CS474 Intro to Natural Language Processing

Question Answering

Question answering

- Overview and task definition
- History
- Open-domain question answering
- Basic system architecture
- **Predictive indexing methods**
  - Slides based on those of Jamie Callan, CMU
- Pattern-matching methods
- Advanced techniques

Indexing with predictive annotation

- **Some answers belong to well-defined semantic classes**
  - People, places, monetary amounts, telephone numbers, addresses, organizations
- **Predictive annotation: index a document with “concepts” or “features” that are expected to be useful in (many) queries**
  - E.g. people names, location names, addresses, etc.
- **Add additional operators for use in queries**
  - E.g. Where does Ellen Vorhees work? “Ellen Vorhees” NEAR/10 *organization

Predictive annotation

In the early part of this century, the only means of transportation for travelers and mail between **<LOCATION> Europe </LOCATION>** and **<LOCATION> North America </LOCATION>** was by passenger steamship. By <DATE> 1907 </DATE>, the **<COMPANY> Cunard Steamship Company </COMPANY>** introduced the largest and fastest steamers in the **<LOCATION> North Atlantic </LOCATION>** service: the **<NAME> Lusitania </NAME>** and the **<NAME> Mauritania </NAME>**. Each had a gross tonnage of **<WEIGHT> 31,000 tons </WEIGHT>** and a maximum speed of **<SPEED> 26 knots </SPEED>**.

– From K. Felkins, H.P. Leighly, Jr., and A. Jankovic. “The Royal Mail Ship Titanic: Did a Metallurgical Failure Cause a Night to Remember?” *JOM*, 50 (1), 1998, pp. 12-18.

Predictive annotation

- **How is annotated text stored in the index?**

In the early part of this century, the only means of transportation for travelers and mail between <$LOCATION, Europe> and <$LOCATION North> <$LOCATION America> was by passenger steamship. By <$DATE 1907>, the <$COMPANY, Cunard> <$COMPANY, Steamship> <$COMPANY, Company> introduced the largest and fastest steamers in the <$LOCATION, North> <$LOCATION, Atlantic> service: the <$NAME, Lusitania> and the <$NAME, Mauritania>. Each had a gross tonnage of <$WEIGHT, 31,000> <$WEIGHT, tons> and a maximum speed of <$SPEED, 26> <$SPEED, knots>.

- **Treat <$QA-token, term> as meaning that $QA-token and term occur at the same location in the text**
  – Or use phrase indexing approach to index as a single item

Issues for predictive annotation

- **What makes a good QA-token?**
  – Question that would use the token
    - Can be recognized with high reliability (high precision)
    - Occurs frequently enough to be worth the effort
- **How do you want the system to make use of the QA-tokens?**
  – Filtering step?
  – Transform original question into an ad-hoc retrieval question that incorporates QA-tokens and proximity operators?
- **Common approaches to recognizing QA-tokens**
  – Tables, lists, dictionaries
  – Heuristics
  – Hidden Markov models

Advantages and disadvantages

\+ **Most of the computational cost occurs during indexing**
  - Allows use of more sophisticated methods

\+ **Annotator has access to complete text of document**
  - Important for recognizing some types of features

– **Must know ahead of time which types of features/concepts are likely to be important**

– **Increases size of index considerably**
  - E.g. by an order of magnitude if many features

- **Used (in varying amounts) by almost all open-domain Q/A systems**

Question answering

- Overview and task definition
- History
- Open-domain question answering
- Basic system architecture
- Predictive indexing methods
- **Pattern-matching methods**
  – Slides based on those of Jamie Callan, CMU
- Advanced techniques

Simple pattern-based QA

- **Observation: there are many questions…but fewer <u>types</u> of questions**
- **Each type of question can be associated with**
  - Expectations about answer string characteristics
  - Strategies for retrieving documents that might have answers
  - Rules for identifying answer strings in documents

Example

- **Who is the President of Cornell?**
  - Expectation: answer string contains a person name
    - Named entity identification
  - Search query: "president Cornell *PersonName"
  - Rule: "*PersonName, President of Cornell"
    - Matches "…Hunter Rawlings, President of Cornell"
    - Answer = "Hunter Rawlings"

Question analysis

- **Input: the question**
- **Output**
  - Search query
  - Answer expectations
  - Extraction strategy
- **Requires**
  - Identifying named entities
  - Categorizing the question
  - Matching question parts to templates
- **Method: pattern-matching**
  - Analysis patterns created manually these days…

Question analysis example

- **"Who is Elvis?"**
  - Question type: "who"
  - Named-entity tagging: "Who is <person-name>Elvis</person-name>"
  - Analysis pattern: if question type = "who" and question contains <person-name> then
    - Search query doesn't need to contain a *PersonName operator
    - Desired answer probably is a description
    - Likely answer extraction patterns
      - "Elvis, the X"
        - » "…Elvis, the <u>king of rock and roll</u>…"
      - "the X Elvis"
        - » "the <u>legendary entertainer</u> Elvis"

Question analysis

Frequency of question types on an Internet search engine

- 42% what
- 21% where
- 20% who
- 8% when
- 8% why
- 2% which
- 0% how

Relative difficulty of question types

- What is difficult
  - What time…
  - What country…
- Where is easy
- Who is easy
- When is easy
- Why is hard
- Which is hard
- How is hard

Example: What is Jupiter?

1. **What We Will Learn from Galileo**
2. **The Nature of Things: Jupiter's shockwaves—How a comet's bombardment has sparked activity on Earth**
3. **Jupiter-Bound Spacecraft Visits Earth on 6-Year Journey**
4. **STAR OF THE MAGI THEORIES ECLIPSED?**
5. **Marketing & Media: Hearst, Burda to Scrap New Astrology Magazine**
6. **Greece, Italy Conflict On Cause Of Ship Crash That Kills 2, Injures 54**
7. **Interplanetary Spacecraft To `Visit` Earth With LaserGraphic**
8. **A List of Events During NASA's Galileo Mission to Jupiter**
9. **SHUTTLE ALOFT, SENDS GALILEO ON 6-YEAR VOYAGE TO JUPITER**
10. **Rebuilt Galileo Probe readied For Long Voyage To Jupiter**

Answer extraction

- **Select highly ranked sentences from highly ranked documents**
- **Perform named-entity tagging (or extract from index) and perform part of speech tagging**
  - "The/DT planet/NN <location>Jupiter/NNP</location> and/CC its/PRP moons/NNS are/VBP in/IN effect/NN a/DT mini-solar/JJ system/NN ,/, and/CC <location>Jupiter/NNP</location> itself/PRP is/VBZ often/RB called/VBN a/DT star/NN that/IN never/RB caught/VBN fire/NN ./."
- **Apply extraction patterns**
  - the/DT X Y, Y=Jupiter -> the planet Jupiter -> "planet"

Simple pattern-based Q/A: assessment

- **Extremely effective when**
  - Question patterns are predictable
    - Fairly "few" patterns cover the most likely questions
      - Could be several hundred
    - Not much variation in vocabulary
      - Simple word matching works
    - The corpus is huge (e.g., Web)
      - Odds of finding an answer document that matches the vocabulary and answer extraction rule improves
- **Somewhat labor intensive**
  - Patterns are created and tested manually

Common problem: matching questions to answers

- **Document word order isn't exactly what was expected**
- **Solution: "soft matching" of answer patterns to document text**
  - Approach: use distance-based answer selection when no rule matches
    - E.g. for "What is Hunter Rawlings' address?"
      - Use the address nearest to the words "Hunter Rawlings"
      - User the address in the same sentence as "Hunter Rawlings"

Common problem: matching questions to answers

- **Answer vocabulary doesn't exactly match question vocabulary**
- **Solution: bridge the vocabulary mismatch**
  - Approach: use WordNet to identify simple relationships
    - "astronaut" is a type of "person"
    - "astronaut" and "cosmonaut" are synonyms

Common problem: improving the set of retrieved documents

- **Sometimes the IR system can't find any documents that have answers (even though the right documents are in the corpus)**
- **Solution: get a broader set of documents**
  - Approach: if answer extractor fails to find an answer, kick the question back to the search engine with instructions to widen the search
    - Assumes answer extractors can tell when they fail
  - Approach: use a variety of retrieval strategies to retrieve documents
    - E.g., all words within one sentence, then all words within one paragraph, then within same document, …
    - E.g. add synonyms to query or do query expansion
    - Simple, but much higher computational expense

Common problem: improving answer extraction patterns

- **Word sequence patterns have limited power**
- **Solution: create patterns that use syntactic information**
  - Partial syntactic parsing of documents
    - Is this noun the subject or the object of the sentence?
  - Allows more complex patterns
    - Question: "Who shot Kennedy?"
    - "Who" implies a person that should be subject of answer sentence/clause
    - "Kennedy" should be direct object of answer
    - Pattern: <subject> shot Kennedy
    - Matching text: "Oswald shot Kennedy"

Common problem: selecting/ranking the answer

- **Multiple answer candidates**
- **Solutions**
  - – Features used to represent answer candidates
    - Frequency
    - Distance to question words
    - Location in answer passage(s)
    - …
  - – Selection functions
    - Created manually
    - Learned from training data

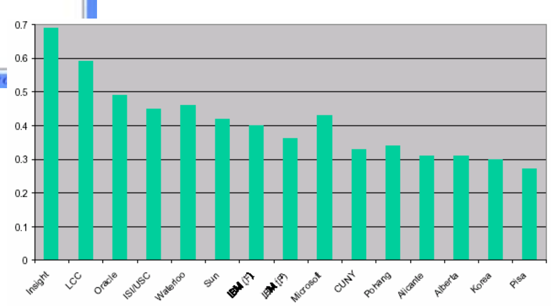
Question answering

- Overview and task definition
- History
- Open-domain question answering
- Basic system architecture
- Predictive indexing methods
- Pattern-matching methods
- Advanced techniques
  - – Slides based on Pasca & Harabagiu (SIGIR 2001) and the slides of J. Callan and K. Czuba, CMU

SMU/LCC system

TREC 2000

TREC 2001

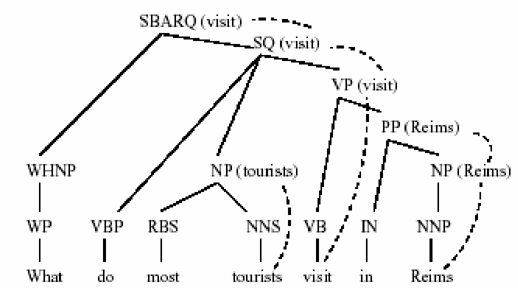


Multi-strategy approach

- **State of the art in QA is the SMU/LCC Falcon system**
  - – Employs informed use of standard IR techniques
  - – Use of broad ontology (extended WordNet)
  - – Lots of NLP
  - – Answer verification
- **Similar to most other systems in architecture except for**
  - – Much more careful tuning of algorithms and resources
  - – More sophisticated control of IR and NLP
  - – Feedback loops

Question analysis

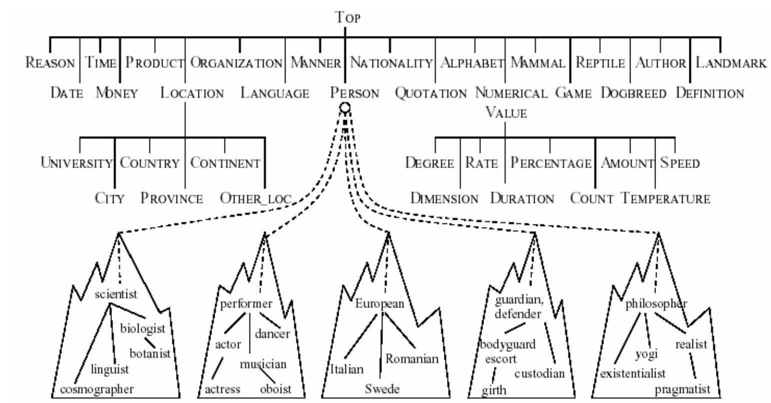
- **Parsing and named entity recognition**
- **Expected answer type determined by parsing**



- **Exceptions for "special cases"**

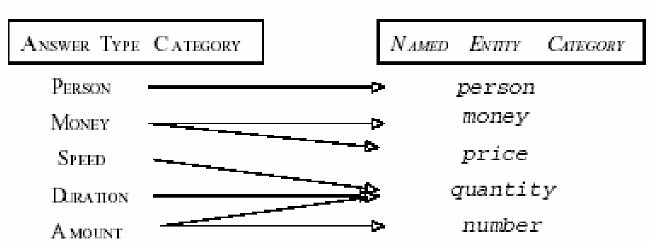
(Q-P1): *What {is|are} <phrase_to_define>?*
(Q-P2): *What is the definition of <phrase_to_define>?*
(Q-P3): *Who {is|was|are|were} < person_name(s)>?*

Expected answer types



Expected answer types

- **Answer types are mapped to named-entity categories that can be recognized in text**

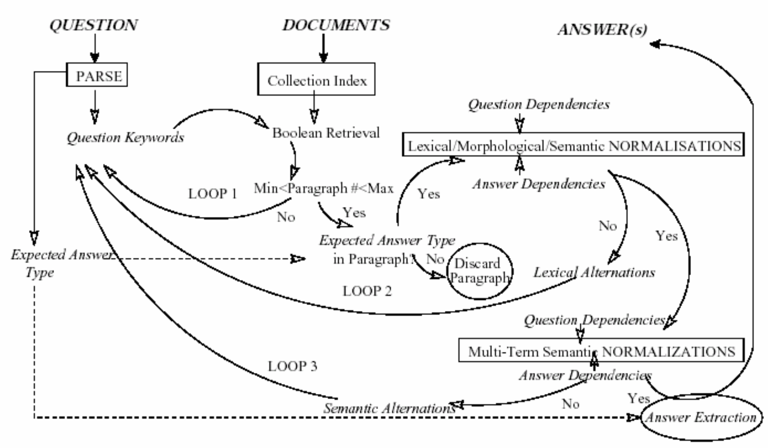


- **Answer types drive processing of paragraphs**
  - Passages need to contain the expected answer type

Paragraph retrieval

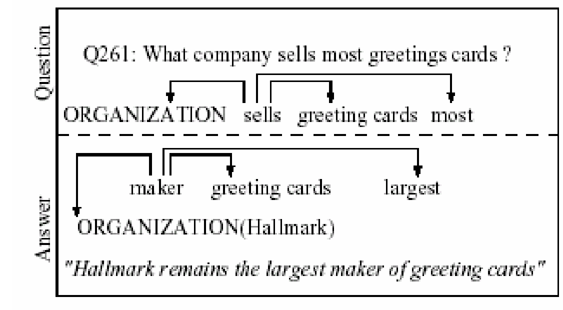
- **Boolean retrieval with loops**
  - Different from multiple queries in that system only uses additional queries when necessary
  - Fewer candidates for analysis components to consider
- **Loop 1: query keyword loop**
  - Keywords added/dropped to make query more/less specific
- **Loop 2: keyword alternations**
  - Try morphological variants and synonyms
- **Loop 3: semantic alternations**
  - Try semantic alternatives

Feedback loops



Answer verification

- **Parse passages to create a dependency tree among words**
- **Attempt to unify logical forms of question and answer text**



Assessment

- **Strengths**
  - Controlled use of IR system
    - Query expansion via lexical and semantic equivalents
    - Believed to be the major power of the system
  - Tailored resources (see paper)
    - WordNet, parser, NE identifier, etc.
  - Answer verification
    - Initially thought to be the key component of the system
    - Now…not so clear
- **Weaknesses**
  - Complex system, contribution of each component unclear