Breaking through the syntax barrier: Searching with entities and relations

Soumen Chakrabarti
IIT Bombay
www.cse.iitb.ac.in/~soumen

Wish upon a textbox, 1996

Wish upon a textbox, 1998

"A rising tide of data lifts all algorithms"

Wish upon a textbox, post-IPO

- Indexing ~~4,285,199,774~~ 8,058,044,651 pages
- Same interface, therefore same 2-word queries
- Mind-reading wizard-in-black-box saves the day

If music had been invented ten
years ago along with the Web,
we would all be playing
one-string instruments
(and not making great music).

Udi Manber, A9.com
Plenary speech
WWW 2004

Telegraphic queries, music not great

- Which produces better responses?
  - Opera fails to connect to secure IMAP tunneled through SSH
  - opera connect imap ssh tunnel

configuring an application to connect to a ... work required by the maintenance opera-tions ... servers Business application protection Secure remote administration ...

I load the signed applet it can still not connect to any ... simple local tunnels, such as to use imap, smtp etc ... to run … an applet in Opera

- Unable to express many details of information need
  - Opera the email client, not a kind of music
  - The problem is with Opera, not ssh, imap, applet
  - “Secure” is an attribute of imap, but may not juxtapose

Why telegraphic queries fail

- Information need relates to entities and relationships in the real world
- But the search engine gets only strings
- Risk over-/under- specified queries
  - Never know true recall
  - No time to deal with poor precision
- Query word distribution dramatically different from corpus distribution
  - Query is inherently incomplete
  - Fix some known info, look for unknown info

Past the syntax barrier: early steps

1. Taking the question apart
   - Question has known parts and unknown “slots”
   - Query-dependent information extraction (IE)
2. Searching entity-relationship graphs
   - Identify (personalized) information networks from semi-structured textual content
   - Enable “mildly-typed” query languages
3. Compiling basic relations from the Web
   - is-instance-of (is-a), is-subclass-of
   - is-part-of, has-attribute

Working harder on the question

Atypes and ground constants

- Specialize given domain to a token related to ground constants in the query
  - What animal is Winnie the Pooh?
    - instance-of("animal") NEAR "Winnie the Pooh"
  - When was television invented?
    - instance-of("time") NEAR "television" NEAR synonym("invented")
- FIND x NEAR GroundConstants(question)
  WHERE x IS-A Atype(question)
  - Ground constants: Winnie the Pooh, television
  - Atypes: animal, time

Taking the question apart

- Atype: the type of the entity that is an answer to the question
- Problem: don't want to compile a classification hierarchy of entities
  - Laborious, can't keep up
  - Offline rather than question-driven
- Instead
  - Identify spans of question as "atype informers"
  - Set up a very large basis of features
  - "Project" question and corpus to basis

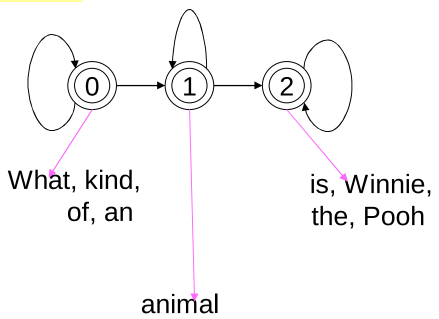
Marking atype informer spans

In which ocean did the *Titanic* sink?

How much RAM can the X40 Thinkpad support?

What is Kofi Annan's son's profession?

Idea: Set up as a sequence-labeling problem, with features extracted from a shallow parse

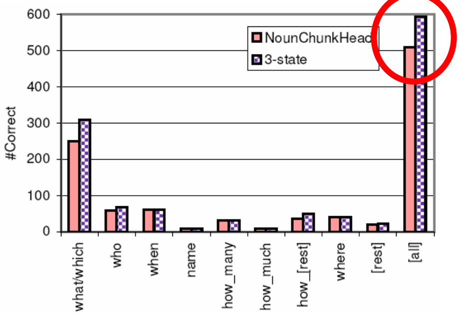


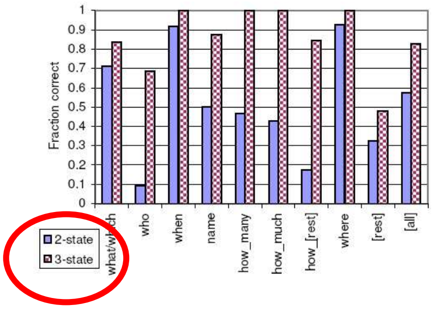
Pre | Informer span | Post

What kind of an animal is Winnie the Pooh?

Atype extraction results

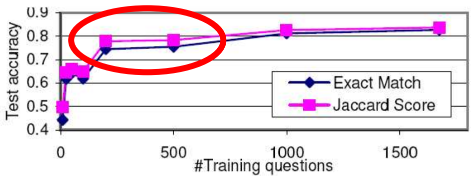
Machine learning approach 14% better at identifying informer span than hand-coded rule-base





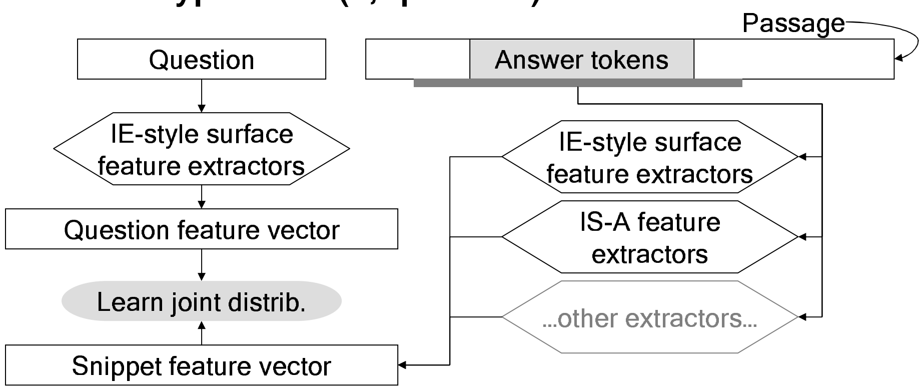
Careful design of state transition model is critical

A few hundred questions are enough to train the system to over 80% accuracy



Scoring tokens for correct Atypes

- FIND x “NEAR” GroundConstants(question)
  WHERE x IS-A **Atype(question)**
- No fixed question or answer type system
- Convert “x IS-A Atype(question)” to a soft match **DoesAtypeMatch(x, question)**



Features for Atype matching

- Question features: 1, 2, 3-token sequences starting with standard wh-words
  - where, when, who, how_X, …
- Passage surface features: hasCap, hasXx, isAbbrev, hasDigit, isAllDigit, lpos, rpos,…
- Passage IS-A features: all generalizations of all noun senses of token
  - Use WordNet: horse→equid→ungulate, hoofed mammal→placental mammal→animal…→entity
  - These are node IDs (“synsets”) in WordNet, not strings

Learning q–a feature connections

- Surface and WordNet features complement each other
- General concepts get *negative* params: use in predictive annotation
- Learning is symmetric (Q↔A)

E. how_far	
entity#n#1	-0.007
object#noun#1	-0.006
hasCap	-0.005
hasXxx	-0.004
measure#n#3	0.01
linear_unit#n#1	0.02
linear_measure#n#1	0.02
hasDigit	0.02
nautical_mile#n#2	0.02

F. linear_unit#n#1	
what	-0.007
how_many	-0.005
what_city	-0.004
when	-0.004
whom	-0.002
how_long	0.003
what_speed	0.005
where_is	0.009
how_far	0.02

G. location#n#1	
who	-0.178
name	-0.113
when	-0.043
how	-0.0314
year	-0.0230
what_tourist	0.004
what_state	0.012
country	0.015
province	0.029
city	0.109
where	0.249

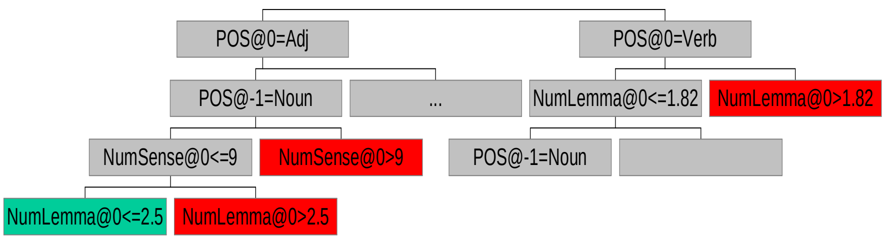
H. hasDigit	
who	-0.21
where	-0.10
name	-0.09
city	-0.05
company	-0.03
how_far	0.02
how_hot	0.02
which_date	0.05
how_much	0.09
how_many	0.16
what_year	.18
when	0.65

Taking the question apart

✓ Atype: the type of the entity that is an answer to the question

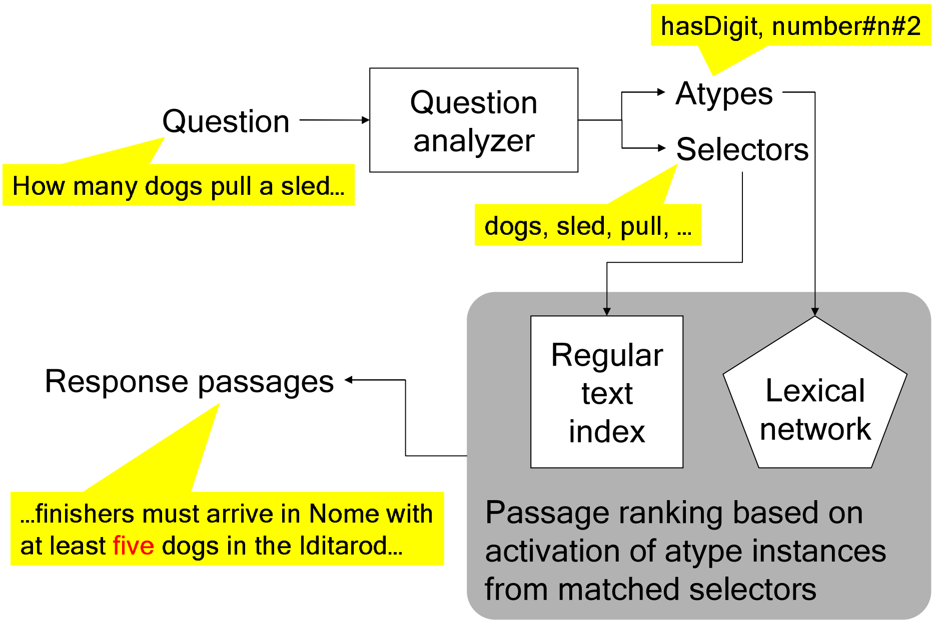
- Ground constants: Which question words are likely to appear (almost) unchanged in an answer passage?
- Arises in Web search sessions too
  - Opera login fails
  - problem with login Opera email
  - Opera login accept password
  - Opera account authentication
  - …

Spotting ground constants: sample result



- @-1, @0, @+1…features at token positions
- NumSense: how many WordNet senses does the word have?
- NumLemma: how many other words describe the same concept?
- F1 score: 71–73% with local features, 81% with local and global (NumSense, NumLemma)

Overall search and ranking architecture



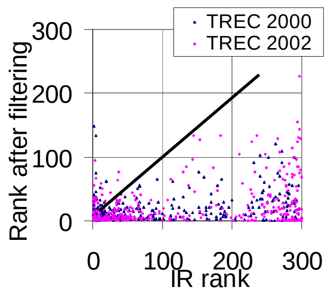
A9/Amazon Chakrabarti 2005 19

Evaluation: Mean reciprocal rank (MRR)

- n_q = smallest rank among answer passages
- MRR = $(1/|Q|)\ \Sigma_{q\in Q}(1/n_q)$
  - Dropping passage from #1 to #2 as bad as dropping it from #2 to not reporting it at all

Experiment setup:

- 300 top IR score passages
- If Pr(Y=1|token) < threshold reject token
- If tokens rejected reject passage
- Points below diagonal are good



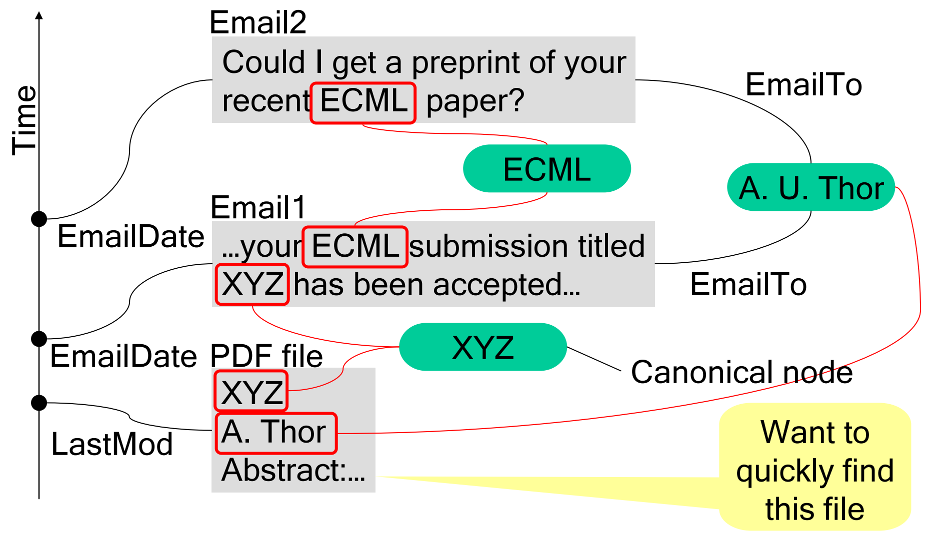
A9/Amazon Chakrabarti 2005 20

SPIN: Searching Personal Information Networks

The Web within

- Personal/desktop search: the first step
  - Corpus = email, files, contacts
  - Anachronism given Web search history
- The second step: searching with entities and relations (people, organizations, papers, time, works-for, wrote-email, advised, …)
  - Need to exploiting clean, non-adversarial data
  - Expose search with fine-grained structure
  - Exploit entity and relation types when possible
  - …without burdening user with schema-enforcing query languages

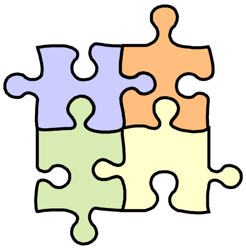
Benefits of connectionist search



More example scenarios

- Student "Ravi" graduated two years ago, is looking for industry jobs
  - type=person NEAR person=Ravi org=.com
  - Connections: person…paper…person…org
- This paper is suited for which conference?
  - type=conference NEAR paper=[uploaded file]
  - Connections: text…old papers…conference or text…citations…authors…committees…conference
- Given a list of accepted papers, locate and watch Web pages where they might appear

Compiling fragments of soft schema



Extracting is-instance-of info

- Which researcher built the WHIRL system?
  - WordNet may not know Cohen **IS-A** researcher
- Google has over 8 billion pages
  - “william cohen” on 86100 (p_1=86.1k/4.2B)
  - researcher on 4.55M (p_2=4.55M/4.2B)
  - +researcher +"william cohen“ on 1730: 18.55x more frequent than expected if independent
- Pointwise mutual information PMI
- Can add high-precision, low-recall patterns
  - “cities such as New York” (26600 hits)
  - “professor Michael Jordan” (101 hits)

Bootstrapping lists of instances

- Hearst 1992, Brin 1997, Etzioni 2004
- A "propose-validate" approach
  - Using existing patterns, generate queries
  - For each web page *w* returned
    - Extract potential fact *e* and assign confidence score
    - Add fact to database if it has high enough score
- Example patterns
  - NP1 {,} {<u>such as</u>|<u>and other</u>|<u>including</u>} NPList2
  - NP1 <u>is a</u> NP2, NP1 <u>is the</u> NP2 <u>of</u> NP3
  - <u>the</u> NP1 <u>of</u> NP2 <u>is</u> NP3
- Start with NP1 = researcher etc.

System details

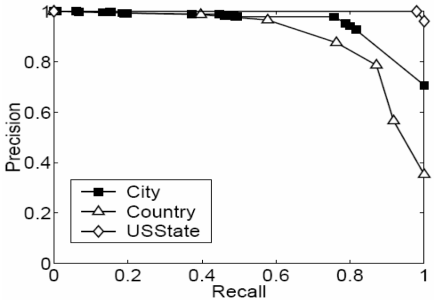
- The importance of shallow linguistics working together with statistical tests
  - China is a (**country**)$_{NP}$ in Asia
  - Garth Brooks is a (country$_{ADJ}$ (**singer**)$_{N}$)$_{NP}$

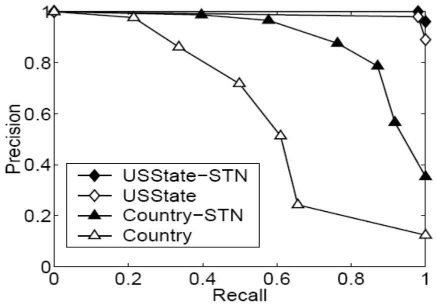
"Head" of phrase

- Unary relation example
  - NP1 <u>such as</u> NPList2 &
    head(NP1)=plural(name(Class1)) &
    properNoun(head(each(NPList2))) ⇒
    instanceOf(Class1, head(each(NPList2)))

Bootstrapping performance

- Recall-vs-precision exposes size and difficulty of domain
  - "US state" is easy
  - "Country" is difficult
- To improve signal-to-noise (STN) ratio, stop when confidence score is lower than threshold
  - Substantially improves recall-vs-precision





Concluding messages

- Work much harder on questions
  - Break down into what's known, what's not
  - Find fragments of structure when possible
  - Exploit user profiles and sessions
- Perform limited pre-structuring of corpus
  - Difficult to anticipate all needs and applications
  - Extract graph structure where possible (e.g. is-a)
  - Do not insist on specific schema
- Design indices and ranking strategies for matching strings and semantics annotations
  - "Tip of the iceberg" under very complex ranking functions