QA (Part II)

Hal Daumé III

School of Computing
University of Utah

me@hal3.name

THE UNIVERSITY OF UTAH

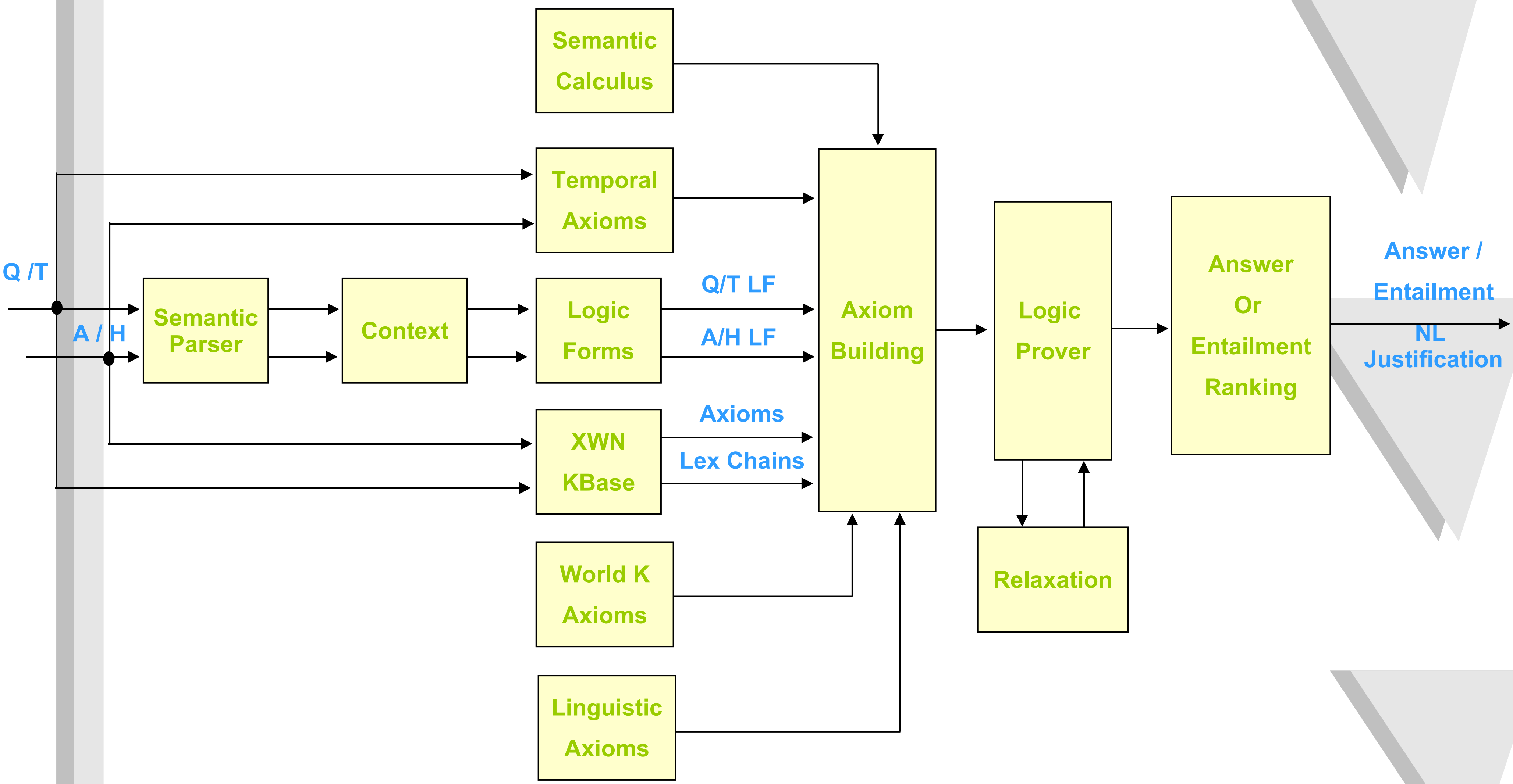
Approaches of Specific Systems

- LCC
- USC-ISI
- Microsoft
- IBM

LCC

- Moldovan & Rus, 2001
- Uses Logic Prover for answer justification
  - Question logical form
  - Candidate answers in logical form
  - XWN glosses
  - Linguistic axioms
  - Lexical chains
- Inference engine attempts to verify answer by negating question and proving a contradiction
- If proof fails, predicates in question are gradually relaxed until proof succeeds or associated proof score is below a threshold.

Semantically Enhanced COGEX



Output of Semantic Parser

Question: What is the Muslim Brotherhood's goal?

The output of the semantic parser:

PURPOSE(x, Muslim Brotherhood)

Answer: The Muslim Brotherhood, Egypt's biggest fundamentalist group established in 1928, advocates turning Egypt into a strict Muslim state by political means, setting itself apart from militant groups that took up arms in 1992.

The output of the semantic parser:

AGENT(Muslim Brotherhood, advocate)

PURPOSE(turning Egypt into a strict Muslim state, advocate)

TEMPORAL(1928, establish)

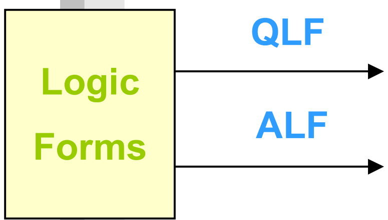
TEMPORAL(1992, took up arms)

PROPERTY(strict, Muslim state)

MEANS(political means, turning Egypt into a strict Muslim state)

SYNONYMY(Muslim Brotherhood, Egypt's biggest fundamentalist group)

Generation of Logical Forms



Question: What is the Muslim Brotherhood's goal?

Question Logical Form (QLF):

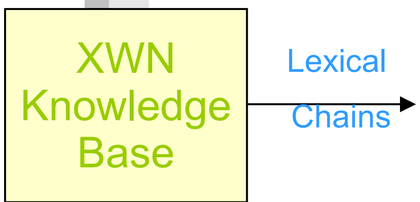
(exists x0 x1 x2 x3 (Muslim_NN(x0) & Brotherhood_NN(x1) & nn_NNC(x2,x0,x1) & PURPOSE_SR(x3,x2))).

Answer: The Muslim Brotherhood, Egypt's biggest fundamentalist group established in 1928, advocates turning Egypt into a strict Muslim state by political means, setting itself apart from militant groups that took up arms in 1992.

Answer Logical Form (AFL):

(exists e1 e2 e3 e4 e5 e6 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 (Muslim_NN(x1) & Brotherhood_NN(x2) & nn_NNC(x3,x1,x2) & Egypt_NN(x4) & _s_POS(x5,x4) & biggest_JJ(x5) & fundamentalist_JJ(x5) & group_NN(x5) & SYNONYMY_SR(x3,x5) & establish_VB(e1,x20,x5) & in_IN(e1,x6) & 1928_CD(x6) & TEMPORAL_SR(x6,e1) & advocate_VB(e2,x5,x21) & AGENT_SR(x5,e2) & PURPOSE_SR(e3,e2) & turn_VB(e3,x5,x7) & Egypt_NN(x7) & into_IN(e3,x8) & strict_JJ(x15,x14) & Muslim_NN(x8) & state_NN(x13) & nn_NNC(x14,x8,x13) & PROPERTY_SR(x15,x14) & by_IN(e3,x9) & political_JJ(x9) & means_NN(x9) & MEANS_SR(x9,e3) & set_VB(e5,x5,x5) & itself_PRP(x5) & apart_RB(e5) & from_IN(e5,x10) & militant_JJ(x10) & group_NN(x10) & take_VB(e6,x10,x12) & up_IN(e6,x11) & arms_NN(x11) & in_IN(e6,x12) & 1992_CD(x12) & TEMPORAL_SR(x12,e6)).

Lexical Chains from XWN



- **Lexical chains**
  - Lexical Chains establish connections between semantically related concepts, i.e. WordNet synsets.

    (note concepts, not words which means Word Sense Disambiguation is necessary)
  - Concepts and relations along the lexical chain explain the semantic connectivity of the end concepts
  - Lexical chains start by using WordNet relations (ISA, Part-Whole) and gloss co-occurrence (weak relation)
  - XWN Knowledge Base then adds more meaningful (precise) relations
    - "Tennis → a game played with rackets by two or four players…"
    - Prior to XWN-KB: 'tennis' → 'two or four' (gloss co-occurrence)
    - With XWN-KB: 'tennis' → 'game' ← 'play' → 'player' → 'two or four'

ISA **THM** **AGT** **MEA**

Examples of Lexical Chains

Question: How were biological agents acquired by bin Laden?

Answer: On 8 July 1998 , the Italian newspaper Corriere della Serra indicated that members of The World Front for Fighting Jews and Crusaders , which was founded by Bin Laden , purchased three chemical and biological_agent production facilities in

Lexical Chain: (V - buy#1, purchase#1) – HYPERNYM → (V - get#1, acquire#1)

Question: How did Adolf Hitler die?

Answer: ... Adolf Hitler committed suicide ...

Lexical Chain: (N - suicide#1, self-destruction#1, self-annihilation#1) – GLOSS → (V - kill#1) – GLOSS → (V - die#1, decease#1, perish#1, go#17, exit#3, pass_away#1, expire#2, pass#25)

Propagating syntactic structures along the chain

The goal is to filter out unacceptable chains, and to improve the ranking of chains when multiple chains can be established

Example 1:

AGENT

Q: Who did Floyd Patterson beat to win the title?

PATIENT

WA: He saw Ingemar Johanson knock down Floyd Patterson seven times there in winning the title.

V - beat#2 – entail → V - hit#4 – derivation → N - hitting#1,striking#2 – derivation → V - strike#2 – hyponym → V - knock-down#2

Example 2:

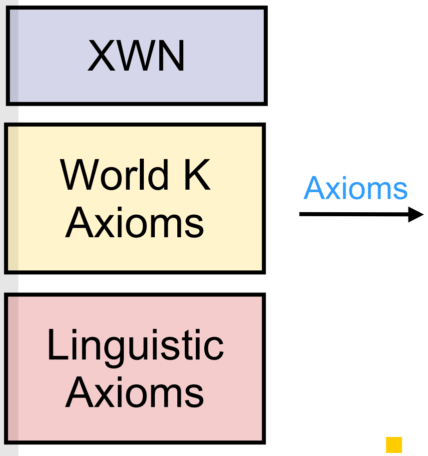
	AGENT		THEME	MEASURE
S1: John	bought	a cowboy hat	for $50.	
	AGENT	MEASURE		THEME
S2: John	paid	$50	for	a cowboy hat.

V - buy#1 – entail → V - pay#1

Axioms on Demand (1/3)

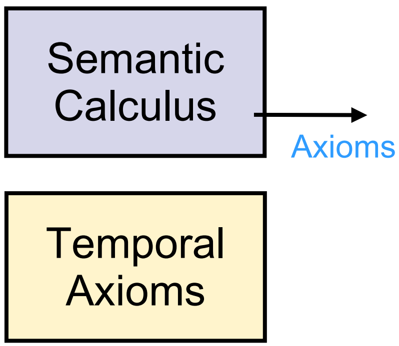
- Extract world knowledge, in the form of axioms, from text or other resources automatically and "on demand"
  - When the logic prover runs out of rules to use, it can request one from external knowledge sources
    - Will ask for a rule connecting two concepts
  - Generate axioms on the fly from multiple knowledge sources
    - WordNet and eXtended WordNet: glosses and lexical chains
    - Instantiation of NLP rules
    - Open text from a trusted source (dictionary, encyclopedia, textbook on a relevant topic, etc.)
    - An automatically-built knowledge base

Axioms on Demand (2/3)



- eXtended WordNet axiom generator
  - Question: What all can a 'player' do?
    - Look at all contexts with 'player' as AGT
    - Gloss of 'tennis': a 'player' can 'hit' (a ball), 'play' (a game)
    - Gloss of 'squash': A 'player' can 'strike' (a ball), etc
  - Connect related-concepts
    - kidnap_VB(e_1,x_1,x_2) -> kidnapper_NN(x_1)
    - (asian_JJ(x1,x2) $\rightarrow$ asia_NN(x1) & _continent_NE(x1))
- World Knowledge axioms
  - WordNet glosses
  - jungle_cat_NN(x_1) -> small_JJ(x_2,x_1) & Asiatic_JJ(x_3,x_1) & wildcat_NN(x_1)
- NLP axioms
  - Linguistic rewriting rules
  - Gilda_NN(x_1) & Flores_NN(x_2) & nn_NNC(x_3,x_1,x_2) -> Flores_NN(x_3)

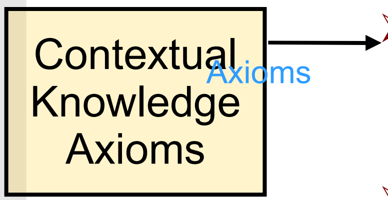
Axioms on Demand (3/3)



- Semantic Relation Calculus
  - Combine two or more local semantic relations to establish broader semantic relations
  - Increase the semantic connectivity
  - Mike is a rich man → Mike is rich
    - ISA_SR(Mike,man) & PAH_SR(man,rich) →PAH_SR(Mike,rich)
  - John lives in Dallas, Texas → John lives in Texas.
    - LOC(John,Dallas) & PW(Dallas,Texas) -> LOC(John, Texas)

- Temporal Axioms
  - Time Transitivity of Events
    - during_CTMP(e_1,e_2) & during_CTMP(e_2,e_3) →during_CTMP(e_1,e_3)
  - Dates entail more general times
    - October 2000 → year 2000

Contextual Knowledge Axioms



Examples

- If someone boards a plane and the flight takes 3 hours, then that person travels for 3 hours
- The person leaves at the same time and arrives at the same time with the traveling plane
- If the departure of a vehicle has a destination and the vehicle arrives at the destination then the arrival is located at the destination
- If something is exactly located somewhere, then nothing else is exactly located in the same place
- If a Process is located in an area, then all sub Processes of the Process are located in the same area

Logic Prover (1/2)

- A first order logic **resolution style** theorem prover
- Inference rule sets are based on **hyperresolution** and **paramodulation**
- Transform the two text fragments into 4-layered **logic forms** based upon LCC's Syntactic, Semantic, Contextual and Event Processing and Analysis
- Automatically create **"Axioms on Demand"** to be used during the proof
  - Lexical Chains axioms
  - World Knowledge axioms
  - Linguistic transformation axioms
  - Contextual / Temporal axioms

Logic Prover (2/2)

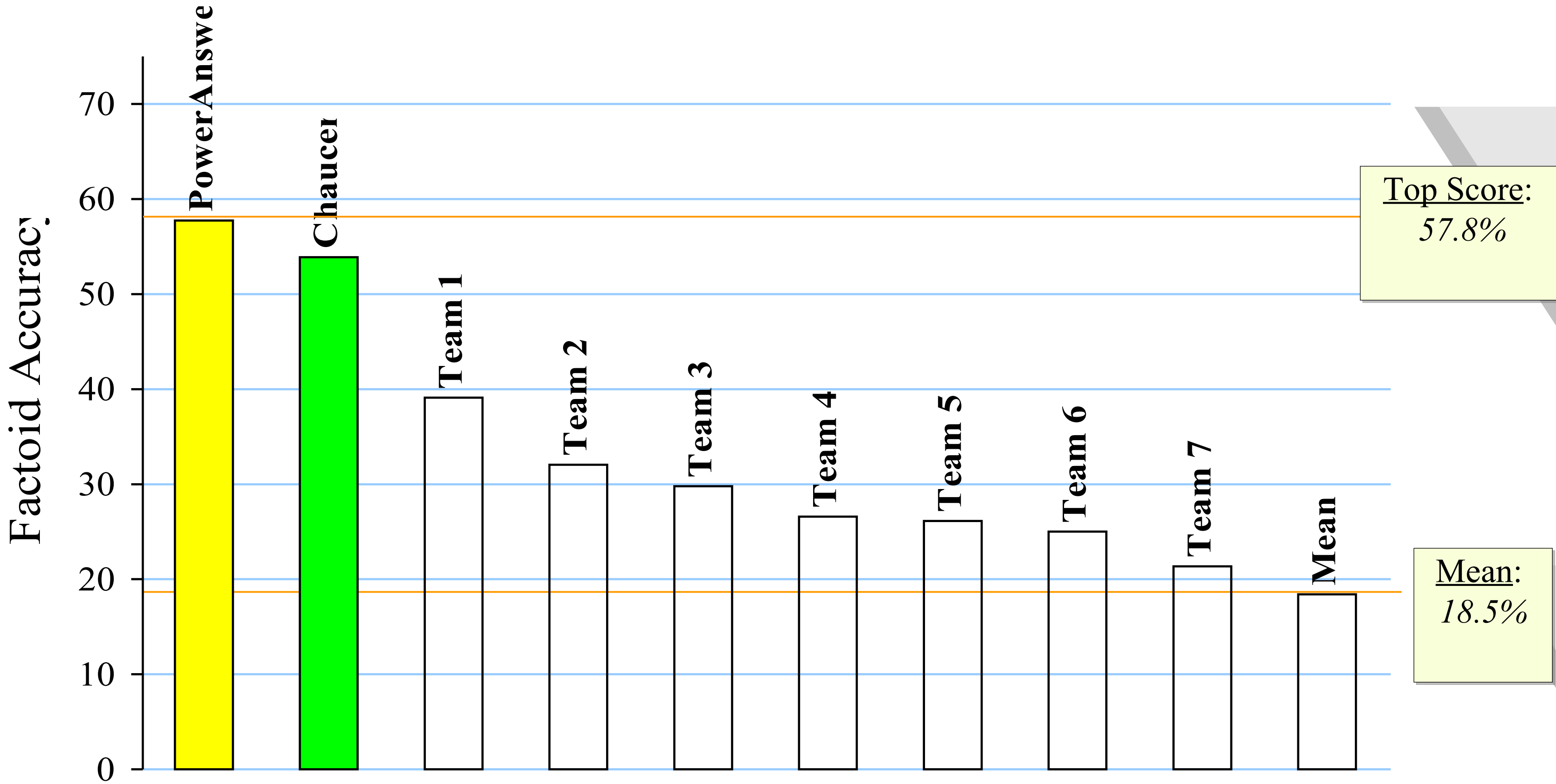
- Load COGEX's **SOS (Set of Support)** with Candidate Answer Passage(s) A and ¬Question Q and its **USABLE** list of clauses with the generated axioms, semantic and temporal axioms
- Search for a proof by iteratively removing clauses from **SOS** and searching the **USABLE** for possible inferences until a refutation is found
  - If no contradiction is detected
    - Relax arguments
    - Drop entire predicates from H
- Compute **"Proof Score"** for each Candidate
- Select best **Result** & Generate **NL Justification**

LCC: Logic Prover

- Question
  - *Which company created the Internet Browser Mosaic?*
  - QLF: (_organization_AT(x2)) & company_NN(x2) & create_VB(e1,x2,x6) & Internet_NN(x3) & browser_NN(x4) & Mosaic_NN(x5) & nn_NNC(x6,x3,x4,x5)
- Answer passage
  - *... Mosaic , developed by the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana - Champaign ...*
  - ALF: ... Mosaic_NN(x2) & develop_VB(e2,x2,x31) & by_IN(e2,x8) & National_NN(x3) & Center_NN(x4) & for_NN(x5) & Supercomputing_NN(x6) & application_NN(x7) & nn_NNC(x8,x3,x4,x5,x6,x7) & NCSA_NN(x9) & at_IN(e2,x15) & University_NN(x10) & of_NN(x11) & Illinois_NN(x12) & at_NN(x13) & Urbana_NN(x14) & nn_NNC(x15,x10,x11,x12,x13,x14) & Champaign_NN(x16) ...
- Lexical Chains develop <-> make and make <->create
  - exists x2 x3 x4 all e2 x1 x7 (develop_vb(e2,x7,x1) <-> make_vb(e2,x7,x1) & something_nn(x1) & new_jj(x1) & such_jj(x1) & product_nn(x2) & or_cc(x4,x1,x3) & mental_jj(x3) & artistic_jj(x3) & creation_nn(x3)).
  - all e1 x1 x2 (make_vb(e1,x1,x2) <-> create_vb(e1,x1,x2) & manufacture_vb(e1,x1,x2) & man-made_jj(x2) & product_nn(x2)).
- Linguistic axioms
  - all x0 (mosaic_nn(x0) -> internet_nn(x0) & browser_nn(x0))

Evaluations: QA (TREC-06)

- LCC's **PowerAnswer** Question Answering (QA) system finished 1st on Factoid Questions and Overall Combined Score. A second LCC QA system, **Chaucer,** finished 2nd in both categories in the TREC QA 2006 evaluation.
- An LCC QA system has finished 1st every year that the TREC QA Evaluation has been conducted (Annually since TREC-8 in 1999)



USC-ISI

- Textmap system
  - Ravichandran and Hovy, 2002
  - Hermjakob et al. 2003
- Use of Surface Text Patterns
- When was X born ->
  - Mozart was born in 1756
  - Gandhi (1869-1948)

  Can be captured in expressions
  - <NAME> was born in <BIRTHDATE>
  - <NAME> (<BIRTHDATE> -
- These patterns can be learned

USC-ISI TextMap

- Use bootstrapping to learn patterns.
- For an identified question type ("When was X born?"), start with known answers for some values of X
  - Mozart 1756
  - Gandhi 1869
  - Newton 1642
- Issue Web search engine queries (e.g. "+Mozart +1756")
- Collect top 1000 documents
- Filter, tokenize, smooth etc.
- Use suffix tree constructor to find best substrings, e.g.
  - Mozart (1756-1791)
- Filter
  - Mozart (1756-
- Replace query strings with e.g. <NAME> and <ANSWER>

- Determine precision of each pattern
  - Find documents with just question term (Mozart)
  - Apply patterns and calculate precision

USC-ISI TextMap

- Finding Answers
  - Determine Question type
  - Perform IR Query
  - Do sentence segmentation and smoothing
  - Replace question term by question tag
    - i.e. replace Mozart with <NAME>
  - Search for instances of patterns associated with question type
  - Select words matching <ANSWER>
  - Assign scores according to precision of pattern

Microsoft

- Data-Intensive QA. Brill et al. 2002
- "Overcoming the surface string mismatch between the question formulation and the string containing the answer"
- Approach based on the assumption/intuition that someone on the Web has answered the question in the same way it was asked.
- Want to avoid dealing with:
  - Lexical, syntactic, semantic relationships (bet. Q & A)
  - Anaphora resolution
  - Synonymy
  - Alternate syntax
  - Indirect answers
- Take advantage of redundancy on Web, then project to TREC corpus (Answer-based QA)

Microsoft AskMSR

- Formulate multiple queries – each rewrite has intrinsic score. E.g. for *"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]
- Get top 100 documents from Google
- Extract n-grams from document summaries
- Score n-grams by summing the scores of the rewrites it came from
- Use tiling to merge n-grams
- Search for supporting documents in TREC corpus

Microsoft AskMSR

- Question is: "What is the rainiest place on Earth"
- Answer from Web is: "Mount Waialeale"
- Passage in TREC corpus is: "… In misty Seattle, Wash., last year, 32 inches of rain fell. Hong Kong gets about 80 inches a year, and even Pago Pago, noted for its prodigious showers, gets only about 196 inches annually. (The titleholder, according to the National Geographic Society, is Mount Waialeale in Hawaii, where about 460 inches of rain falls each year.) …"
- Very difficult to imagine getting this passage by other means

IBM Statistical QA (Ittycheriah, 2001)

$$p(c|q,a) = \Sigma_e\, p(c,e|q,a)$$
$$= \Sigma_e\, p(c|e,q,a)\; p(e|q,a)$$

q = question
a = answer
c = "correctness"
e = answer type

- ATM predicts, from the question and a proposed answer, the answer type they both satisfy
- Given a question, an answer, and the predicted answer type, ASM seeks to model the correctness of this configuration.
- Distributions are modelled using a maximum entropy formulation

- Training data = human judgments
  - For ATM, 13K questions annotated with 31 categories
  - For ASM, ~ 5K questions from TREC plus trivia

IBM Statistical QA (Ittycheriah)

- Question Analysis (by ATM)
  - Selects one out of 31 categories
- Search
  - Question expanded by Local Context Analysis
  - Top 1000 documents retrieved
- Passage Extraction: Top 100 passages that:
  - Maximize question word match
  - Have desired answer type
  - Minimize dispersion of question words
  - Have similar syntactic structure to question
- Answer Extraction:
  - Candidate answers ranked using ASM