Using the Web to Enhance a QA System

- The application
  - We want to do QA on a specific corpus
    - e.g. the question-answering track of TREC
  - How can we use the Web?
- Approaches
  - Extract a set of answer candidates from the corpus and then use the Web to validate the answers or refine their scores
    - e.g. count hits of a Web search with the question and the answer candidate
  - Use the Web to find the answer candidates and use the candidate that has the best support in our corpus

Today's Topics

- Finding Candidate Documents
- Question Analysis
- Answer Extraction
- Using the Web for QA
- ExtrAns
- The Future of QA

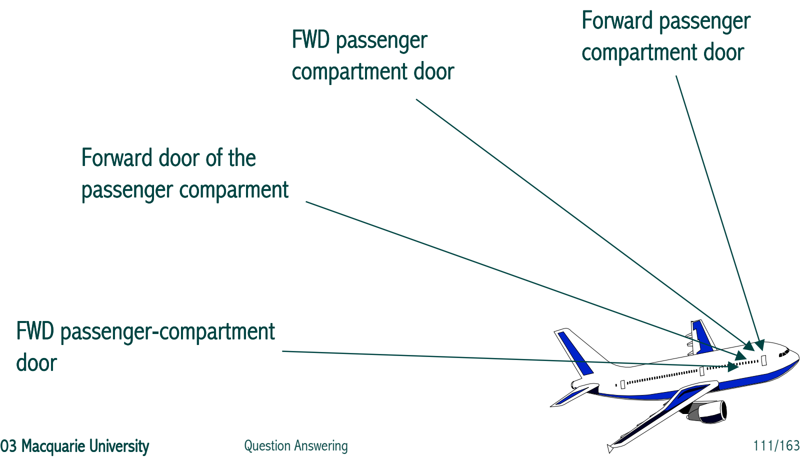
Technical Domains

- Large documents, but still much smaller than the TREC datasets (or the web!)
  - [~100MB << GB << TB]
- Cannot make use of redundancy
- Have to cope with specific formats and sublanguage
- Terminology is a key problem
- Cannot use Web as a "last-resort" resource
  - Problems of coverage

Terminology: The Problems

- The Parsing Problem
  - Multi-word terms "confuse" the parser
    - Complex internal structure
    - Possible combinations with external elements
  - Domain specific lexical items are NOT included in generic lexica (as used by the parser)
- The Paraphrase Problem
  - The query could contain a variant of a term used in the domain (possibly completely new)
    - *How are the cargo compartment doors opened?*
    - *To open the doors of the compartment cargo use …*

Synonymous Terms



FWD passenger compartment door

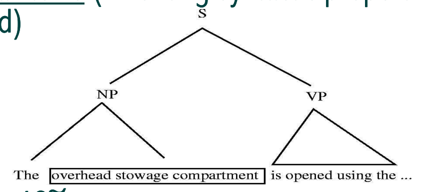
Forward passenger compartment door

Forward door of the passenger comparment

FWD passenger-compartment door

Terminology: Solution

- Identify all terms in a preprocessing phase and collect them in a domain specific thesaurus
  - Encode information about synonyms and hyponyms
- Recognize terms while analyzing the documents and treat them as single lexical items (inheriting syntactic properties from their head word)

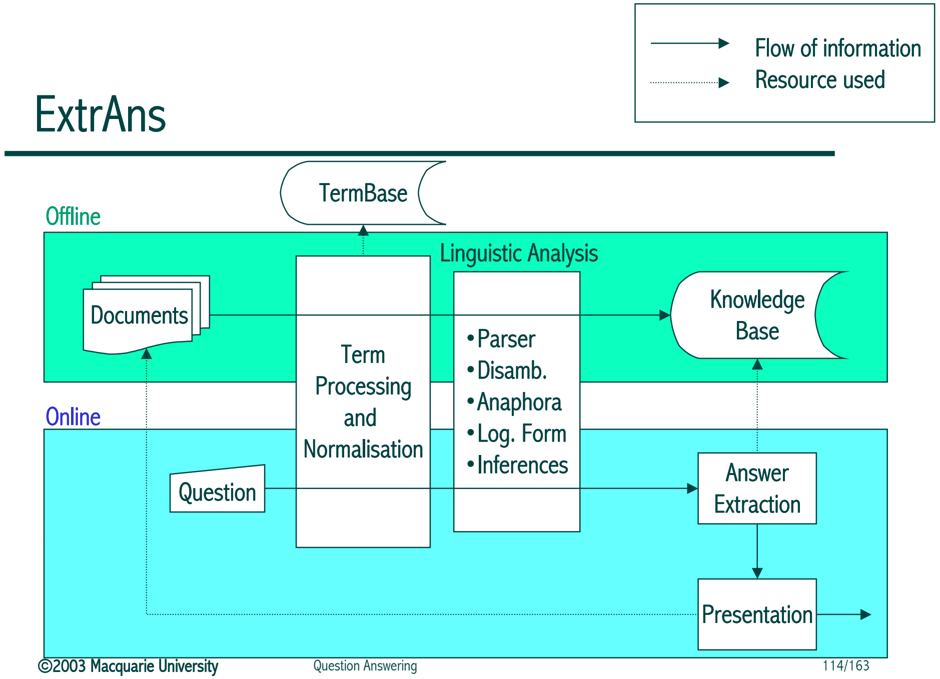


- Parsing simplified by 46%

ExtrAns

- A Question Answering system targeted at technical domains
- Convert document and queries in a semantic representation (documents are processed off-line)
  - [From "*bag of words*" to "*bag of semantic relations*"]
- Match (sem rep of) queries against documents
- Return the matched answers in the context where they originally appear

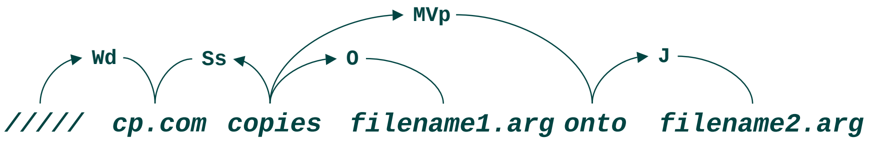
ExtrAns



Link Grammar

- Link Grammar is written in the spirit of dependency grammars and consists of
  - a very fast parser,
  - a grammar/dictionary with about 60,000 word forms.
- LG parser returns dependency relations between pairs of words.
- By default, the direction of the dependency is not given.
- For the construction of the minimal logical forms (MLF) the direction of the dependency is important (see next slide).

Link Grammar



- The link **Wd** connects the subject **cp.com** to the wall **/////**.
- The link **Ss** connects the transitive verb with the subject.
- The transitive verb and its object are connected by the link **O**.
- The link **MVp** connects the verb to its modifying prepositional phrase.
- The link **J** connects the preposition **onto** to its object.

Link Grammar

- LG's coverage: 76% full parses for 2,781 test sentences from Unix manual pages.
- LG parser allows robust parsing by ignoring words until a valid dependency structure is found.
- Ignored words are not lost – they result in keywords.
- LG handles unknown words by making guesses from the context about syntactic categories.
- Nevertheless, the result is always better when the words have been categorized in advance (ExtrAns: 650 new domain-specific words).

Minimal Logical Forms

A first approximation:

cp copies long files.

```
object(cp,[x1])
evt(copy,[x1,x2])
object(file,[x2])
prop(long,[x2])
```

Minimal Logical Forms

Observation:

– in Unix man pages we find sentences like

cp refuses to copy a file onto itself.

Therefore:

– reification of events (refusing, copying)

– encode the existence of the refusing event

– but not the existence of the copying event.

Reification of Events

Reification of events:

cp refuses to copy a file onto itself.

```
object(cp,[x1])
evt(refuse,e1,[x1,e2])
evt(copy,e2,[x1,x2])
object(file,[x2])
prop(onto,[e2,x2])
```

Reification of Events

Reification to encode the (actual) existence of concepts:

cp refuses to copy a file onto itself.

```
hold(e1)
object(cp,[x1])
evt(refuse,e1,[x1,e2])
evt(copy,e2,[x1,x2])
object(file,[x2])
prop(onto,[e2,x2])
```

Reification of Adjectives

Reification at work:

cp copies very long files.

```
holds(e1)
object(cp,[x1])
evt(copy,e1,[x1,x2])
object(file,[x2])
prop(long,p1,[x2])
prop(very,p2,[p1])
```

Adjective modifies the object.

Reification of Objects

Reification at work:

cp copies possible files.

```
holds(e1)
object(cp,o1,[x1])
evt(copy,e1,[x1,x2])
object(file,o2,[x2])
prop(possible,p1,[o2])
```

Adjective modifies the concept.

Reification at Work

Reification at work:

cp copies files quickly.

```
holds(e1)
object(cp,o1,[x1])
evt(copy,e1,[x1,x2])
object(file,o2,[x2])
prop(quickly,p1,[e1])
```

Reification at Work

- Logical operators are translated as predicates over reified concepts:

  cp does not copy a file onto itself.

```
not(op1,e1)
object(cp,o1,[x1])
evt(copy,e1,[x1,x2])
object(file,o2,[x2])
prop(onto,e1,[x2])
```

Reification at Work

- Another example:

  If the user types y, then cp copies the files.

```
if(op2,[e1],[e2])
...
evt(type,e1,[x1,x2])
...
evt(copy,e2,[x3,x4])
```

Nominal Compounds

- Nominal compounds result in underspecified representations:

  computer design systems

```
object(computer,o1,[x1])
object(design,o2,[x2])
object(system,o3,[x3])
nominal_compound(i1,[o1,o2,o3])
```

- If the term processor identifies the term then only one predicate is produced

```
object(computer_design_system,o1,[x1])
```

  – And how do we know that we are talking about a type of system…? We'll see later

Minimal Logical Forms – Summary

- Minimal logical forms (MLFs) consist of
  - flat existentially closed conjunctions of atomic formulae
  - with reified event, object and property concepts and
  - a particular interpretation of existence and logical operators.
- MLFs are designed to encode the part of the semantic information that we need for the AE task (i.e. underspecification).
- MLFs are incrementally extensible – further refinements can be added without destroying old information.

Term Variations Producing Strict Synonymy

- Orthographic
  - *cargo compartment door*
  - *Cargo-compartment door(s)*
- Morpho-Syntactic
  - *Cargo compartment door*
  - *doors of the cargo compartment*

Three Weaker Synonymy Relations

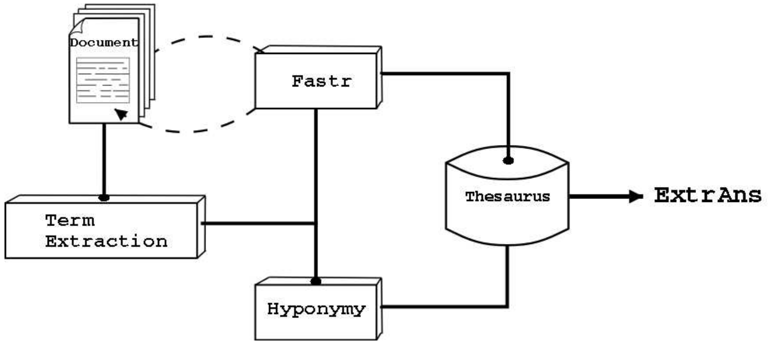
- Head
  - *electrical cable*
  - *electrical line*
- Modifier
  - *attachment strip*
  - *fastener strip*
- Head and Modifier
  - *functional test*
  - *operational check*

Lexical/Terminological Paraphrases

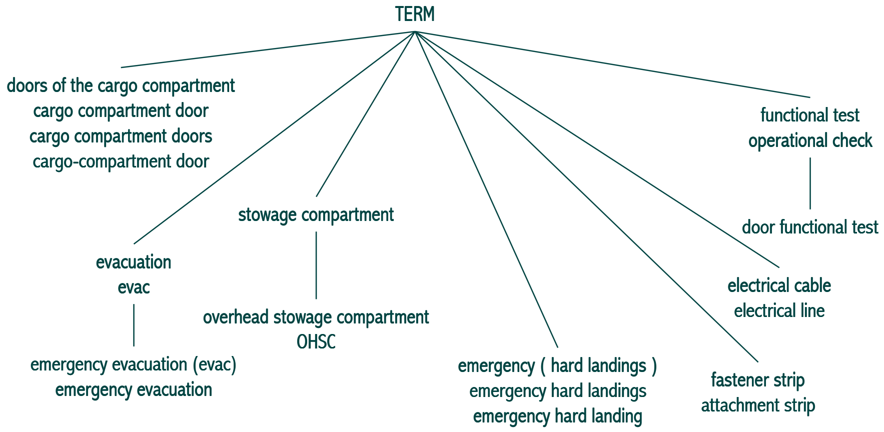
- Thesauri
  - Use of a WordNet-like thesaurus
    - Synonymy and synsets
    - Hyponymy
  - Term normalisation
- Process:

1. Identify all terms
2. Remove punctuation and check for acronyms
3. Identify synonyms (Fastr)
   - apply metarules based on part of speech, morphology, and WordNet synonyms
4. Identify hyponyms

Building the TermBase



A Fragment of the TermBase



TERM

doors of the cargo compartment
cargo compartment door
cargo compartment doors
cargo-compartment door

stowage compartment

overhead stowage compartment
OHSC

evacuation
evac

emergency evacuation (evac)
emergency evacuation

emergency (hard landings)
emergency hard landings
emergency hard landing

functional test
operational check

door functional test

electrical cable
electrical line

fastener strip
attachment strip

Synonymy and Hyponymy in the LFs

- Synonymy as synsets
- Hyponymy as disjunctions in the logical form
  - *Where are the stowage compartments installed?*

```
– object(s_stowage_compartment,A,[B]),
  evt(install,C,[D,B]), object(E,F,[D|G])
–  (object(s_stowage_compartment,A,[B]) ;
  object(s_overhead_stowage_compartment,A,[B])),
  evt(install,C,[D,B]), object(E,F,[D|G])
```

Additional Information in the Logical Forms

If the sentence with identifier s1 is *cp copies very long files*, then:

```
holds(e1/s1)/s1~[1,2,3,4,5]
object(cp,o1/s1,[x1/s1])/s1~[1]
object(s_command,o2/s1,[x1/s1])/s1~[1]
evt(s_copy,e1/s1,[x1/s1,x2/s1])/s1~[1,2,3,4,5]
object(s_file,o3/s1,[x2/s1])/s1~[3,4,5]
prop(long,p1/s1,[x2/s1])/s1~[4]
prop(very,p2/s1,[p1/s1])/s1~[3]
```

What is the additional information?

```
evt(s_copy,e1/s1,[x1/s1,x2/s1])/s1~[1,2,3,4,5]
```

- A thesaurus (WordNet) lets us find the synset to which “copy” belongs
- A sentence identifier let us find the sentence in the original manual page
- A list of tokens to highlight will help the user find the exact phrase in the sentence

- Domain knowledge about manpages and the way they are formatted lets us know that cp is a command. Therefore we add the predicate:

```
object(s_command,o2/s1,[x1/s1])/s1~[1]
```

Questions

- The question

  Which command copies files?

  translates into the following Prolog query:

```
?- findall( [S,P1,P2,P3], (
      evt(s_copy,A,[B,C])/S~P1,
      object(s_file,D,[C])/S~P2,
      object(s_command,E,[B])/S~P3 ),
    Results ).
```

Questions

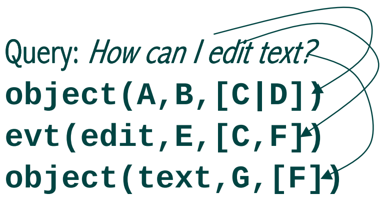
- Questions such as
  - *Which command can copy files?*
  - *Which command copies files?*
  - *Which command can copy a file?*
  - *Which command can copy all my files?*

  should retrieve …

Answers

- … the following <u>informative</u> answers:
  - *cp copies files.*
  - *cp does not copy a file onto itself.*
  - *cp refuses to copy a file onto itself.*
  - *if the user types y, then cp copies the files.*
- But not:
  - *XYZ files a copy.*

Nominalization



- Query: *How can I edit text?*

```
object(A,B,[C|D])
evt(edit,E,[C,F])
object(text,G,[F])
```

- Data: *vi is a (display-oriented) text editor …*

```
object(vi,o1,[x1])
evt(edit,e1,[x1,x2])
object(editor,o2,[x1,x2])
object(text,o3,[x2])
```

Fall-back Strategy

1. First, try with synonyms
   - use the synset information
2. If not enough answers, try with hyponyms
   - hyponyms add disjunctions to the query
3. If not enough answers, try with approximate matching
   - compute the overlap between logical forms
4. If everything fails, try a keyword approach
   - an answer guaranteed

Approximate Matching

- Based on overlap of logical forms
  - *A man named Richard Sears has been playing a joke on shoppers.*
    holds(o10), object('man',o2,[x2]), evt('name',e3,[x3,x2,x5]),
    evt(‘play_on',o10,[x2,x9,x12]), object('richard',o4,[x4]),
    object('sear',o5,[x5]), object('shopper',o12,[x12])
  - *Who played a joke on shoppers?*
    holds(e2), object(WHO,o1,[x1]), evt('play_on',e2,[x1,x4,x6]),
    object('joke',v_o4,[v_x4]), object('shopper',v_o6,[v_x6])
- Further research:
  - give weights to the predicates and the matching process
  - convert this into a process of abduction

Ambiguities

Observations:

- many ambiguities can be resolved (on account of lexical or syntactic information and some domain knowledge),
- some will survive,
- humans are good at resolving them.

Ambiguities

Cases of ambiguities:

- A document sentence may be ambiguous,
- A document sentence can have multiple (not exclusive) interpretations (e.g. enumerations),
- A logical form may provide multiple answers because different sets of facts can answer a query (e.g. coordination),
- The user question can be ambiguous.

Surviving Ambiguities

Therefore:

- assert logical forms of all surviving ambiguities,
- find all proofs for a question,
- the more often a term was used in a proof the more pertinent the term is,
- express pertinence by selective highlighting the retrieved phrases in the context of the document.

Selective Highlighting – How it Works . . .

- Translation of

  rm, rmdir – remove files or directories.

  results in the assertion of

```
evt(s_remove,e1/P1,[x3/P1,x6/P1])/P1~[1,4,5,6,7],
...
evt(s_remove,e1/P2,[x3/P2,x6/P2])/P2~[2,4,5,6,7],
...
```

- Both interpretations are stored in the knowledge base as independent MLFs.
- They can lead to more than one correct proof.

Selective Highlighting – How it Works . . .

- The proof of the question

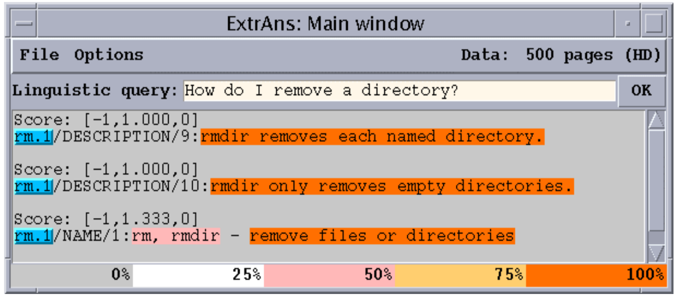
  How do I remove a directory?

  extracts the underlined information

```
rm, rmdir - remove files or directories
rm, rmdir - remove files or directories
```

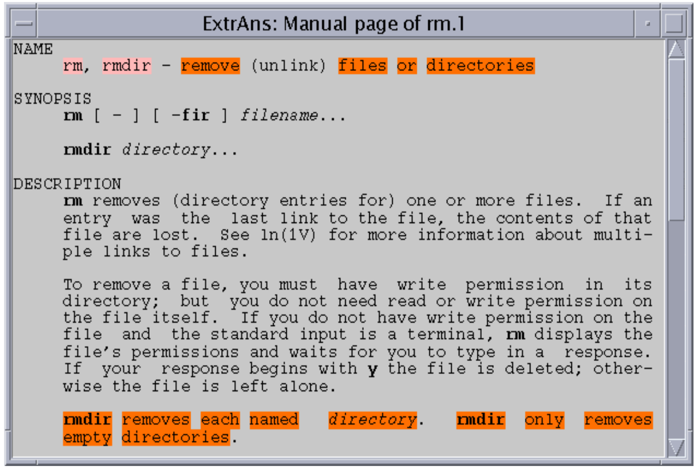
- In the first case *rm* is used during the proof, in the second case *rmdir* is used.
- As a result *rm* and *rmdir* get 50% retrieval relevance and the rest of the sentence 100%.
- This result is converted into a graded colouring scheme.

Selective Highlighting: What it looks like . . .



```
Data: 500 pages (HD)
Linguistic query: How do I remove a directory?
Score: [-1,1.000,0]
rm.1/DESCRIPTION/9:rmdir removes each named directory.

Score: [-1,1.000,0]
rm.1/DESCRIPTION/10:rmdir only removes empty directories.

Score: [-1,1.333,0]
rm.1/NAME/1:rm, rmdir - remove files or directories
```

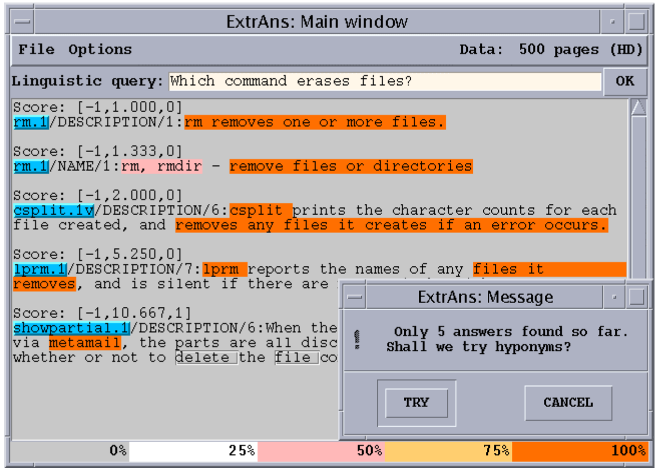
Selective Highlighting: Answers in Context



ExtrAns: Manual page of rm.1

```
NAME
     rm, rmdir - remove (unlink) files or directories

SYNOPSIS
     rm [ - ] [ -fir ] filename...

     rmdir directory...

DESCRIPTION
     rm removes (directory entries for) one or more files.  If an
     entry  was  the  last link to the file, the contents of that
     file are lost.  See ln(1V) for more information about multi-
     ple links to files.

     To remove a file, you must  have  write  permission  in  its
     directory;  but  you do not need read or write permission on
     the file itself.  If you do not have write permission on the
     file  and  the standard input is a terminal, rm displays the
     file's permissions and waits for you to type in a  response.
     If  your  response begins with y the file is deleted; other-
     wise the file is left alone.

     rmdir removes each named  directory.   rmdir  only  removes
     empty directories.
```

Fall-back Strategy



ExtrAns: Main window

Data: 500 pages (HD)

Linguistic query: Which command erases files?

```
Score: [-1,1.000,0]
rm.1/DESCRIPTION/1:rm removes one or more files.

Score: [-1,1.333,0]
rm.1/NAME/1:rm, rmdir - remove files or directories

Score: [-1,2.000,0]
csplit.1v/DESCRIPTION/6:csplit prints the character counts for each
file created, and removes any files it creates if an error occurs.

Score: [-1,5.250,0]
lprm.1/DESCRIPTION/7:lprm reports the names of any files it
removes, and is silent if there are

Score: [-1,10.667,1]
showpartial.1/DESCRIPTION/6:When the
via metamail, the parts are all disc
whether or not to delete the file co
```

ExtrAns: Message

Only 5 answers found so far.
Shall we try hyponyms?

Current Research

- AnswerFinder combines ExtrAns technology with TREC QA technology
  - http://www.comp.mq.edu.au/~diego/answerfinder/
  - try the demo of ExtrAns and give us feedback so that we can improve the system
- Current research
  - A revision of the MLF notation
  - The analysis of nominalization and other linguistic phenomena
  - The processing of larger documents over varied domains
  - The integration of document retrieval techniques for the preselection of documents
  - The use of abduction methods
  - The integration of external resources and the Web

Today's Topics

- Finding Candidate Documents
- Question Analysis
- Answer Extraction
- Using the Web for QA
- ExtrAns
- The Future of QA

The Future of QA

- Question Answering track in TREC
  - by DARPA, ARDA and NIST
  - QA on large open-domain sets of documents
- QA Roadmapping
  - by DARPA, ARDA and NIST
  - Determines specific issues to be addressed in QA research
  - Sets a tight schedule to handle these issues
  - TREC QA (partially) implements the QA roadmapping
- AQUAINT
  - Advanced Question Answering for Intelligence
  - By ARDA
  - QA from heterogeneous data sources and multiple languages

Roadmapping: TREC-10

- Requirements
  1. Find exact answer without extraneous information
  2. The answer is scattered across two or more documents
  3. The answer is not guaranteed to be present in the text collection

Roadmapping: TREC-10

- Challenges
  1. Fusion of answers from different documents into a single answer
  2. Detection of overlapping or contradictory information
  3. Time stamping
  4. Event tracking
  5. Recognition of questions with no answers in the collection

Roadmapping: TREC-11

- New Requirements
  1. The Q/A process will take place “within a context”
  2. A first step towards a dialogue Q/A environment

Roadmapping: TREC-11

- Challenges
  1. The comprehension of the context
  2. Update of the context
  3. The common ground

Roadmapping: TREC-12

- New Requirements
  1. The answer will be the product of text generation
  2. The generation may comprise an explanation of possible ambiguities and a justification of the answer

Roadmapping: TREC-12

- Challenges
  1. The quantification of the “naturalness” of the generated answer
     - When does the answer require an explanation or a justification?
  2. The recognition and explanation of ambiguities
  3. The generation of answer justifications in natural language

Roadmapping: TREC-13

- New Requirements
  1. More complex questions will be asked, requiring the answers to be *summaries:*
     - Context-based summary-generating questions
     - Stand-alone summary-generating questions
     - Example-based summary-generating questions

Roadmapping: TREC-13

- Challenges

1. The interaction between the context, the question and the context-based summary
2. The interaction between the example's context, the question and the quality of the example-based summary
3. Measuring the "informativeness" of stand-alone summaries

Roadmapping: TREC-14

- New Requirements

1. The questions will be asked at expert-level

How likely is that the Fed will raise the interest rates at their next meeting?

Data regarding the past decisions of the Fed, given certain values of inflation, stock market performance, employment data and other major political factors are used to make a prediction of the Fed's expected actions. The answer should also formulate comparisons to previous situations and the Fed's action on the interest rates.

Roadmapping: TREC-14

- Challenges

1. The heterogeneity of domain data
2. Techniques for mining information on the fly and at expert-level
3. The evaluation of the expertise displayed in the answer