Question Answering: Overview of Tasks and Approaches

Horacio Saggion
Department of Computer Science
University of Sheffield
England, United Kingdom
http://www.dcs.shef.ac.uk/~saggion

Outline

- QA Task
- QA in TREC
- QA Architecture
- Collection Indexing
- Question Analysis
- Document Retrieval
- Answer Extraction
- Linguistic Analysis
- Pattern-based Extraction
- N-gram based approach
- Evaluation
- Finding Definitions

QA Task (Burger&al'02)

- Given a question in natural language and a given text collection (or data base)
- Find the answer to the question in the collection (or data base)
- A collection can be a fixed set of documents or the Web
- Different from Information or Document retrieval which provides lists of documents matching specific queries or users' information needs

QA Task (Voorhees'99)

- In the Text Retrieval Conferences (TREC) Question Answering evaluation, 3 types of questions are identified
- Factoid questions such as:
  - "Who is Tom Cruise married to?"
- List questions such as:
  - "What countries have atomic bombs?"
- Definition questions such as:
  - "Who is Aaron Copland?" or "What is aspirin?"

(Changed name to "other" question type)

QA Task

- A collection of documents is given to the participants
  - AP newswire (1998-2000), New York Times newswire (1998-2000), Xinhua News Agency (English portion, 1996-2000)
  - Approximately 1,033,000 documents and 3 gigabytes of text

QA Task

- In addition to answer the question systems have to provide a "justification" for the answer, e.g., a document where the answer occurs and which gives the possibility of fact checking
  - Who is Tom Cruise married to?
  - Nicole Kidman

  …Batman star George Clooney and Tom Cruise's wife Nicole Kidman …

QA Examples

- Q1984: How far is it from Earth to Mars?

<DOC DOCNO="APW19980923.1395"> After five more months of aerobraking each orbit should take less than two hours. Mars is currently 213 million miles (343 million kilometers) from Earth.</DOC>

<DOC DOCNO="NYT19990923.0365">

its farthest point in orbit, it is 249 million miles from Earth. And, so far as anyone knows, there isn't a McDonalds restaurant on the place. And yet we keep trying to get there. Thirty times in the past 40 years, man has sent a spacecra

</DOC>

- Correct answer is given by patterns:

(190|249|416|440)(\s|\-)million(\s|\-)miles?

QA Task

- Question can be stated in a "context-free" environment
  - "Who was Aaron Copland?"
  - "When was the South Pole reached for the first time?"
- Question may depend on previous question or answer
  - "What was Aaron Copland first ballet?"
  - "When was its premiere?"
  - "When was the South Pole reached?"
  - "Who was in charge of the expedition?"

TREC/QA 2004 question example

```
<target id = "3" text = "Hale Bopp comet">
  <qa>
    <q id = "3.1" type="FACTOID">
      When was the comet discovered?
    </q>
  </qa>
  <qa>
    <q id = "3.2" type="FACTOID">
      How often does it approach the earth?
    </q>
  </qa>
  <qa>
    <q id = "3.3" type="LIST">
      In what countries was the comet visible on its last return?
    </q>
  </qa>
  <qa>
    <q id = "3.4" type="OTHER">
      Other
    </q>
  </qa>
</target>
```

QA Challenge

- Language variability (paraphrase)
  - Who is the President of Argentina?
  - Kirchner is the President of Argentina
  - The President of Argentina, N. Kirchner
  - N. Kirchner, the Argentinean President
  - The presidents of Argentina, N. Kirchner and Brazil, I.L da Silva…
  - Kirchner is elected President of Argentina…
- Note: the answer has to be supported by the collection, not by the current state of the world…

QA Challenge

- How to locate the information given the question keywords
  - there is a gap between the wording of the question and the answer in the document collection
- Because QA is open domain it is unlikely that a system will have all necessary resources pre-computed to locate answers
  - should we have encyclopaedic knowledge in the system? all bird names, all capital cities, all drug names…
  - current systems exploit web redundancy in order to find answers, so vocabulary variation is not an issue…because of redundancy it is possible that one of the variations will exist on the Web…but what occurs in domains where information is unique…

QA Challenge

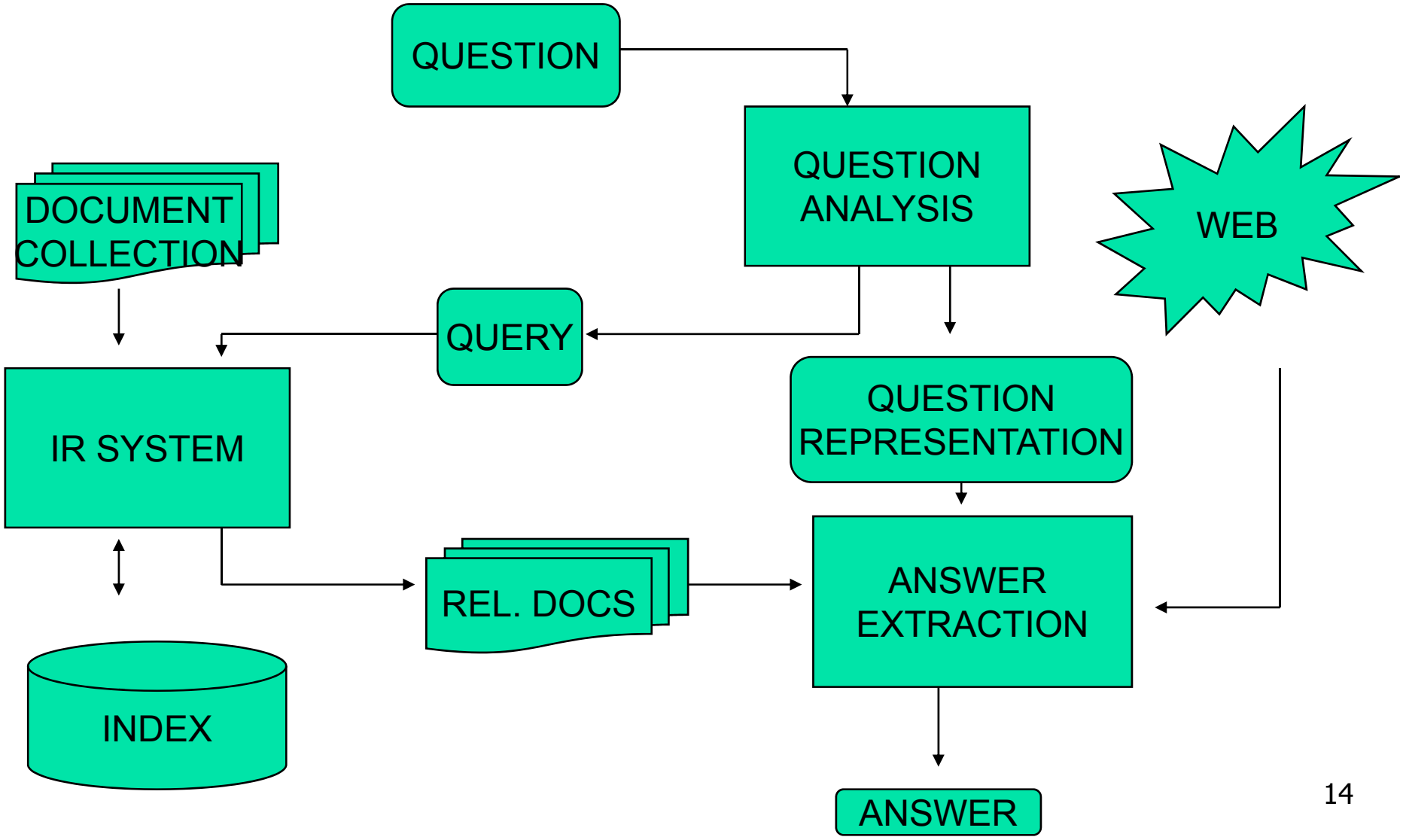
- Sometimes the task requires some deduction or extra linguistic knowledge:
  - What was the most powerful earthquake to hit Turkey?
  1. Find all earthquakes in Turkey
  2. Find intensity for each of those
  3. Pick up the one with higher intensity

  (some text-based QA systems will find the answer because it is explicitly expressed in text: "The most powerful earthquake in the history of Turkey….")

How to attack the problem?

- Given a question, we could go document by document verifying if it contains the answer
- However, a more practical approach is to have the collection pre-indexed (so we know what terms belong to which document) and use a query to find a set of documents matching the question terms
- This set of matching documents is (depending on the system) further ranked to produce a list where the top document is the most likely to match the question terms
- The document ranking is generally used to inform answer extraction components

QA Architecture



Collection Indexing

- Index full documents, paragraphs, sentences, etc.
- Index the collection using the words of the document – possibly ignoring stop words
- Index using stems – using an stemmer process
  - heroin ~ heroine
- Index using word lemmas - using morphological analysis
  - heroin <> heronie
- Index using additional information: syntactic/semantic information
  - named entities, named entity types
  - triples: X-lsubj-Y; X-lobj-Y; etc.

Question Analysis

- Two types of analysis are required
  - First, the question needs to be transformed in a query to the document retrieval system
    - each IR system has its own query language so we need to perform this mapping
    - identify useful keywords; identify type of answer sought, etc.
  - Second, the question needs to be analysed in order to create features to be used during answer extraction
    - identify keywords to be matched in document sentences; identify answer type to match answer candidates and select a list of useful patterns from a pattern repository
    - identify question relations which may be used for sentence analysis, etc.

Answer Type Identification

- What is the expected type of entity?
- One may assume a fixed inventory of possible answer types such as: person, location, date, measurement, etc.
- There may be however types we didn't think about before seen the questions: drugs, atoms, birds, flowers, colors, etc. So it is unlikely that a fixed set of answer types would cover open domain QA

Pattern Based Approach (Greenwood'04)

- Devise a number of regular patterns or sequence of filters to detect the most likely answer type
  - question starts with "who"
  - question starts with "how far"
  - question contains word "born"...
  - question does not contain the word "how"

Learning Approach

- We may have an inventory of questions and expected answer types and so we can train a classifier
  - features for the classifier may include the words of the question or the lemmas question; relevant verb (born) or semantic information (named entity)
- We can use a question retrieval approach (Li&Roth'02)
  - index the <question,qtypes> in a training corpus
  - retrieve set of n <question,qtypes> given a new question
  - decide based on the majority of qtypes returned the qtype of the new question

Linguistic Analysis of Question

- The type of the answer may be extracted from a process of full syntactic parsing (QALaSIE - Gaizauskas&al'04)
  - Question grammar required (in our case implemented in Prolog – attribute value context free grammar)
  - How far from Denver to Aspen?

    name(e2,'Denver') location(e2) city(e2) name(e3,'Aspen') qvar(e1) qattr(e1,count) qattr(e1,unit) measure(e1) measure_type(e1,distance)

    2 QA rules used to obtain this:

    Q -> HOWADJP(How far) VPCORE(be) PPS(it) IN(from) NP TO(to) NP

    HOWADJP1a: HOWADJP -> WRB(how) JJ(far|wide|near|close|…|huge)

    (these are not the actual rules in Prolog, but pseudo rules)

Linguistic Analysis of Question

- What is the temperature of the sun's surface?
  - qvar(e1) lsubj(e2,e1) be(e2), temperature(e1) sun(e4) of(e3,e4) surface(e3) of(e1,e3)
  - Some relations are computed: of(X,Y) and lsubj(X,Y) which might be relevant for scoring answer hypothesis
- More of this latter

Question Analysis

- If collection indexed with stems, then stem the question, if with lemmas, then lemmatise the question, …
  - if a document containing "heroine" has been indexed with term "heroin", then we have to use "heroin" to retrieve it
  - if a document containing "laid" has been indexed with lemma "lay", then we have to use "lay" to retrieve the document
- Question transformation when words are used in the index: Boolean case
  - "What lays blue eggs?"
  - non-stop-words: lays, blue, eggs
  - stems: lay, blue, egg
  - morphs (all verbs forms, all nominal forms): lay, lays, laid, laying; blue; egg, eggs

Question Analysis

- In Boolean retrieval queries are composed of terms combined with operators 'and' 'or' and 'negation'
  - lays AND blue AND eggs (may return very few documents)
  - lay AND blue AND egg (if index contains stemmed forms, query may return more documents because 'eggs' and 'egg' are both mapped into 'egg')
  - (lay OR lays OR laid OR laying) AND blue AND (egg OR eggs)
- Other more sophisticated strategies are possible:
  - one may consider to expand word forms with synonyms: film will be expanded with film OR movie
  - one may need to disambiguate each word first
  - nouns and derived adjectives (Argentina ~ Argentinean) can also be used
  - the type of the question might be used for expansion. Looking for a measurement? then, look for documents containing "inches", "metres", "kilometres", etc.

Iterative Retrieval

- Sometimes it is necessary to carry out an iterative process because not enough documents/passages have been returned
  - initial query: lay AND blue AND egg (too restrictive)
  - modified queries: lay AND blue; lay AND egg; blue AND egg… but which one to chose
  1. delete from query a term with higher document frequency (less informative)
  2. delete from query a term with lowest document frequency (most informative) – we found this to help more

Iterative Retrieval

- One may consider the status of information in the question
  - "What college did Magic Johnson attend?"
  - One should expect "Magic Johnson" to be a more relevant term than any other in the question ("Magic Johnson went to…", "Magic Johnson studied at…"). So, common words might be discarded from the query before than proper nouns in an iterative process.

Getting the Answer

- Question/answer text word overlap
  - Retrieve candidate answer bearing docs using IR system
  - Slide a window (e.g. 250 bytes) over the docs
  - Select the window with the highest word overlap with question

Getting the Answer

- Semantic tagging + semantic or grammatical relational constraints
  - Analyse question to identify semantic type of answer (*who* → person)
  - Retrieve candidate answer texts and semantically tag
  - Window + score based on question/window word overlap + presence of correct answer type
  - Optionally, parse + derive semantic/grammatical constraints to further inform the scoring/matching process

Getting the Answer

- Learning answer patterns (Soubbotin&Soubbotin'01; Ravichandran&Hovy'02)
  - From training data derive question-answer sentence pairs
  - Induce (e.g. regular expression) patterns to extract answers for specific question types

Answer Extraction

- Given question Q and documents Ds
- Analyse the question marking all named entities and identify the class of the answer (ET)
- Analyse documents in Ds and retain sentences containing entities identified in Q
- Extract all entities of type ET (but are not in Q)
- Cluster entities and return the most frequent one

Answer Extraction

- "Who is Tom Cruise married to?"
  - Tom Cruise is married to Nicole Kidman
  - Demi Moore and Tom Cruise's wife Nicole Kidman went to…
  - Claire Dickens, Tom Cruise, and wife Nicole attended a party.
- 3 answer candidates equivalent to "Nicole Kidman"; it is our best guess

An Example

Q: *How high is Everest?*

A: *Everest's 29,035 feet is 5.4 miles above sea level…*

Semantic Type:

```
If Q contains 'how' and 'high' then the
semantic class, S, is measurement:distance
```

Known Entities:

```
location('Everest'),
measurement:distance('29,035 feet')
measurement:distance('5.4 miles')
```

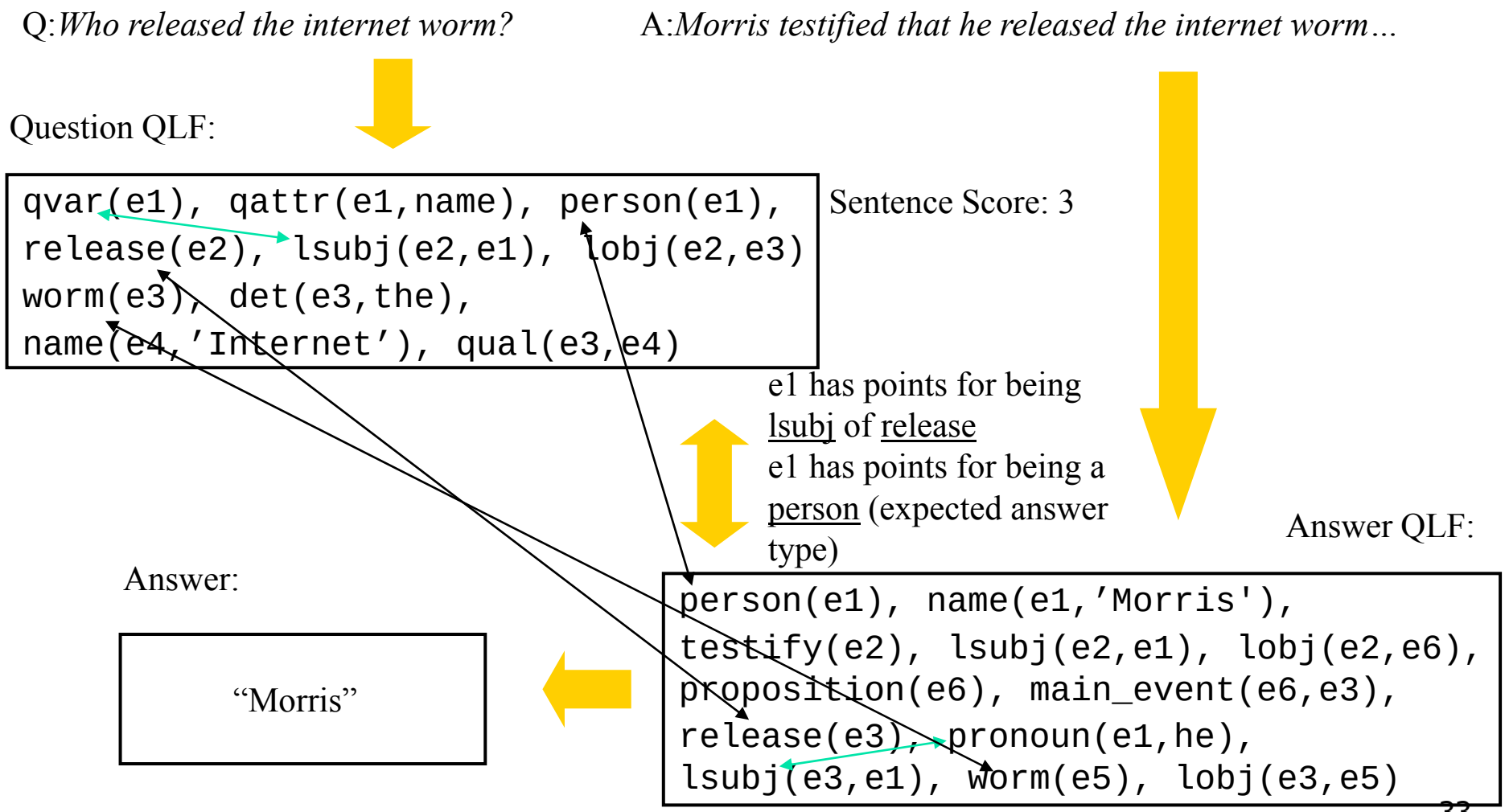
Answer:

"29,035 feet"

Linguistic Processing

- Parse and translate into logical form Q (-> Q1) and each text T (-> T1)
  - Identify in Q1 the sought entity (SE)
- Solve coreference in T1
- For each sentence S1 in T1
  - Count number of shared entities/events (verbs and nouns); this is one score
- For each entity E in S1
  - calculate a score based on
    - semantic proximity between E and SE
    - the number of "constraints" E shares with SE (e.g. subject/object of the same verb)
  - calculate a normalized, combined score for E based on the two scores
- return top scoring entity as answer

An Example


Q:Who released the internet worm?
A:Morris testified that he released the internet worm…
Question QLF:
qvar(e1), qattr(e1,name), person(e1),
release(e2), lsubj(e2,e1), lobj(e2,e3)
worm(e3), det(e3,the),
name(e4,'Internet'), qual(e3,e4)
Sentence Score: 3
e1 has points for being lsubj of release
e1 has points for being a person (expected answer type)
Answer QLF:
person(e1), name(e1,'Morris'),
testify(e2), lsubj(e2,e1), lobj(e2,e6),
proposition(e6), main_event(e6,e3),
release(e3), pronoun(e1,he),
lsubj(e3,e1), worm(e5), lobj(e3,e5)
Answer:
"Morris"

Learning Answer Patterns

- Soubboutin and Soubboutin (2001) introduced a technique for learning answer matching patterns
  - Using a training set consisting of questions, answers and answer bearing contexts from previous TRECs

Learning Answer Patterns

- Answer is located in the context and a regular expression proposed in which a wildcard is introduced to match the answer
  - Question: *When was Handel born?*
  - Answer: *1685*
  - Context: *Handel (1685-1750) was one of the…*
  - Learned RE: *|w+|(|d|d|d|d-*
- Highest scoring system in TREC20001; high scoring in TREC2002

Learning Answer Patterns

- Generalised technique (Greenwood'03)
- Allow named entity typed variables (e.g. Person, Location,Date) to occur in the learned RE's as well as literal text
- Shows significant improvement over previous results for limited question types

Learning Patterns

Suppose a question such as "When was X born?"

- A collection of twenty example questions, of the correct type, and their associated answers is assembled.
- For each example question a pair consisting of the question and answer terms is produced.
  - For example "Abraham Lincoln" – "1809".
- For each example the question and answer terms are submitted to Google, as a single query, and the top 10 documents are downloaded

Learning Patterns

- Each retrieved document then has the question term (e.g. the person) replaced by the single token AnCHoR.
- Depending upon the question type other replacements are then made for dates, persons, locations, and organizations (DatE, LocatioN, OrganizatioN and PersoN) and AnSWeRDatE is used for the answer
- Any remaining instances of the answer term are then replaced by AnSWeR.
- Sentence boundaries are determined and those sentences which contain both AnCHoR and AnSWeR are retained.

Learning Patterns

- A suffix tree is constructed using the retained sentences and all repeated substrings containing both AnCHoR and AnSWeR and which do not span a sentence boundary are extracted.
- This produces a set of patterns, which are specific to the question type.

  for the example of the date of birth the following patterns are induced
  - from AnCHoR (AnSWeRDatE - DatE)
  - AnCHoR , AnSWeRDatE -
  - - AnCHoR (AnSWeRDatE
  - from AnCHoR (AnSWeRDatE –
- these patterns have no information on how accurate they are; so a second step is needed to measure their fitness to answer questions

Learning Pattern Accuracy

- A second set of twenty question-answer pairs are collected and each question is submitted to Google and the top ten documents are downloaded.
- Within each document the question term is replaced by AnCHoR
- The same replacements as carried out in the acquisition phase are made and a table is constructed of the inserted tags and the text they replace.

Learning Pattern Accuracy

- Each of the previously generated patterns is converted to a standard regular expression
- Each of the previously generated patterns is then matched against each sentence containing the AnCHoR tag. Along with each pattern, P, two counts are maintained:
  - CPa(P) , which counts the total number of times the pattern has matched against the text
  - CPc(P) , which counts the number of matches which had the correct answer or a tag which expanded to the correct answer as the text extracted by the pattern.

Learning Pattern Accuracy

- After a pattern, P, has been matched against all the sentences if CPc(P) is less than five it is discarded. The remaining patterns are assigned a precision score calculated as:

  CPc(P)/CPa(P)

- If the pattern's precision is less than or equal to 0.1 then it is also discarded.

Using the Patterns

- Given a question patterns are applied to identify which set of patterns to use
- The patterns are used to match against retrieved passages
- The answer is extracted with the score associated to the pattern
- The best answer is returned

How it performed?

- Patterns learned for the following "questions"
  - *What is the abbreviation for X?*
  - *When was X born?*
  - *What is the capital of X?*
  - *What country is X the capital of?*
  - *When did X die?*
  - *What does X stand for?*
- 49% accuracy
- Works well over the Web
- Patterns are different over other collections such as AQUAINT

Scoring entities

- Index the paragraphs of the AQUAINT collection using the Lucene IR retrieval system
- Apply NE recognition and parsing to the question and perform iterative retrieval using the terms from the question
- Apply NE recognition and parsing to the retrieved documents

Scoring entities

- identify expected answer type from the question
  - qvar(e1) location(e1) then location is the expected answer type
- identify in sentence semantics all 'events'
  - eat(e2) time(e2,pres) then e2 is an event
  - create an annotation of type 'Event' and store the entity identifier as a feature
- identify in sentence semantics all 'objects'
  - everything that is not an 'event'
  - create an annotation of type 'Mention' and store the entity identifier as a feature

Scoring entities

- Identify which 'events' in sentence occur in the question semantics and mark them in the annotation
  - eat(e1) (in question) and eat(e4) (in sentence)
- Identify which 'objects' in sentence occur in the question semantics and mark them in the annotation
  - bird(e2) (in question) and bird(e6) (in sentence)

Scoring entities

- For each ‘object’ identify relations in which they are involved (lsubj, lobj, of, in, etc.) and if they are related to any entity which was marked, then record the relation with value 1 as a feature of the ‘object’
  - release(e1) (in question)
  - release(e3) and lsubj(e3,e2) and name(e2,’Morris) then mark e2 as having a relation lsubj=1

Scoring entities

- Compute 'WordNet' similarity between the expected answer type and each 'object'
  - EAT = location and city(e2) is in sentence the similarity is 0.66 using Lin similarity metric from the JWordNetSim package developed by M. Greenwood

Scoring entities

- For each sentence count how many shared events and objects the sentence has with the question
  - add that score to each 'object' in the sentence – feature 'constrains'
- Score each sentence with a formula which takes into account
  - constrains; similarity; some matched relations (adjust weights on training data)
- Use score to rank entities
- In case of ties use external sources for example

N-gram Techniques (Brill&al'01)

- Do not use any sophisticated technique but redundancy on the Web
- Locate possible answers on the Web and then project over a document collections
- Given a question, patterns are generated which can locate the answer
  - "Who is Tom Cruise married to?"
  - <"Tom Cruise is married to", right, 5>
  - < text, where to look for answer, confidence>

N-gram Techniques

- Use the text to locate documents and summaries (snippets)
- Generate n-grams (n<=3) from the summaries
- n-grams scored (n-grams occurring in multiple summaries score higher)

N-gram example

President Adamkus will meet with **the President of Argentina** Ms. Cristina Fernández

Ms., Cristina, Fernandez, Ms. Cristina, Cristina Fernandez, Ms. Cristina Fernandez

Speech by **the President of Argentina**, Dr. Néstor Kirchner

Dr., Nestor, Kirchner, Dr. Nestor, Nestor Kirchner, …

The President of Argentina: Néstor Kirchner Vice President: Daniel Scioli.

Nestor, Kirchner, Vice,…,Nestor Kirchner,…

the president of Argentina, Nestor Kirchner, is outdoing both leaders

Nestor, Kirchner, Nestor Kirchner,…

Nestor Kirchner **the Argentine president**…

Nestor, Kirchner, Nestor Kirchner

Ms. Kirchner **the Argentine president**….

Ms., Kirchner, Ms. Kirchner

Dr. Menem **the Argentine president**

Dr., Menem, Dr. Menem

She is not the daughter of **the Argentine president**

She, is, not, the, daughter, of, She is, ….the daughter,

N-gram Techniques

- Filtering for type of sought entity is applied to modify the statistical score
  - for example if person is sought, then n-gram should contain person name
- Tilling is applied to combine multiple n-grams
  - A B C and B C D produce A B C D with a new score
- Best n-grams are used to find documents which can be used as justification for the answer
- System has very good performance in TREC/QA

Metrics and Scoring – MRR (Voorhees'00)

- The principal metric for TREC8-10 was Mean Reciprocal Rank (MRR)
  - Correct answer at rank 1 scores 1
  - Correct answer at rank 2 scores 1/2
  - …

  Sum over all questions and divide by number of questions

$$MRR = \frac{\sum_{i=1}^{N} r_i}{N}$$

Metrics and Scoring – MRR

- where

  N = # questions, r_i = the reciprocal of the best (lowest) rank assigned by a system at which a correct answer is found for question i, or 0 if no correct answer was found
- Judgements made by human judges based on answer string alone (lenient evaluation) and by reference to documents (strict evaluation)

Metrics and Scoring – CWS (Voorhees'02)

- The principal metric for TREC2002 was Confidence Weighted Score

$$\text{confidence weighted score} = \frac{\sum_{i=1}^{Q} \#\text{correct in first } i \text{ positions}/i}{Q}$$

where Q is number of questions

Answer Accuracy (Voorhees'03)

- When only one answer is accepted per question, the metric used is answer accuracy: percent of correct answers

Answering Definition Questions (Voorhees'03)

- text collection (e.g., AQUAINT)
- definition question (e.g., "*What is Goth*?", *"Who is Aaron Copland?")*
  - Goth is the definiendum or term to be defined
- answer for Goth: "*a subculture that started as one component of the punk rock scene"* or *"horror/mystery literature that is dark, eerie, and gloomy"* or *...*
- architecture: Information Retrieval + Information Extraction
- definiendum gives little information for retrieving definition-bearing passages

Gold standard by NIST

Qid 1901: Who is Aaron Copland?

1901 1	vital	american composer
1901 2	vital	musical achievements ballets symphonies
1901 3	vital	born brooklyn ny 1900
1901 4	okay	son jewish immigrant
1901 5	okay	american communist
1901 6	okay	civil rights advocate
1901 7	okay	had senile dementia
1901 8	vital	established home for composers
1901 9	okay	won oscar for "the Heiress"
1901 10	okay	homosexual
1901 11	okay	teacher tanglewood music center boston symphony

BBN Approach (Yang et al'03) – best approach in TREC 2003

1. Identify type of question (who or what) and the question target
2. Retrieve 1000 documents using an IR system and the target as query
3. For each sentence in the documents decide if it mention the target
4. Extract *kernel facts* (phrases) from each sentence
5. Rank all kernel facts according to type and similarity to a question profile (centroid)
6. Detect redundant facts – facts that are different from already extracted facts are added to the answer set

BBN Approach (cont.)

- Check if document contains target
  - First...Last for who, full match for what
  - Sentence match can be direct or through coreference; name match uses last name only
- Extract kernel facts
  - appositive and copula constructions
    - "George Bush, the president..." "George Bush is the president..." (this is done using parsed sentences)

BBN Approach (cont.)

- Extract kernel facts
  - special and ordinary propositions: pred(role:arg,.....role:arg) for example love(subj:mary,obj:john) for "Mary loves John" – an special proposition would be "born in" of "educated in"
  - ~ 40 structured patterns typically used to define terms (TERM is NP)
  - Relations – 24 specific types of binary relations such as the staff of an organization
  - Full sentences used as fall back – do not match any of the above

BBN Approach (cont.)

- Ranking kernel facts
  - 1) appositives and copula ranked higher; 2) structured patterns; 3) special props; 4) relations; 5) props and sentences
  - Question profile: centroid of definitions from on-line dictionaries (e.g., Wikipedia); centroid of set of biographies; or centroid of all kernel facts
  - a similarity metric using tf*idf is used to rank the facts

BBN Approach (cont.)

- Redundancy removal
  - for propositions to be equivalent, same predicate and same argument head
  - for structured patterns, if the sentence was selected by a pattern used at least two times, then redundant
  - for other facts, check word overlap (>0.70 overlap is redundant)

BBN Approach (cont.)

- Algorithm for generating definitions
  - S={}
  - Rank all kernel facts based on profile similarity; iterate over the facts and discard redundant until there are m facts in S
  - Rank all remaining based on type (first) and similarity (second) add to S until maximum allowance reached or number of sentences and ordinary props greater than n
  - return S
- there is also a fall back approach when the above procedure does not produce any results – this is based on information retrieval

Other Techniques

- Off-line strategies for identification in news paper articles of cases of <Concept, Instance> such as "Bush, President of the United States" (Fleishman&al'03)
  - use 2 types of patterns common noun (CN) proper noun (PN) constructions (English goalkeeper Seaman) and appositive constructions (Seaman, the English goalkeeper)
  - use a filter (classifier) to weed out noise
    - a number of features are used for the classifier including the pattern used; the semantic type of the head noun in the pattern; the morphology of the headnoun (e.g. spokesman); etc.

Other techniques

- DefScriber: definitional predicates and data-driven techniques (Blair-Goldensohn&al'03)
  - predicates = genus, species, non-specific – ML techniques over annotated corpus and patterns (manual)
  - centroid-based similarity and clustering

Other techniques

- Best TREC QA 2006 def system used the Web to collect word frequencies (Kaisser'07)
  - Given a target obtain snippets from the web for queries containing the target words
  - Create a list of word frequencies
  - Retrieve docs from collection using target
  - Score sentences using the word frequencies
  - Pick up top ranked sentence and re-rank the rest of the sentences
  - Continue until termination

QA-definition approach (Saggion&Gaizauskas'04)

- linguistic patterns:
  - "*is a*" , "*such as*", "*consists of*", etc.
  - many forms in which definitions are expressed in texts
  - match definitions and non-definitions
  - "*Goth is a subculture*" & "*Becoming a Goth is a process that demands lots of effort*"

QA-definition approach

- Secondary terms
  - Given multiple definitions of a specific definiendum, key defining terms are observed to recur across the definitions
  - For example
    - On the Web "*Goth*" seems to be associated with "*subculture*" in definition passages
  - Can we exploit known definitional contexts to assemble terms likely to co-occur with the definiendum in definitions?

Approach: use external sources

- Knowledge capture
  - identify definition passages (outside target collection) for the definiendum using patterns
  - WordNet, Wikipedia, Web in general
  - identify (secondary) terms associated to the definiendum in those passages
- During Answer extraction
  - use definiendum & secondary terms during IR
  - use secondary terms & patterns during IE from collection passages

Examples of Passages

Definiendum: aspirin

Pattern		Passage	
Uninstantiated	**Instantiated**	**Relevant**	**Not Relevant**
TERM is a	aspirin is a	Aspirin is a weak monotripic acid	Aspirin is a great choice for active people
such as TERM	such as aspirin	blood-thinners such as aspirin...	Look for travel size items such as aspirin
like TERM	like aspirin	non-steroidal antinflamatory drugs like aspirin	a clown is like aspirin, only he works twice as fast

Term List

- create a list of secondary terms
  - all WordNet terms, terms with count > 1 from web

Definiendum	WordNet	Encyclopedia	Web
aspirin	analgesic; anti-inflammatory; antipyretic; drug; …	inhibit; prostaglandin; ketofren; synthesis; …	drug; drugs; blood; ibuprofen; medication; pain; …
Aum Shirikyo	* NOTHING *	* NOTHING *	group; groups; cult; religious; japanese; etc.

Definition extraction

- perform query expansion & retrieval
- analyse retrieved passages
  - look-up of definiendum, secondary terms, definition patterns
  - identify definition-bearing sentences
- identify answer
  - "*Who is Andrew Carnegie*?"
    - *In a question-and-answer session after the panel discussion, Clinton cited philanthropists from an earlier era such as Andrew Carnegie, J.P. Morgan, and John D. Rockefeller...*
    - *philanthropists from an earlier era such as Andrew Carnegie, J.P. Morgan, and John D. Rockefeller...*
- filter out redundant answers
  - vector space model and cosine similarity with threshold

What can go wrong

- many things…
  - *Akbar the Great* — Proper Noun
  - *Abraham in the Old Testament* — definiendum
    Problem
  - *Andrea Bocceli* — no such person
  - *Antonia Coelho Novello* — name alias
  - *Charles Lindberg* — aviator/aviation
  - *medical condition shingles* — no patterns
  - *Alexander Pope*
    irrelevant docs

Gold standard by NIST

Qid 1901: Who is Aaron Copland?

1901 1	vital	american composer
1901 2	vital	musical achievements ballets symphonies
1901 3	vital	born brooklyn ny 1900
1901 4	okay	son jewish immigrant
1901 5	okay	american communist
1901 6	okay	civil rights advocate
1901 7	okay	had senile dementia
1901 8	vital	established home for composers
1901 9	okay	won oscar for "the Heiress"
1901 10	okay	homosexual
1901 11	okay	teacher tanglewood music center boston symphony

Evaluation

- NIST
  - matching system answers to human answers
- Metrics
  - « nugget recall » (NR) ~ traditional recall
  - « nugget precision » (NP) ~ space used by system answer is important
    - it is better to save space
  - « F-score » (F) harmonic mean of NR and NP where NR is 5 times more important than NP