Statistical NLP
Spring 2007

Lecture 21: Coreference

Dan Klein – UC Berkeley

Includes examples from Johnson, Jurafsky and Gildea, Luo, Palmer

Next Few Weeks

- April 23: John DeNero on Syntactic MT
- April 25: No class
- April 30 / May 2: Presentations
- May 7: Conclusions

Next Few Weeks

- In-class presentations: April 30 (M), May 2 (W)
  - Sign up with me
  - You've got 10-20 minutes, one slot per group
  - Tell us:
    - The problem: why do we care?
    - Your concrete task: input, output, evaluation
    - A simple baseline for the task
    - Your method (half the time here)
    - Any serious surprises, challenges, etc?
    - Headline results (if any)
  - Or: describe a paper you've chosen to present
- Hint: put any slides on the web before class

CCG Parsing

- Combinatory Categorial Grammar
  - Fully (mono-) lexicalized grammar
  - Categories encode argument sequences
  - Very closely related to the lambda calculus
  - Can have spurious ambiguities (why?)

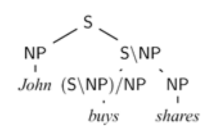
$John \vdash \mathrm{NP} : john'$

$shares \vdash \mathrm{NP} : shares'$

$buys \vdash (\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP} : \lambda x.\lambda y.buys'xy$

$sleeps \vdash \mathrm{S}\backslash\mathrm{NP} : \lambda x.sleeps'x$

$well \vdash (\mathrm{S}\backslash\mathrm{NP})\backslash(\mathrm{S}\backslash\mathrm{NP}) : \lambda f.\lambda x.well'(fx)$



Reference Resolution

- Noun phrases refer to entities in the world, many pairs of noun phrases co-refer:

John Smith, CFO of Prime Corp. since 1986,

saw his pay jump 20% to $1.3 million

as the 57-year-old also became

the financial services co.'s president.

Kinds of Reference

- Referring expressions
  - *John Smith*
  - *President Smith*
  - *the president*
  - *the company's new executive*

More common in newswire, generally harder in practice

- Free variables
  - Smith saw *his pay* increase
- Bound variables
  - Every company trademarks its name.

More interesting grammatical constraints, more linguistic theory, easier in practice

Grammatical Constraints

- Gender / number
  - Jack gave Mary a gift. She was excited.
  - Mary gave her mother a gift. She was excited.
- Position (cf. binding theory)
  - The company's board polices itself / it.
  - Bob thinks Jack sends email to himself / him.
- Direction (anaphora vs. cataphora)
  - She bought a coat for Amy.
  - In her closet, Amy found her lost coat.

Discourse Constraints

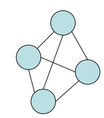
- Recency
- Salience
- Focus
- Centering Theory [Grosz et al. 86]

Other Constraints

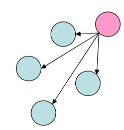
- Style / Usage Patterns
  - *Peter Watters* was named CEO. *Watters'* promotion came six weeks after his brother, *Eric Watters*, stepped down.
- Semantic Compatibility
  - Smith had bought *a used car* that morning. *The used car dealership* assured him it was in good condition.

Two Kinds of Models

- Mention Pair models
  - Treat coreference chains as a collection of pairwise links
  - Make independent pairwise decisions and reconcile them in some way (e.g. clustering or greedy partitioning)



- Entity Mention models
  - A cleaner, but less studied, approach
  - Posit single underlying entities
  - Each mention links to a discourse entity [Pasula et al. 03], [Luo et al. 04]



Mention Pair Models

- Most common machine learning approach
- Build classifiers over pairs of NPs
  - For each NP, pick a preceding NP or NEW
  - Or, for each NP, choose link or no-link
- Clean up non transitivity with clustering or graph partitioning algorithms
  - E.g.: [Soon et al. 01], [Ng and Cardie 02]
  - Some work has done the classification and clustering jointly [McCallum and Wellner 03]
- Kind of a hack, results in the 50's to 60's on all NPs
  - Better number on proper names and pronouns
  - Better numbers if tested on gold entities
- Failures are mostly because of insufficient knowledge or features for hard common noun cases

Pairwise Features

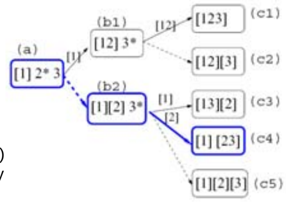
Category	Features	Remark
Lexical	exact_strm	1 if two mentions have the same spelling; 0 otherwise
	left_subsm	1 if one mention is a left substring of the other; 0 otherwise
	right_subsm	1 if one mention is a right substring of the other; 0 otherwise
	acronym	1 if one mention is an acronym of the other; 0 otherwise
	edit_dist	quantized editing distance between two mention strings
	spell	pair of actual mention strings
	ncd	number of different capitalized words in two mentions
Distance	token_dist	how many tokens two mentions are apart (quantized)
	sent_dist	how many sentences two mentions are apart (quantized)
	gap_dist	how many mentions in between the two mentions in question (quantized)
Syntax	POS_pair	POS-pair of two mention heads
	apposition	1 if two mentions are appositive; 0 otherwise
Count	count	pair of (quantized) numbers, each counting how many times a mention string is seen
Pronoun	gender	pair of attributes of {female, male, neutral, unknown }
	number	pair of attributes of {singular, plural, unknown}
	possessive	1 if a pronoun is possessive; 0 otherwise
	reflexive	1 if a pronoun is reflexive; 0 otherwise

	Devtest		Feb02		Sep02	
Model	ACE-val(%)	ECM-F(%)	ACE-val(%)	ECM-F(%)	ACE-val(%)	ECM-F(%)
MP	89.8	73.2 (±2.9)	90.0	73.1 (±4.0)	88.0	73.1 (±6.8)
EM	89.9	71.7 (±2.4)	88.2	70.8 (±3.9)	87.6	72.4 (±6.2)

[Luo et al. 04]

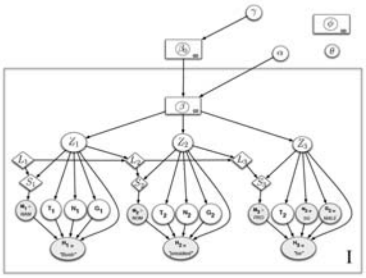
An Entity Mention Model

- Example: [Luo et al. 04]
- Bell Tree (link vs. start decision list)
- Entity centroids, or not?
  - Not for [Luo et al. 04], see [Pasula et al. 03]
  - Some features work on nearest mention (e.g. recency and distance)
  - Others work on "canonical" mention (e.g. spelling match)
  - Lots of pruning, model highly approximate
  - (Actually ends up being like a greedy-link system in the end)



A Generative Model

- [Haghighi and Klein 07]



Question Answering

- Following largely from Chris Manning's slides, which includes slides originally borrowed from Sanda Harabagiu, ISI, Nicholas Kushmerick.

Question Answering from Text

- Question Answering:
  - Give the user a (short) answer to their question, perhaps supported by evidence.
  - An idea originating from the IR community
  - With massive collections of full-text documents, simply finding *relevant documents* is of limited use: we want *answers* from textbases
- The common person's view? [From a novel]
  - "I like the Internet. Really, I do. Any time I need a piece of shareware or I want to find out the weather in Bogota … I'm the first guy to get the modem humming. But as a source of information, it sucks. You got a billion pieces of data, struggling to be heard and seen and downloaded, and anything I want to know seems to get trampled underfoot in the crowd."
    - M. Marshall. *The Straw Men.* HarperCollins Publishers, 2002.

People *want* to ask questions?

Examples from AltaVista query log
who invented surf music?
how to make stink bombs
where are the snowdens of yesteryear?
which english translation of the bible is used in official catholic liturgies?
how to do clayart
how to copy psx
how tall is the sears tower?
Examples from Excite query log (12/1999)
how can i find someone in texas
where can i find information on puritan religion?
what are the 7 wonders of the world
how can i eliminate stress
What vacuum cleaner does Consumers Guide recommend
Around 10–15% of query logs

AskJeeves (Classic)

- Probably the most hyped example of "question answering"
- It largely does pattern matching to match your question to their own knowledge base of questions
- If that works, you get the human-curated answers to that known question
- If that fails, it falls back to regular web search
- A potentially interesting middle ground, but not full QA

A Brief (Academic) History

- Question answering is not a new research area
- Question answering systems can be found in many areas of NLP research, including:
  - Natural language database systems
    - A lot of early NLP work on these
  - Spoken dialog systems
    - Currently very active and commercially relevant
- The focus on open-domain QA is new
  - MURAX (Kupiec 1993): Encyclopedia answers
  - Hirschman: Reading comprehension tests
  - TREC QA competition: 1999–

Question Answering at TREC

- Question answering competition at TREC consists of answering a set of 500 fact-based questions, e.g., *"When was Mozart born*?".
- For the first three years systems were allowed to return 5 ranked answer snippets (50/250 bytes) to each question.
  - IR think
  - Mean Reciprocal Rank (MRR) scoring:
    - 1, 0.5, 0.33, 0.25, 0.2, 0 for 1, 2, 3, 4, 5, 6+ doc
  - Mainly Named Entity answers (person, place, date, …)
- From 2002 the systems are only allowed to return a single *exact* answer and the notion of confidence has been introduced.

The TREC Document Collection

- The current collection uses news articles from the following sources:
  - AP newswire, 1998-2000
  - New York Times newswire, 1998-2000
  - Xinhua News Agency newswire, 1996-2000
- In total there are 1,033,461 documents in the collection. 3GB of text
- Clearly this is too much text to process entirely using advanced NLP techniques so the systems usually consist of an initial information retrieval phase followed by more advanced processing.
- Many supplement this text with use of the web, and other knowledge bases

Sample TREC questions

1. Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?
2. What was the monetary value of the Nobel Peace Prize in 1989?
3. What does the Peugeot company manufacture?
4. How much did Mercury spend on advertising in 1993?
5. What is the name of the managing director of Apricot Computer?
6. Why did David Koresh ask the FBI for a word processor?
7. What debts did Qintex group leave?
8. What is the name of the rare neurological disease with symptoms such as: involuntary movements (tics), swearing, and incoherent vocalizations (grunts, shouts, etc.)?

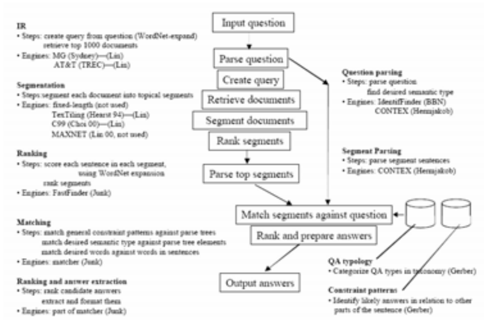
Top Performing Systems

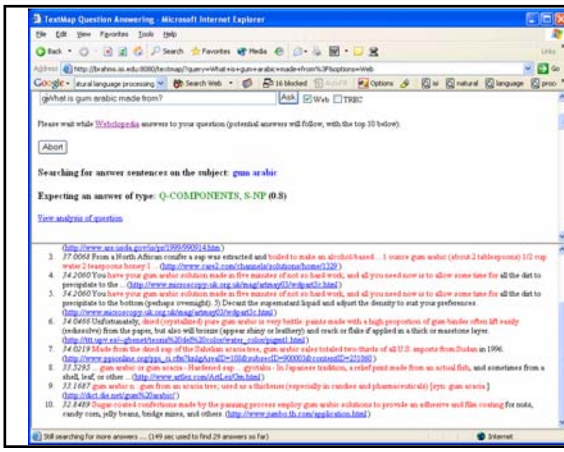
- Currently the best performing systems at TREC can answer approximately 70% of the questions
- Approaches and successes have varied a fair deal
  - Knowledge-rich approaches, using a vast array of NLP techniques stole the show in 2000, 2001
    - Notably Harabagiu, Moldovan et al. – SMU/UTD/LCC
  - AskMSR system stressed how much could be achieved by very simple methods with enough text (and now various copycats)
  - Middle ground is to use large collection of surface matching patterns (ISI)

Online QA System Examples

- Examples
  - **AnswerBus** is an open-domain question answering system: www.answerbus.com
  - **Ionaut:** http://www.ionaut.com:8400/
  - **LCC:** http://www.languagecomputer.com/
  - **EasyAsk, AnswerLogic, AnswerFriend, Start, Quasm, Mulder, Webclopedia, etc.**
  - **ISI TextMap** http://brahms.isi.edu:8080/textmap/

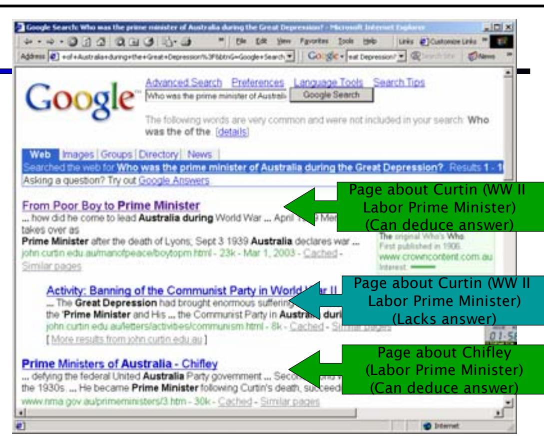
Webclopedia Architecture





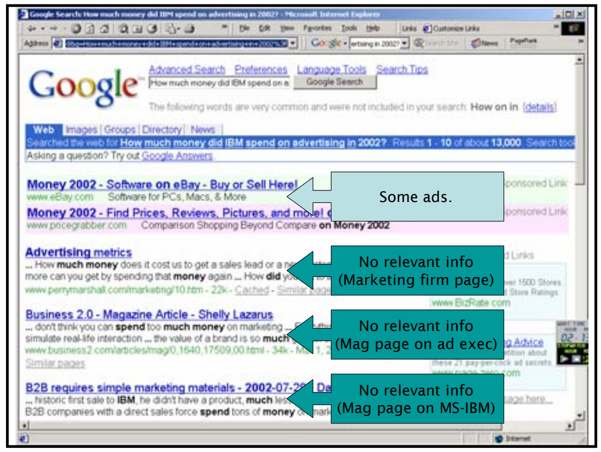
The Google Answer #1

- Include question words etc. in your stop list
- Do standard IR
- Sometimes this (sort of) works:
- Question: *Who was the prime minister of Australia during the Great Depression?*
- Answer: *James Scullin (Labor) 1929–31.*



But often it doesn't…

- Question: *How much money did IBM spend on advertising in 2002?*
- Answer: *I dunno, but I'd like to …*



The Google Answer #2

- Take the question and try to find it as a string on the web
- Return the next sentence on that web page as the answer
- Works brilliantly if this exact question appears as a FAQ question, etc.
- Works lousily most of the time
- Reminiscent of the line about monkeys and typewriters producing Shakespeare
- But a slightly more sophisticated version of this approach has been revived in recent years with considerable success…

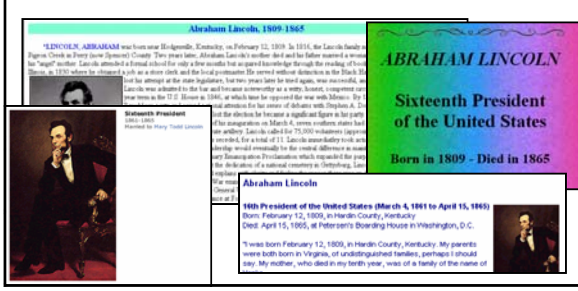
AskMSR

- ***Web Question Answering: Is More Always Better?***
  - Dumais, Banko, Brill, Lin, Ng (Microsoft, MIT, Berkeley)

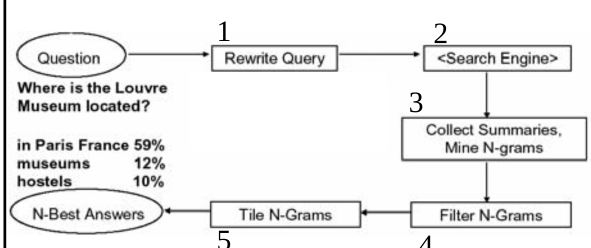
- **Q: "Where is the Louvre located?"**
- Want "Paris" or "France" or "75058 Paris Cedex 01" or a map
- Don't just want URLs

AskMSR: Shallow approach

- *In what year did Abraham Lincoln die?*
- Ignore hard documents and find easy ones



AskMSR: Details



Step 1: Rewrite queries

- Intuition: The user's question is often syntactically quite close to sentences that contain the answer
  - Where is the Louvre Museum located?
  - The Louvre Museum is located in ***Paris***
  - Who created the character of Scrooge?
  - ***Charles Dickens*** created the character of Scrooge.

Query Rewriting: Variations

- Classify question into seven categories
  - **Who** is/was/are/were…?
  - **When** is/did/will/are/were …?
  - **Where** is/are/were …?

  a. Category-specific transformation rules
  eg "For Where questions, move 'is' to all possible locations"
  "Where is the Louvre Museum located"
  → "is the Louvre Museum located"
  → "the is Louvre Museum located"
  → "the Louvre is Museum located"
  → "the Louvre Museum is located"
  → "the Louvre Museum located is"

  Nonsense, but who cares? It's only a few more queries

  b. Expected answer "Datatype" (eg, Date, Person, Location, …)
  When was the French Revolution? → DATE
- Hand-crafted classification/rewrite/datatype rules
  (Could they be automatically learned?)

Query Rewriting: Weights

- One wrinkle: Some query rewrites are more reliable than others

Where is the Louvre Museum located?

Weight 1
Lots of non-answers could come back too

+Louvre +Museum +located

Weight 5
if we get a match, it's probably right

+"the Louvre Museum is located"

Step 2: Query search engine

- Send all rewrites to a Web search engine
- Retrieve top N answers (100?)
- For speed, rely just on search engine's "snippets", not the full text of the actual document

Step 3: Mining N-Grams

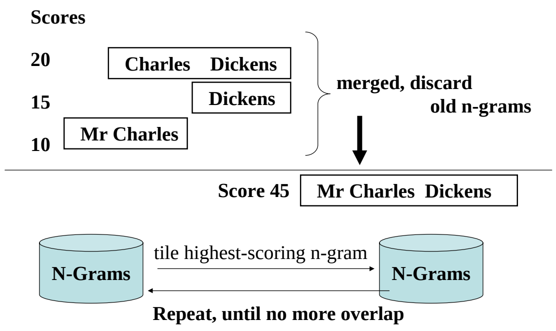
- Simple: Enumerate all N-grams (N=1,2,3 say) in all retrieved snippets
  - Use hash table and other fancy footwork to make this efficient
- Weight of an n-gram: occurrence count, each weighted by "reliability" (weight) of rewrite that fetched the document
- Example: "Who created the character of Scrooge?"
  - Dickens - 117
  - Christmas Carol - 78
  - Charles Dickens - 75
  - Disney - 72
  - Carl Banks - 54
  - A Christmas - 41
  - Christmas Carol - 45
  - Uncle - 31

Step 4: Filtering N-Grams

- Each question type is associated with one or more "**data-type filters**" = regular expression
- When… → **Date**
- Where… → **Location**
- What … → **Date**, **Location**
- Who … → **Person**

- Boost score of n-grams that do match regexp
- Lower score of n-grams that don't match regexp
- Details omitted from paper….

Step 5: Tiling the Answers



Results

- Standard TREC contest test-bed:
  ~1M documents; 900 questions
- Technique doesn't do too well (though would have placed in top 9 of ~30 participants!)
  - MRR = 0.262 (ie, right answered ranked about #4-#5 on average)
  - Why? Because it relies on the enormity of the Web!
- Using the Web as a whole, not just TREC's 1M documents… MRR = 0.42 (ie, on average, right answer is ranked about #2-#3)

Issues

- In many scenarios (e.g., monitoring an individuals email…) we only have a small set of documents
- Works best/only for "Trivial Pursuit"-style fact-based questions
- Limited/brittle repertoire of
  - question categories
  - answer data types/filters
  - query rewriting rules

Ravichandran and Hovy 2002
Learning Surface Patterns

- Use of Characteristic Phrases
- "When was <person> born"
  - Typical answers
    - "Mozart was born in 1756."
    - "Gandhi (1869-1948)..."
  - Suggests phrases like
    - "<NAME> was born in <BIRTHDATE>"
    - "<NAME> (<BIRTHDATE>-"
  - as Regular Expressions can help locate correct answer

Use Pattern Learning

- Example: Start with "Mozart 1756"
  - Results:
    - "The great composer Mozart (1756-1791) achieved fame at a young age"
    - "Mozart (1756-1791) was a genius"
    - "The whole world would always be indebted to the great music of Mozart (1756-1791)"
  - Longest matching substring for all 3 sentences is "Mozart (1756-1791)"
  - Suffix tree would extract "Mozart (1756-1791)" as an output, with score of 3
- Reminiscent of IE pattern learning

Pattern Learning (cont.)

- Repeat with different examples of same question type
  - "Gandhi 1869", "Newton 1642", etc.
- Some patterns learned for BIRTHDATE
  - a. born in <ANSWER>, <NAME>
  - b. <NAME> was born on <ANSWER> ,
  - c. <NAME> (<ANSWER> -
  - d. <NAME> (<ANSWER> -)

Experiments: (R+H, 2002)

- 6 different Question types
  - from Webclopedia QA Typology (Hovy et al., 2002a)
    - BIRTHDATE
    - LOCATION
    - INVENTOR
    - DISCOVERER
    - DEFINITION
    - WHY-FAMOUS

Experiments: pattern precision

- BIRTHDATE table:
  - 1.0 <NAME> (<ANSWER> -)
  - 0.85 <NAME> was born on <ANSWER>,
  - 0.6 <NAME> was born in <ANSWER>
  - 0.59 <NAME> was born <ANSWER>
  - 0.53 <ANSWER> <NAME> was born
  - 0.50 - <NAME> (<ANSWER>
  - 0.36 <NAME> (<ANSWER> -
- INVENTOR
  - 1.0 <ANSWER> invents <NAME>
  - 1.0 the <NAME> was invented by <ANSWER>
  - 1.0 <ANSWER> invented the <NAME> in

Experiments (cont.)

- WHY-FAMOUS
  - 1.0 <ANSWER> <NAME> called
  - 1.0 laureate <ANSWER> <NAME>
  - 0.71 <NAME> is the <ANSWER> of
- LOCATION
  - 1.0 <ANSWER>'s <NAME>
  - 1.0 regional : <ANSWER> : <NAME>
  - 0.92 near <NAME> in <ANSWER>
- Depending on question type, get high MRR (0.6–0.9), with higher results from use of Web than TREC QA collection

Shortcomings & Extensions

- Need for POS &/or semantic types
  - "Where are the Rocky Mountains?"
  - "Denver's new airport, topped with white fiberglass cones in imitation of the Rocky Mountains in <u>the background</u> , continues to lie empty"
  - <NAME> in <ANSWER>
- NE tagger &/or ontology could enable system to determine "background" is not a location

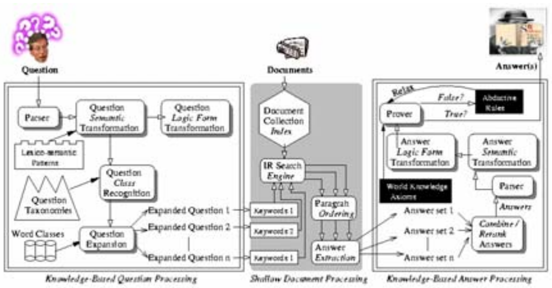
Shortcomings... (cont.)

- Long distance dependencies
  - "Where is London?"
  - "London, which has one of the most busiest airports in the world, lies on the banks of the river Thames"
  - would require pattern like: <QUESTION>, (<any_word>)*, lies on <ANSWER>
- But: abundance & variety of Web data helps system to find an instance of patterns w/o losing answers to long distance dependencies

Shortcomings... (cont.)

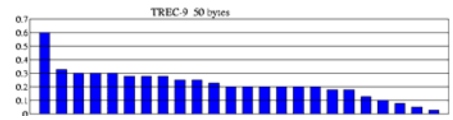
- Their system uses only one anchor word
  - Doesn't work for Q types requiring multiple words from question to be in answer
    - "In which county does the city of Long Beach lie?"
    - "Long Beach is situated in Los Angeles County"
    - required pattern: <Q_TERM_1> is situated in <ANSWER> <Q_TERM_2>
- Does not use case
  - "What is a micron?"
  - "...a spokesman for Micron, <u>a maker of semiconductors</u>, said SIMMs are..."

LCC: Harabagiu, Moldovan et al.

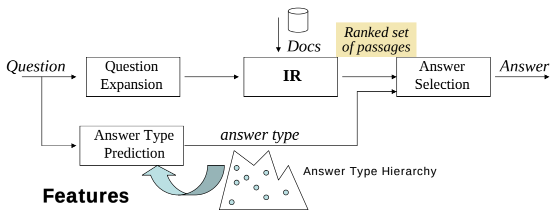


Value from Sophisticated NLP
Pasca and Harabagiu (2001)

- Good IR is needed: SMART paragraph retrieval
- Large taxonomy of question types and expected answer types is crucial
- Statistical parser used to parse questions and relevant text for answers, and to build KB
- Query expansion loops (morphological, lexical synonyms, and semantic relations) important
- Answer ranking by simple ML method



Answer types in SOA QA systems



Features

- Answer type
  - Labels questions with answer type based on a taxonomy
  - Classifies questions (e.g. by using a maximum entropy model)

QA Typology (from ISI USC)

- Typology of typical Q forms—94 nodes (47 leaf nodes)
- Analyzed 17,384 questions (from answers.com)

```
(THING
 ((AGENT
   (NAME (FEMALE-FIRST-NAME (EVE MARY ...))
         (MALE-FIRST-NAME (LAWRENCE SAM ...))))
         (COMPANY-NAME (BOEING AMERICAN-EXPRESS))
         JESUS ROMANOFF ...)
   (ANIMAL-HUMAN (ANIMAL (WOODCHUCK YAK ...))
                 PERSON)
   (ORGANIZATION (SQUADRON DICTATORSHIP ...))
   (GROUP-OF-PEOPLE (POSSE CHOIR ...))
   (STATE-DISTRICT (TIROL MISSISSIPPI ...))
   (CITY (ULAN-BATOR VIENNA ...))
   (COUNTRY (SULTANATE ZIMBABWE ...))))
 (PLACE
   (STATE-DISTRICT (CITY COUNTRY...))
   (GEOLOGICAL-FORMATION (STAR CANYON...))
   AIRPORT COLLEGE CAPITOL ...)
 (ABSTRACT
  (LANGUAGE (LETTER-CHARACTER (A B ...)))
  (QUANTITY
   (NUMERICAL-QUANTITY INFORMATION-QUANTITY
   MASS-QUANTITY MONETARY-QUANTITY
   TEMPORAL-QUANTITY ENERGY-QUANTITY
   TEMPERATURE-QUANTITY ILLUMINATION-QUANTITY

   (SPATIAL-QUANTITY
    (VOLUME-QUANTITY AREA-QUANTITY DISTANCE-QUANTITY)) ...
     PERCENTAGE)))
 (UNIT
  ((INFORMATION-UNIT (BIT BYTE ... EXABYTE))
   (MASS-UNIT (OUNCE ...)) (ENERGY-UNIT (BTU ...))
   (CURRENCY-UNIT (ZLOTY PESO ...))
   (TEMPORAL-UNIT (ATTOSECOND ... MILLENIUM))
   (TEMPERATURE-UNIT (FAHRENHEIT KELVIN CELCIUS))
   (ILLUMINATION-UNIT (LUX CANDELA))
   (SPATIAL-UNIT
    ((VOLUME-UNIT (DECILITER ...))
     (DISTANCE-UNIT (NANOMETER ...))))
     (AREA-UNIT (ACRE)) ... PERCENT))
 (TANGIBLE-OBJECT
  ((FOOD (HUMAN-FOOD (FISH CHEESE ...)))
   (SUBSTANCE
    ((LIQUID (LEMONADE GASOLINE BLOOD ...))
     (SOLID-SUBSTANCE (MARBLE PAPER ...))
     (GAS-FORM-SUBSTANCE (GAS AIR)) ...))
   (INSTRUMENT (DRUM DRILL (WEAPON (ARM GUN)) ...)
   (BODY-PART (ARM HEART ...))
   (MUSICAL-INSTRUMENT (PIANO)))
   ... *GARMENT *PLANT DISEASE)
```

Extracting Answers for Factoids

- In TREC 2003 the LCC QA system extracted 289 correct answers for factoid questions
- The Name Entity Recognizer was responsible for 234 of them

QUANTITY	55	ORGANIZATION	15	PRICE	3
NUMBER	45	AUTHORED WORK	11	SCIENCE NAME	2
DATE	35	PRODUCT	11	ACRONYM	1
PERSON	31	CONTINENT	5	ADDRESS	1
COUNTRY	21	PROVINCE	5	ALPHABET	1
OTHER LOCATIONS	19	QUOTE	5	URI	1
CITY	19	UNIVERSITY	3		

Special Case of Names

Questions asking for names of authored works

1934: What is the play "West Side Story" based on?
Answer: Romeo and Juliet

1976: What is the motto for the Boy Scouts?
Answer: Be Prepared

1982: What movie won the Academy Award for best picture in 1989?
Answer: Driving Miss Daisy

2080: What peace treaty ended WWI?
Answer: Versailles

2102: What American landmark stands on Liberty Island?
Answer: Statue of Liberty

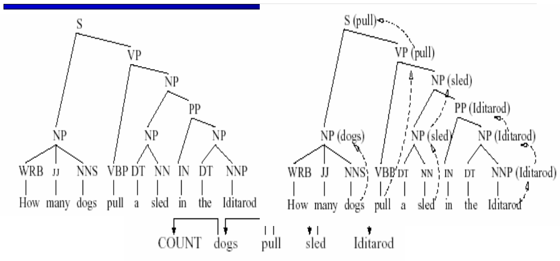
Named Entity Recognition for QA

- The results of the past 5 TREC evaluations of QA systems indicate that current state-of-the-art QA is determined by the recognition of Named Entities:
  - *Precision of recognition*
  - *Coverage of name classes*
  - *Mapping into concept hierarchies*
  - *Participation into semantic relations (e.g. predicate-argument structures or frame semantics)*

Concept Taxonomies

- For 29% of questions the QA system relied on an off-line taxonomy with semantic classes such as:
  - Disease
  - Drugs
  - Colors
  - Insects
  - Games
- *The majority of these semantic classes are also associated with patterns that enable their identification*

Syntax to Logical Forms



•Syntactic analysis plus semantic => logical form
•Mapping of question and potential answer LFs to find the best match

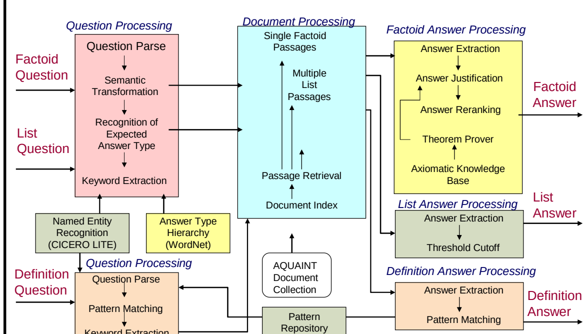
Abductive inference

- System attempts inference to justify an answer (often following lexical chains)
- Their inference is a kind of funny middle ground between logic and pattern matching
- But quite effective: 30% improvement
- *Q: When was the internal combustion engine invented?*
- *A: The first internal-combustion engine was built in 1867.*
- invent->create_mentally->create->build

Question Answering Example

- How hot does the inside of an active volcano get?
- get(TEMPERATURE, inside(volcano(active)))
- "lava fragments belched out of the mountain were as hot as 300 degrees Fahrenheit"
- fragments(lava, TEMPERATURE(degrees(300)),
  belched(out, mountain))
  - volcano ISA mountain
  - lava ISPARTOF volcano
  - lava inside volcano
  - fragments of lava HAVEPROPERTIESOF lava
- The needed semantic information is in WordNet definitions, and was successfully translated into a form that was used for rough 'proofs'

The Architecture of LCC's QA System around 2003



Definition Questions

- They asked about:
  - PEOPLE (most of them starting with "Who")
  - other types of NAMES
  - general concepts
- People questions
  - Many use the PERSON name in the format [First name, Last name]
    - examples: Aaron Copland, Allen Iverson, Albert Ghiorso
  - Some names had the PERSON name in format [First name, Last name1, Last name2]
    - example: Antonia Coello Novello
  - Other names had the name as a single word → very well known person
    - examples: Nostradamus, Absalom, Abraham
  - Some questions referred to names of kings or princes:
    - examples: Vlad the Impaler, Akbar the Great

Answering definition questions

- Most QA systems use between 30-60 patterns
- The most popular patterns:

Id	Pattern	Freq.	Usage	Question
25	person-hyponym QP	0.43%	The doctors also consult with former Italian Olympic skier Alberto Tomba, along with other Italian athletes	1907: Who is Alberto Tomba?
9	QP, the AP	0.28%	Bausch Lomb, the company that sells contact lenses, among hundreds of other optical products, has come up with a new twist on the computer screen magnifier	1917: What is Bausch & Lomb?
11	QP, a AP	0.11%	ETA, a Basque language acronym for Basque Homeland and Freedom _ has killed nearly 800 people since taking up arms in 1968	1987: What is ETA in Spain?
13	QA, an AP	0.02%	The kidnappers claimed they are members of the Abu Sayaf, an extremist Muslim group, but a leader of the group denied that	2042: Who is Abu Sayaf?
21	AP such as QP	0.02%	For the hundreds of Albanian refugees undergoing medical tests and treatments at Fort Dix, the news is mostly good: Most are in reasonable good health, with little evidence of infectious diseases such as TB	2095: What is TB?

Example of Complex Question

How have thefts impacted on the safety of Russia's nuclear navy, and has the theft problem been increased or reduced over time?

Need of domain knowledge

To what degree do different thefts put nuclear or radioactive materials at risk?

Question decomposition

Definition questions:
- *What is meant by nuclear navy?*
- *What does 'impact' mean?*
- *How does one define the increase or decrease of a problem?*

Factoid questions:
- *What is the number of thefts that are likely to be reported?*
- *What sort of items have been stolen?*

Alternative questions:
- *What is meant by Russia? Only Russia, or also former Soviet facilities in non-Russian republics?*

Complex questions

- Characterized by the need of domain knowledge
- There is no single answer type that can be identified, but rather an answer structure needs to be recognized
- Answer selection becomes more complicated, since inference based on the semantics of the answer type needs to be activated
- Complex questions need to be decomposed into a set of simpler questions

Future Directions

- Basic Tools: Parsers, Role-Assigners, Taggers
  - Need to be much faster, more accurate
  - How to make inroads into remaining errors?
- Machine Translation
  - Making effective use of syntax, semantics
  - Methods which don't require parallel texts
  - Efficient algorithms
- Semantic Processing for QA, Summarization
  - Taking a set of sentences, inferring facts from them
  - Probabilistic frameworks, learning methods
- Tasks waiting for new methods:
  - Discourse / Dialog Models
  - ASR and Speech Synthesis
- Unsupervised / Adaptive Methods
- Post Dynamic Programming Methods
  - Understanding Pipelines, Bidirectional Pipelining