Harvesting and Storing Knowledge from the Web

Eduard Hovy

Information Sciences Institute
University of Southern California

USC INFORMATION SCIENCES INSTITUTE

The problem with automated QA...

Webclopedia
(Hovy et al. 01)

- Where do lobsters like to live?
  — *on the table*
- Where are zebras most likely found?
  — *in the dictionary*
- How many people live in Chile?
  — *nine*
- What is an invertebrate?
  — *Dukakis*

BUT: we could quite easily get this from the web!
…need a repository of knowledge plus
commonsense semantic, numerical info

USC INFORMATION SCIENCES INSTITUTE

Imagine…

The web is the world's knowledge store.

But it is disorganized, inconsistent, and constantly changing…

…and it's often hard to find things quickly and accurately.

WHAT IF you could ask your system to create a database of the knowledge you needed, overnight?

You'd need, at least:

- metadata creation
- query input / definition
- data item harvesting
- data relationship harvesting
- data verification
- data updating

USC INFORMATION SCIENCES INSTITUTE

Example applications 1: NLP

- **<u>How to improve accuracy of IR / web search?</u>**
  TREC 98–01: around 40%

★ Understand user query; expand query terms by meaning

- **<u>How to achieve conceptual summarization?</u>**
  Never been done yet, at non-toy level

★ Interpret topic, fuse concepts according to meaning; re-generate

- **<u>How to improve QA?</u>**
  TREC 99–02: around 65%

★ Understand Q and A; match their meanings; know common info

- **<u>How to improve MT quality?</u>**
  MTEval 94: ~70%, depending on what you measure

★ Disambiguate word senses to find correct meaning

Large standardized metadata collections

What is an ontology?
My def: a collection of terms denoting entities, events, and relationships in the domain, taxonomized and interrelated so as to express the sharing of properties. It's a formalized model of the domain, focusing on the aspects of interest for computation.

The need is there…
everybody's making lists:

- SIC and NAICS and other codes
- Yahoo!'s topic classification
- Semantic Web termbanks / ontologies

But how do you:

- Guarantee the freshness and accuracy of the list?
- Guarantee its completeness?
- Ensure commensurate detail in levels of the list?
- Cross-reference elements of the list?

Need automated procedure for creating lists / metadata / ontologies

Credo and methodology

Ontologies (and even concepts) are too complex to build all in one step…

…so build them bit by bit, testing each new (kind of) addition empirically…

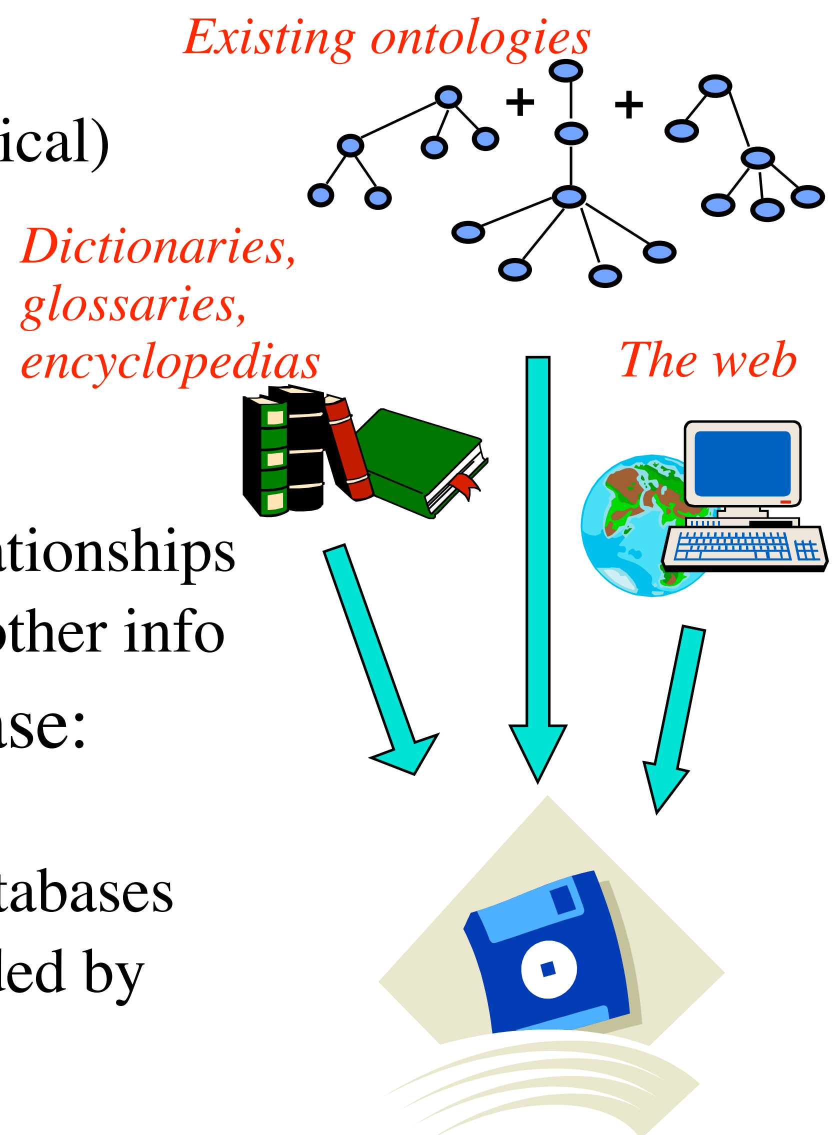
…and develop appropriate learning techniques for each bit, so you can automate the process…

…so next time (since there's no ultimate truth) you can build a new one more quickly

USC INFORMATION SCIENCES INSTITUTE

Plan: stepwise accretion of knowledge

- Initial framework:
  - Start with existing (terminological) ontologies as pre-metadata
  - Weave them together
- Build metadata/concepts:
  - Define/extract concept ‘cores’
  - Extract/learn inter-concept relationships
  - Extract/learn definitional and other info
- Build (large) data/instance base:
  - Extract instance ‘cores’
  - Link into ontology; store in databases
  - Extract more information, guided by parent concept



USC INFORMATION SCIENCES INSTITUTE

Talk overview

1. **Framework**: Ontology as metadata
    - Creating Omega: recent work on connecting ontologies
2. **Metadata terms (concepts)**:
    - Learning concepts by clustering
    - Learning additional relations from text
3. **Data (instances)**:
    - Harvesting seed instances from text
    - Harvesting additional information
4. **Verifying data**:
    - Determining opinions and other epistemic statuses
    - Updating
5. **Conclusion**

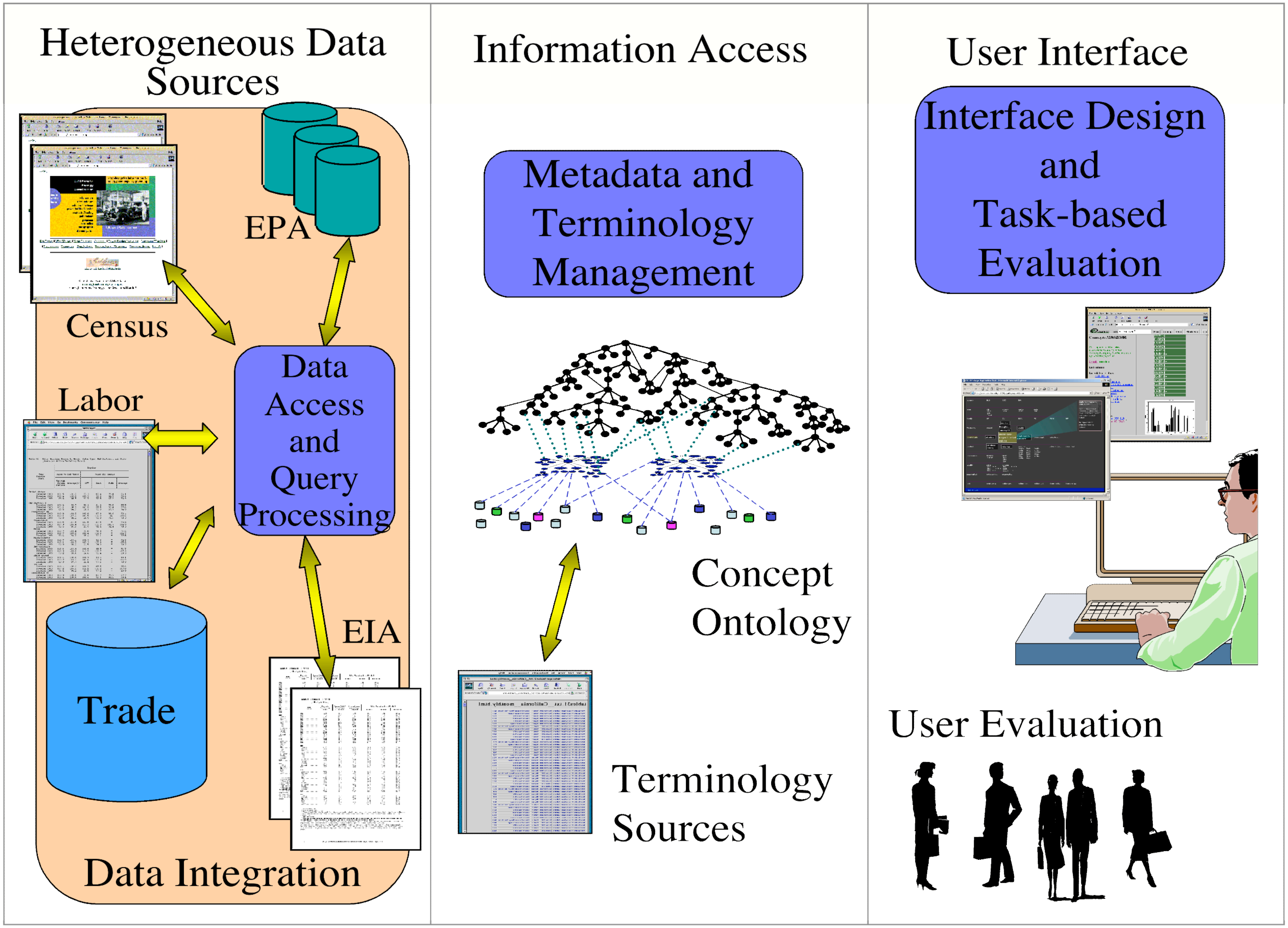
USC INFORMATION SCIENCES INSTITUTE

1. Framework
Ontology as metadata: semi-automated alignment and merging

(This work with Andrew Philpot, Michael Fleischman, and Jerry Hobbs)

USC INFORMATION SCIENCES INSTITUTE

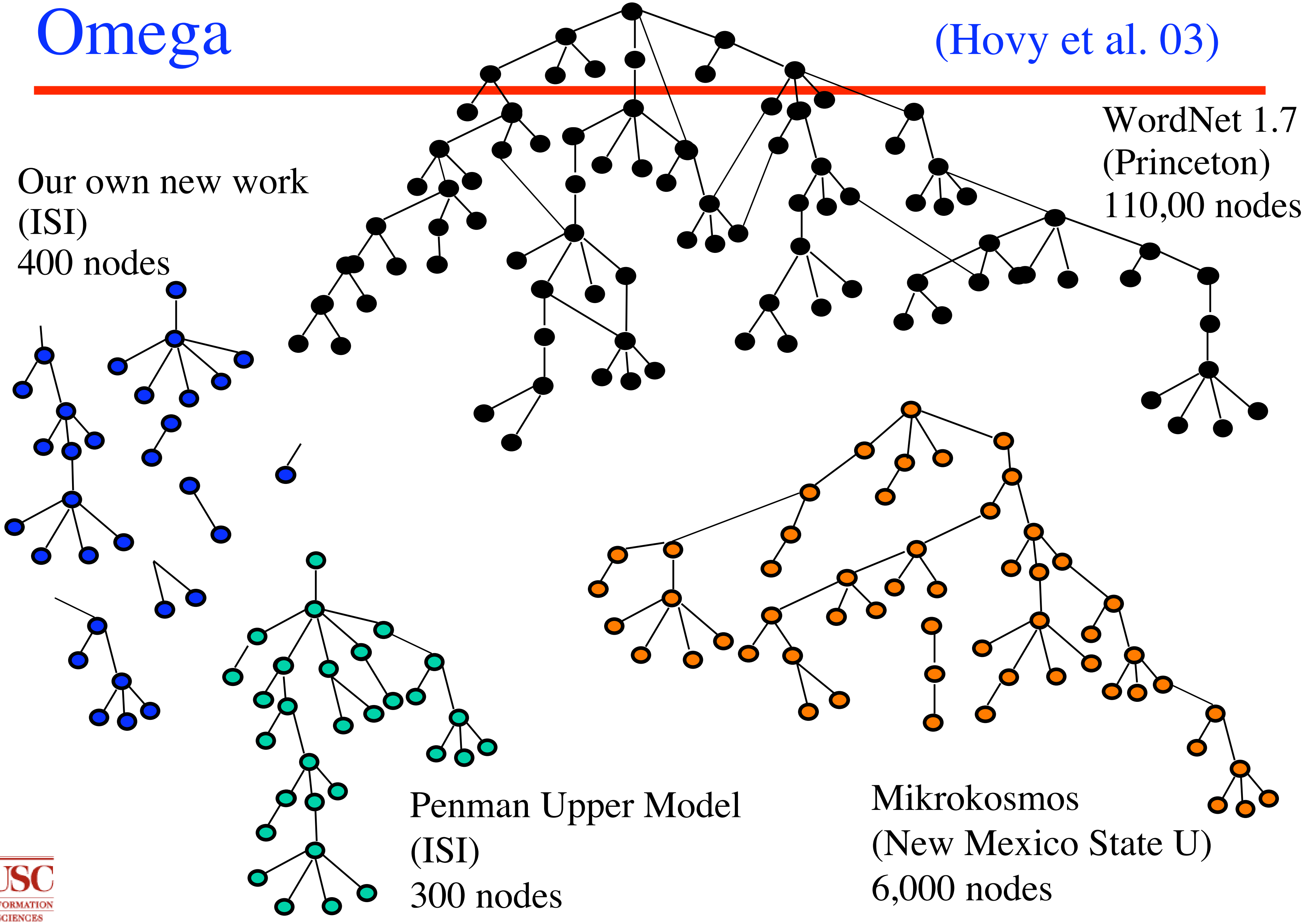
Example application: EDC (Hovy et al. 02)



USC INFORMATION SCIENCES INSTITUTE

Omega

(Hovy et al. 03)

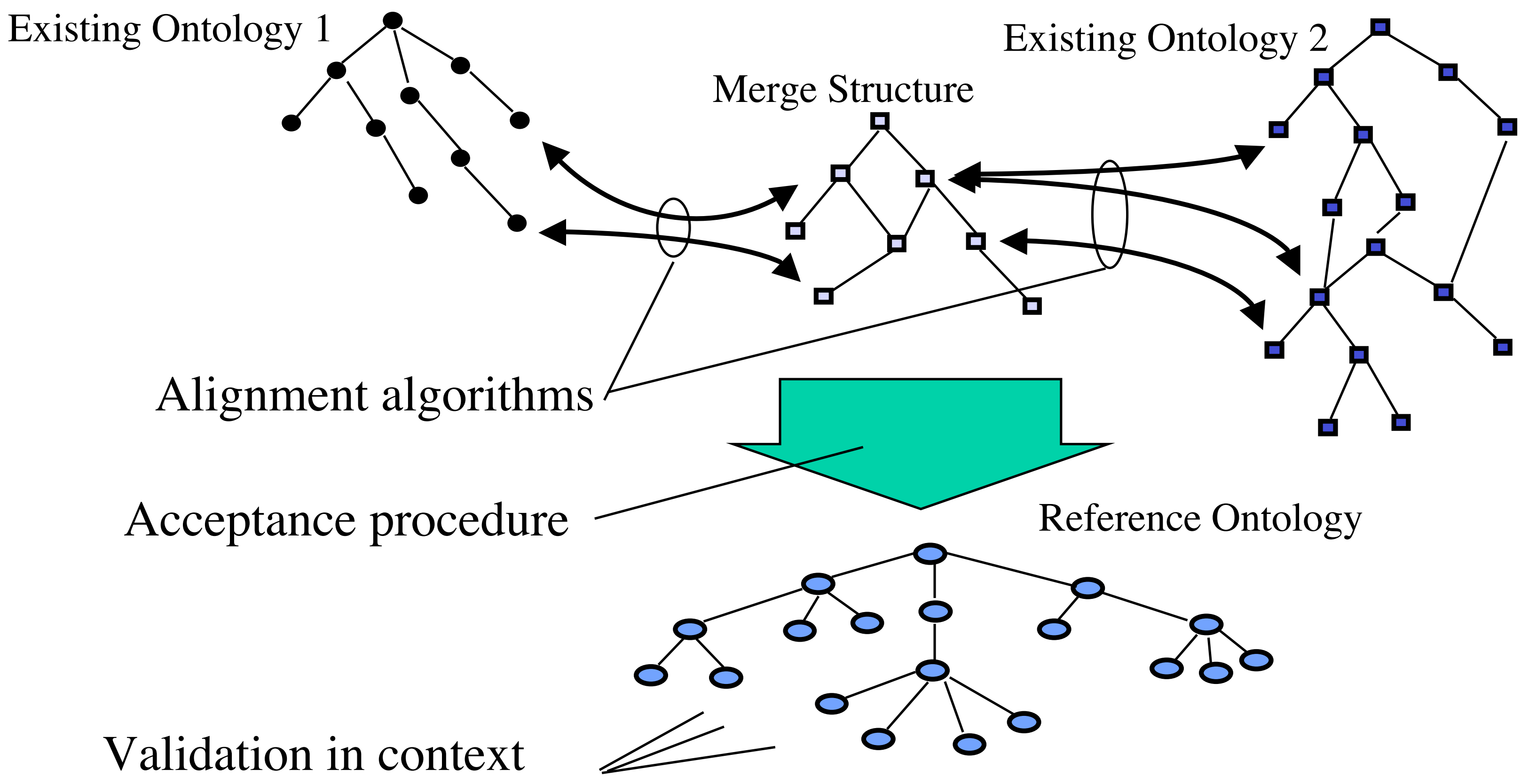


USC INFORMATION SCIENCES INSTITUTE

General alignment and merging problem

Goal: **find attachment point(s) in ontology** for node/term from somewhere else (ontology, website, metadata schema, etc.)

It's hard to do manually; very hard to do automatically — system needs to understand semantics of entities to be aligned



USC INFORMATION SCIENCES INSTITUTE

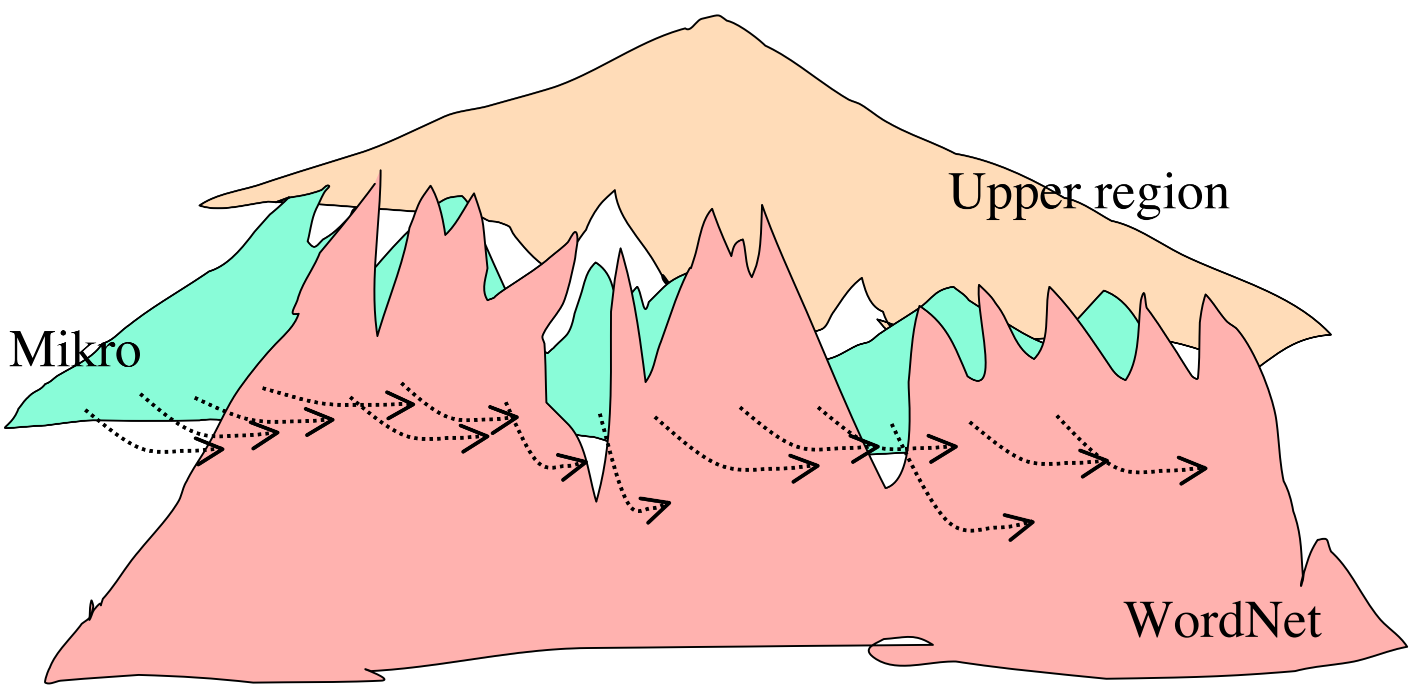
Ontology alignment and merging

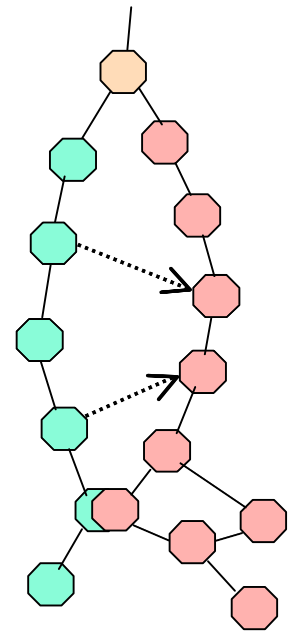
- Goal: find attachment point in ontology for node/term from somewhere else (ontology, website, metadata schema, etc.)
- Procedure:
  - 1. For a new term/concept, extract and format: name, definition, associated text, local taxonomy cluster, etc.
  - 2. apply alignment suggestion heuristics (NAME, DEFINITION, HIERARCHY, DISPERSAL match) against big ontology, to get proposed attachment points with strengths (Hovy 98) — test with numerous parameter combinations, see http://edc.isi.edu/alignment/ (Hovy et al. 01)
  - 3. automatically combine proposals (Fleischman et al 03)
  - 4. apply verification checks
  - 5. bless or reject proposals manually
- Process developed in early 1990s: (Agirre et al. 94; Knight & Luk 94; Okumura & Hovy 96; Hovy 98; Hovy et al. 01)
- Not stunningly accurate, but can speed up manual alignment markedly

USC INFORMATION SCIENCES INSTITUTE

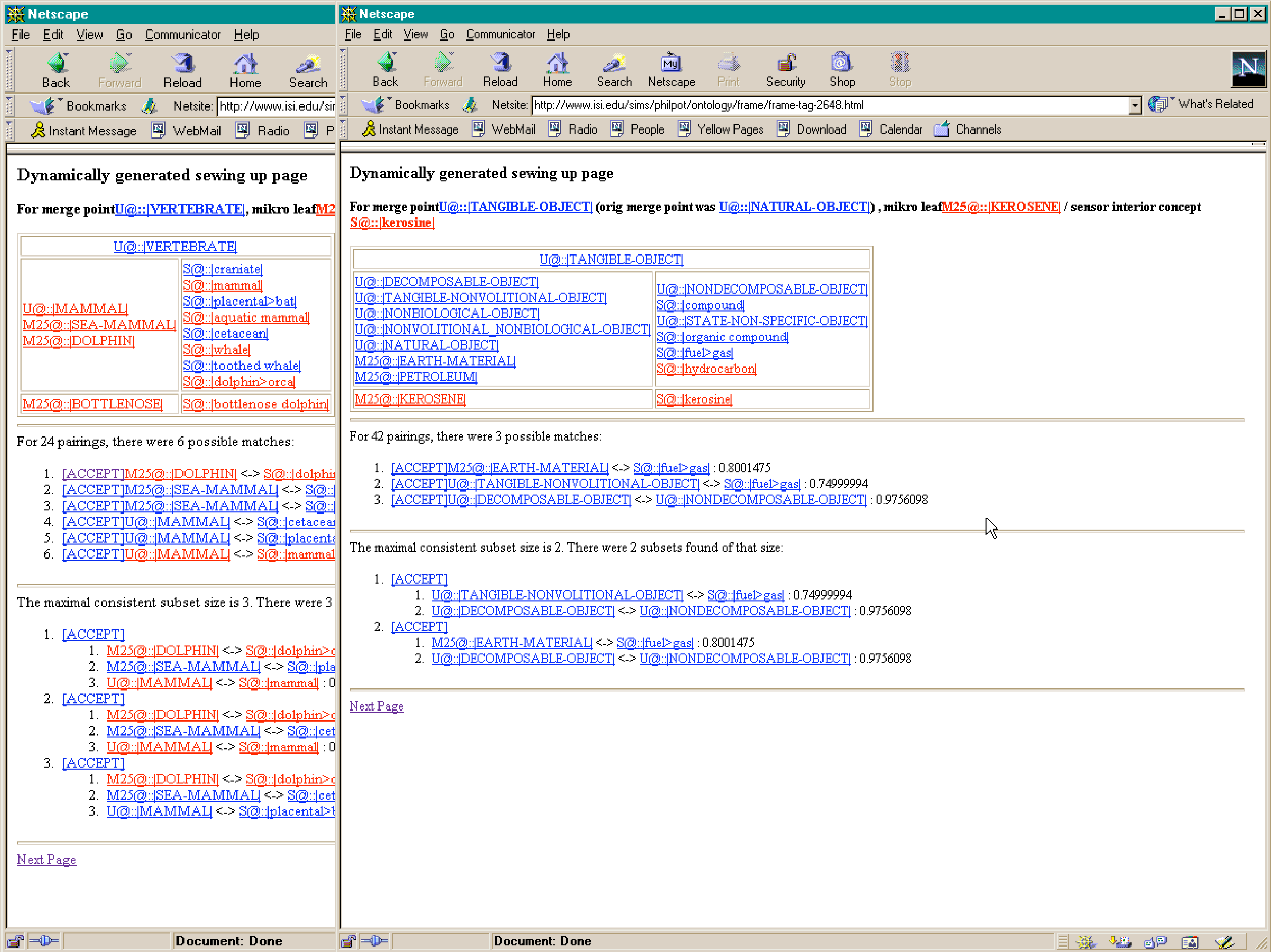
Alignment for Ωmega

- Created Upper Region (400 nodes) manually
- Manually snipped tops off Mikro and WordNet, then attached them to fringe of Upper Region
- Automatically aligned bottom fringe of Mikro into WordNet
- Automatically aligned sides of bubbles



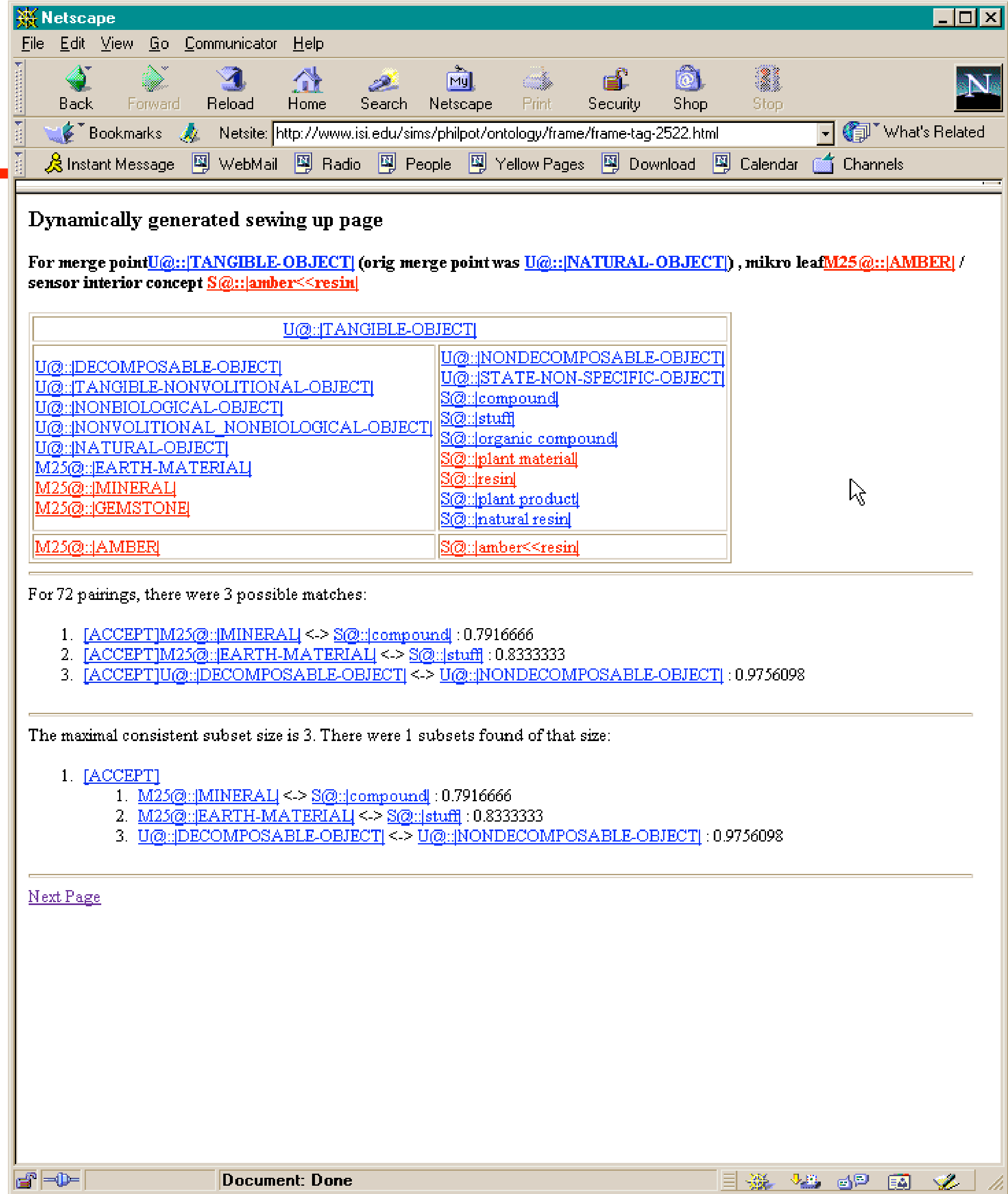


USC INFORMATION SCIENCES INSTITUTE


Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search
Bookmarks Netsite: http://www.isi.edu/si
Instant Message WebMail Radio P
Dynamically generated sewing up page
For merge point U@::|VERTEBRATE|, mikro leaf M2
U@::|VERTEBRATE|
U@::|MAMMAL|
M25@::|SEA-MAMMAL|
M25@::|DOLPHIN|
S@::|craniate|
S@::|mammal|
S@::|placental>bat|
S@::|aquatic mammal|
S@::|cetacean|
S@::|whale|
S@::|toothed whale|
S@::|dolphin>orca|
M25@::|BOTTLENOSE|
S@::|bottlenose dolphin|
For 24 pairings, there were 6 possible matches:
1. [ACCEPT]M25@::|DOLPHIN| <-> S@::|dolphi
2. [ACCEPT]M25@::|SEA-MAMMAL| <-> S@::|
3. [ACCEPT]M25@::|SEA-MAMMAL| <-> S@::|
4. [ACCEPT]U@::|MAMMAL| <-> S@::|cetacea
5. [ACCEPT]U@::|MAMMAL| <-> S@::|placenta
6. [ACCEPT]U@::|MAMMAL| <-> S@::|mammal
The maximal consistent subset size is 3. There were 3
1. [ACCEPT]
1. M25@::|DOLPHIN| <-> S@::|dolphin>
2. M25@::|SEA-MAMMAL| <-> S@::|pla
3. U@::|MAMMAL| <-> S@::|mammal| : 0
2. [ACCEPT]
1. M25@::|DOLPHIN| <-> S@::|dolphin>
2. M25@::|SEA-MAMMAL| <-> S@::|cet
3. U@::|MAMMAL| <-> S@::|mammal| : 0
3. [ACCEPT]
1. M25@::|DOLPHIN| <-> S@::|dolphin>
2. M25@::|SEA-MAMMAL| <-> S@::|cet
3. U@::|MAMMAL| <-> S@::|placental>
Next Page
Document: Done
Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: http://www.isi.edu/sims/philpot/ontology/frame/frame-tag-2648.html What's Related
Instant Message WebMail Radio People Yellow Pages Download Calendar Channels
Dynamically generated sewing up page
For merge point U@::|TANGIBLE-OBJECT| (orig merge point was U@::|NATURAL-OBJECT|) , mikro leaf M25@::|KEROSENE| / sensor interior concept S@::|kerosine|
U@::|TANGIBLE-OBJECT|
U@::|DECOMPOSABLE-OBJECT|
U@::|TANGIBLE-NONVOLITIONAL-OBJECT|
U@::|NONBIOLOGICAL-OBJECT|
U@::|NONVOLITIONAL_NONBIOLOGICAL-OBJECT|
U@::|NATURAL-OBJECT|
M25@::|EARTH-MATERIAL|
M25@::|PETROLEUM|
U@::|NONDECOMPOSABLE-OBJECT|
S@::|compound|
U@::|STATE-NON-SPECIFIC-OBJECT|
S@::|organic compound|
S@::|fuel>gas|
S@::|hydrocarbon|
M25@::|KEROSENE|
S@::|kerosine|
For 42 pairings, there were 3 possible matches:
1. [ACCEPT]M25@::|EARTH-MATERIAL| <-> S@::|fuel>gas| : 0.8001475
2. [ACCEPT]U@::|TANGIBLE-NONVOLITIONAL-OBJECT| <-> S@::|fuel>gas| : 0.74999994
3. [ACCEPT]U@::|DECOMPOSABLE-OBJECT| <-> U@::|NONDECOMPOSABLE-OBJECT| : 0.9756098
The maximal consistent subset size is 2. There were 2 subsets found of that size:
1. [ACCEPT]
1. U@::|TANGIBLE-NONVOLITIONAL-OBJECT| <-> S@::|fuel>gas| : 0.74999994
2. U@::|DECOMPOSABLE-OBJECT| <-> U@::|NONDECOMPOSABLE-OBJECT| : 0.9756098
2. [ACCEPT]
1. M25@::|EARTH-MATERIAL| <-> S@::|fuel>gas| : 0.8001475
2. U@::|DECOMPOSABLE-OBJECT| <-> U@::|NONDECOMPOSABLE-OBJECT| : 0.9756098
Next Page
Document: Done

A puzzle…

- Is Amber Decomposable or Nondecomposable?
- The 'stone' sense of it (Mikro) is; the 'resin' sense (WordNet) is not…
- What to do??



Shishkebobs (Hovy et al. 03)

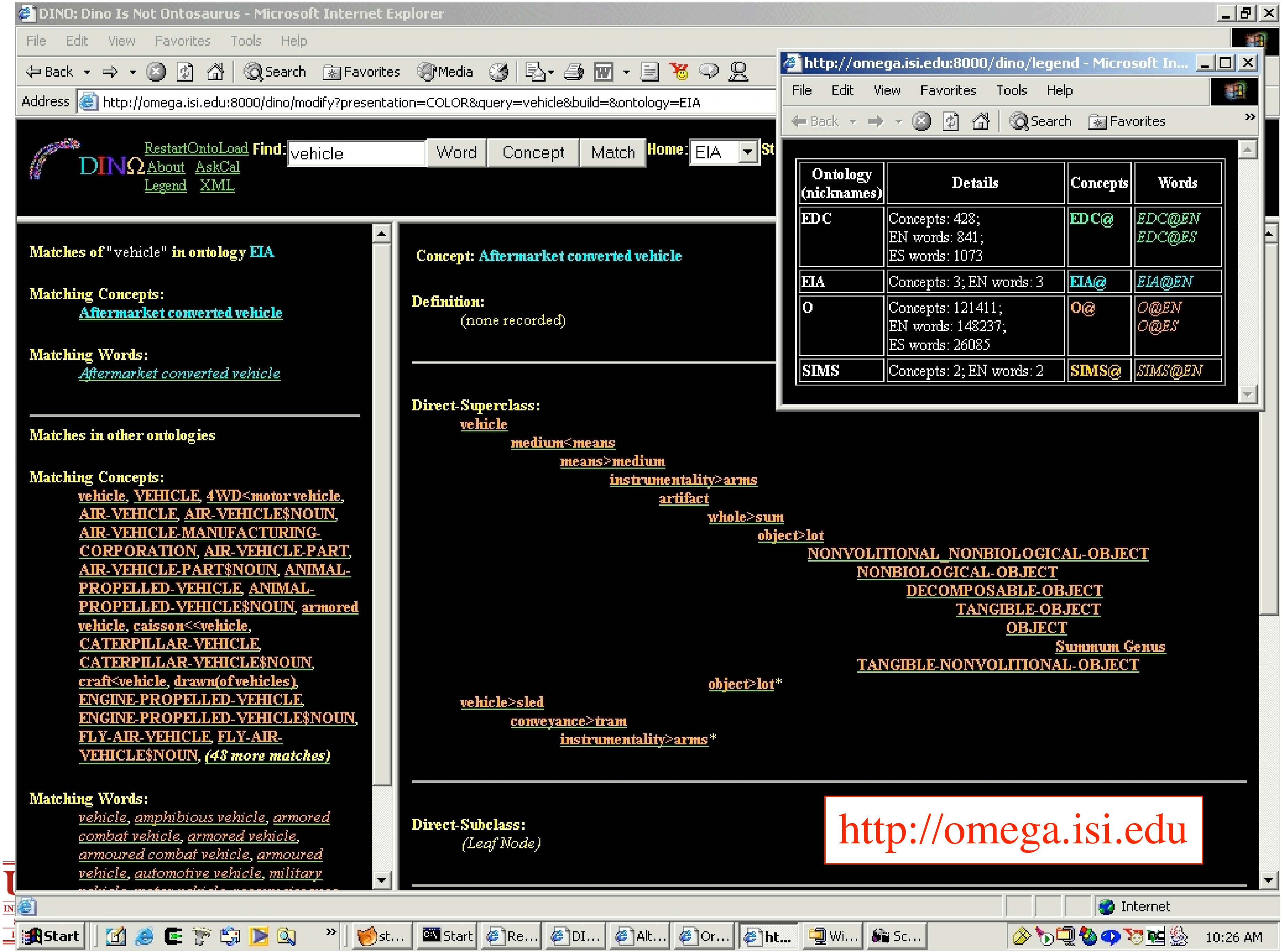
- Library ISA Building (and hence can't buy things)
  Library ISA Institution (and hence can buy things)
  SO: Building ◊ Institution ◊ Location …a Library is *all* these

- Also: Field-of-Study ◊ Activity ◊ Result-of-Process:
  (Science, Medicine, Architecture, Art…)

- Allowing shishkebobs makes merging ontologies easier (possible?): you respect each ontology's perspective

- Continuum: from on-the-fly shadings to metonymy
  (see Guarino's *identity conditions*; Pustejovsky's *qualia*)

- We found about 400 shishkebobs


DINO: Dino Is Not Ontosaurus - Microsoft Internet Explorer
http://omega.isi.edu:8000/dino/modify?presentation=COLOR&query=vehicle&build=&ontology=EIA
DINΩ
RestartOntoLoad About AskCal Legend XML
Find: vehicle Word Concept Match Home: EIA
Matches of "vehicle" in ontology EIA
Matching Concepts:
Aftermarket converted vehicle
Matching Words:
Aftermarket converted vehicle
Matches in other ontologies
Matching Concepts:
vehicle, VEHICLE, 4WD<motor vehicle, AIR-VEHICLE, AIR-VEHICLE$NOUN, AIR-VEHICLE-MANUFACTURING-CORPORATION, AIR-VEHICLE-PART, AIR-VEHICLE-PART$NOUN, ANIMAL-PROPELLED-VEHICLE, ANIMAL-PROPELLED-VEHICLE$NOUN, armored vehicle, caisson<<vehicle, CATERPILLAR-VEHICLE, CATERPILLAR-VEHICLE$NOUN, craft<vehicle, drawn(of vehicles), ENGINE-PROPELLED-VEHICLE, ENGINE-PROPELLED-VEHICLE$NOUN, FLY-AIR-VEHICLE, FLY-AIR-VEHICLE$NOUN, (48 more matches)
Matching Words:
vehicle, amphibious vehicle, armored combat vehicle, armored vehicle, armoured combat vehicle, armoured vehicle, automotive vehicle, military
Concept: Aftermarket converted vehicle
Definition:
(none recorded)
Direct-Superclass:
vehicle
medium<means
means>medium
instrumentality>arms
artifact
whole>sum
object>lot
NONVOLITIONAL_NONBIOLOGICAL-OBJECT
NONBIOLOGICAL-OBJECT
DECOMPOSABLE-OBJECT
TANGIBLE-OBJECT
OBJECT
Summum Genus
TANGIBLE-NONVOLITIONAL-OBJECT
object>lot*
vehicle>sled
conveyance>tram
instrumentality>arms*
Direct-Subclass:
(Leaf Node)
http://omega.isi.edu:8000/dino/legend - Microsoft In...
Ontology (nicknames) | Details | Concepts | Words
EDC | Concepts: 428; EN words: 841; ES words: 1073 | EDC@ | EDC@EN EDC@ES
EIA | Concepts: 3; EN words: 3 | EIA@ | EIA@EN
O | Concepts: 121411; EN words: 148237; ES words: 26085 | O@ | O@EN O@ES
SIMS | Concepts: 2; EN words: 2 | SIMS@ | SIMS@EN
http://omega.isi.edu
Internet
10:26 AM

2a. New Metadata: Learning terms by clustering web information

(This work by Patrick Pantel and Dekang Lin)

USC INFORMATION SCIENCES INSTITUTE

Where/how to find new metadata?

- Potential sources:
  - Existing ontologies (AI efforts, Yahoo!, etc.) and lists (SIC codes, etc.)
  - Manual entry, esp with reference to foreign-language text (EuroWordNet, IL-Annot, etc.)
  - Dictionaries and thesauri (Webster's, Roget's, etc.)
  - Automated discovery by text clustering (Pantel and Lin, etc.)
- Issues:
  - How **large** do you want it? — tradeoff size vs. consistency and ease of use
  - How **detailed**? — tradeoff granularity/domain-specificity vs. portability and wide acceptance (Semantic Web)
  - How **language-independent**? — tradeoff independence vs. utility for non/shallow-semantic NLP applications

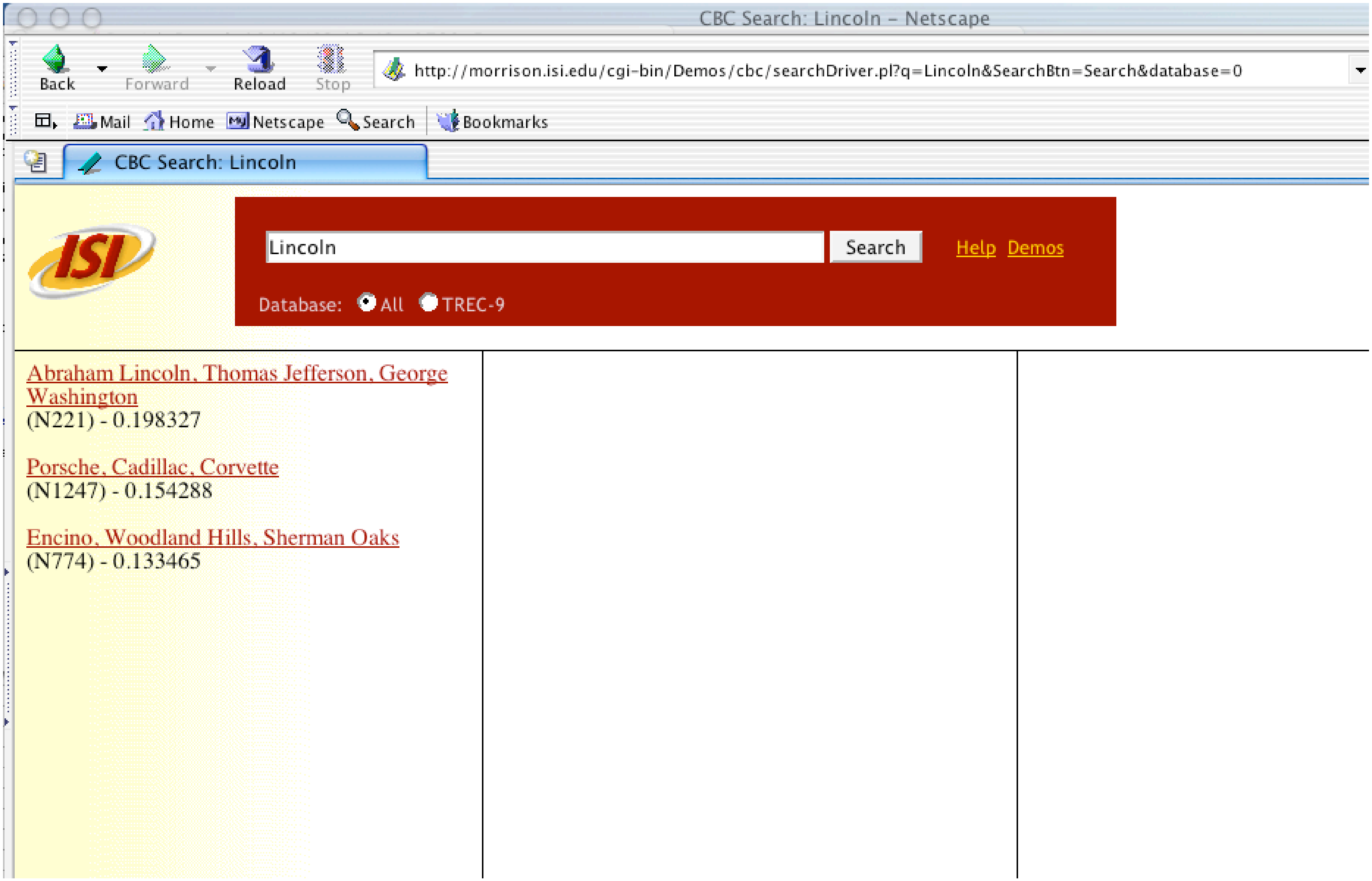
Clustering By Committee (Pantel and Lin 02)

- Very accurate new clustering procedure CBC:
  - Define syntactic/POS patterns as features (*N-N; N-subj-V*)
  - Parse corpus using MINIPAR (D. Lin)
  - Cluster, using MI on features:

    (*e=word, f=pattern*)

$$mi_{ef} = \log \frac{\frac{c_{ef}}{N}}{\frac{\sum_{i=1}^{n} c_{if}}{N} \times \frac{\sum_{j=1}^{m} c_{ej}}{N}}$$

  - Find cluster centroids (using strict criteria): *committee*
  - For non-centroid words, match their pattern features to committee's; if match, include in cluster, remove features
  - If word has remaining features, try to include in other clusters as well — handle ambiguity
- Find name for clusters (superconcepts):
  - Word shared in apposition, nominal-subj, etc. templates
- Complexity: $O(n^2k)$ for *n* words in corpus, *k* features



www.isi.edu/~pantel/

CBC Search: Lincoln – Netscape

Back Forward Reload Stop http://morrison.isi.edu/cgi-bin/Demos/cbc/searchDriver.pl?q=Lincoln&SearchBtn=Searc

Mail Home Netscape Search Bookmarks

CBC Search: Lincoln

ISI

Lincoln Search Help Demos

Database: All TREC-9

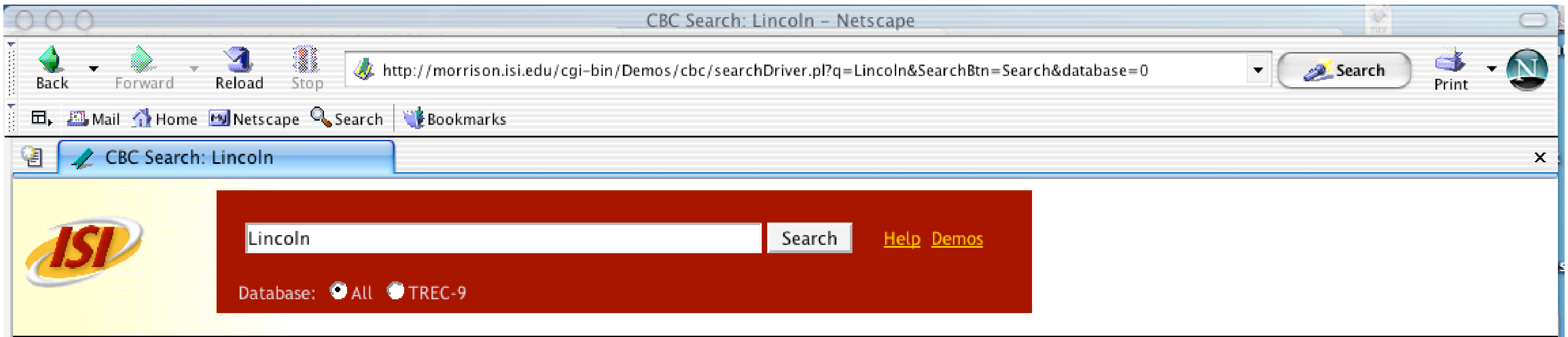
Abraham Lincoln, Thomas Jefferson, George Washington
(N221) - 0.198327

Porsche, Cadillac, Corvette
(N1247) - 0.154288

Encino, Woodland Hills, Sherman Oaks
(N774) - 0.133465

N221: Abraham Lincoln, Thomas Jefferson, George Washington

Abraham Lincoln	0.440408
Thomas Jefferson	0.427359
George Washington	0.390025
Alexander Hamilton	0.357866
Franklin Delano Roosevelt	0.347813
Andrew Jackson	0.346218
John Adams	0.333103
Theodore Roosevelt	0.254847
Jefferson	0.232393
Franklin Roosevelt	0.231732
John Quincy Adams	0.20884
Benjamin Franklin	0.20414
Lincoln	0.198327
James Madison	0.197463
Winston Churchill	0.194458
Robert E. Lee	0.174996
Roosevelt	0.164226
founding father	0.161931
Martin Luther King Jr.	0.136376
Napoleon	0.128418
Christopher Columbus	0.128006
Sam Houston	0.123654
Lenin	0.123188
Mel Karmazin	0.120426
Genghis Khan	0.110591



OMEGA::Lincoln

Grammatical templates

-V:obj:N 1869 times:

- {V1662 offer, provide, make} 156, have 108, {V1650 go, take, fly} 51, sell 45, {V1754 become, remain, seem} 34, … give 24, {V1647 oppose, reject, support} 24, buy 21, {V1653 allocate, earmark, owe} 21, win 20 …

-N:conj:N 536 times:

- {N719 Toyota, Nissan, BMW} 65, {N257 Cadillac, Buick, Lexus} 59, {N549 Philadelphia, Seattle, Chicago} 41, American Continental 20, Cadillacs 11, …

-V:by:N 50 times:

- {V1662 offer, provide, make} 12, own 5, hire 4, target 4, write 3, buy 2, …

SCIENCES INSTITUTE

Finding names for clusters

- Search for repeated appositions
  - “the *President*, Thomas Jefferson, …”
  - “Kobe Bryant, famous *basketball star*…”
- Check against ontology, if present
- Examples for Lincoln:
  - PRESIDENT(N891) - 0.187331
  - UNIT / BORROWER / THRIFT(N724) - 0.166958
  - CAR / DIVISION(N257) - 0.137333

→ Candidate metadata terms

USC INFORMATION SCIENCES INSTITUTE

2b. Metadata Concept (Term) Relations: Learning interconnectivity

(This work with Chin-Yew Lin, Mike Junk, Michael Fleischman, and Tom Murray)

Topic signature

Word family built around inter-word relations.

- **Def**: Head word (or concept), plus set of related words (or concepts), each with strength:

  $$\{ T_k, (t_{k1}, w_{k1}), (t_{k2}, w_{k2}), \ldots, (t_{kn}, w_{kn}) \}$$

- **Problem**: Scriptal co-occurrence, etc. — how to find it?
- Approximate by simple textual term co-occurrence...

<u>Related words in texts show Poisson distribution:</u>

In large set of texts, topic keywords concentrate around topics; so compare topical word frequency distributions against global background counts.

Learning signatures

Procedure:

1. Collect texts, sorted by topic
2. Identify families of co-occurring words
3. Evaluate their purity
4. Find the words' concepts in the Ontology
5. Link together the concept signatures

Need texts, sorted

How to count co-occurrence?

How to evaluate?

Need disambiguator

Calculating weights

Approximate relatedness using various formulas

$$\underline{tf.idf} \; : \; w_{jk} = tf_{jk} * idf_j$$

$$\chi^2 \; : \; w_{jk} = \begin{cases} (tf_{jk} - m_{jk})^2 / m_{jk} & \text{if } tf_{jk} > m_{jk} \\ 0 & \text{otherwise} \end{cases}$$

(Hovy & Lin, 1997)

- tf_{jk} : count of term j in text k ("waiter" often only in some texts).
- $idf_j = log(N/n_j)$: within-collection frequency ("the" often in <u>all</u> texts),
  n_j = number of docs with term j, N = total number of documents.
- *tf.idf* is the best for IR, among 287 methods (Salton & Buckley, 1988).
- $m_{jk} = (\Sigma_j \, tf_{jk} \, \Sigma_k \, tf_{jk}) \, / \, \Sigma_{jk} \, tf_{jk}$: mean count for term j in text k .

$$\underline{\text{likelihood ratio } \lambda} : \; 2log\,\lambda = 2N \, . \, I\,(R\,;T)$$

(Lin & Hovy, 2000)

(more approp. for sparse data; $-2log\lambda$ asymptotic to χ^2).

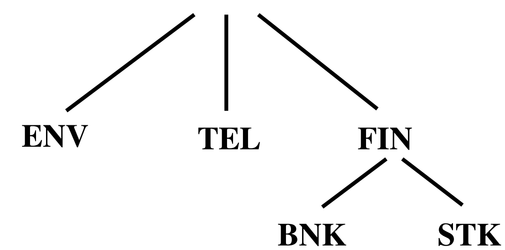
- N = total number terms in corpus.
- I = mutual information between text relevance R and given term T,
  $= H(R) - H(R \mid T)$ for $H(R)$ = entropy of terms over relevant texts R
  and $H(R \mid T)$ = entropy of term T over rel and nonrel texts.

USC INFORMATION SCIENCES INSTITUTE

Early signature study (Hovy & Lin 97)

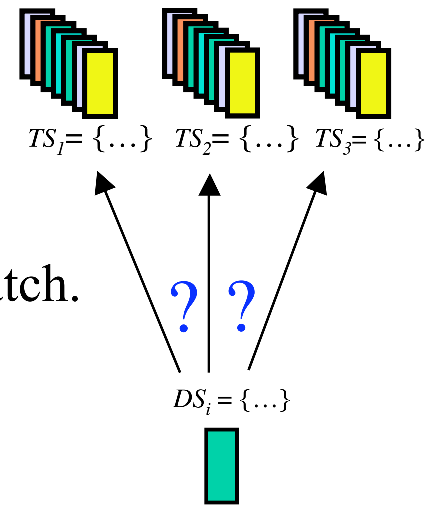
- **Corpus**
  - Training set WSJ 1987:
    - 16,137 texts (32 topics).
  - Test set WSJ 1988:
    - 12,906 texts (31 topics).
  - Texts indexed into categories by humans.
- **Signature data**
  - 300 terms each, using *tf.idf* .
  - Word forms: single words, demorphed words, multi-word phrases.
- **Topic distinctness...**
  - Topic hierarchy.

RANK	ARO	BNK	ENV	TEL
1	contract	bank	epa	at&t
2	air_force	thrift	waste	network
3	aircraft	banking	environmental	fcc
4	navy	loan	water	cbs
5	army	mr.	ozone	cable
6	space	deposit	state	bell
7	missile	board	incinerator	long-distance
8	equipment	fslic	agency	telephone
9	mcdonnell	fed	clean	telecomm.
10	northrop	institution	landfill	mci
11	nasa	federal	hazardous	mr.
12	pentagon	fdic	acid_rain	doctrine
13	defense	volcker	standard	service
14	receive	henkel	federal	news

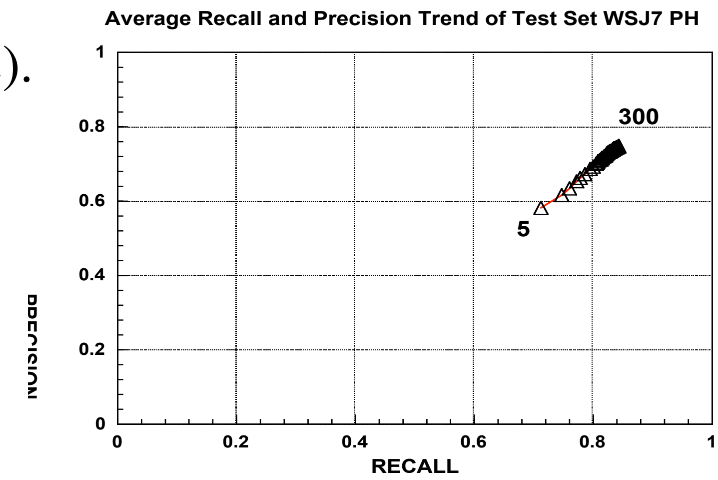


Evaluating signatures

- **Solution**: Perform text categorization task:
  - create N sets of texts, one per topic,
  - create N topic signatures TS_k ,
  - for each new document, create document signature DS_i ,
  - compare DS_i against all TS_k ; assign document to best match.
- **Match function**: vector space similarity measure:
  - Cosine similarity, $\cos\theta = TS_k \cdot DS_i \ / \ |TS_k||DS_i|$.



- **Test 1** (Hovy & Lin, 1997, 1999)
  - Training set: 10 topics; ~3,000 texts (TREC).
  - Contrast set (background): ~3,000 texts.
  - Conclusion: *tf.idf* and χ^2 signatures work ok but depend on signature length.
- **Test 2** (Lin & Hovy, 2000):
  - 4 topics; 6,194 texts; uni/bi/trigram signats.
  - Evaluated using SUMMARIST: $\lambda > tf.idf$.



USC INFORMATION SCIENCES INSTITUTE

Text pollution on the web

Goal: Create word families (signatures) for *each concept in the Ontology*. Get texts from Web.

Main problem: text pollution. What's the search term?

<MORTICE,w=33.7982>	<STAR, w=75.1358>	<AIRCRAFT, w=207.998>
<WOODWORKING, w=20.9227>	<ORION,w=55.8937>	<ENGINE, w=178.677>
<TENNON, w=20.9227>	<PYRAMID,w=42.1494>	<WING, w=138.36>
<JOINERY, w=17.7038>	<DNA,w=41.2331>	<PROPELLER, w=122.317>
<WOOD, w=15.8356>	<SOUL,w=31.1539>	<FLY, w=103.187>
<HARDWOOD, w=14.4849>	<IMPLOSION,w=23.8236>	<AIRPLANE, w=98.0431>
<JASON, w=14.4849>	<KHUFU,w=19.3133>	<AVIATION, w=96.5663>
<DOTH, w=12.8755>	<GOLD,w=18.3897>	<FLIGHT, w=85.3079>
<BRASH, w=12.8755>	<RECURSION,w=18.3258>	<AIR, w=80.1996>
<OAK, w=12.8281>	<BELLATRIX,w=17.7038>	<WARBIRDS, w=72.4247>
<WEDGE, w=11.9118>	<OSIRIS,w=17.7038>	<PILOT, w=71.4707>
<FURNITURE, w=10.0792>	<PHI,w=16.4932>	<MPH, w=65.987>
<TOOL, w=9.19486>	<EMBED,w=16.4932>	<CONTROL, w=65.9729>
<SHAFT, w=8.17321>	<MAGNETIC,w=16.4932>	<FUEL, w=62.3078>

Purifying: In later work, used Latent Semantic Analysis

Purifying with Latent Semantic Analysis

- Technique used in Psychologists to determine basic cognitive conceptual primitives (Deerwester et al., 1990; Landauer et al., 1998).
- Singular Value Decomposition (SVD) used for text categorization, lexical priming, language learning…
- LSA automatically creates collections of items that are correlated or anti-correlated, with strengths:

  ice cream, drowning, sandals ⇨ *summer*

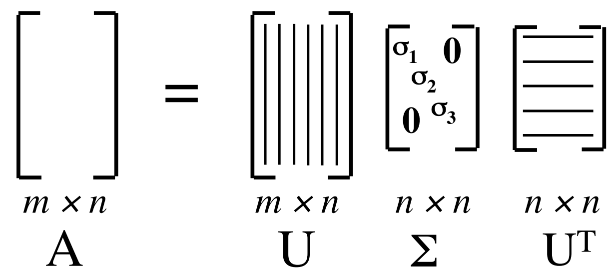
- Each such collection is a 'semantic primitive' in terms of which objects in the world are understood.

- We tried LSA to find most reliable signatures in a collection—reduce number of signatures in contrast set.

USC INFORMATION SCIENCES INSTITUTE

LSA for signatures

- Create matrix A, one signature per column (words × topics).
- Apply SVDPAC to compute U so that $A = U \Sigma U^T$:
  - U : $m \times n$ orthonormal matrix of left singular vectors that span space
  - U^T : $n \times n$ orthonormal matrix of right singular vectors
  - Σ : diagonal matrix with exactly *rank(A)* nonzero singular values; $\sigma_1 > \sigma_2 > \ldots > \sigma_n$



- Use only the first *k* of the new concepts: $\Sigma' = \{\sigma_1, \sigma_2 \ldots \sigma_k\}$.
- Create matrix A′ out of these *k* vectors: $A' = U \Sigma' U^T \approx A$.

A′ is a new (words × topics) matrix, with different weights and new ‘topics’. Each column is a purified signature.

Some results with LSA (Hovy and Junk 99)

- Contrast set (for *idf* and χ^2): set of documents on very different topic, for good *idf*.
- Partitions: collect documents within each topic set into partitions, for faster processing. $/n$ is a collecting parameter.
- U function: function for creation of LSA matrix.

Results:
- Demorphing helps.
- χ^2 better than *tf* and *tf.idf*.
- LSA improves results, but not dramatically.

TREC texts

Function	Demorph?	Partitions	U function	Recall	Precision
Without contrast set					
tf	no			0.748447	0.628782
tf	yes			0.766428	0.737976
tf	yes	10	*tf*	0.820609	0.880663
tf	yes	20	*tf*	0.824180	0.882533
tf	yes	30	*tf*	0.827752	0.884352
With contrast set					
tf.idf	no	10	*tf.idf*	0.626888	0.681446
tf.idf	no	20	*tf.idf*	0.635875	0.682134
tf.idf	yes	10	*tf.idf*	0.718177	0.760925
tf.idf	yes	20	*tf.idf*	0.715399	0.762961
X^2	no	10	X^2	0.847393	0.841513
X^2	no	20	X^2	0.853436	0.849575
X^2	yes	10	X^2	0.822615	0.828412
X^2	yes	20	X^2	0.839114	0.839055
Varying partitions					
X^2	yes	30/0	X^2	0.912525	0.881494
X^2	yes	30/3	X^2	0.903534	0.879115
X^2	yes	30/6	X^2	0.903611	0.873444
X^2	yes	30/9	X^2	0.899407	0.868053

USC INFORMATION SCIENCES INSTITUTE

Web signature experiment

Procedure:

1. Create query from Ontology concept (word + defn. words)
2. Retrieve ~5,000 documents (8 web search engines)
3. Purify results (remove duplicates, html, etc.)
4. Extract word family (using *tf.idf*, χ^2, LSA, etc.)
5. Purify
6. Compare to siblings and parents in the Ontology

Problem: raw signatures overlap…

- average parent-child node overlap: ~50%
- Bakery—Edifice: ~35% …too far: missing generalization.
- Airplane—Aircraft: ~80% …too close?

Remaining problem: web signatures still not pure...

WordNet: In 2002–04, Agirre and students (U of the Basque Country) built signatures for all WordNet nouns

USC INFORMATION SCIENCES INSTITUTE

Recent work using signatures

- **Multi-document summarization** (Lin and Hovy, 2002)
  - Create λ signature for each set of texts.
  - Create IR query from signature terms; use IR to extract sentences.
  - (Then filter and reorder sentences into single summary).
  - **Performance:** DUC-01: tied first; DUC-02: tied second place
- **Wordsense disambiguation** (Agirre, Ansa, Martinez, Hovy, 2001)
  - Try to use WordNet concepts to collect text sets for signature creation: *(word+synonym* > *def-words* > *word .AND. synonym .NEAR. def-word* > etc…).
  - Built competing signatures for various noun senses:
    (a) WordNet synonyms; (b) SemCor tagged corpus (χ^2);
    (c) web texts (χ^2); (d) WSJ texts (χ^2).
  - **Performance**: Web signatures > random, WordNet baseline.
- **Email clustering** (Murray and Hovy, in prep)
  - Social Network Analysis: Cluster emails and create signatures.
  - Infer personal expertise, project structure, experts omitted, etc.
  - Corpora: ENRON (240K emails), ISI corpus, NSF eRulemaking corpus.

USC INFORMATION SCIENCES INSTITUTE

3a. Data Instances:
Extracting seed instances from text

(This work with Michael Fleischman)

USC INFORMATION SCIENCES INSTITUTE

What kinds of knowledge?

- **Goal 1**: Add instantial knowledge:
  - *Sofia is a city*
  - *Sofia is a woman's name*
  - *Cleopatra was a queen*
  - *Everest is a mountain*
  - *Varig is an airline company*

- **Goal 2**: Add definitional / descriptive knowledge:
  - *Mozart was born in 1756*
  - *Bell invented the telephone*
  - *Pisa is in Italy*
  - *The Leaning Tower is in Pisa*
  - *Columbus discovered America*

Create links between concepts

Classify instances under types

- **Uses**:
  - QA (answer suggestion and validation)
  - Wordsense disambiguation for MT
- **Sources**:
  - Existing lists (CIA factbook, atlases, phone books…)
  - Dictionaries and encyclopedias
  - The Web

USC INFORMATION SCIENCES INSTITUTE

Learning about locations… (Fleischman 01)

- Challenge ex.—*region, state/territory*, or *city*?
  - The company, which is based in *Dpiyj Dsm Gtsmdodvp*, *Vsaog*., said an antibody prevented development of paralysis.
  - The situation has been strained ever since *Yplup* began waging war in *Dpiyj Rsdy Sdos*.
  - The pulp and paper operations moved to *Dpiyj Vstpaomos* in 1981.
- Try to learn instances of 8 types (*country, region, territory, city, street, artifact, mountain, water*):
  - (we have lists of these already, so finding sentences for training data is easy).
- Uses:
  - QA: corroborating evidence for answer.
  - IR: query expansion and signature enrichment.

South San Francisco—region; Calif.—state; Tokyo—city; South East Asia—region; South Carolina—state

Learning procedure

- **Approach**:
  - Training: For each location, identify features in context; try to learn features that indicate each type
  - Usage: For new material, use learned features to classify type of location; place results with high confidence into ontology
- **Training**:
  - Applied BBN's IdentiFinder to bracket locations
  - Chose 88 features (unigrams,bigrams,trigrams in fixed positions before and after location instance; later added signatures, etc.)
  - 3 approaches: Bayesian classifier, neural net, decision tree (C4.5)
  - MemRun procedure: store good examples and prefer later

USC INFORMATION SCIENCES INSTITUTE

Memrun

Initial results:

- Bayesian classifier not very accurate; neural net ok.
- D-tree better, but still multiple classes for each instance.

Memrun: record best example of each instance.

Algorithm with Memrun:

- **Pass 1**: for each text,
  - preprocess with POS tagger and IdentiFinder
  - apply D-tree to classify instance
  - if *score* > THRESH1, save (*instance,tag,score*) in Memrun
- **Pass 2**: for each text,
  - again apply D-tree
  - if *score* < THRESH2, replace tag by Memrun value

Examples

City
Aachen
Abadan
Abassi Madani
Abbassi Madani
Abbreviations AZ
Abdullojonov
Aberdeen
Abidjan
Abidjan Radio Cote d'Ivoire Chaine
Abiko
Abrahamite
Abramenkov
Abu Dhabi
Abuja
Abyssinia
Acari
Accom
Accordance
...

Mountain
…
Wicklow Mountains
Wudang Mountain
Wudangshan Mountain
Wuling mountains
Wuyi Mountains
Xiao Hinggan Mountains
Yimeng Mountains
Zamborak mountain
al-Marakishah mountain
al-Maraqishah mountains
al-Nubah mountains
al-Qantal mountain

Water
Abuna River
Adriatic
Adriatic Sea
Adriatic sea
Aegean Sea
Aguapey river
Aguaray River
Akhtuba River
Akpa Yafe River
Akrotiri salt lake
Aksu River
Alma-Atinka River
Almendares River
Alto Maranon River
Amazon River
Amur river
Andaman Sea
Angara River
Angrapa river
Anna River
Arabian Gulf
Arabian Sea
...

Territory
…
General Robles
Ghanaians
Gilan Province
Gilan Province Sha'ban
Gitega Province
Glencore
Goias State
Goias State of Brazil
Gongola State
Granma Province
Great Brotherly Russia
Greytown
Guanacaste Province
Guandong province
Guangdong Province
Guangxi Province
Guangzhou Shipyards
Guantanamo Province
Guayas Province
Guerrero State
Guiliano Amato
Guizhou Province
Gwent
...

Results for locations

Decision Tree

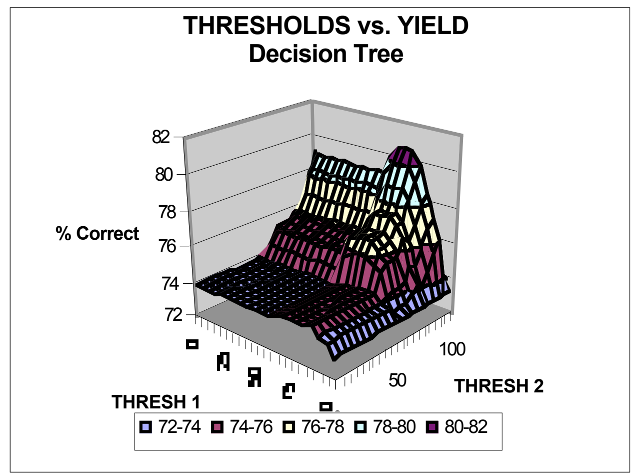
	After C4.5		After MemRun	
Class	# corr / tot	%	# corr / tot	%
City	260 / 373	69.70	309 / 373	82.80
Country	411 / 482	85.30	440 / 482	91.20
Street	6 / 10	60.00	6 / 10	60.00
Region	44 / 65	67.70	44 / 65	67.70
Water	6 / 7	85.70	6 / 7	85.70
Artifacts	4 / 8	50.00	4 / 8	50.00
Territory	71 / 148	48.00	79 / 148	53.40
Mount	0 / 3	0.00	0 / 3	0.00
Total	802 / 1096	73.20	888 / 1096	81.00

Confusion Matrix After MemRun

Actual	CITY	COUNTR	STR	REGN
CITY	309	51	0	3
COUNTRY	29	440	0	6
STREET	3	1	6	0
REGION	12	7	0	44
WATER	0	1	0	0
ARTS	0	3	0	0
TER	32	33	0	4
MOUNT	0	3	0	0

Human Competence

	Subject 1		Subject 2		Subject 3		Avg.
Class	# corr / tot	%	#corr / tot	%	# corr / tot	%	%
City	6 / 7	71.4	5 / 7	71.4	4 / 7	57.1	66.7
Country	27 / 37	73	25 / 37	67.6	17 / 37	45.9	62.2
Street	2 / 5	40	2 / 5	40	1 / 5	20	33.3
Region	4 / 8	50	4 / 8	50	3 / 8	37.5	45.8
Water	0 / 3	0	0 / 3	0	0 / 3	0	0
Artifacts	2 / 2	0	0 / 2	0	2 / 2	100	66.7
Territory	8 / 11	72.7	5 / 11	45.5	10 / 11	90.9	69.7
Mount	1 / 2	50	0 / 2	0	0 / 2	0	16.7
Total	49 / 75	65.3	40 / 75	53.3	40 / 75	53.3	57.3



NB: test samples are small

THRESH1 = 77%; THRESH2 = 98%

People

(Fleischman and Hovy 02)

Goal: Collecting training data about 8 **types of people**: politicians, entertainers (movie stars, etc.), athletes, businesspeople...

Procedure: as before, with added features using signature of each category and WordNet hypernyms.

athlete	
458.029	;
398.626	perez
392.904	rogers
368.441	carlos
351.686	points
333.083	roy
311.042	andres
284.927	scored
273.197	chris
252.172	hardee's
239.747	george
223.879	games
222.202	mark
217.711	mike
...	

businessperson	
4428.267	greenspan
3999.135	alan
2774.682	reserve
2429.129	chairman
1786.783	federal
1709.120	icahn
1665.358	fed
1252.701	carl
827.0291	board
682.420	rates
662.510	investor
651.529	twa
531.907	kerkorian
522.072	interest
...	

cleric	
1133.793	rabbi
1074.785	cardinal
1011.190	paul
809.128	archbishop
798.372	john
748.170	bishop
714.173	catholic
688.291	church
613.625	roman
610.287	tutu
584.720	desmond
460.057	pope
309.923	kahane
300.236	meir
...	

entertainer	
1902.178	"
1573.695	actor
1083.622	actress
721.929	movie
618.947	george
607.466	film
553.659	singer
541.235	president
536.962	her
536.856	keating
528.226	star
448.524	(
433.065	)
404.008	said
...	

Some results for people

Total count	1030	
Total Correct	839	0.815
Total Incorrect	191	0.185
miscCorrect	0/20	0.0
lawyerCorrect	13/44	0.295
policeCorrect	11/17	0.647
doctorCorrect	48/50	0.96
entertainerCorrect	150/173	0.867
athleteCorrect	11/13	0.846
businessCorrect	120/166	0.722
militaryCorrect	14/21	0.666
clergyCorrect	11/11	1.0
politicianCorrect	461/515	0.895

Best results: using signatures and WordNet hyperlinks (but no synset expansion).

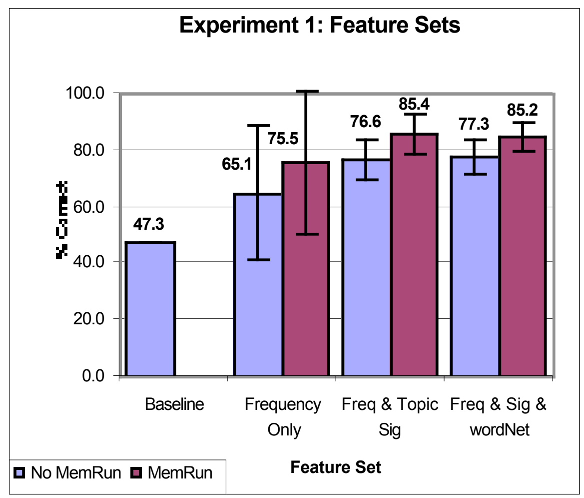
Problems:

- Training and test data skewed.
- Genuine ambiguity

  often, politician = military leader.

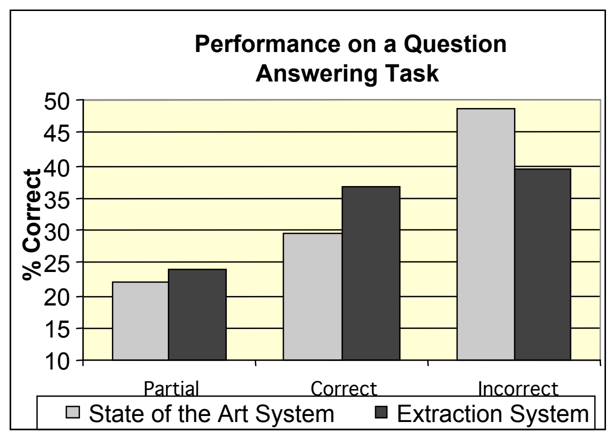
Confusion Matrix

Type	BUS	POL	ENT	MIL	DOC	ATH	CLR	LAW	COP	MISC
BUS	120	23	7	1	11	0	0	1	3	0
POL	8	461	34	0	5	0	7	0	0	0
ENT	12	6	150	0	5	0	0	0	0	0
MIL	0	3	4	14	0	0	0	0	0	0
DOC	0	0	2	0	48	0	0	0	0	0
ATH	0	1	1	0	0	11	0	0	0	0
CLR	0	0	0	0	0	0	11	0	0	0
LAW	0	29	0	0	2	0	0	13	0	0
COP	0	4	0	2	0	0	0	0	11	0
MISC	9	2	4	0	0	1	1	0	3	0



Instance extraction++ (Fleischman & Hovy 03)

- **Goal**: extract *all* instances from the web
- **Method**:
  - Download text from web (15GB)
  - Identify named entities (BBN's IdentiFinder (Bikel et al. 93))
  - Extract ones with descriptive phrases (<APOS>, <CN/PN>)
    ("the vacuum manufacturer Horeck" / "Saddam's physician Abdul")
  - Cluster them, and categorize in ontology
- **Result**: over 900,000 instances
  - Average: 2 mentions per instance, 40+ for George W. Bush
- **Evaluation**:
  - Tested with 200 "*who is X?*" questions
  - Better than TextMap: 25% more
  - Faster: 10 sec ↔ 9 hr !



USC INFORMATION SCIENCES INSTITUTE

3b. Learning Relations: Harvesting additional information from text

(This work with Deepak Ravichandran and Patrick Pantel)

USC INFORMATION SCIENCES INSTITUTE

Shallow patterns for information

- **Goal**: learn relationship data from the web
  - (when was someone born? Where does he live?)
- Procedure: automatically learn <u>word-level patterns</u>

  When was Mozart born?

  "Mozart (1756–1792)…"

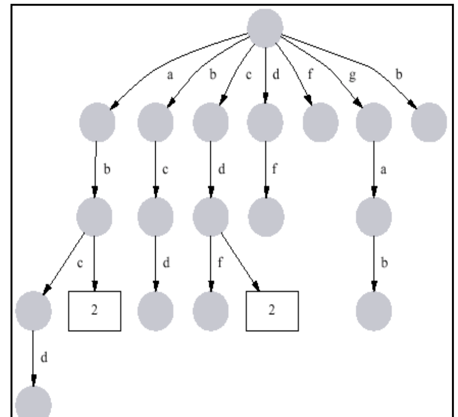
  [<NAME> (<BIRTHYEAR> – <DEATHYEAR>)]
- Apply patterns to Omega concepts/instances
- Evaluation: test in TREC QA competition
- **Main problem**: learning patterns
  - (In TREC QA 2001, Soubbotin and Soubbotin got very high score with over 10,000 patterns built by hand)

USC INFORMATION SCIENCES INSTITUTE

Learning extraction patterns from the web

(Ravichandran and Hovy 02)

- Prepare:
  - Select example for target relation: Q term (Mozart) and A term (1756)
- Collect data:
  - Submit Q and A terms as queries to a search engine (Altavista)
  - Download top 1000 web documents
- Preprocess:
  - Apply a sentence breaker to the documents
  - Retain only sentences with both Q and A terms
  - Pass retained sentences through suffix tree constructor



Suffix Tree for the strings abcd, cdf and gab
Substrings of length atleast 2 with a score greater than 1
are ab and cd as shown

- Select and create patterns:
  - Filter each phrase in the suffix tree to retain only those phrases that contain both Q and A terms
  - Replace the Q term by the tag “<NAME>” and the A term by the term by “<ANSWER>”

USC INFORMATION SCIENCES INSTITUTE

Some results

<u>BIRTHYEAR</u>:

1.0 <NAME> (<ANS> —

0.85 <NAME> was born on <ANS>

0.6 <NAME> was born in <ANS>

…

<u>DEFINITION</u>:

1.0 <NAME> and related <ANS>s

1.0 <ANS> (<NAME>,

0.9 as <NAME> , <ANS> and

…

<u>LOCATION</u>:

1.0 <ANS>'s <NAME> .

1.0 regional : <ANS> : <NAME>

0.9 the <NAME> in <ANS> ,

…

Testing (TREC-10 questions)

Question type	Num Qs	TREC MRR	Web MRR
BIRTHYEAR	8	0.479	0.688
INVENTOR	6	0.167	0.583
DISCOVERER	4	0.125	0.875
DEFINITION	102	0.345	0.386
WHY-FAMOUS	3	0.667	0.0
LOCATION	16	0.75	0.864

Regular expressions (Ravichandran et al. 2004)

- New process: learn regular expression patterns

Surface	Babe	Ruth	was	born	in	Baltimore	,	on	February	6 , 1895
NE Tags	<NAME>					<LOCATION>			<DATE>	
Part of Speech	NNP	NNP	VBD	VBN	IN	NNP	,	IN	NNP	CD

Surface	George	Herman	"Babe"	Ruth	was	born	here	in	1895
NE Tags	<NAME>								<DATE>
Part of Speech	NNP	NNP	NNP	NNP	VBD	VBN	RB	IN	CD

<NAME>
"was born"
<?>
_IN
<DATE>

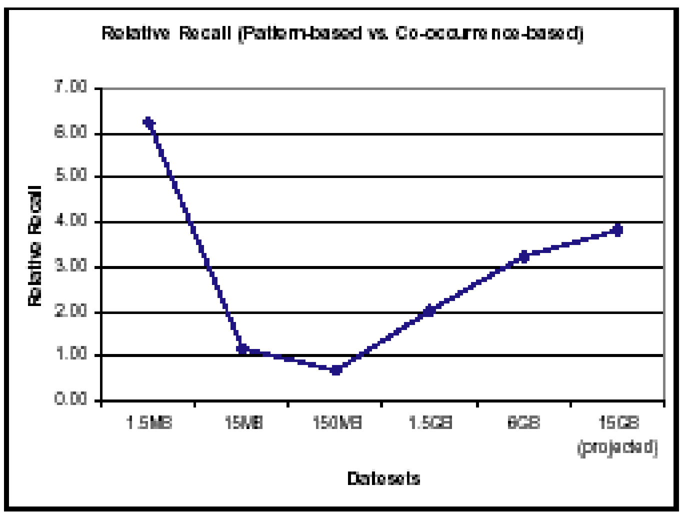
- Complexity: $O(y^2)$, for max string length y
- Results: over 2 million instances from 15GB corpus

USC INFORMATION SCIENCES INSTITUTE

Comparing clustering and surface patterns

- Precision: took random 50 words, each with system's learned superconcepts (top 3 of system); added top 3 from WordNet, 1 human superconcept. Used 2 judges (Kappa = 0.78–0.85)
- Recall: *Relative Recall* = $Recall_{Patt}$ / $Recall_{Co\text{-}occ}$ = C_P / C_C
- TREC-03 def'ns: Patt up to 52%; Co-Occ up to 44% MRR

Relative Recall



(Ravichandran and Pantel 2004)

Precision (correct+partial)

	Pattern System			Co-occurrence System		
Training	Prec	Top-3	MRR	Prec	Top-3	MRR
1.5MB	56.6%	60.0%	60.0%	12.4%	20.0%	15.2%
15MB	57.3%	63.0%	61.0%	23.2%	50.0%	37.3%
150MB	50.7%	56.0%	55.0%	60.6%	78.0%	73.2%
1.5GB	52.6%	51.0%	51.0%	69.7%	93.0%	85.8%
15GB	61.8%	69.0%	67.5%	78.7%	92.0%	86.2%
150GB	67.8%	67.0%	65.0%	Too large to process		

4a. Verifying Data: Determining opinion (and other epistemic statuses, one day)

(This work with Soo-Min Kim)

USC INFORMATION SCIENCES INSTITUTE

Summary of Opinion detection (Kim and Hovy, 2004)

- Our definition of *Opinion*:
  - A quadruple **[Topic, Holder, Claim, Sentiment]**
  - Holder = person or organization
  - Claim = statement about Topic ("abortion should be banned")
  - Sentiment (only with affective opinions):
    - *Positive* (Claim is GOOD) or *Negative* or *Mixed* or *Neutral* ("I don't care one way or the other") or *Unstated* ("they had strong feelings about that")
- Approach:
  1. Opinion recognition: Find opinion-bearing expressions in texts
  2. Sentiment recognition: For affective opinions, determine the Sentiment of these expressions
- Algorithms:
  - Word sentiment classification (for individual verbs, adjectives)
  - Sentence sentiment classification (tested various models, for different combinations of word sentiment classifiers)

USC INFORMATION SCIENCES INSTITUTE

Evaluation results (accuracy)

- Human-human agreement: 88.9% (adjs) and 85.1% (verbs)
- Human-system agreement: 76.8% (adjs) and 80.1% (verbs), with recall 97.8% and 93.2%

M*i*: sentence classifier model

P*i*: word classifier model

	Full sent	Holder–Topic	H–T ±2 words	H/T to sent end
m0p1	0.772727	0.742424	0.712121	0.80303
m0p3	0.666667	0.727273	0.712121	**0.818182**
m1p1	0.651515	0.712121	0.69697	0.712121
m1p2	0.69697	0.742424	0.742424	0.772727
m1p3	0.787879	0.757576	0.727273	0.757576
m1p4	0.681818	0.772727	0.787879	0.787879
m2p1	0.69697	0.727273	0.727273	0.787879
m2p2	0.590909	0.681818	0.651515	0.681818
m2p3	0.636364	0.742424	0.742424	0.742424
m2p4	0.590909	0.69697	0.666667	0.681818

USC INFORMATION SCIENCES INSTITUTE

5. Conclusion

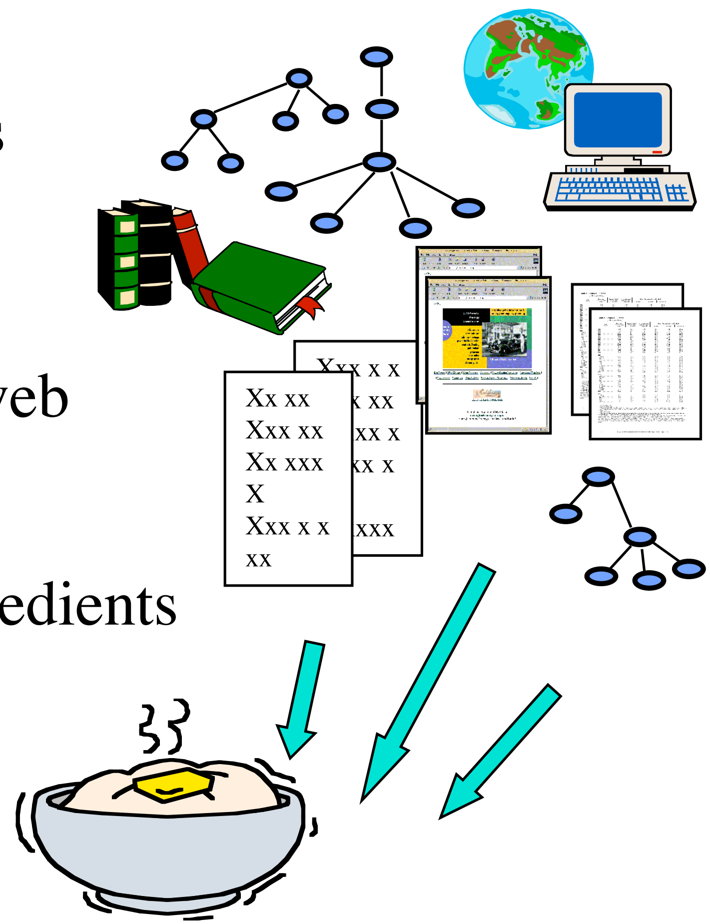
Summary

Ingredients:

– small ontologies and metadata sets
– concept families (signatures)
– information from dictionaries, etc.
– additional info from text and the web

Method:

1. Into a large database, pour all ingredients
2. Stir together in the right way
3. Bake



Evaluate—IR, QA, MT, and so on!

USC INFORMATION SCIENCES INSTITUTE

What would be nice?

- Databases that support rapid reorganization when new metadata is learned
- Databases that accommodate possibly inconsistent, partial, and tenuous data
- Very very large databases that can grow rapidly

USC INFORMATION SCIENCES INSTITUTE

Current status

- Omega:
  - Approx. 110,000 concepts
  - Approx. 1.1 mill instances
  - Subject information from TAP (Guha et al.)
  - Additional information from various sources
- Tools:
  - Alignment algorithms
  - Concept spotting, clustering, glossary parsing algorithms
  - Instance harvesting algorithm
  - Algorithms to learn inter-concept/instance relations
- Infrastructure:
  - Instances (and concepts?) into RDF (also database form?)
  - Online access and DINO browser

Next steps

- Collect **instances** of many other entities (not only locations, organizations, and people)
- Learn **more details** about each person, location, organization, etc., using patterns: date of birth, nationality, occupation, spouse, etc.
- Into Omega, incorporate **WordNet extensions** (inference rules) from (Moldovan 03)
- Merge **OpenCyc** and perhaps SIC code terms into Omega
- Build **access tools** and inline access methods to support QA, summarization, etc.

Vision

- Many people could use something like Omega:
  - The Semantic Web needs a large standardized well-organized multi-lingual termset
  - MT systems need a language-independent (or at least neutral) ontology
  - Many HLT systems can use the semantic and instantial information in Omega for better performance
  - Database integration and access systems might use something like Omega
  - AI systems might take subsets of it
- People should be encouraged to build their own!

Thank you