Advanced Internet Systems

CSE 454
Daniel Weld

Project Presentations - Monday

10:30 - BestBet
10:45 - WikiTruthiness
11:00 - TwitEvents
11:15 - Freshipes
11:30 - One Click Books
11:45 - ProjectNomNom
12:00 - Read.me
12:15 - Developing Regions

- Times are approximate
- 12 min presentation
- 2 min questions

Next groups sets up during Qs

Presentation Guidelines

- **Every group member should talk**
- **Practice, practice**
  - Use time wisely: 12 min
  - Get length right
- **Assumption: own laptop**
  - Else…. 8:30am

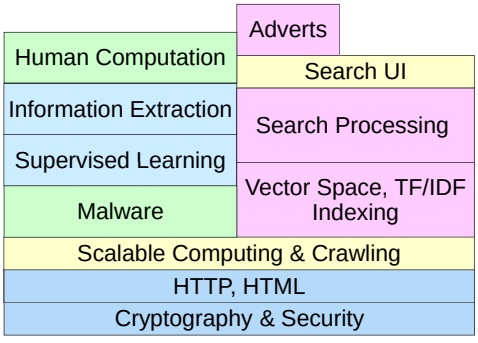
Presentation Content

- Aspirations & Reality
- Demo
- Suprises
  - **What was harder or easier than expected?**
- What You learned
- Experiments & Validation
- Division of Labor

Final Reports

- **Due Friday 12/17 12:00 noon**
  - Hardcopy of paper
  - Digital copy of paper
  - Digital copy of in-class presentation
  - Code
- **Content**
  - See Web (and see past examples)
- **Length**

CSE 454 Overview



Cyptography

- Symmetric + asymmetric ciphers
- Stream + block ciphers; 1-way hash
- $Z=Y^X \bmod N$

DNS, HTTP, HTML

- Get, put, post
- Cookies, log file analysis

DNS

- Hierarchical namespace
- Susceptible to DOS attacks
- Recent news: Google Public DNS

HTTP / HTTPS

- Get, put, post
- Cookies, log file analysis

SSL / PCT / TLS

You (client) **Merchant (server)**

Here are the protocols + ciphers I know

Let's use this protocol; here's my pub key, a cert + nonce

Using your key, I've encrypted a random sym key + nonce

Symmetric Ciphers
- Stream + block

Asymmetric ciphers
- $Z=Y^X \bmod N$
- Discrete root finding = hard w/o factors of N

1-way hash

Securing Requires Threat Modeling

- Who are potential attackers?
- What are their
  Resources?
  Capabilities?
  Motives?
- What assets are you trying to protect?
- What might attackers try, to get assets?

- Must answer these questions early!

Slide by Yoshi Kohno

HTML

- "Semantic Formatting"
- Link Extraction

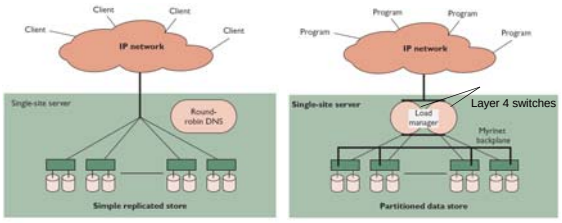
AJAX

- Javascript
- Async communication of XML

HTML5

- Form handling
- Drag & draw canvas
- Native video

Common Types of Clusters

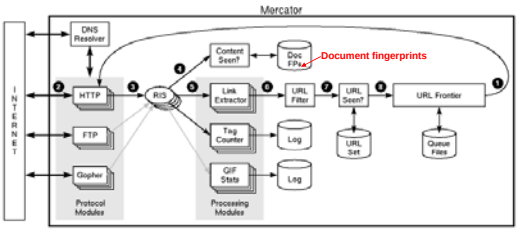


Simple Web Farm | Search Engine Cluster

Inktomi (2001) Supports programs (not users) Persistent data is partitioned across servers:
⇑ capacity, but ⇓ data loss if server fails

From: Brewer *Lessons from Giant-Scale Services*

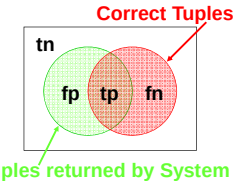
Structure of Mercator Spider

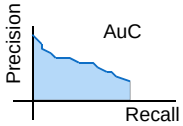


1. Remove URL from queue
2. Simulate network protocols & REP
3. Read w/ RewindInputStream (RIS)
4. Has document been seen before? (checksums and fingerprints)
5. Extract links
6. Download new URL?
7. Has URL been seen before?
8. Add URL to frontier

The Precision / Recall Tradeoff

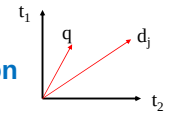
- **Precision** $\frac{tp}{tp + fp}$
  - Proportion of selected items that are correct
- **Recall** $\frac{tp}{tp + fn}$
  - Proportion of target items that were selected
- **Precision-Recall curve**
  - Shows tradeoff





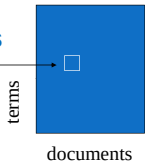
Vector Space Representation

- Dot Product as Similarity Metric



TF-IDF for Computing Weights

- $w_{ij} = f(i,j) * log(N/n_i)$
- Where q = … $word_i$…
- N = |docs| n_i = |docs with $word_i$|



- **How Process Efficiently?**

Thinking about Efficiency

- **Clock cycle: 2 GHz**
  - Typically *completes* 2 instructions / cycle
    - ~10 cycles / instruction, but pipelining & parallel execution
  - Thus: 4 billion instructions / sec
- **Disk access: 1-10ms**
  - Depends on seek distance, published average is 5ms
  - Thus perform 200 seeks / sec
  - (And we are ignoring rotation and transfer times)
- **Disk is *20 Million* times slower !!!**

Inverted Files for Multiple Documents

LEXICON

WORD	NDOCS	PTR
jezebel	20	
jezer	3	
jezerit	1	
jeziah	1	
jeziel	1	
jezliah	1	
jezoar	1	
jezrahliah	1	
jezreel	39	

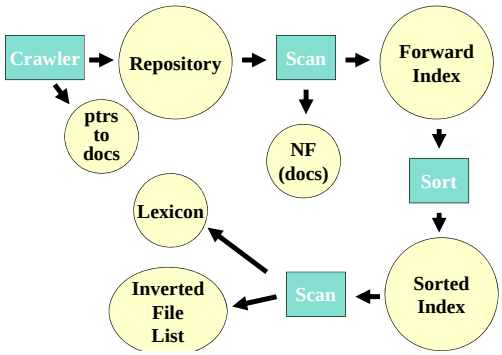
"jezebel" occurs 6 times in document 34, 3 times in document 44, 4 times in document 56 . . .

DOCID	OCCUR	POS 1	POS 2	…			
34	6	1	118	2087	3922	3981	5002
44	3	215	2291	3010			
56	4	5	22	134	992		
566	3	203	245	287			
67	1	132					
. . .							
107	4	322	354	381	405		
232	6	15	195	248	1897	1951	2192
677	1	481					
713	3	42	312	802			

OCCURENCE INDEX

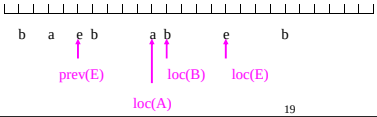
- **One method. Alta Vista uses alternative**

How Inverted Files are Created



AltaVista

- **Basic Framework**
  - Flat 64-bit address space
  - Index Stream Readers: Loc, Next, Seek, Prev
  - Constraints
- **Let E be ISR for word** enddoc
- **Constraints for conjunction a AND b**
  - prev(E) ≤ loc(A)
  - loc(A) ≤ loc(E)
  - prev(E) ≤ loc(B)
  - loc(B) ≤ loc(E)



Beyond Size - Search User Interfaces

- **Specialized Search**
- **Suggestions**
- **Spelling Correction**

Web Search at 15

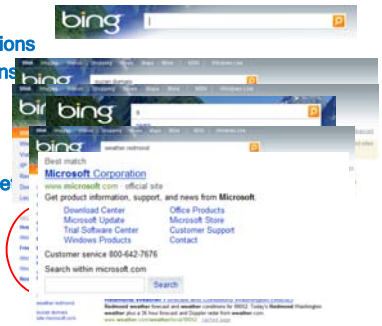
How search is accessed

- **Number of pages indexed**
  - 7/94 Lycos -
  - 95 - 10^6 millions
  - 97 - 10^7
  - 98 - 10^8
  - 01 - 10^9 billions
  - 05 - 10^10 …
- **Types of content**
  - Web pages, newsgroups
  - Images, videos, maps
  - News, blogs, spaces
  - Shopping, local, desktop
  - Books, papers, many formats
  - Health, finance, travel …

Slide by Susan Dumais

Support for Searchers

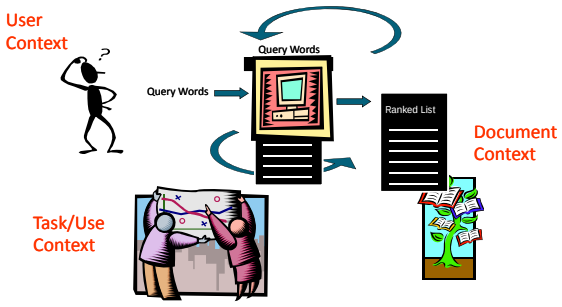
- **The search box**
- **Spelling suggestions**
- **Query suggestions**
- **Auto complete**
- **Inline answers**
- **Richer snippets**
- **But, we can do be**



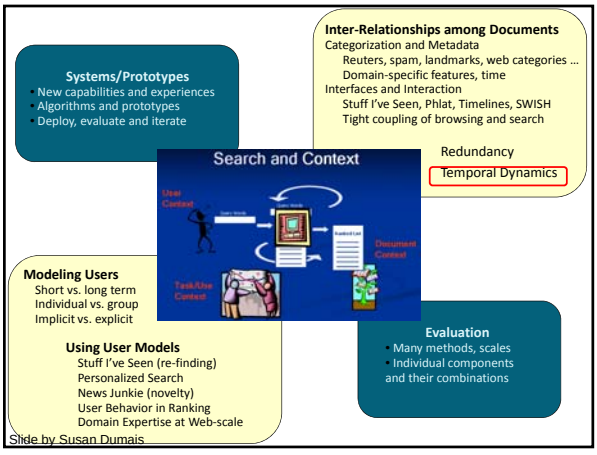
Slide by Susan Dumais

Search and Context

Search Today



Slide by Susan Dumais

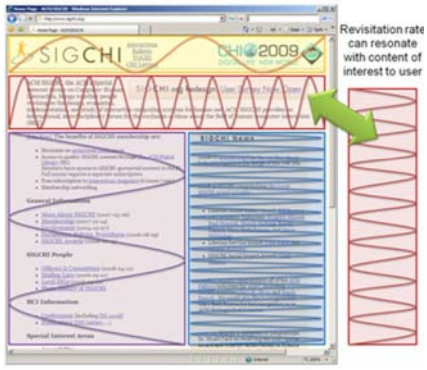


Slide by Susan Dumais

50-80% Page Views = Revisits

Cluster Group	Name	Shape	Description
Fast Revisits (< hour) 23611 pages	F1		Pornography & Spam, Hub & Spoke, Shopping & Reference Web sites, Auto refresh, Fast monitoring
	F2		
	F3		
	F4		
	F5		
Medium (hour to day) 9421 pages	M1		Popular homepages, Communication, .edu domain, Browser homepages
	M2		
Slow Revisits (> day) 18422 pages	S1		Entry pages, Weekend activity, Search engines used for revisitation, Child-oriented content, Software updates
	S2		
	S3		
	S4		
Hybrid 3334 pages	H1		Popular but infrequently used, Entertainment & Hobbies, Combined Fast & Slow

Resonance



A-B testing

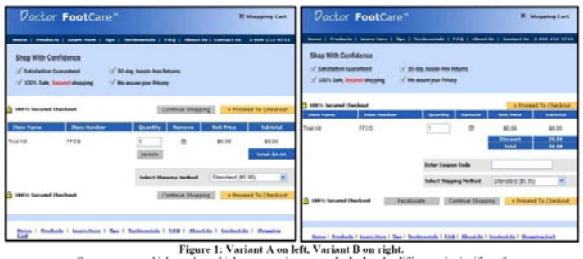
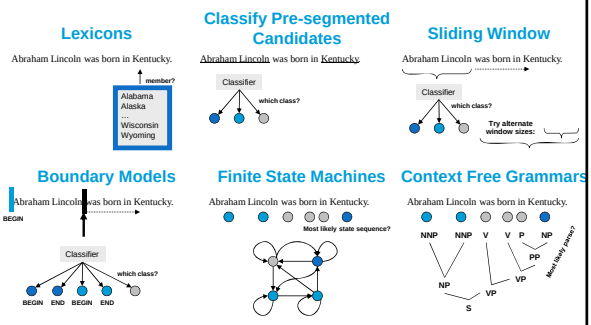


Figure 1: Variant A on left, Variant B on right.
Can you guess which one has a higher conversion rate and whether the difference is significant?

Nine Changes in Site Above

Landscape of IE Techniques: Models



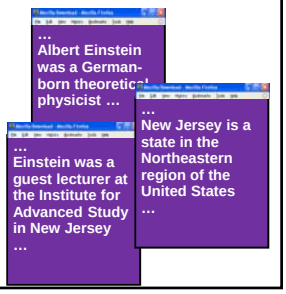
Any of these models can be used to capture words, formatting or both.

Slides from Cohen & McCallum

Motivating Vision

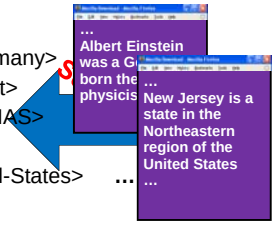
Next-Generation Search = Information Extraction
+ Ontology
+ Inference

Which German Scientists Taught at US Universities?



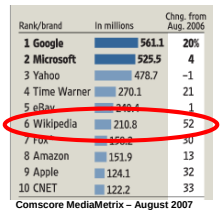
Next-Generation Search

- Information Extraction
  - <Einstein, Born-In, Germany>
  - <Einstein, ISA, Physicist>
  - <Einstein, Lectured-At, IAS>
  - <IAS, In, New-Jersey>
  - <New-Jersey, In, United-States> ...
- Ontology
  - Physicist (x) ➔ Scientist(x) ...
- Inference
  - Einstein = Einstein ...



Why Wikipedia?

- **Comprehensive**
- **High Quality**
  [Giles Nature 05]
- **Useful Structure**
  - Unique IDs & Links
  - Infoboxes
  - Categories & Lists
  - First Sentence
  - Redirection pages
  - Disambiguation pages
  - Revision History
  - Multilingual



Rank/brand	In millions	Chng. from Aug. 2006
1 Google	561.1	20%
2 Microsoft	525.5	4
3 Yahoo	478.7	–1
4 Time Warner	270.1	21
5 eBay	248.4	1
6 Wikipedia	210.8	52
7 Fox	158.2	30
8 Amazon	151.9	13
9 Apple	124.1	32
10 CNET	122.2	33

Comscore MediaMetrix – August 2007

Cons

- Natural-Language
- Missing Data
- Inconsistent
- Low Redundancy

The Intelligence in Wikipedia Project

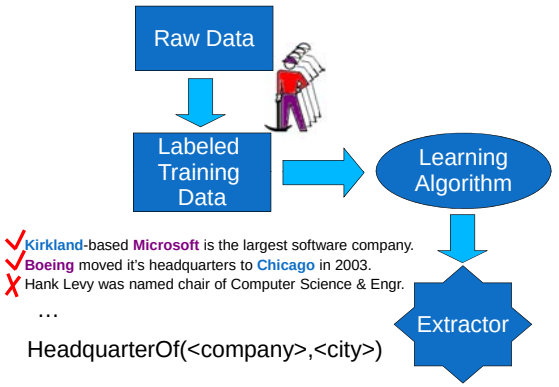
Outline

1. **Self-Supervised Extraction** from Wikipedia text (bootstrapping to the greater Web)
2. Automatic **Taxonomy Generation**
3. Scalable **Probabilistic Inference** for Q/A

Outline

1. **Self-Supervised Extraction** from Wikipedia text
   - Training on Infoboxes
   - Improving Recall – Shrinkage, Retraining, Web Extraction
   - Community Content Creation
2. Automatic **Taxonomy Generation**
3. Scalable **Probabilistic Inference** for Q/A

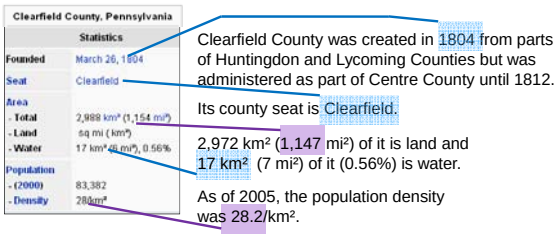
Traditional, Supervised I.E.



Kylin: Self-Supervised Information Extraction from Wikipedia

[Wu & Weld CIKM 2007]

From infoboxes to a training set



Clearfield County, Pennsylvania	
Statistics	
Founded	March 26, 1804
Seat	Clearfield
Area	
- Total	2,988 km² (1,154 mi²)
- Land	sq mi (km²)
- Water	17 km² (6 mi²), 0.56%
Population	
- (2000)	83,382
- Density	28/km²

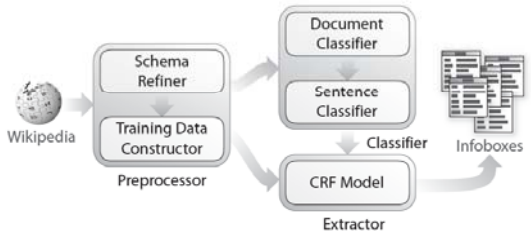
Clearfield County was created in 1804 from parts of Huntingdon and Lycoming Counties but was administered as part of Centre County until 1812.

Its county seat is Clearfield.

2,972 km² (1,147 mi²) of it is land and 17 km² (7 mi²) of it (0.56%) is water.

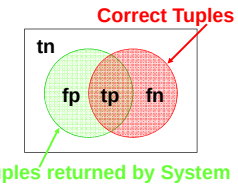
As of 2005, the population density was 28.2/km².

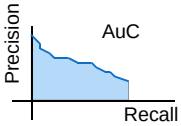
Kylin Architecture



The Precision / Recall Tradeoff

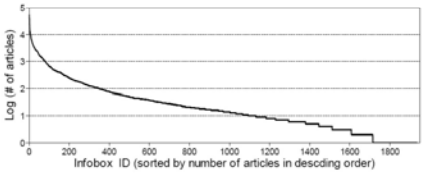
- **Precision** $\frac{tp}{tp + fp}$
  - Proportion of selected items that are correct
- **Recall** $\frac{tp}{tp + fn}$
  - Proportion of target items that were selected
- **Precision-Recall curve**
  - Shows tradeoff





Preliminary Evaluation

- Kylin Performed Well on Popular Classes:
  - Precision: mid 70% ~ high 90%
  - Recall: low 50% ~ mid 90%
- ... Floundered on Sparse Classes – Little Training Data

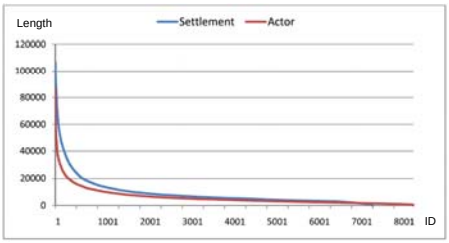


82% < 100 instances; *40% <10 instances*

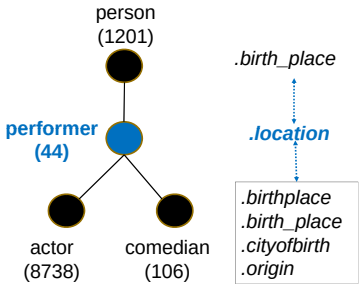
Long-Tail 2: Incomplete Articles

- Desired Information Missing from Wikipedia

800,000/1,800,000(44.2%) *stub* pages *[July 2007 of Wikipedia]*



Shrinkage?

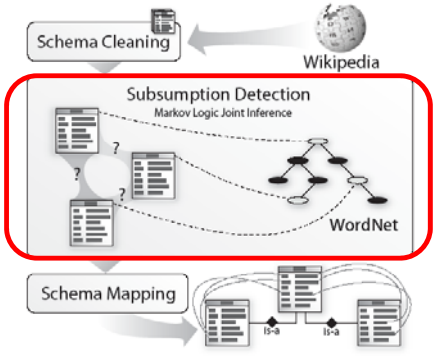


Outline

1. **Self-Supervised Extraction** from Wikipedia Text
   - Training on Infoboxes
   - Improving Recall – Shrinkage, Retraining, Web Extraction
   - Community Content Creation
2. Automatic **Taxonomy Generation**
3. Scalable **Probabilistic Inference** for Q/A

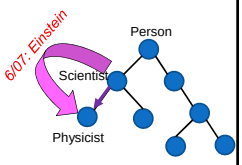
KOG: Kylin Ontology Generator

[Wu & Weld, WWW08]

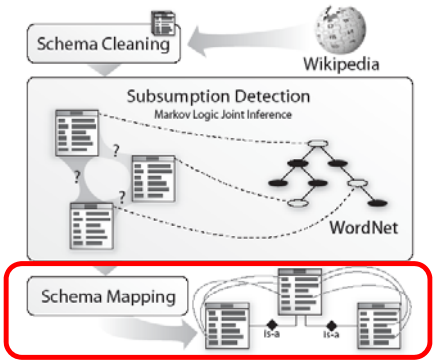


Subsumption Detection

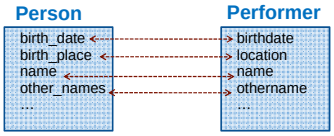
- **Binary Classification Problem**
- **Nine Complex Features**
  E.g., String Features
  … IR Measures
  … Mapping to Wordnet
  … Hearst Pattern Matches
  … Class Transitions in Revision History
- **Learning Algorithm**
  SVM & MLN Joint Inference



KOG Architecture



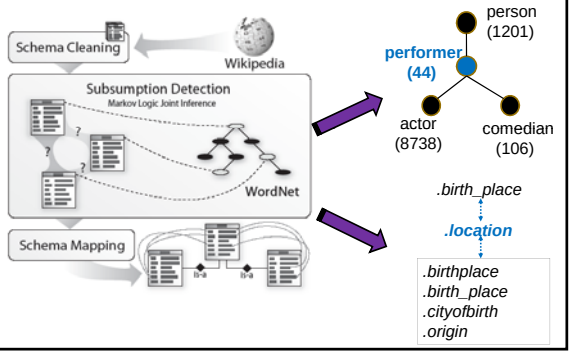
Schema Mapping



- **Heuristics**
  - Edit History
  - String Similarity
- **Experiments**
  - Precision: 94% Recall: 87%
- **Future**
  - Integrated Joint Inference

KOG: Kylin Ontology Generator

[Wu & Weld, WWW08]



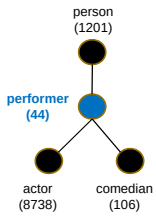
Outline

1. **Self-Supervised Extraction** from Wikipedia Text
   - Training on Infoboxes
   - **Improving Recall** – Shrinkage, Retraining, Web Extraction
   - Community Content Creation
2. Automatic **Ontology Generation**
3. Scalable **Probabilistic Inference** for Q/A

Improving Recall on Sparse Classes

[Wu et al. KDD-08]

- **Shrinkage**
  - Extra Training Examples from Related Classes
  - How Weight New Examples?



Improving Recall on Sparse Classes

[Wu et al. KDD-08]

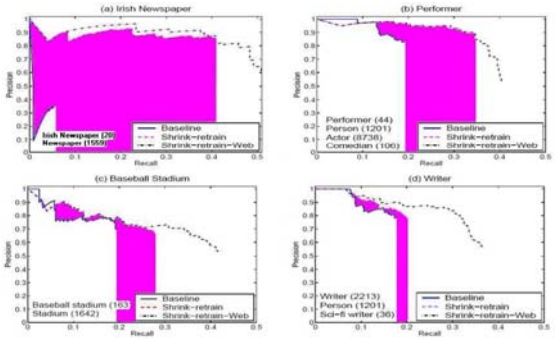
Retraining

- Compare Kylin Extractions with Tuples from Textrunner
- Additional Positive Examples
- Eliminate False Negatives

TextRunner [Banko et al. IJCAI-07, ACL-08]

- Relation-Independent Extraction
- Exploits Grammatical Structure
- CRF Extractor with POS Tag Features

Recall after Shrinkage / Retraining…

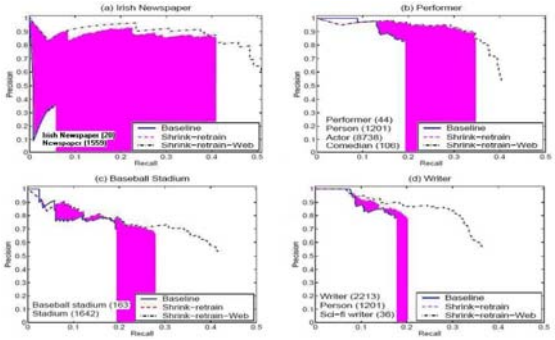


Improving Recall on Sparse Classes

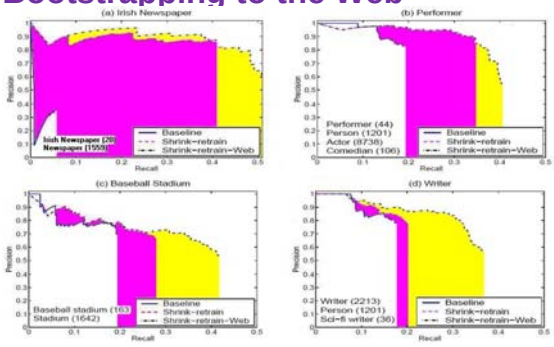
[Wu et al. KDD-08]

- **Shrinkage**
- **Retraining**
- **Extract from Broader Web**
  - **44% of Wikipedia Pages = "stub"**
    - **Extractor quality irrelevant**
  - **Query Google & Extract**
    - **How maintain high precision?**
    - **Many Web pages noisy, describe multiple objects**
    - **How integrate with Wikipedia extractions?**

Recall after Shrinkage / Retraining…



Bootstrapping to the Web



Outline

1. **Self-Supervised Extraction** from Wikipedia Text
   - Training on Infoboxes
   - Improving Recall – Shrinkage, Retraining, Web Extraction
   - **Community Content Creation**
2. Automatic **Ontology Generation**
3. Scalable **Probabilistic Inference** for Q/A

Problem

- **Information Extraction is Imprecise**
  - Wikipedians Don't Want 90% Precision
- **How Improve Precision?**
  - People!

Motivation

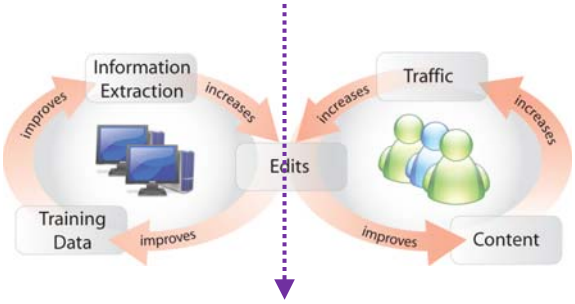
Altruism

Self-Interest

Self-Esteem, Community

Money

Accelerate



Contributing as a Non-Primary Task

- **Encourage contributions**
- **Without annoying or abusing readers**
  - Compared 5 different interfaces

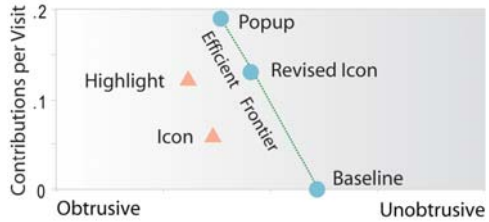
Adwords Deployment Study

[Hoffman *et al.* 2008]

- **2000 articles containing writer infobox**
- **Query for "ray bradbury" would show**

- **Redirect to mirror with injected JavaScript**
- **Round-robin interface selection:**
  - baseline, popup, highlight, icon
- **Track clicks, load, unload, and show survey**

Results



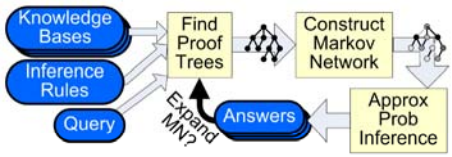
- Contribution Rate
  1.6% → 13%
- Additional Training Data
  Improved Extraction Performance

Outline

1. **Self-Supervised Extraction** from Wikipedia
2. Automatic **Ontology Generation**
3. Scalable **Probabilistic Inference** for Q/A

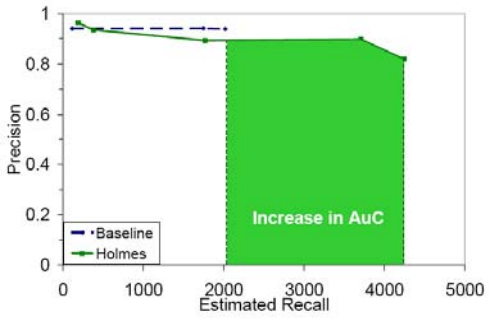
Scalable Probabilistic Inference

[Schoenmacker *et al.* 2008]

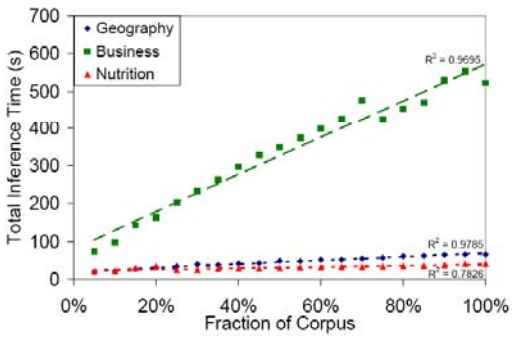


- **Eight MLN Inference Rules**
  - **Transitivity of predicates, *etc.***
- **Knowledge-Based Model Construction**
- **Tested on 100 Million Tulples**
  - **Extracted by Textrunner from Web**

Effect of Limited Inference



Inference Appears Linear in |Corpus|



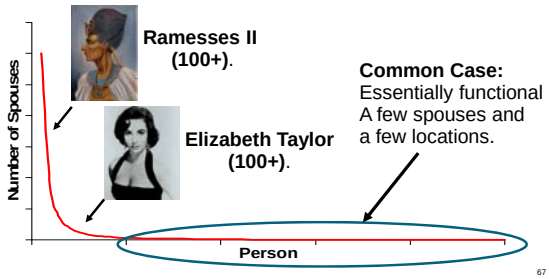
How Can This Be True?

- **Q(X,Y,Z) <= `Married`(X,Y) ∧ `LivedIn`(Y,Z)**
- **Worst Case: Some person y' married *everyone*, and lived in *every* place:**

  |Q(X,y',Z)| = |`Married`|*|`LivedIn`| = $O(n^2)$

Why is inference expensive?

- Q(X,Y,Z) <= Married(X,Y) ∧ LivedIn(Y,Z)



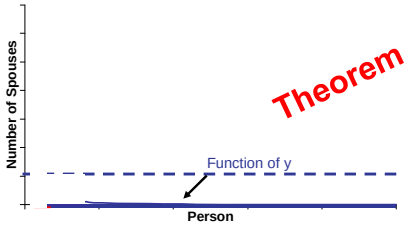
67

Approximately Pseudo-Functional Relations

E.g. Married(X,Y) Most Y have only 1 spouse mentioned

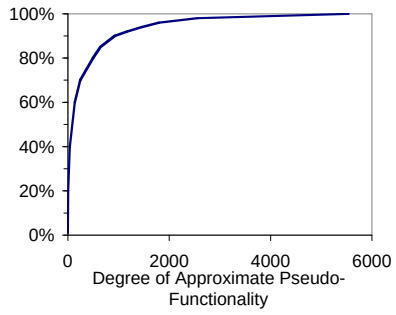
People in $\mathcal{Y}_6$ have at most a constant k_M spouses each

People in $\mathcal{Y}_6$ have at most k_M*log $|\mathcal{Y}_6|$ spouses in total



68

Prevalence of APF relations



69

Related Work

- Unsupervised Information Extraction
  - SNOWBALL *[Agichtein & Gravano ICDL00]*
  - MULDER *[Kwok et al. TOIS01]*
  - AskMSR *[Brill et al. EMNLP02]*
  - KnowItAll *[Etzioni et al. WWW04, ...]*
  - TextRunner *[Banko et al. IJCAI07, ACL-08]*
  - KNEXT *[VanDurme et al. COLING-08]*
  - WebTables [Cafarella *et al. VLDB-08]*
- Ontology Driven Information Extraction
  - SemTag and Seeker *[Dill WWW03]*
  - PANKOW *[Cimiano WWW05]*
  - OntoSyphon *[McDowell & Cafarella ISWC06]*

Related Work II

- Other Uses of Wikipedia
  - **Semantic Distance Measure** [Ponzetto&Strube07]
  - **Word-Sense Disambiguation** [Bunescu&Pasca06, Mihalcea07]
  - **Coreference Resolution** [Ponzetto&Strube06, Yang&Su07]
  - **Ontology / Taxonomy** [Suchanek07, Muchnik07]
  - **Multi-Lingual Alignment** [Adafre&Rijke06]
  - **Question Answering** [Ahn *et al.*05, Kaisser08]
  - **Basis of Huge KB** [Auer *et al.*07]

Conclusion

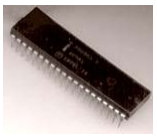
- **Self-Supervised Extraction** from Wikipedia
  - Training on Infoboxes
    - Works well on popular classes
  - Improving Recall – Shrinkage, Retraining, Web Extraction
    - High precision & recall - even on sparse classes, stub articles
  - Community Content Creation
- Automatic **Ontology Generation**
  - Probabilistic Joint Inference
- Scalable **Probabilistic Inference** for Q/A
  - Simple Inference - Scales to Large Corpora
  - Tested on 100 M Tuples

Internet History 1960s

1960 Nelson → Xanadu
1965 Salton "SMART Doc Retrieval" CACM
1969 First ARPAnet message

Internet History 1970s

1960 Nelson → Xanadu
1965 Salton "SMART Doc Retrieval" CACM
1969 First ARPAnet message
1972 Email
1974 Intel 8080 CPU; Design of TCP



Internet History 1980s

1960 Nelson → Xanadu
1965 Salton "SMART Doc Retrieval" CACM
1969 First ARPAnet message
1972 Email
1974 Intel 8080 CPU; Design of TCP
1983 ARPAnet (1000 nodes) uses TCP/IP; DNS
1990 Berners-Lee creates WWW

Internet History 1990s

1960 Nelson → Xanadu
1965 Salton "SMART Doc Retrieval" CACM
1969 First ARPAnet message
1972 Email
1974 Intel 8080 CPU; Design of TCP
1983 ARPAnet (1000 nodes) uses TCP/IP; DNS
1990 Berners-Lee creates WWW
1994 Pinkerton's Webcrawler; YHOO, AMZN, Netscape
1997 Goto.com
1998 Google

Google

Internet History 2000s

1960 Nelson → Xanadu
1965 Salton "SMART Doc Retrieval" CACM
1969 First ARPAnet message
1972 Email
1974 Intel 8080 CPU; Design of TCP
1983 ARPAnet (1000 nodes) uses TCP/IP; DNS
1990 Berners-Lee creates WWW
1994 Pinkerton's Webcrawler; YHOO, AMZN, Netscape
1997 Goto.com
1998 Google

2001 Wikipedia; Bittorrent (by 2009 → 55% all traffic)
2004-6 Facebook, Digg; Youtube, MTurk; Twitter

Adoption