



IIT-H

## Web Mining

# Lecture 18: Mining Structured Information from the Web (Part 2)

Manish Gupta

1<sup>st</sup> Oct 2013

Slides borrowed (and modified) from  
Michael Caferella's slides on WebTables  
Sunita Sarawagi and Rahul Gupta's slides on WWT

# Recap of Lecture 17: Mining Structured Information from the Web (Part 1)

- Introduction to Information Extraction
- Wrapper Induction
- List Extraction using Automatic Wrapper Generation
- ~~Information Extraction for Unstructured Data~~

# Announcements

- No Midsem-2 for our course.
- How was assignment 3?
  - Do not do find email id manually!

# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - Answering Table Augmentation Queries from Unstructured Lists on the Web
  - Annotating Tables with Ontological Links
- Extracting Sets from the Web

# Today's Agenda

- **Extracting Top-K Lists from the Web**
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - Answering Table Augmentation Queries from Unstructured Lists on the Web
  - Annotating Tables with Ontological Links
- Extracting Sets from the Web

# Information Extraction from Top-K List Webpages

- Zhixian Zhang, Kenny Q. Zhu, Haixun Wang, Hongsong Li. Automatic Extraction of Top-k Lists from the Web. ICDE 2013.
- Typical titles
  - 20 Most Influential Scientists Alive Today
  - Twelve Most Interesting Children's Books in USA
  - 10 Hollywood Classics You Shouldn't Miss
  - .net Awards 2011: top 10 podcasts
- Title of a top-K page contains
  - A number k
  - A topic or concept the items belongs to
  - A ranking criterion (*Influential, Interesting, and You Shouldn't Miss, Top, Best*)
  - Two optional pieces of information: time and location

# Top-K List Webpage

## 10 Top Windows 8 Apps

By Sandy Berger - Tech Page One January 15 2013



[aNewDomain.net](#) — Using Windows 8 apps is a first for many of us. Yet these tiny programs can be very beneficial and can greatly enhance your computing experience. Here are my top ten Windows 8 apps.




### 1. Jujuba's free Clock app


Microsoft didn't include a clock in the Windows 8 Metro desktop, and many people miss it. I like this one because it's a live tile that shows both the date and time. When open, it gives you a large analog clock and calendar.



### 2. News Bento

This news app offers a great Windows 8 app experience. You can select from its growing list of news sources or add an RSS feed from any website. Bento's news articles expand to fill the screen making them easily readable and you can use the Share charm to share articles.





**Sandy Berger**  
Contributor at Tech  
Page One

Sandy Burger, based in Pinehurst, N.C., is a veteran tech journalist and regular contributor to Tech Page One. As senior editor at aNewDomain.net, Sandy covers tech tips and tricks, apps and gadgets in general. Email her at [Sandy@aNewDomain.net](mailto:Sandy@aNewDomain.net).

About  
Tech Page One

### Trending

- Top 7 social media marketing trends for 2014
- PowerEdge VRTX: Powerful, compact, ROBO-friendly
- No More Mouse: Meet the Future of Computer's Oldest Sidekick
- LinkedIn tips for reaching affluent audiences
- Slideshow: A look back at VMworld 2013

# Why Target Top-K Pages for Information Extraction

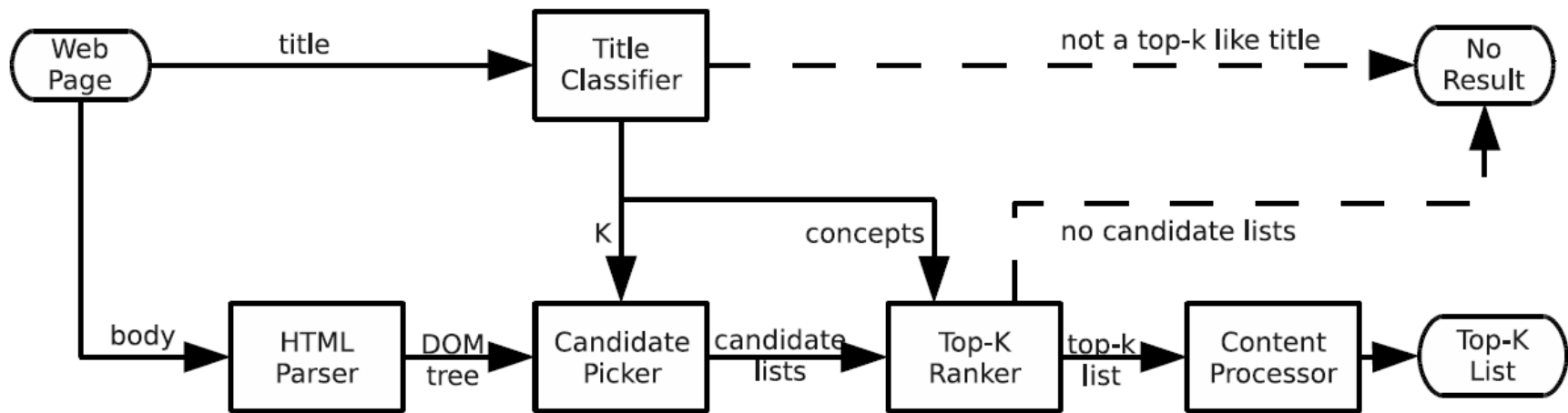
- Top-k data on the web is *large and rich*. 1.7 million top-k lists from a web corpus of 1.6 billion web pages
- Top-k data is of *high quality*
  - All top-k pages have a common style: the page title contains the number and the concept of items in the list.
- Top-k data is *ranked*
- Top-k data has *interesting semantics*. Top-k lists are often manually composed.
- Top-k data acquisition is an important step towards automatically constructing a universal knowledge base that includes a large number of known concepts and their instances.



# Top-K Extraction Steps

- *Recognize top-k pages*
  - *Title recognition*: Convert each top-k title into a 5-tuple:  $\langle k, \text{concept}, \text{ranking criterion}, \text{time}, \text{location} \rangle$  where time and location are optional
- *List extractor: Extract top-k lists*
  - Page is usually in natural language text and is not formatted using tags such as  $\langle \text{ul} \rangle$ ,  $\langle \text{li} \rangle$ , and  $\langle \text{table} \rangle$
  - Knowing k and using a general purpose knowledge base helps.
    - Each item represents an entity that is an instance of the concept c in an is-a taxonomy
- *Content extractor: Understand list content*
  - *Find schema of list*
  - *Extract rich attributes for each item in the list*

# System Overview

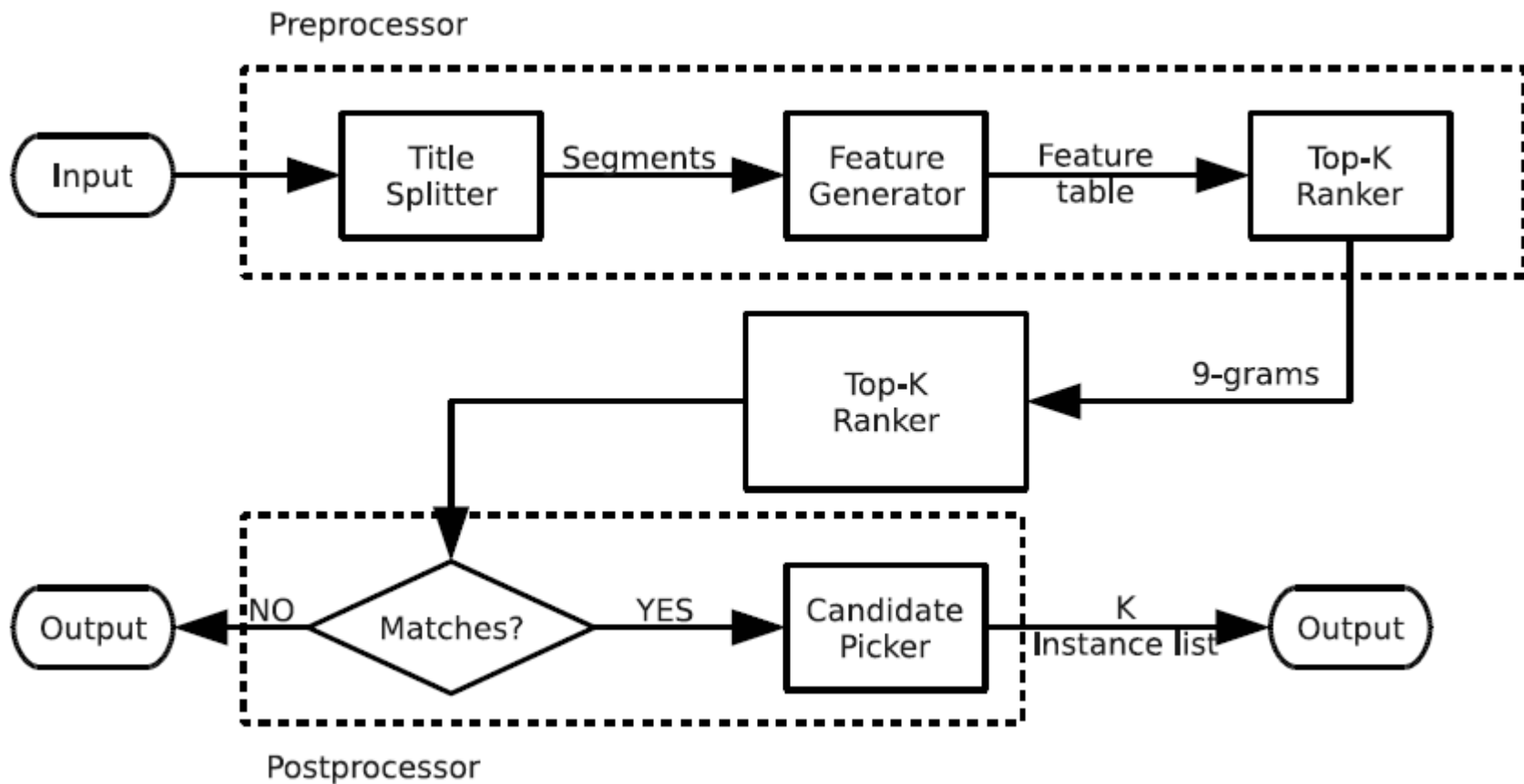


- Title Classifier, which attempts to recognize the page title of the input web page
- Candidate Picker, which extracts all potential top-k lists from the page body as candidate lists
- Top-K Ranker, which scores each candidate list and picks the best one
- Content Processor, which post-processes the extracted list to further produce attribute values, etc.

# Title Classifier

- CRF model
- Feature set: Lexical features like word, lemma (root form of word), POS tag, concept (whether the word forms a string suffix of a concept in a knowledge base)
- Classifier also returns information like  $k$  and set of concepts mentioned in title
- “ $k$  recognition task” is a sequence labeling problem: Each word in the title is considered a token in a sequence, and is  $k$  or *not*  $k$ .

# Title Classifier



# Candidate Picker

- List should have k items
- Visually, it should be rendered as k vertically or horizontally aligned regular patterns
- Structurally, it is presented as a list of HTML nodes with identical *tag path*
  - A *tag path* is the path from the root node to a certain tag node
  - *Tag Path Clustering Method*: algorithm recursively computes the tag path for each node, and groups text nodes with an identical tag path into one node list.
- One of these three conditions should be satisfied
  - **Index**: There exists an integer number (in sequence) in front of every list item, serving as a rank or index
  - **Highlighting Tag**: The tag path of the candidate list contains at least one tag among *<b>*, *<strong>*, *<h1-h6>* for highlighting
  - **Table**: The candidate list is shown in a table format

# Top-K Ranker

- Rank candidate set and pick top-ranked list with highest score
  - P-Score: correlation between the list and title
    - $\geq 1$  list items should be instances of the central concept in title
    - $$P - Score = \frac{1}{k} \sum_{n \in L} \frac{LMI(n)}{Len(n)}$$
      - LMI(n) is the word count of the longest matched instance in the text of node n
      - Len(n) is the word count of the whole text in node n
  - V-Score: visual area occupied by a list
    - main list of the page tends to be larger
    - $$Area(L) = \sum_{n \in L} (TextLength(n) \times FontSize(n)^2)$$
  - Other Features

Name	Type	Description	Positives	Negatives
Word	Boolean	Existence of a certain word in the list text	Indexes (e.g., "25.", "12.")	"Contact Us", "Privacy Policy"
Tag Name	Boolean	The tag name of the list nodes	<h2>, <strong>, ...	<input>, <iframe>
Attribute	Boolean	Existence of a attribute token in the list nodes	"articleBody", "main"	"comment", "breadcrumb"
Word Count	Integer	The average word count of the list items	/	/
Length Variance	Float	The standard variance of the lengths of the list items	/	/

# Content Processor

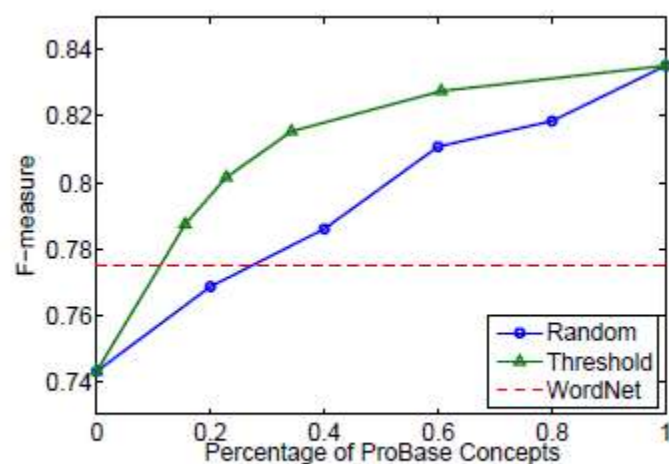
USA Today (Arlington, Va.)
Wall Street Journal (New York, N.Y.)
Times (New York, N.Y.)
Times (Los Angeles)
Post (Washington, DC)
Tribune (Chicago)
Daily News (New York, N.Y.)
Inquirer (Philadelphia)
Post/Rocky Mountain News (Denver)



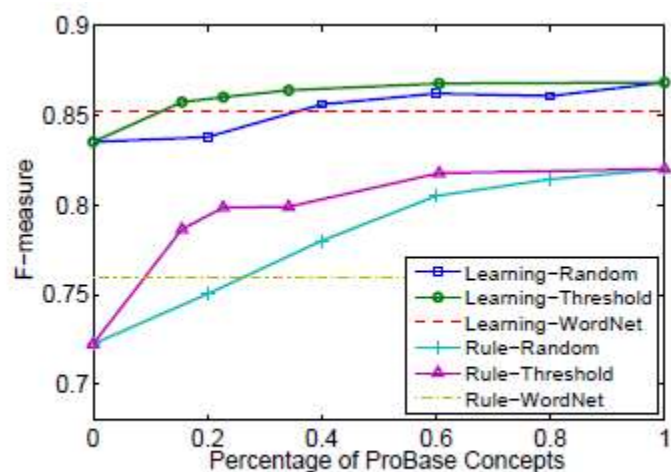
Newspaper	American city
USA Today	Arlington, Va.
Wall Street Journal	New York, N.Y.
Times	New York, N.Y.
Times	Los Angeles
Post	Washington, DC
Tribune	Chicago
Daily News	New York, N.Y.
Inquirer	Philadelphia
Post/Rocky Mountain News	Denver

- *Infer the structure of text nodes*
  - *Using delimiters*
- *Conceptualize the list attributes*
  - *Infer schema for attributes*
  - *Three methods*
    - **Table head: <th>**
    - **Attribute/value pair:** If every element of a column contains the same text and ends with a colon, we will consider that column as the attribute column and the column to the right as the value column.
    - **Column content:** Use Probase to identify the best matching concept for the column
- *Detect when and where*
  - *NER*
  - The location entity must be in the main segment of the title.
  - The previous word of the location entity must be a proper preposition such as “in”, “at”, “of”, etc.
  - The previous word of the time entity must be “during”, “before” and “after”.

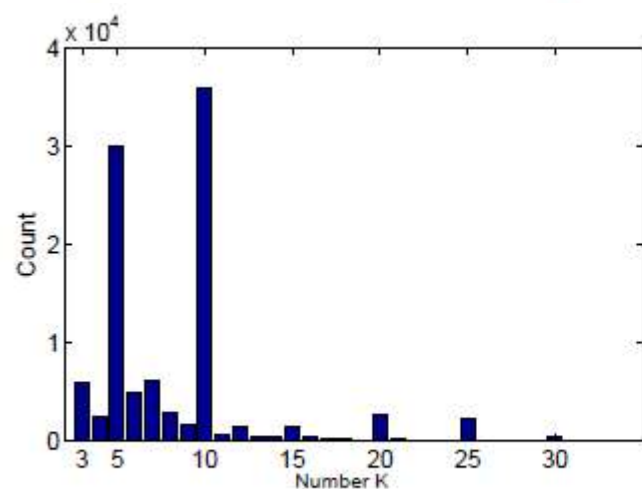
# Results



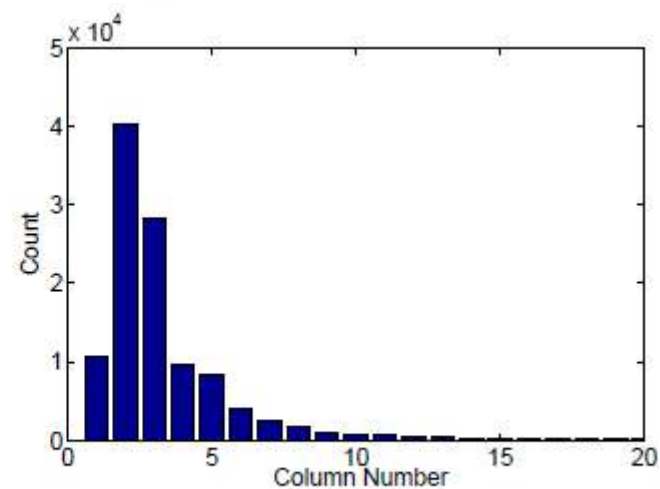
(a) Title Recognition with ProBase subsets



(b) List Extraction with ProBase subsets



(e) The Dist. of Number K



(f) The Dist. of Number of Attributes



# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - Answering Table Augmentation Queries from Unstructured Lists on the Web
  - Annotating Tables with Ontological Links
- Extracting Sets from the Web

# Extraction of Records containing User-Generated Content (UGC)

- Xinying Song, Jing Liu, Yunbo Cao, Chin-Yew Lin, and Hsiao-Wuen Hon. Automatic Extraction of Web Data Records Containing User-Generated Content. CIKM 2010

## Data record

- post-date
- author
- author-profile
- content
- ...



UGC



UGC

# Extracting Web Data Records Containing UGC

- When a web data record includes a large portion of free-format UGC, the similarity test between records may fail, which in turn results in lower performance
  - Mining Data Records (MDR) identifies a list of records by conducting a similarity test against a pre-defined threshold for two sub-trees in the DOM tree of a web page.
- Certain domain constraints (e.g., *post-date*) can be used to design better similarity measures capable of circumventing the influence of UGC
- Anchor points provided by the domain constraints can also be used to improve the extraction process, which ends in an algorithm called MiBAT (Mining data records Based on Anchor Trees)

# Tree Alignment of the Two Posts

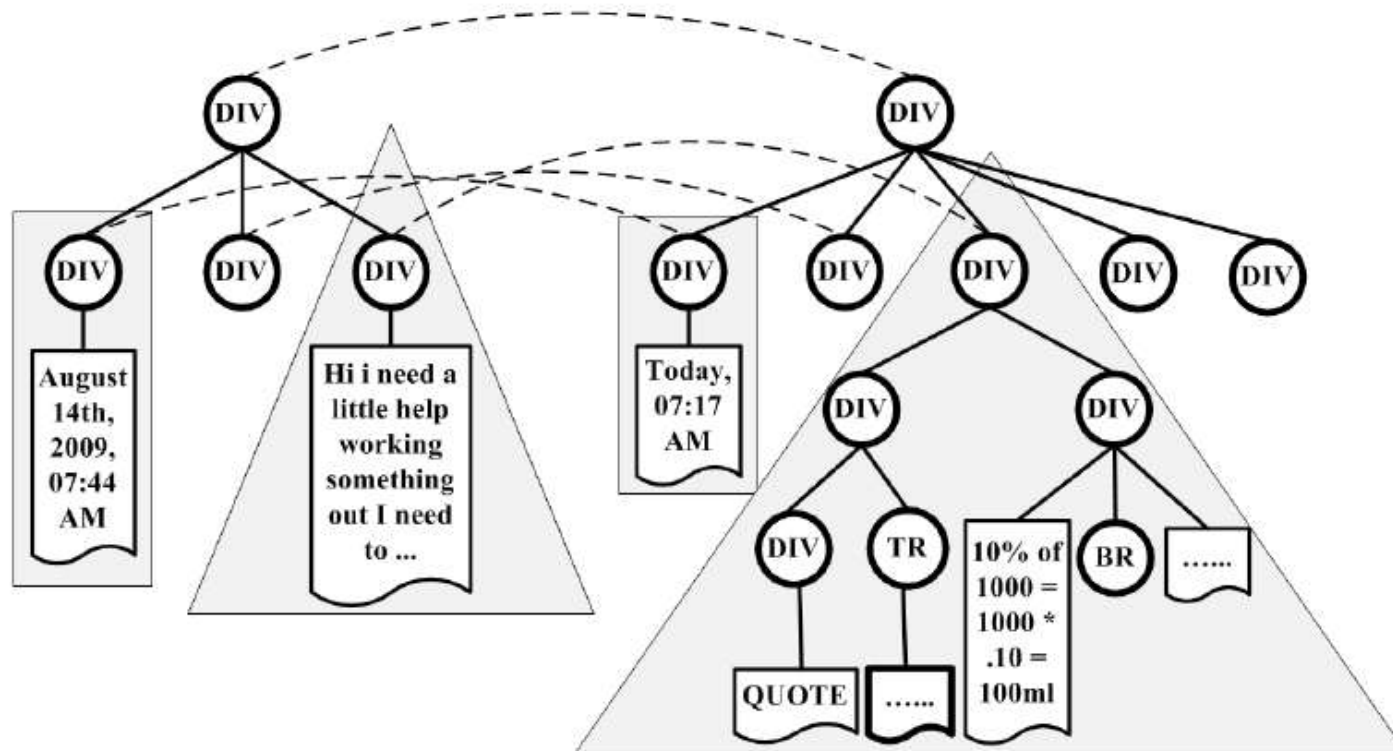
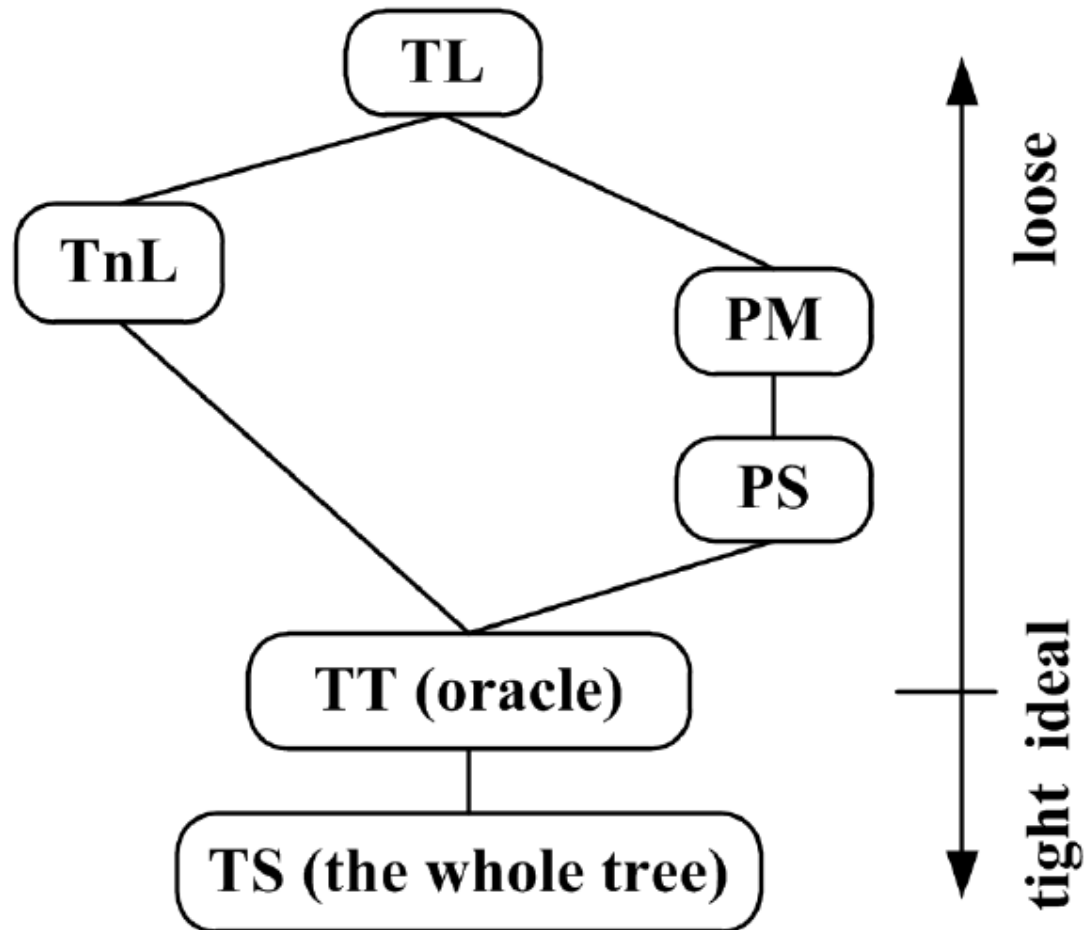


Figure 2: Tree match of two posts (gray triangles denote UGC while gray rectangles denote *post-date*)

# Similarity Measures

- Top Level Similarity (TL): MDR compares only the tag strings of the roots of two trees
  - Erroneous as it ignores the lower level structure
- Tree Similarity (TS): Compare two trees completely (along with the text)
  - Cannot handle UGC
- Template Tree (TT): Differentiate the regular template part from the irregular UGC part and compute the similarity on the template part to serve as the similarity measure
  - Difficult to differentiate UGC from the template
- Top N Level (TnL): Compute similarity using the top N levels of nodes only
- Pivot Match (PM): Compute similarity using pivot (e.g., timestamp)
  - easily identified (using regex)
  - always occurring as key structured data in every data record even across different types of media
- Pivot and Siblings (PS): Enlarge the tree fragment to consider to pivot+its siblings

# Relationships between Similarity Measures



# MiBAT (Mining Based on Anchor Trees): Assumptions

- **Same parent:** A list of data records are formed by child sub-trees under the same parent node
- **Same (record) length:** Each data record consists of the same number (maybe more than one) of adjacent child sub-trees.
- **Non-contiguity:** The data record list does not have to be consecutive. (ads could be in between)
- **Similarity:** Data records must be structurally similar with each other to some extent
  - all pairs of corresponding sub-trees have the same HTML tag at root
  - one pair of corresponding sub-trees, i.e. the anchor trees, must be judged as similar with respect to the domain-constraint guided similarity measure in use

# Mining Based on Anchor Trees (MiBAT)

- Along a traversal on the DOM tree, for each parent node
  - Find the anchor trees
  - Determine the record boundary, i.e. start offset and length, and extract data records around each anchor tree

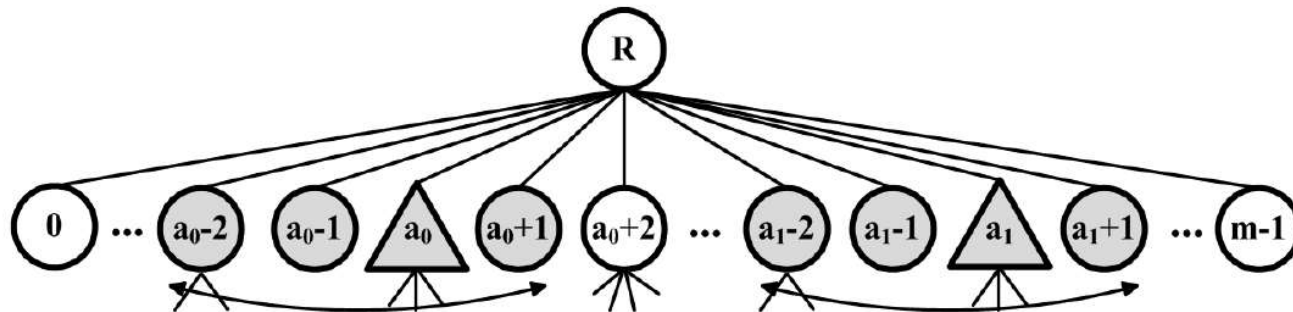


Figure 5: Mining data records based on anchor trees (triangle nodes represent anchor trees while every four consecutive gray nodes shows a data record)



# Finding Anchor Trees

- Candidate pivots: Nodes containing text in pivot format
  - Not all candidate pivots are real pivots, e.g., UGC nodes might also contain timestamp
  - Candidate pivots are real pivots only if they match between all data records

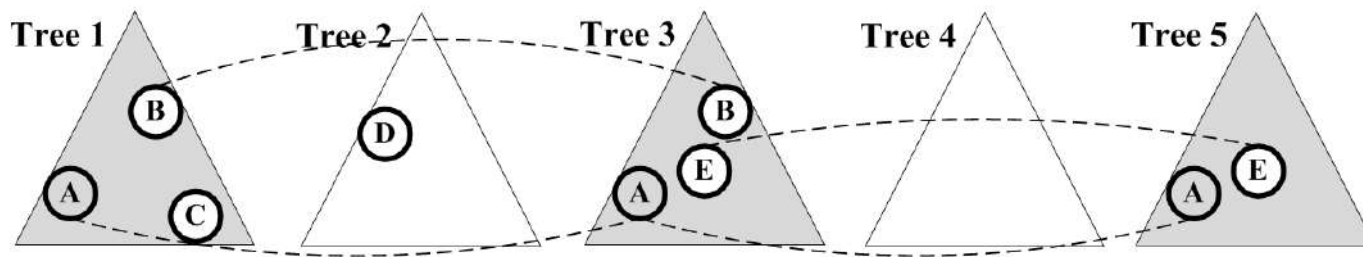
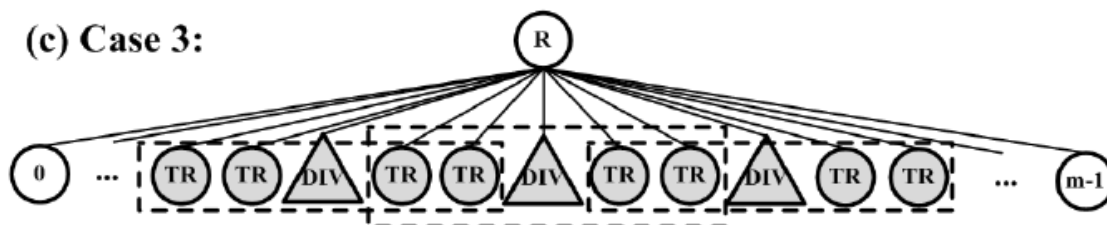
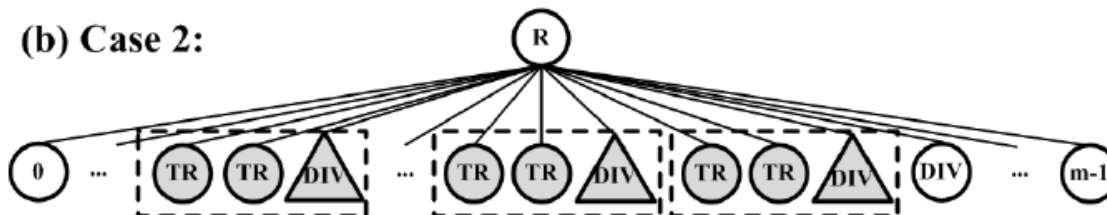
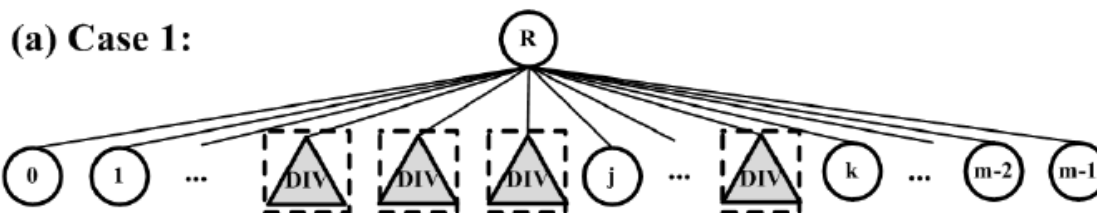


Figure 6: Finding anchor trees (each triangle denotes a tree; each circle denotes a candidate pivot; the gray node is the real pivot; gray triangles are anchor trees)

# Determine Record Boundary

- Given an anchor tree find
  - the start offset of a record relative to each anchor tree
  - the record length



**Case 1:** Two or more anchor trees are adjacent. Every single anchor tree forms a data record

**Case 2:** The length of each expansion is less than or equal to the minimal distance between two anchor trees. Sub-trees within each expansion form a data record.

**Case 3:** If the length of each expansion is greater than the minimal distance between two anchor trees, there must be two expansion regions overlapping on a few sub-trees.

(TR TR DIV TR TR)

Largest record length is the minimal distance of two anchor trees.

Find the start offset leading to the maximum similarity among all possible choices (-2, -1, 0).

# Main Region Selection and Results

- 2 requirements for the main region
  - The list of posts must have *post-date* in each record
  - The list of posts should occupy a majority of the page and each record should be somewhat similar to each other

Table 1: Post extraction (post level, prec./rec.)

	MDR	MDR2Pass	MiBAT
TL	55.8% / 70.0%	47.0% / 71.1%	-/-
TS	80.8% / 73.0%	60.2% / 80.5%	-/-
T3L	76.1% / 77.1%	55.2% / 79.8%	-/-
PM	90.0% / 85.4%	90.4% / 87.5%	97.5% / 96.2%
PS	90.4% / 86.2%	91.2% / 88.2%	98.9% / 97.3%

MDR2Pass: First pass=basic MDR. Second pass= check those non-consecutive siblings and put them back to the existing region if they also meet the comparison criterion, regardless of whether they are adjacent to the existing records or not.

Table 3: Comment extraction

	Precision	Recall	Perfect	Extract $\leq 2$ wrong	Miss $\leq 2$ golden
Blog comments					
B1	52.5%	76.6%	45.7%	74.2%	73.8%
B2	58.5%	79.9%	65.2%	77.8%	81.9%
M	95.8%	91.1%	78.3%	96.4%	89.6%
Review comments					
B1	89.3%	80.0%	63.7%	85.8%	79.6%
B2	91.8%	81.4%	72.2%	84.1%	81.2%
M	94.1%	81.8%	73.9%	87.3%	82.4%

B1 = MDR2Pass+TS

B2 = MDR2Pass+PS

M = MiBAT+PS.

# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- **Extracting Tables from the Web**
  - **WebTables: Exploring the Power of Tables on the Web**
  - Answering Table Augmentation Queries from Unstructured Lists on the Web
  - Annotating Tables with Ontological Links
- Extracting Sets from the Web

# Identifying Relational DBs from the Web

Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, Yang Zhang. Uncovering the Relational Web. WebDB 2008

The Presidents of the USA - EnchantedLearning.com - Mozilla Firefox

http://www.enchantedlearning.com/history/us/pres/list.shtml


As a thank-you bonus, [site members](#) have access to a banner-ad-free version of the site, with print-friendly pages.

(Already a member? [Click here.](#))

 [US Flags](#) [EnchantedLearning.com](#) [US History](#)  [US Geography](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

[AFRICAN-AMERICANS](#) [ARTISTS](#) [Explorers of the US](#) [Inventors](#) [US Presidents](#) [US Symbols](#) [US States](#)

 [President's Day Activities](#) [EnchantedLearning.com](#) **The Presidents of the United States of America**  [Abraham Lincoln](#)

[In the order in which they served](#) [Alphabetical order](#) [Short table of Data](#)

The President and Vice-President are elected every four years. They must be at least 35 years of age, they must be native-born citizens of the United States, and they must have been residents of the U.S. for at least 14 years. (Also, a person cannot be elected to a third term as President.)

President	Party	Term as President	Vice-President
1. <a href="#">George Washington</a> (1732-1799)	None, Federalist	1789-1797	<a href="#">John Adams</a>
2. <a href="#">John Adams</a> (1735-1826)	Federalist	1797-1801	<a href="#">Thomas Jefferson</a>
3. <a href="#">Thomas Jefferson</a> (1743-1826)	Democratic-Republican	1801-1809	Aaron Burr, George Clinton
4. <a href="#">James Madison</a> (1751-1836)	Democratic-Republican	1809-1817	George Clinton, Elbridge Gerry
5. James Monroe (1758-1831)	Democratic-Republican	1817-1825	Daniel Tompkins
6. <a href="#">John Quincy Adams</a> (1767-1848)	Democratic-Republican	1825-1829	John Calhoun
7. Andrew Jackson (1767-1845)	Democrat	1829-1837	John Calhoun, Martin van Buren
8. Martin van Buren (1782-1862)	Democrat	1837-1841	Richard Johnson
9. William H. Harrison (1773-1841)	Whig	1841	John Tyler
10. John Tyler (1790-1862)	Whig	1841-1845	
11. James K. Polk (1795-1849)	Democrat	1845-1849	George Dallas
12. Zachary Taylor (1784-1850)	Whig	1849-1850	Millard Fillmore
13. Millard Fillmore (1800-1874)	Whig	1850-1853	
14. Franklin Pierce (1804-1869)	Democrat	1853-1857	William King
15. James Buchanan (1791-1868)	Democrat	1857-1861	John Breckinridge





This page contains 16 distinct HTML tables,  
but only one relational database

Each relational database has its own schema, usually with labeled columns.

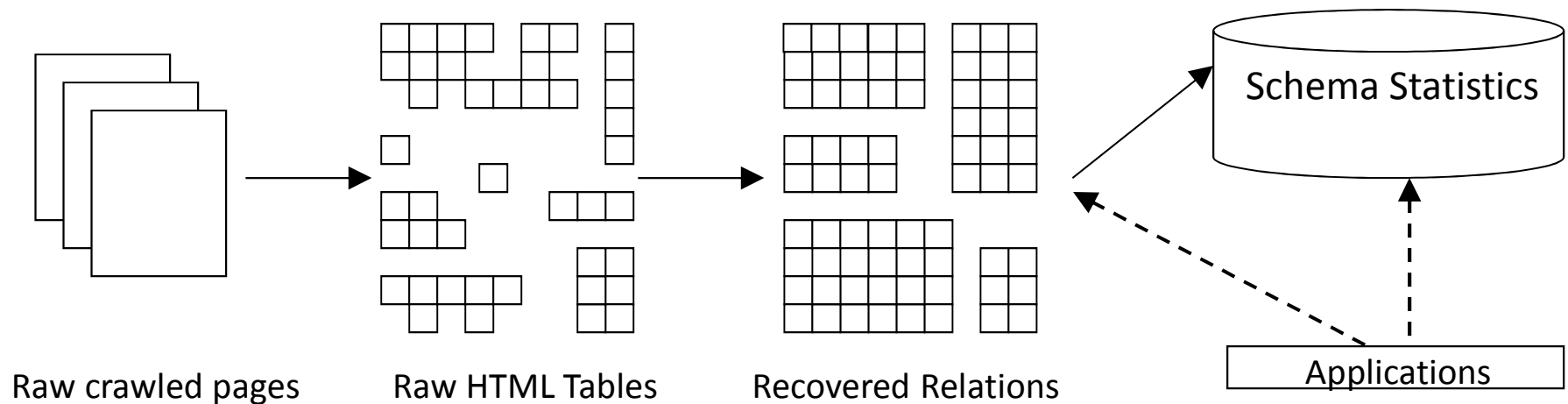
[illegible]

# WebTables

- WebTables system automatically extracts dbs from web crawl

*[WebDB08, “Uncovering...”, Cafarella et al]*

*[VLDB08, “WebTables: Exploring...”, Cafarella et al]*

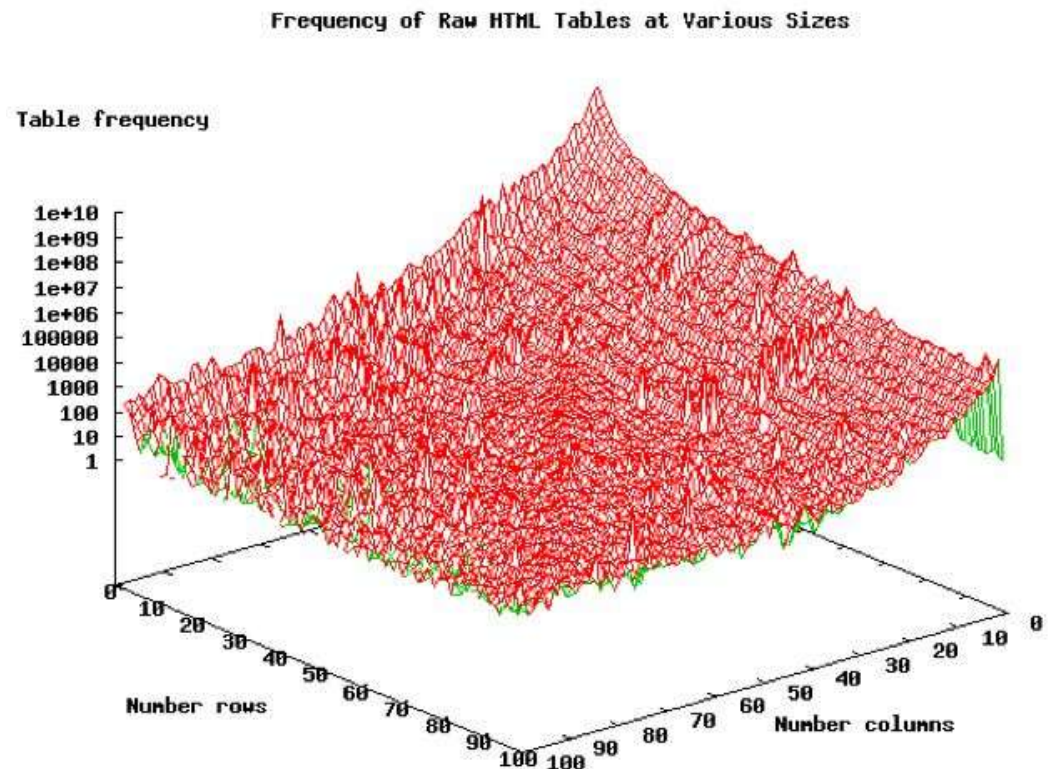


- An extracted relation is one table plus labeled columns
- Estimate that our crawl of **14.1B** raw HTML tables contains **~154M** good relational dbs

# The Table Corpus

Table type	% total	count
Small tables	88.06	12.34B
HTML forms	1.34	187.37M
Calendars	0.04	5.50M
Obvious non-rel	89.44	12.53B
Other non-rel (est.)	9.46	1.33B
Rel (est.)	1.10	154.15M

- Many tables are embedded inside HTML forms and are generally used for visual layout of user input fields.
- Some tables are used to draw a calendar onscreen, and consist of nothing but a column for each day of the week and a number in each cell.
- Non-relations include tables used for page layout, tables with an enormous number of blank cells, tables that are really simple lists presented in two dimensions, and “property sheets” that consist of attribute value pairs for a single data entity





# Recovering Relations from Raw HTML Tables

- Filter out all non-relational tables
  - Features
    - Table layout: average #rows, std. dev. of #rows, average #columns, std dev of #columns, average cell length, std. dev. of cell length, Average Cumulative length consistency across all table cells
    - Content consistency: #columns containing non-string basic datatypes, histogram of content type (Image, Form, Hyperlink, Alphabetical, Digit, Empty, Others), Average content type consistency
    - Word group
      - TF-IDF of words wrt appearance in genuine vs. non-genuine tables
      - Cosine similarity of word vector with genuine/non-genuine word vector
  - Yalin Wang, Jianying Hu. A Machine Learning Based Approach for Table Detection on The Web. WWW2002
- Recover metadata (in the form of attribute labels)

# Table Metadata Recovery

- Case 1: There is already a header row
  - Classifier to detect if the header row is present or not
    - Features: # rows, # cols, % cols w/lower-case in row 1, % cols w/punctuation in row 1, % cols w/non-string data in row 1, % cols w/non-string data in body, % cols w/ $|\text{len}(\text{row } 1) - \mu| > 2\sigma$ , % cols w/ $\sigma \leq |\text{len}(\text{row } 1) - \mu| \leq 2\sigma$ , % cols w/ $\sigma > |\text{len}(\text{row } 1) - \mu|$
- Case 2: There is no header row
  - Match the column values with domain-specific dictionaries
    - 6.8M tuples in 849 separate domains

## Attribute Co-occurrence Statistics Database (ACSDb)

- Co-occurrence count of attributes
- Schema Coherence Score  $S(R)$  for relation  $R$  is

$$S(R) = \frac{\sum_{A,B \in R, A \neq B} \log\left(\frac{p(A,B)}{p(A)p(B)}\right)}{|R|(|R|-1)}$$

true class	Precision	Recall
relational	0.41	0.81
non-relational	0.98	0.87

Detector	header?	Precision	Recall
<b>Detect</b>	has-header	0.79	0.84
	no-header	0.65	0.57
<b>Detect-ACSDb</b>	has-header	0.89	0.85
	no-header	0.75	0.80

# Relation Recovery

[File](#) [Edit](#) [View](#) [History](#) [Bookmarks](#) [Tools](#) [Help](#) [http://www.enchantedlearning.com/us/history/print.shtml](#)

As a thank-you bonus, [this century](#) has access to a **hundred and five** versions of the site, with print-friendly pages.

(Already a member? [Click here.](#))

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

American Literature   Artists   Explorers of the U.S.   Geography   History   Maps   Native Americans   Presidents of the U.S.   States   US Geography

[President's Day Activities](#)

## The Presidents of America [In order in which they served](#)   [Alphabetical order](#)   [Short table of Data](#) The President of the United States is elected every four years. They must be at least 35 years of age, they must be native born citizens of the United States, and they must have been residents of the U.S. for at least 14 years. (Also, a person cannot be elected to two terms as President.) | President | Party | Term as President | John Adams | Vice-President | |------------------------------------|-----------------------|-------------------|------------------|----------------| | 1. George Washington (1733-1799) | None, Federalist | 1789-1797 | Thomas Jefferson | None | | 2. John Adams (1735-1826) | Federalist | 1797-1801 | Thomas Jefferson | Elbridge Gerry | | 3. Thomas Jefferson (1743-1826) | Democratic-Republican | 1801-1809 | James Madison | George Clinton | | 4. James Madison (1751-1836) | Democratic-Republican | 1809-1817 | James Monroe | George Clinton | | 5. James Monroe (1758-1831) | Democratic-Republican | 1817-1823 | James Monroe | Elbridge Gerry | | 6. John Quincy Adams (1767-1848) | Democratic-Republican | 1823-1829 | John Adams | Elbridge Gerry | | 7. Andrew Jackson (1767-1845) | Democratic | 1829-1837 | John Adams | Elbridge Gerry | | 8. Martin van Buren (1767-1862) | Democratic | 1837-1841 | John Adams | Elbridge Gerry | | 9. William H. Harrison (1773-1841) | Whig | 1841 | John Adams | Elbridge Gerry | | 10. John Tyler (1790-1862) | Whig | 1841-1845 | John Adams | Elbridge Gerry | | 11. James K. Polk (1795-1849) | Whig | 1845-1849 | John Adams | Elbridge Gerry | | 12. Zachary Taylor (1784-1850) | Whig | 1850-1850 | John Adams | Elbridge Gerry | | 13. Millard Fillmore (1800-1874) | Whig | 1850-1855 | John Adams | Elbridge Gerry | | 14. Franklin Pierce (1803-1869) | Democratic | 1853-1857 | John Adams | Elbridge Gerry |

## Step 1. Relational Filtering

Recall 81%, Precision 41%

- Output
  - 271M databases, about 125M are good
  - Five orders of magnitude larger than previous largest corpus [WWW02, “A Machine Learning...”, Wang & Hu]
  - 2.6M unique relational schemas
- What can we do with those schemas?  
[VLDB08, “WebTables: Exploring...”, Cafarella et al]

[File](#) [Edit](#) [View](#) [History](#) [Bookmarks](#) [Tools](#) [Tools](#) [Help](#)

[http://www.enrichmentlearning.com/usa/usa-president.shtml](#)

[Go](#) [Google](#)

As a thank-you bonus, [this website](#) has access to a banner-and-live version of the site, with print-friendly pages.

(Already a member? [Click here.](#))

[EnrichmentLearning.com](#)  
**US History**




[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

[American Revolution](#) [Antislavery](#) [Emigration of the 19th C](#) [Immigration](#) [US Settlement](#)




[President's Day Activities](#)

## The Presidents of the United States of America

[In the order in which they served](#) [Alphabetically by name](#) [Short lists of Dates](#)

The President and Vice-President are elected every four years. They must be at least 35 years of age, they must be native-born citizens of the United States, and they must have been born in the U.S. (or at least, in England - America's earliest settlers are elected to this position as Presidents.)

President	Party	Term as President	Vice-President
1. George Washington (1789-1797)	None; Federalist	1789-1797	John Adams
2. Thomas Jefferson (1797-1809)	Democratic-Republican	1801-1809	Aaron Burr, George Clinton
3. James Madison (1809-1817)	Democratic-Republican	1809-1817	George Clinton, Elbridge Gerry
4. James Monroe (1793-1803)	Democratic-Republican	1817-1825	James Tompkins
5. John Quincy Adams (1797-1848)	Democratic-Republican	1825-1829	John Calhoun
6. Andrew Jackson (1767-1845)	Democratic	1829-1837	John Calhoun, Martin van Buren
7. Martin van Buren (1782-1862)	Democratic	1837-1841	Richard Johnson
8. William H. Harrison (1773-1841)	Whig	1841	John Tyler
9. John Tyler (1790-1862)	Whig	1841-1845	
10. James K. Polk (1795-1846)	Democratic	1845-1849	George Dallas
11. Zachary Taylor (1774-1850)	Whig	1849-1850	Millard Fillmore
12. Millard Fillmore (1800-1874)	Whig	1850-1855	
13. Franklin Pierce (1804-1879)	Democratic	1855-1857	William King
14. James Buchanan (1791-1868)	Democratic	1857-1861	Jefferson Davis

## Step 2. Metadata Detection

Recall 85%, Precision 89%

# App #1: Relation Search

- Problem: Keyword search for high-quality extracted databases
- Output depends on both quality of extracted tables and the ranking function
- Schema statistics can help improve both:
  - Relation Recovery (Metadata detection)
  - Ranking
- Ranking: compared 4 rankers on test set
  - **Naïve**: Top-10 *pages* from google.com
  - **Filter**: Top-10 *good tables* from google.com
  - **Rank**: Trained ranker
  - **Rank-Stats**: Trained ranker with coherency score

k	Naïve	Filter	Rank	Rank-Stats
<b>10</b>	0.26	0.35	0.43	0.47
<b>20</b>	0.33	0.47	0.56	0.59
<b>30</b>	0.34	0.59	0.66	0.68

## App #2: Schema Autocomplete

- Input: topic attribute (e.g., make)
- Output: relevant schema {make, model, year, price}
- “tab-complete” for your database
- For input set  $I$ , output  $S$ , threshold  $t$ 
  - while  $p(S-I|I) > t$ 
    - $\text{newAttr} = \max p(\text{newAttr}, S-I | I)$
    - $S = S \cup \text{newAttr}$
    - emit newAttr
- Asked experts for schemas in 10 areas
- What was autocompleter’s recall?
- 3 tries for autocompletion (without replacement)

Input	1	2	3
name	0	0.6	0.8
instructor	0.6	0.6	0.6
elected	1.0	1.0	1.0
ab	0	0	0
stock-symbol	0.4	0.8	0.8
company	0.22	0.33	0.44
director	0.75	0.75	1.0
album	0.5	0.5	0.66
sqft	0.5	0.66	0.66
goals	0.66	0.66	0.66
Average	0.46	0.59	0.62

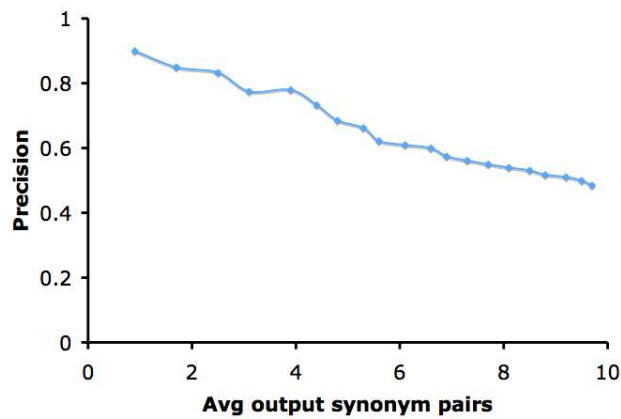
## App #3: Synonym Discovery

- Input: topic attribute (e.g., address)
- Output: relevant synonym pairs (telephone = tel-#)
- Observations
  - synonymous attributes  $a$  and  $b$  will never appear together in the same schema, i.e.,  $p(a, b) = 0$
  - odds of synonymity are higher if  $p(a, b) = 0$  despite a large value for  $p(a)p(b)$
  - two synonyms will appear in similar contexts: i.e., for  $a$  and  $b$  and a third attribute  $z \notin C$ ,  $p(z|a, C) \approx p(z|b, C)$

$$\text{syn}(a, b) = \frac{p(a)p(b)}{\epsilon + \sum_{z \in A} (p(z|a, C) - p(z|b, C))^2}$$

## App #3: Synonym Discovery

name	e-mail email, phone telephone, e-mail_address email_address, date last_modified
instructor	course-title title, day days, course course-#, course-name course-title
elected	candidate name, presiding-officer speaker
ab	k so, h hits, avg ba, name player
sqft	bath baths, list list-price, bed beds, price rent





# Multiple Tables: Project Octopus

- Can we combine tables to create new data sources?
- Data integration for the Structured Web
- Octopus: Provides “workbench” of data integration operators to build target database
  - **SEARCH**(“VLDB program committee members”)
  - **CONTEXT()** – Recovers relevant data
  - Union()
  - **EXTEND()**
    - Input: table and source page
    - Output: data values to add to table

# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - **Answering Table Augmentation Queries from Unstructured Lists on the Web**
  - Annotating Tables with Ontological Links
- Extracting Sets from the Web

# Table Augmentation Queries

- Rahul Gupta, Sunita Sarawagi. Answering Table Augmentation Queries from Unstructured Lists on the Web. VLDB 2009.
- Example: Compiling a List of famous CS inventors and inventions

Person	Concept/Invention
Alan Turing	Turing Machine
Seymour Cray	Supercomputer
E. F. Codd	Relational Databases
Tim Berners-Lee	WWW
Charles Babbage	Babbage Engine

Web Images Maps News Orkut Groups Gmail more ▼



computer science concept inventor year

Search

Advanced Search  
Preferences

Search: ☒ the web ☐ pages from India

Web [Show options...](#)

### Coding Horror: The Greatest Invention in Computer Science ★

I'd say the single greatest **invention** in **computer science** is the **concept** of .... Didn't you mention the blog post 'Design Patterns of 1972' last **year**? ...

[www.codinghorror.com/blog/archives/001129.html](http://www.codinghorror.com/blog/archives/001129.html) - [Cached](#) - [Similar](#) -

### History of computer science - Wikipedia, the free encyclopedia ★

Alan Turing, known as the Father of **Computer Science**, **invented** such a logical ... 1936 was a key **year** for **computer science**. Alan Turing and Alonzo Church ... This **concept**, of utilizing the properties of electrical switches to do logic, ...

[en.wikipedia.org/wiki/History\\_of\\_computer\\_science](http://en.wikipedia.org/wiki/History_of_computer_science) - [Cached](#) - [Similar](#) -

### ENIAC Computer History - Invention of the ENIAC Computer ★

Aiken never trusted the **concept** of storing a program within the **computer**. .... is provided courtesy of Department of **Computer Science**, Virginia Tech, ... and I can assure you that data processing is a fad that won't last out the **year**. ...

[www.ideatinder.com/history/inventions/story072.htm](http://www.ideatinder.com/history/inventions/story072.htm) - [Cached](#) - [Similar](#) -

### Great names in computer science ★ - 2 visits - 11 Aug

4 Sep 2005 ... Babbage is considered one of the forefathers of **computer science** for having ... Edsger Dijkstra is the **inventor** of the **concept** of semaphore, ...

[www.madore.org/~david/computers/greatnames.html](http://www.madore.org/~david/computers/greatnames.html) - [Cached](#) - [Similar](#) -

### Who invented computers? - Yahoo! Answers ★

1 **year** ago. Sign in to vote! 0 Rating: Good Answer; 3 Rating: Bad Answer ... The **invention** of the 'modern' stored-program digital **computer** is usually ... philosopher, **inventor** and mechanical engineer who originated the **concept** of a ... Parts of his uncompleted mechanisms are on display in the London **Science Museum**. ...

[answers.yahoo.com/question/index?qid...](http://answers.yahoo.com/question/index?qid...) - [Cached](#) - [Similar](#) -

Correct answer is not one click away.

Verbose articles, not structured tables

Desired records spread across many documents

The only document with an unstructured list of some desired records

# The Only List in One of the Retrieved Pages

Harold Abelson

[\(web page\)](#) Professor of Computer Science and Engineering at the [Massachusetts Institute of Technology](#). Abelson is the author of *Structure and Interpretation of Computer Processes*.

Eric Allman

[\(web page\)](#) Eric Allman is the main author of the [sendmail](#) program, which handles email (emails), although certain alternatives have become popular, such as [Duke](#) and [McKusick's](#) partner.

Charles Babbage

Born: Monday, December 26, 1791, in London (England). Died: Wednesday, September 9, 1871, in London (England). Babbage is considered one of the forefathers of computer science for having designed (with the help of [Ada Lovelace](#)) the [analytical engine](#), which, although it was never built, was a (mechanical) computer. See also Babbage's [biography](#) on the [MacTutor History of Mathematics archive](#).




John W. Backus

Born: Wednesday, December 3, 1924, in Philadelphia, Pennsylvania (USA). Died: Wednesday, September 12, 2007, in Philadelphia, Pennsylvania (USA). Backus, who gave birth to the language FORTRAN (the oldest programming language), was also involved in the development of ALGOL 60, ALGOL 68, and, of course, assembler. John Backus is the 1977 recipient of the [IEEE Computer Society's Pioneer Award](#).

Tim Berners-Lee

[\(web page\)](#) Born: Wednesday, June 8, 1955, in London (UK). Tim Berners-Lee is the inventor of the World Wide Web and the [World Wide Web Consortium](#).

# Highly relevant Wikipedia table not retrieved in the top-kIdeal

Person 	Achievement 	Ach. Date 
<a href="#">John Atanasoff</a>	Built the first electronic digital computer, the <a href="#">Atanasoff–Berry Computer</a> , though it was neither programmable nor Turing-complete.	1939
<a href="#">Charles Babbage</a>	Designed the <a href="#">Analytical Engine</a> and built a prototype for a less powerful <a href="#">mechanical calculator</a> .	1822 1837
<a href="#">John Backus</a>	Invented <a href="#">FORTRAN</a> ( <i>Formula Translation</i> ), the first practical high-level programming language, and he formulated the <a href="#">Backus-Naur form</a> that described the formal language <a href="#">syntax</a> .	1954 1963
<a href="#">George Boole</a>	Formalized <a href="#">Boolean algebra</a> , the basis for <a href="#">digital logic</a> and computer science.	1830~

Ideal answer should be integrated from these incomplete sources



## Attempt 2: Include samples in query

- New query: “alan turing machine codd relational database” ← Known example
- Results: Documents relevant only to the keywords
- Ideal answer still spread across many documents
- What to do?
- Can use table augmentation technique to augment Freebase/Wikipedia tables

# Table Augmentation Problem

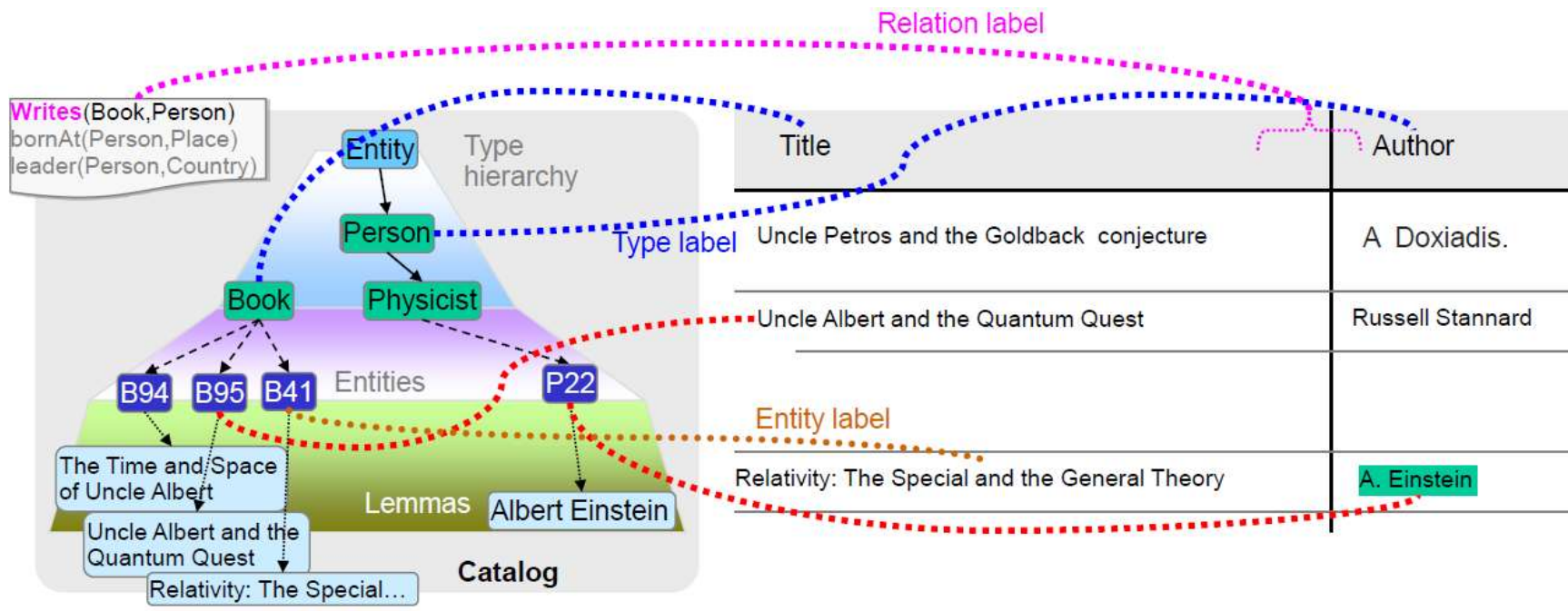
- A user provides a few ( $\sim 3$ ) structured records.
- Goal is to return a single table with more such records ranked by relevance.
- Large number of relevant records can be extracted from multiple semi/unstructured sources like HTML lists/tables.
- Have to integrate across multiple sources.
- Multi-attribute version of Google Sets.
- Goal similar to Google-Squared (launched mid-May 2009; now no longer available publicly).



# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - Answering Table Augmentation Queries from Unstructured Lists on the Web
  - **Annotating Tables with Ontological Links**
- Extracting Sets from the Web

# Entity, Type, and Relation links



# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - Answering Table Augmentation Queries from Unstructured Lists on the Web
  - Annotating Tables with Ontological Links
- **Extracting Sets from the Web**

# WebSets

- Bhavana Dalvi, William W. Cohen, Jamie Callan.  
WebSets: Extracting Sets of Entities from the Web  
Using Unsupervised Information Extraction. WSDM  
2012
- Extracting concept-instance pairs from an HTML corpus
  - Cluster terms found in HTML tables
  - Assign concept names to these clusters using Hearst patterns
- Past approaches
  - Hyponym patterns (Hearst patterns) (like “Xs such as Y”) indicate that X, Y are a concept-instance pair
  - Two terms i and j are coordinate terms if i and j are instances of the same concept

# Experiments

<b>Religions:</b> Buddhism, Christianity, Islam, Sikhism, Taoism, Zoroastrianism, Jainism, Bahai, Judaism, Hinduism, Confucianism
<b>Government:</b> Monarchy, Limited Democracy, Islamic Republic, Parliamentary Self Governing Territory, Parliamentary Republic, Constitutional Republic, Republic Presidential Multiparty System, Constitutional Democracy, Democratic Republic, Parliamentary Democracy
<b>International Organizations:</b> United Nations Children Fund UNICEF, Southeast European Cooperative Initiative SECI, World Trade Organization WTO, Indian Ocean Commission INOC, Economic and Social Council ECOSOC, Caribbean Community and Common Market CARICOM, Western European Union WEU, Black Sea Economic Cooperation Zone BSEC, Nuclear Energy Agency NEA, World Confederation of Labor WCL
<b>Languages:</b> Hebrew, Portuguese, Danish, Anzanian, Croatian, Phoenician, Brazilian, Surinamese, Burkinabe, Barbadian, Cuban

<b>Instruments:</b> Flute, Tuba , String Orchestra, Chimes, Harmonium, Bassoon, Woodwinds, Glockenspiel, French horn, Timpani, Piano
<b>Intervals:</b> Whole tone, Major sixth, Fifth, Perfect fifth, Seventh, Third, Diminished fifth, Whole step, Fourth, Minor seventh, Major third, Minor third
<b>Genres:</b> Smooth jazz, Gothic, Metal rock, Rock, Pop, Hip hop, Rock n roll, Country, Folk, Punk rock
<b>Audio Equipments:</b> Audio editor , General midi synthesizer , Audio recorder , Multichannel digital audio workstation , Drum sequencer , Mixers , Music engraving system , Audio server , Mastering software , Soundfont sample player

## Take-away Messages

- Web is an awesome source of information.
- We looked at mechanisms to extract
  - Top-k lists
  - UGC data records
  - Tables
  - Sets
- We also looked at applications of structured data

## Further Reading

- Chapter 9 of “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data” by Bing Liu. <http://rd.springer.com/book/10.1007/978-3-642-19460-3//page/1>
- Rahul Gupta, Sunita Sarawagi. Answering Table Augmentation Queries using Unstructured Lists on the Web. VLDB 2009
- Arvind Arasu, Hector Garcia-Molina. Extracting Structured Data from Web Pages. SIGMOD 2003.
- M. Cafarella, N. Khoussainova, D. Wang, E. Wu, Y. Zhang, and A. Halevy. Uncovering the Relational Web. In WebDB, 2008.
- R. Gupta and S. Sarawagi. Curating probabilistic databases from information extraction models. In VLDB, 2006.
- Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. WebTables: exploring the power of tables on the web. VLDB 2008.
- Zhixian Zhang and Kenny Q. Zhu and Haixun Wang and Hongsong Li. Automatic extraction of top-k lists from the web. ICDE 2013.
- Bhavana Dalvi, William W. Cohen, Jamie Callan. WebSets: Extracting Sets of Entities from the Web Using Unsupervised Information Extraction. WSDM 2012.

# Preview of Lecture 19: Entity Semantics Mining (Part 1)

- Entity Synonyms
- Entity Attribute Discovery and Augmentation
- Entity Linking



# Disclaimers

- This course represents opinions of the instructor only. It does not reflect views of Microsoft or any other entity (except of authors from whom the slides have been borrowed).
- Algorithms, techniques, features, etc mentioned here might or might not be in use by Microsoft or any other company.
- Lot of material covered in this course is borrowed from slides across many universities and conference tutorials. These are gratefully acknowledged.

**Thanks!**

# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - **Answering Table Augmentation Queries from Unstructured Lists on the Web**
  - Annotating Tables with Ontological Links
- Extracting Sets from the Web

# Table Augmentation Queries

- Rahul Gupta, Sunita Sarawagi. Answering Table Augmentation Queries from Unstructured Lists on the Web. VLDB 2009.
- Example: Compiling a List of famous CS inventors and inventions

Person	Concept/Invention
Alan Turing	Turing Machine
Seymour Cray	Supercomputer
E. F. Codd	Relational Databases
Tim Berners-Lee	WWW
Charles Babbage	Babbage Engine

Web Images Maps News Orkut Groups Gmail more ▼



computer science concept inventor year

Search

Advanced Search  
Preferences

Search: ☒ the web ☐ pages from India

Web [+ Show options...](#)

### Coding Horror: The Greatest Invention in Computer Science ★

I'd say the single greatest **invention** in **computer science** is the **concept** of .... Didn't you mention the blog post 'Design Patterns of 1972' last **year**? ...

[www.codinghorror.com/blog/archives/001129.html](http://www.codinghorror.com/blog/archives/001129.html) - [Cached](#) - [Similar](#) -

### History of computer science - Wikipedia, the free encyclopedia ★

Alan Turing, known as the Father of **Computer Science**, **invented** such a logical ... 1936 was a key **year** for **computer science**. Alan Turing and Alonzo Church ... This **concept**, of utilizing the properties of electrical switches to do logic, ...

[en.wikipedia.org/wiki/History\\_of\\_computer\\_science](http://en.wikipedia.org/wiki/History_of_computer_science) - [Cached](#) - [Similar](#) -

### ENIAC Computer History - Invention of the ENIAC Computer ★

Aiken never trusted the **concept** of storing a program within the **computer**. .... is provided courtesy of Department of **Computer Science**, Virginia Tech, ... and I can assure you that data processing is a fad that won't last out the **year**. ...

[www.ideatinder.com/history/inventions/story072.htm](http://www.ideatinder.com/history/inventions/story072.htm) - [Cached](#) - [Similar](#) -

### Great names in computer science ★ - 2 visits - 11 Aug

4 Sep 2005 ... Babbage is considered one of the forefathers of **computer science** for having ... Edsger Dijkstra is the **inventor** of the **concept** of semaphore, ...

[www.madore.org/~david/computers/greatnames.html](http://www.madore.org/~david/computers/greatnames.html) - [Cached](#) - [Similar](#) -

### Who invented computers? - Yahoo! Answers ★

1 **year** ago. Sign in to vote! 0 Rating: Good Answer; 3 Rating: Bad Answer ... The **invention** of the 'modern' stored-program digital **computer** is usually ... philosopher, **inventor** and mechanical engineer who originated the **concept** of a ... Parts of his uncompleted mechanisms are on display in the London **Science** Museum. ...

[answers.yahoo.com/question/index?qid...](http://answers.yahoo.com/question/index?qid...) - [Cached](#) - [Similar](#) -

Correct answer is not one click away.

Verbose articles, not structured tables

Desired records spread across many documents

The only document with an unstructured list of some desired records

# The Only List in One of the Retrieved Pages

Harold Abelson

[\(web page\)](#) Professor of Computer Science and Engineering at the [Massachusetts Institute of Technology](#). Abelson is the author of *Structure and Interpretation of Computer Systems*.

Eric Allman

[\(web page\)](#) Eric Allman is the main author of the [sendmail](#) program, which handles email (emails), although certain alternatives have become popular, such as [Duke](#) and [McKusick's](#) partner.

Charles Babbage

Born: Monday, December 26, 1791, in London (England). Died: Wednesday, September 26, 1871, in London (England). Babbage is considered one of the forefathers of computer science for having designed (with the help of [Ada Lovelace](#)) the [analytical engine](#), which, although it was never built, was a (mechanical) computer. See also Babbage's [biography](#) on the [MacTutor History of Mathematics archive](#).



John W. Backus

Born: Wednesday, December 3, 1924, in Philadelphia, Pennsylvania (USA). Died: Wednesday, September 12, 2007, in Philadelphia, Pennsylvania (USA). Backus, who gave birth to the language FORTRAN (the oldest programming language), was also involved in the development of ALGOL 60, ALGOL W, and Calculus, and, of course, assembler). John Backus is the 1977 recipient of the [IEEE Computer Society's Pioneer Award](#).

Tim Berners-Lee

[\(web page\)](#) Born: Wednesday, June 8, 1955, in London (UK). Tim Berners-Lee is the inventor of the World Wide Web and the [World Wide Web Consortium](#).

# Highly relevant Wikipedia table not retrieved in the top-kIdeal

Person 	Achievement 	Ach. Date 
<a href="#">John Atanasoff</a>	Built the first electronic digital computer, the <a href="#">Atanasoff–Berry Computer</a> , though it was neither programmable nor Turing-complete.	1939
<a href="#">Charles Babbage</a>	Designed the <a href="#">Analytical Engine</a> and built a prototype for a less powerful <a href="#">mechanical calculator</a> .	1822 1837
<a href="#">John Backus</a>	Invented <a href="#">FORTRAN</a> ( <i>Formula Translation</i> ), the first practical high-level programming language, and he formulated the <a href="#">Backus-Naur form</a> that described the formal language <a href="#">syntax</a> .	1954 1963
<a href="#">George Boole</a>	Formalized <a href="#">Boolean algebra</a> , the basis for <a href="#">digital logic</a> and computer science.	1830~

Ideal answer should be integrated from these incomplete sources



## Attempt 2: Include samples in query

- New query: “alan turing machine codd relational database” ← Known example
- Results: Documents relevant only to the keywords
- Ideal answer still spread across many documents
- What to do?
- Can use table augmentation technique to augment Freebase/Wikipedia tables



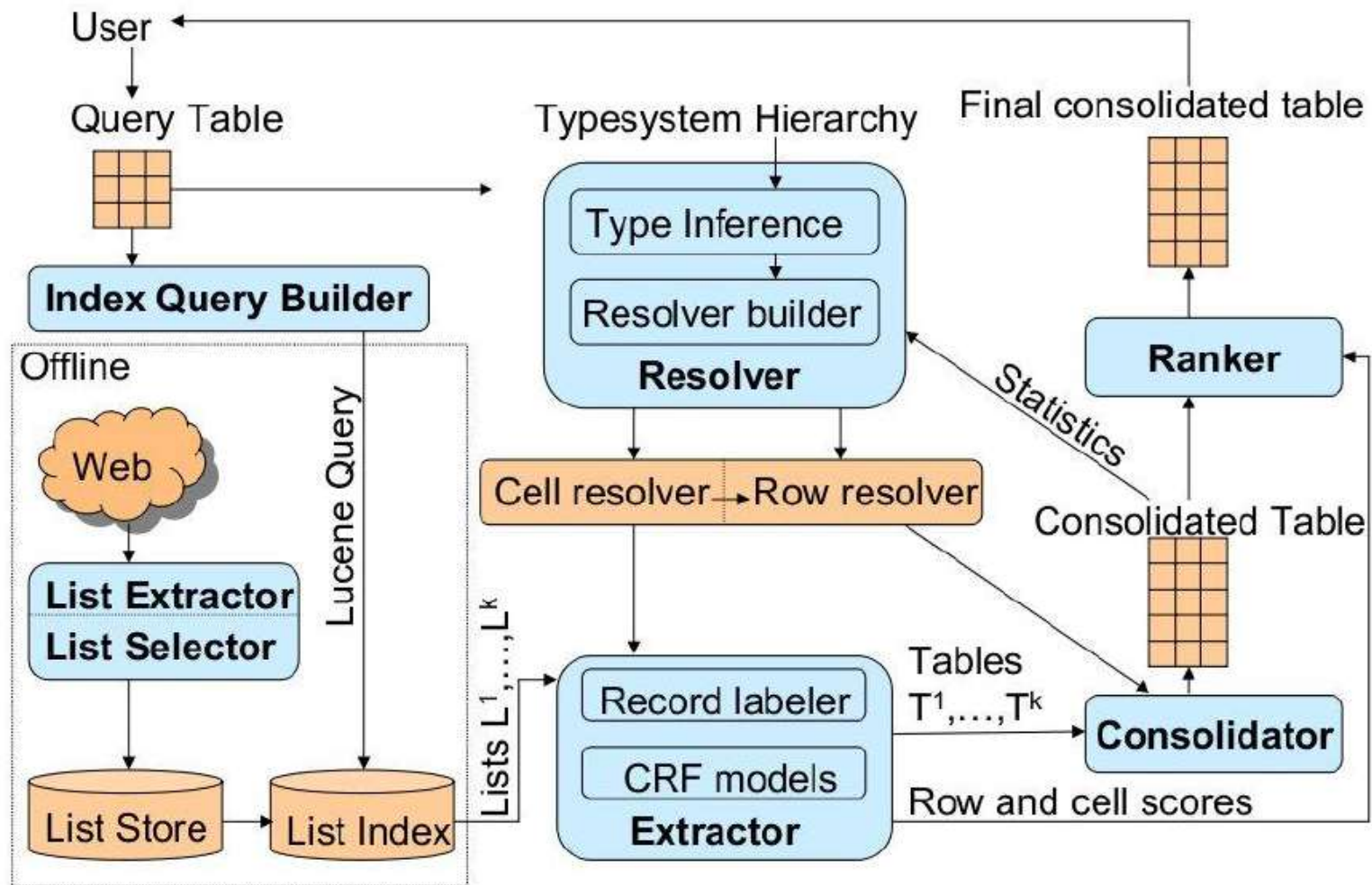
# Table Augmentation Problem

- A user provides a few ( $\sim 3$ ) structured records.
- Goal is to return a single table with more such records ranked by relevance.
- Large number of relevant records can be extracted from multiple semi/unstructured sources like HTML lists/tables.
- Have to integrate across multiple sources.
- Multi-attribute version of Google Sets.
- Goal similar to Google-Squared (launched mid-May 2009; now no longer available publicly).

## World Wide Tables (WWT) system

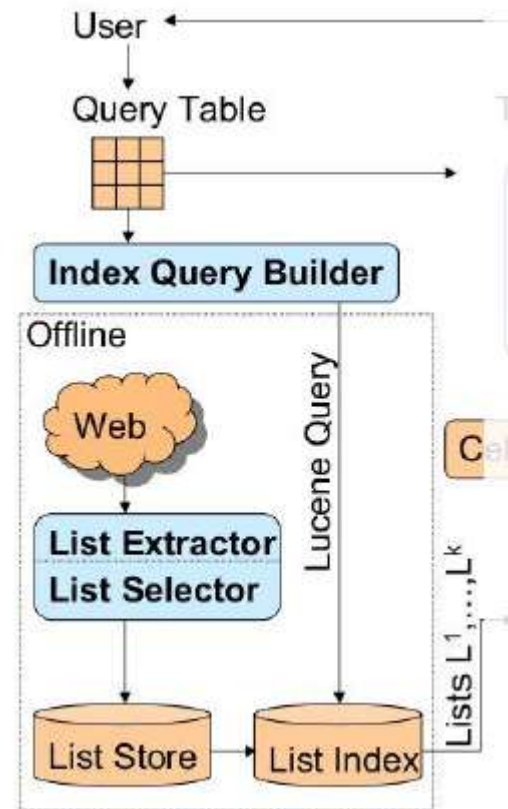
- answers table augmentation queries
- Achieves a runtime of ~30s. Recall ~ 55% when reconstructing Wikipedia tables from 3 samples.
- Many lists have records that are absent in tables
- Lists harder to process than tables
- This section: Answering queries only using HTML list sources
- Current WWT system uses both lists and tables

# WWT: Architecture



# Offline List Extraction and Indexing+Index Probe

- Retrieving and processing HTML lists instead of documents
  - Indexed 16M lists extracted from 500M documents.
- Step 0: Index Probe



# Step 1: Extraction

- Extracting required columns from list records

Cornell University	Ithaca
State University of New York	Stony Brook
New York University	New York

- New York University (NYU), New York City, founded in 1831.
- Columbia University, founded in 1754 as King's College.
- Binghamton University, Binghamton, established in 1946.
- State University of New York, Stony Brook, New York, founded in 1957
- Syracuse University, Syracuse, New York, established in 1870
- State University of New York, Buffalo, established in 1846
- Rensselaer Polytechnic Institute (RPI) at Troy.

- Rule-based extractor insufficient. Statistical extractor needs training data.
  - How to generate training data?

# Extraction

- Adapt Conditional Random Fields for extraction
  - No explicitly labeled training data. Generate reliable training data from the small query.
  - Exploit regularity inside a source using multiple sequence alignment.
  - Use content overlap across sources to strengthen sources with less labeled data.

## Extraction: Labeled data generation

- Lists are unlabeled. Labeled records needed to train a CRF
- A fast but naïve approach for generating labeled records

New York University	New York
Monroe College	Brighton
State University of New York	Stony Brook

Query about colleges in NY

- New York Univ. in NYC
- Columbia University in NYC
- Monroe Community College in Brighton
- State University of New York in Stony Brook, New York.

Fragment of a relevant list source

## Extraction: Labeled data generation

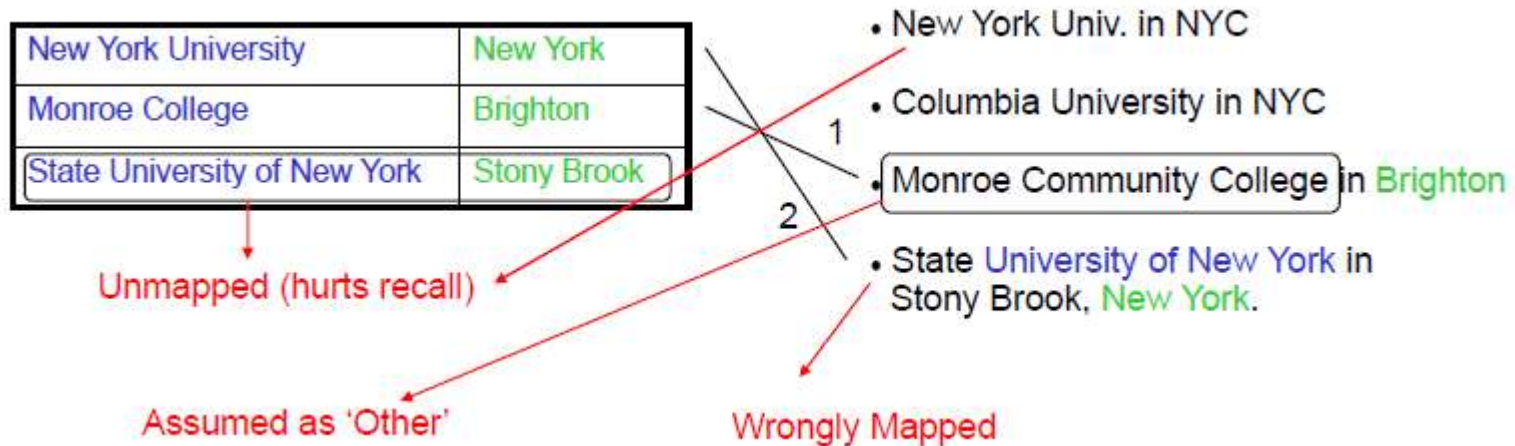
New York University	New York
Monroe College	Brighton
State University of New York	Stony Brook

- New York Univ. in NYC
  - Columbia University in NYC
  - Monroe Community College in Brighton
  - State University of New York in Stony Brook, New York.
- 1
- 2

- In the list, look for matches of every query cell.
- Greedily map each query row to the best match in the list

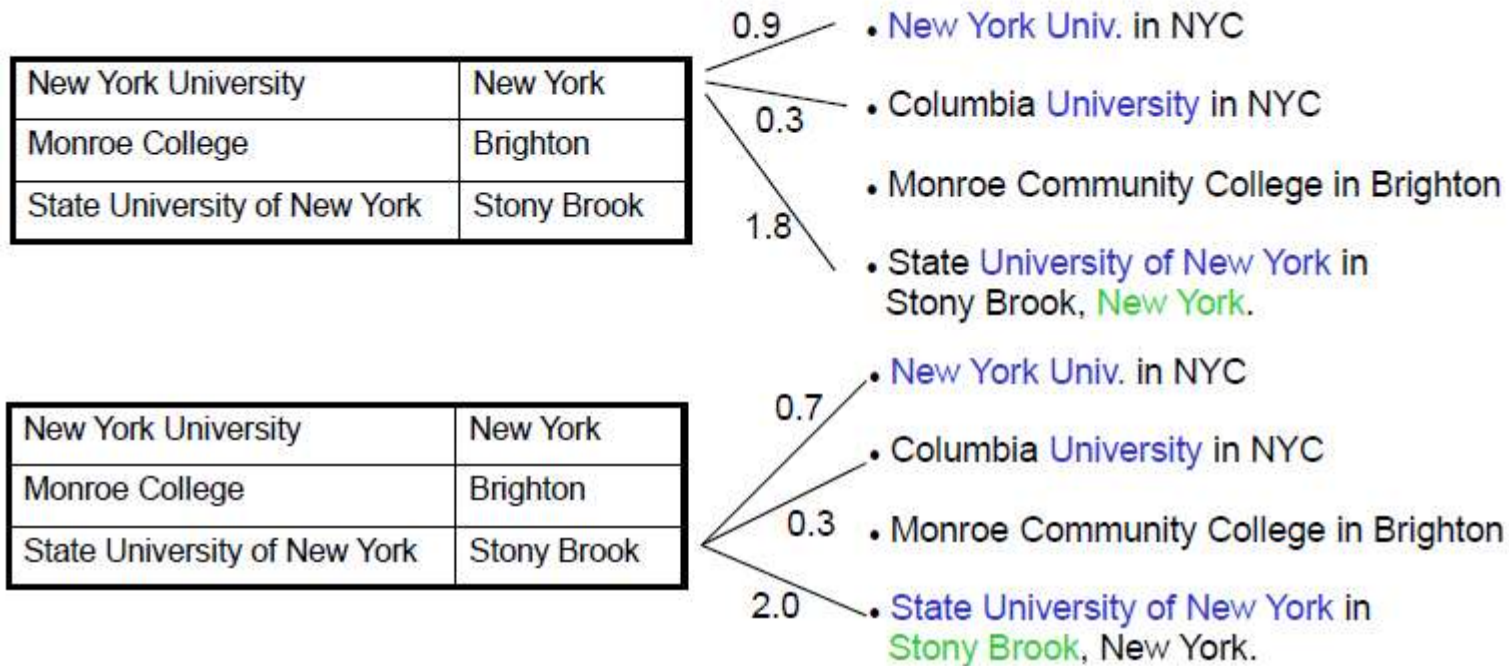


# Extraction: Labeled data generation



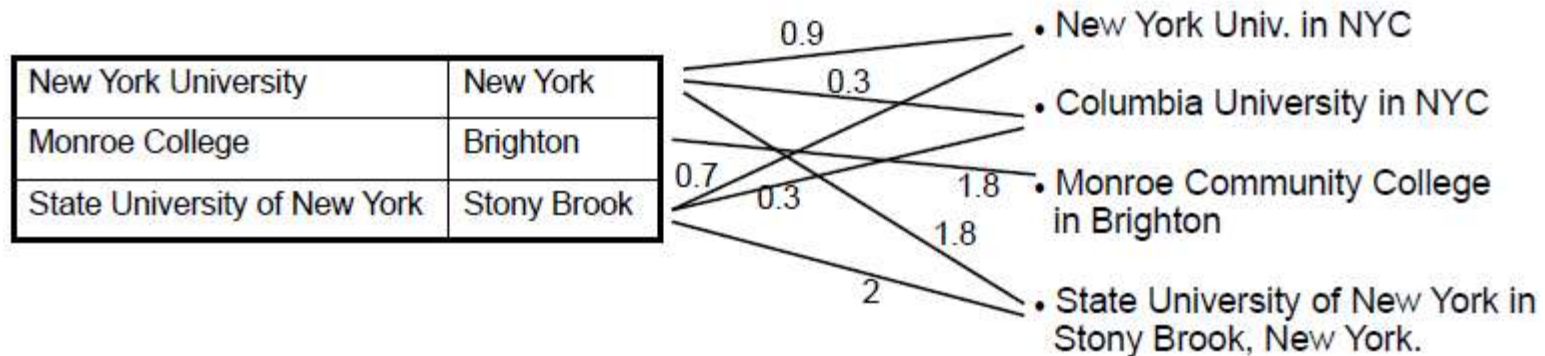
- Hard matching criteria has significantly low recall
  - Missed segments.
  - Does not use natural clues like Univ = University
- Greedy matching can be lead to really bad mappings

# Generating labeled data: Soft approach



- Compute the best match score for each query and source row
  - Score not hard but a continuous value
  - Computed as probabilities  $P(\text{cell}, \text{string})$  by cell-string resolvers.
  - Resolver uses similarity functions specific to column-type

# Generating labeled data: Soft approach



- Compute the maximum weight matching
  - Better than greedily choosing the best match for each row
- Soft string-matching increases the labeled candidates significantly
  - Vastly improves recall, leads to better extraction models.

# Extractor

- Use CRF on the generated labeled data
- Feature Set: delimiter and HTML tokens in a window around labeled segments.
- Extractor: Overlap across sources
  - The extractor until now independently transforms each list into table without exploiting overlap among lists
  - Order lists from strong to weak.
  - Build the model for next list as before
  - Extract high confidence records from the list
  - Merge high confidence records with the query (done by consolidator)
  - Re-label weak sources with enhanced query
  - Repeat for all weak sources

# Exploiting Overlap: Staged Extraction

New York University	New York City
University of Buffalo	Buffalo
RPI	Troy

- New York Univ. in New York City.
- Columbia University, New York City.
- Binghamton University.
- Cornell University in Ithaca.
- Syracuse University in Syracuse.
- RPI in Troy.



New York Univ.	New York City	✓
Columbia University	New York City	
Binghamton University	-	
Cornell University	Ithaca	✓
Syracuse University	Syracuse	✓
RPI	Troy	✓

Query

New York University	New York City
University of Buffalo	Buffalo
RPI	Troy

+

New York Univ.	New York City
Cornell University	Ithaca
Syracuse University	Syracuse
RPI	Troy

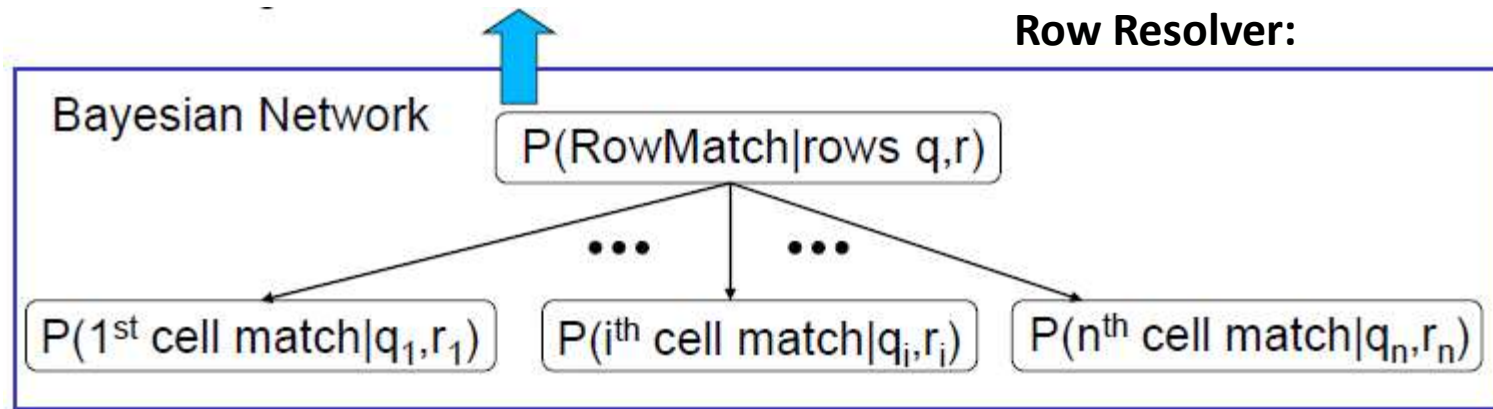
=

New York University OR New York Univ.	New York City
Cornell University	Ithaca
Syracuse University	Syracuse
University of Buffalo	Buffalo
RPI	Troy

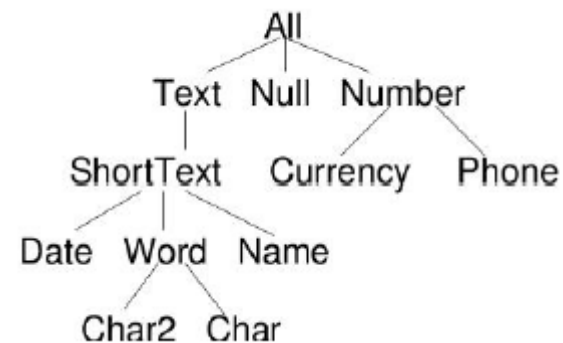
(Enhanced Query)

# Resolver

- Used during consolidation



- Cell Resolver:** Cell-level probabilities
  - Parameters automatically set using list statistics
  - Derived from user-supplied type-specific similarity functions





## Step 2: Consolidation

- Merging the extracted tables into one

Cornell University	Ithaca
State University of New York	Stony Brook
New York University	New York City
Binghamton University	Binghamton

+

SUNY	Stony Brook
New York University (NYU)	New York
RPI	Troy
Columbia University	New York
Syracuse University	Syracuse

=

Cornell University	Ithaca
State University of New York OR SUNY	Stony Brook
New York University OR New York University (NYU)	New York City OR New York
Binghamton University	Binghamton
RPI	Troy
Columbia University	New York
Syracuse University	Syracuse

Merging duplicates

# Consolidation

- Design a Bayesian Network for resolution
  - Parameters set automatically on a per-source basis.
  - Naturally handles missing columns



## Step 3: Ranking

- Just sort by support across lists
  - Junk records that only have spam columns (city/state) come on top. (NY, NYC)
  - All columns assumed equally important.
  - Ignores confidence of extraction (Rochester vs Cornell)
  - Also, confidence decreases when more columns present

School	Location	State	Merged Row Confidence	Support
-	-	NY	0.99	9
-	NYC	New York	0.95	7
New York Univ. OR New York University	New York City OR New York	New York	0.85	4
University of Rochester OR Univ. of Rochester,	Rochester	New York	0.50	2
University of Buffalo	Buffalo	New York	0.70	2
Cornell University	Ithaca	New York	0.76	1

## Step 3: Ranking

- Ordering consolidated records by relevance

		Support	Total Row Confidence
-	NYC	9	0.95
Cornell University	Ithaca	2	0.90
State University of New York <b>OR</b> SUNY	Stony Brook	2	0.80
New York University <b>OR</b> New York University (NYU)	New York City <b>OR</b> New York	3	0.82
Binghamton University	Binghamton	1	0.90
RPI	Troy	1	0.94
Columbia University	New York	2	0.91
Syracuse University	Syracuse	1	0.85

- Reward records supported by multiple sources
- Penalize records with only common “spam” columns e.g. City, State
- Reward records confidently extracted by the statistical extractor

# Ranking: “Cell-SoftMax” Criteria

- Score(Row r):  $\underbrace{\text{Importance}(\text{column } c)}_{\substack{\text{More for text,} \\ \text{long strings}}} \times \underbrace{\text{Cell-extraction-confidence}(c, \text{support})}_{\substack{\text{Obtained from CRFs.} \\ \text{Support included via noisy-OR}}}$   

↓

Favors more non-null cells

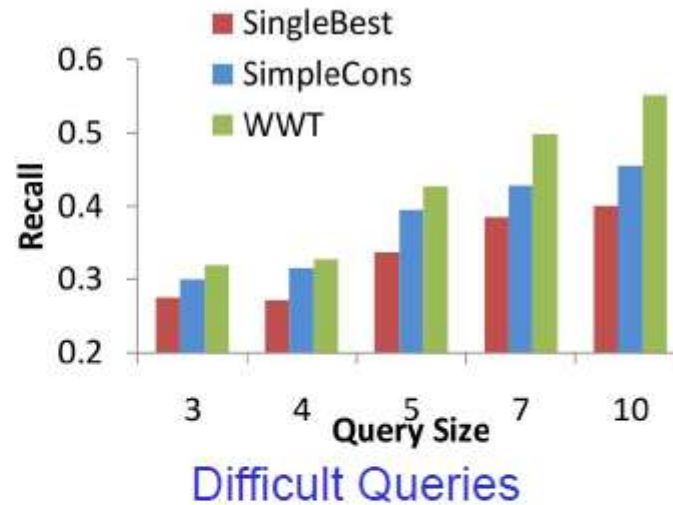
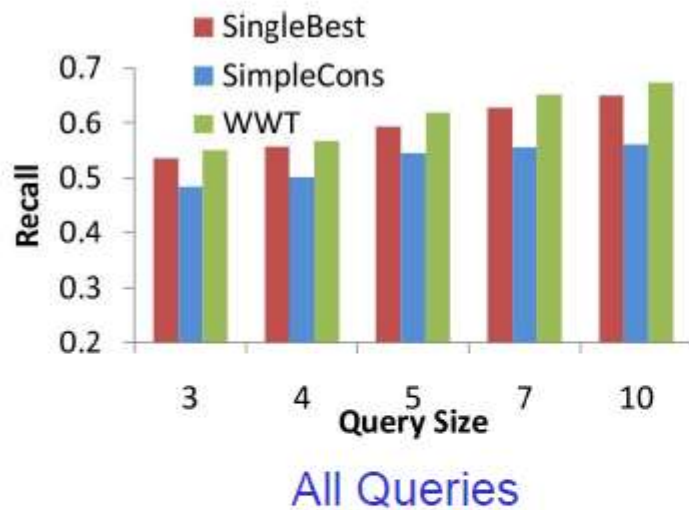
School	Location	State	Merged Row Confidence	Support
New York Univ. OR New York University (0.90)	New York City OR New York (0.95)	New York (0.98)	0.85	4
University of Buffalo (0.88)	Buffalo (0.99)	New York (0.99)	0.70	2
Cornell University (0.92)	Ithaca (0.95)	New York (0.99)	0.76	1
University of Rochester OR Univ. of Rochester, (0.80)	Rochester (0.95)	New York (0.99)	0.50	2
-	-	NY (0.99)	0.99	9
-	NYC (0.98)	New York (0.98)	0.95	7

Gain in recall at 10% error vs Additive: +10%

# Experiments

- Corpus
  - 16M lists from 500M pages.
  - 45% of lists retrieved by index probe are irrelevant.
- Query workload
  - 65 queries. Ground truth hand-labeled by 10 users over 1300 lists.
  - 27% queries not answerable with one list (difficult).
  - True consolidated table = 75% of Wikipedia table, 25% new rows not present in Wikipedia.

# Overall performance



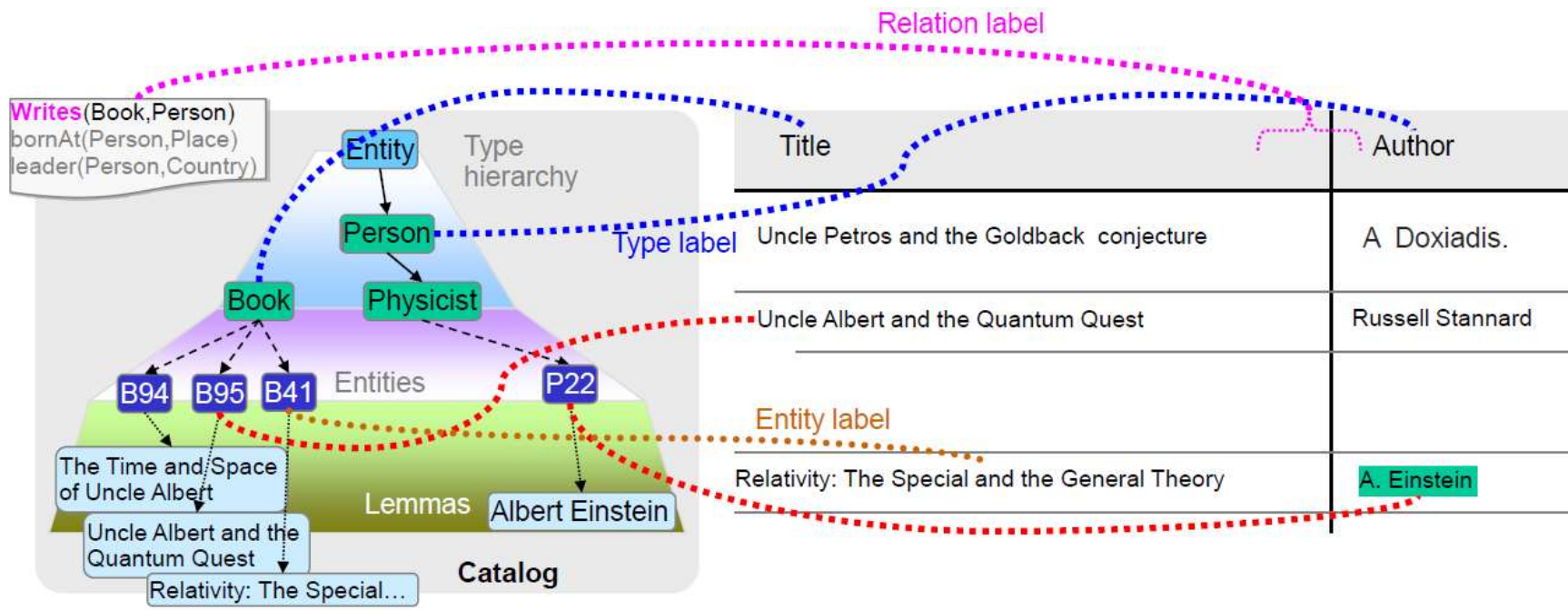
- Justify sophisticated consolidation and resolution. So compare with:
  - Processing only the magically known single best list  
=> no consolidation/resolution required.
  - Simple consolidation. No merging of approximate duplicates.
- WWT has > 55% recall, beats others. Gain bigger for difficult queries.

# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - Answering Table Augmentation Queries from Unstructured Lists on the Web
  - **Annotating Tables with Ontological Links**
- Extracting Sets from the Web



# Entity, Type, and Relation links



# Challenges

- Ambiguity of entity names
  - “Hydrogen” both a chemical element and a place name
- Noisy mentions of entity names
  - A. Einstein Vs Albert Einstein
- Multiple labels
  - YAGO Ontology has average 2.2 types per entity
- Missing type links in Ontology → cannot use least common ancestor
  - Universities in Toronto → Universities in Ontario.
  - SatyajitRay → Indian film directors



# Collective labeling via graphical models

- Variables

$e_{rc}$  = Entity label in row  $r$  column  $c$

$t_c$  = Type label of column  $c$

$b_{cc'}$  = Relation between columns  $c$  and  $c'$

- Potentials

- Entity  $\phi_1(r, c, e_{rc}) = \exp(\mathbf{w}_1^\top \mathbf{f}_1(r, c, e_{rc}))$ .

- Similarity between cell  $(r, c)$  in table and lemmas of entity  $e_{rc}$  in catalog

- Type  $\phi_2(c, t_c) = \exp(\mathbf{w}_2^\top \mathbf{f}_2(c, t_c))$

- Similarity between header & context of column  $c$  in table and lemmas of type  $t_c$  in catalog
- The specificity of a type  $t_c$ :  $1/|\text{Entities in } t_c|$

# Potentials (continued)

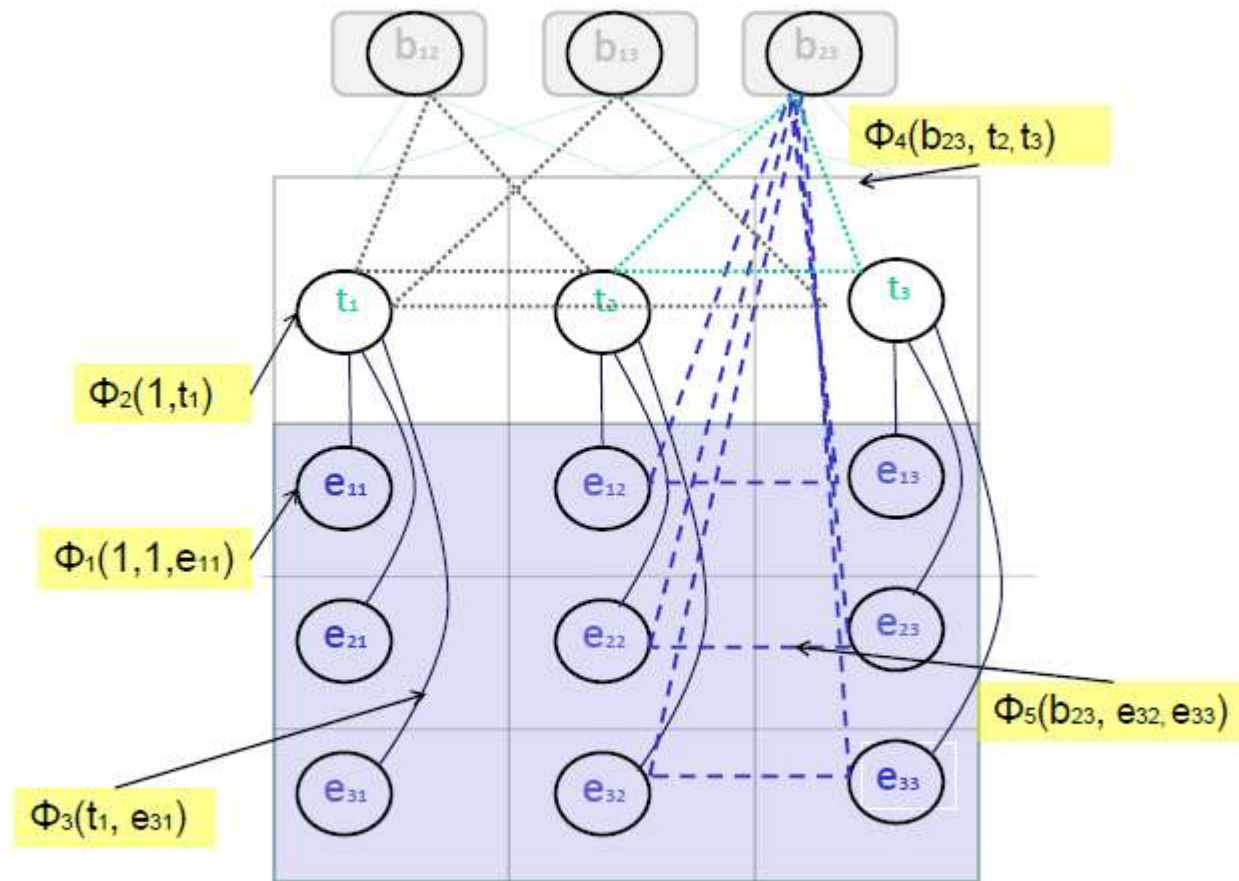
Entity-Type  $\phi_3(t_c, e_{rc}) = \exp(\mathbf{w}_3^\top \mathbf{f}_3(t_c, e_{rc}))$

- 1▷  $e_{rc}$  has path to  $t_c$ : (Einstein, Physicist)
  - Inverse distance between them
    - Penalizes over-generalization
- 2▷  $e_{rc}$  has no path to  $t_c$ : (Julius Plucker, German Physicist)
  - Fraction of  $e_{rc}$ 's siblings with  $t_c$ 
    - Julius Plucker is a Mathematician, many mathematicians are physicists
    - Newton is an English Physicist, overlap with German Physicists zero.
- 3▷  $e_{rc}$  not in the catalog
  - Constant.
    - Allows NA label for columns with many unmatched entities

## Potentials (continued)

- **Relation-Type-Type**  $\phi_4(b_{cc'}, t_c, t_{c'}) = \exp(\mathbf{w}_4^\top \mathbf{f}_4(b_{cc'}, t_c, t_{c'}))$ 
  - Frequency of occurrence of this relationship in the catalog
- **Relation-Entity-Entity**  
 $\phi_5(b_{cc'}, e_{rc}, e_{rc'}) = \exp(\mathbf{w}_5^\top \mathbf{f}_4(b_{cc'}, e_{rc}, e_{rc'}))$ 
  - 1 if this triple exists in the catalog
  - -1: if the catalog refutes this triple
  - 0: if the catalog is neutral

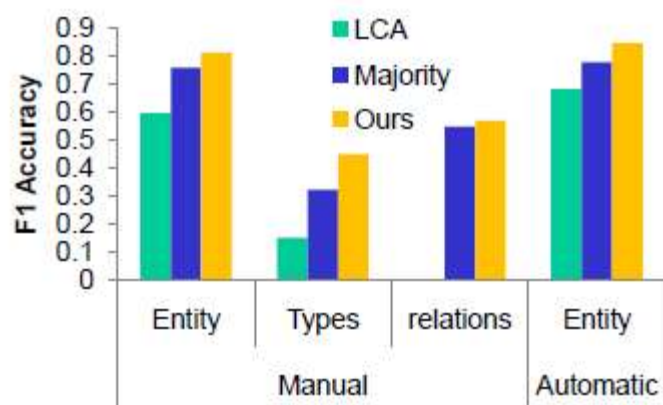
# Inference



Exact: NP-hard, Belief propagation on factor graph

# Accuracy of joint labeling

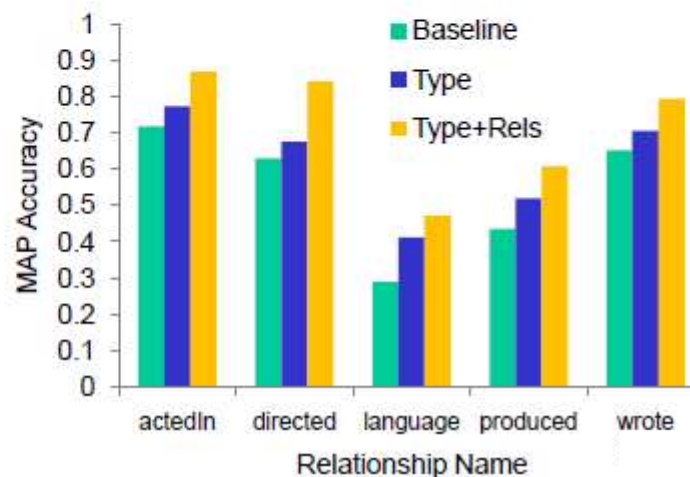
- Dataset
  - Manually labeled
    - 450 tables spanning general Web and Wikipedia
  - Automatically labeled
    - 650 tables from Wikipedia where cells have entity links



# Impact on query accuracy

Given inputs  $R, T_1, T_2, E_2 \in^+ T_2$ , return all  $E_1 \in^+ T_1$  such that  $R(E_1, E_2)$  holds.

- Movies: directed-by person=“George Clooney”
- Countries: hasOfficial language=“Spanish”
- Workload:
  - Five relations,
  - 40 queries per relation
- Ground truth: DBPedia





# Today's Agenda

- Extracting Top-K Lists from the Web
- Extracting Web Data Records Containing User-Generated Content
- Extracting Tables from the Web
  - WebTables: Exploring the Power of Tables on the Web
  - Answering Table Augmentation Queries from Unstructured Lists on the Web
  - Annotating Tables with Ontological Links
- **Extracting Sets from the Web**

# WebSets

- Bhavana Dalvi, William W. Cohen, Jamie Callan.  
WebSets: Extracting Sets of Entities from the Web  
Using Unsupervised Information Extraction. WSDM  
2012
- Extracting concept-instance pairs from an HTML corpus
  - Cluster terms found in HTML tables
  - Assign concept names to these clusters using Hearst patterns
- Past approaches
  - Hyponym patterns (Hearst patterns) (like “Xs such as Y”) indicate that X, Y are a concept-instance pair
  - Two terms i and j are coordinate terms if i and j are instances of the same concept



# Approach

- Novel clustering algorithm that finds extremely precise coordinate-term clusters by merging table columns that contain overlapping triplets of instances, and show that this clustering method outperforms k-means
- New method for combining hyponym and coordinate-term information, and show its effectiveness on table-rich corpora
- Allowing a small amount of user input for each coordinate-term cluster can produce concept-instance pairs with accuracy in the high 90's for four different corpora

# WebSets Overview

- Observations
  - entities appearing in a table column possibly belong to the same concept
  - Clustering the table columns will yield sets of entities, each of which potentially belongs to a coherent concept.
- System components
  - Table Identification: Extracting tables from the corpus that are likely to have relational data
  - Entity Clustering: Efficiently clustering the extracted table cells to generate coherent sets of entities.
  - Hypernym Recommendation: Labeling each cluster with an appropriate concept-name (hypernym).

# Entity Clustering

- The triplets data representation

Country	Capital City
India	Delhi
China	Beijing
Canada	Ottawa
France	Paris

Country	Capital City
China	Beijing
Canada	Ottawa
France	Paris
England	London

: TableId= 21, domain= “www.dom1.com”    TableId= 34, URL= “www.dom2.com”

Entities	Tid:Cids	Domains
India,China,Canada	21:1	www.dom1.com
China, Canada, France	21:1, 34:1	www.dom1.com, www.dom2.com
Delhi, Beijing, Ottawa	21:2	www.dom1.com
Beijing, Ottawa, Paris	21:2, 34:2	www.dom1.com, www.dom2.com
Canada, England, France	34:1	www.dom2.com
London, Ottawa, Paris	34:2	www.dom2.com

Triplet records created by WebSets

# Entity Clustering

- Bottom-up Clustering
  - Scan through each triplet record  $t$  which has occurred in at least  $\text{minUniqueDomain}$  distinct domains
  - A triplet and a cluster are represented with the same data-structure
    - a set of entities
    - a set of columnIds in which the entities co-occurred
    - a set of domains in which the entities occurred
  - The clusterer compares the overlap of triplet  $t$  against each cluster  $C_i$ . The triplet  $t$  is added to the first  $C_i$  so that either of the following two cases is true
    - At least 2 entities from  $t$  appear in cluster  $C_i$
    - At least 2 columnIds from  $t$  appear in cluster  $C_i$  (i.e.  $\text{minEntityOverlap} = 2$  and  $\text{minColumnOverlap} = 2$ )
  - If no such overlap is found with existing clusters, the algorithm creates a new cluster and initializes it with the triplet  $t$ .
- This clustering algorithm is order dependent.
  - Order triplets in the descending order of number of distinct domains.

# Hypernym Recommendation

- Hyponym Concept Dataset
  - NELL KB built using data extracted from ClueWeb09 corpus
  - Contains (entity set, filler, entity set) triplets with frequency
    - The filler “\_ and vacations in \_” occurs with the pair “Holidays, Thailand” with a count of one and “Hotels, Italy” with a count of six

Id	Regular expression
1	arg1 such as (w+ (and or))? arg2
2	arg1 (w+ )? (and or) other arg2
3	arg1 include (w+ (and or))? arg2
4	arg1 including (w+ (and or))? arg2

Hyponym	Concepts
USA	country:1000
India	country:200
Paris	city:100, tourist_place:50
Monkey	animal:100, mammal:60
Sparrow	bird:33

Table 4: Regular expressions used to create Hyponym Concept Dataset

An example of Hyponym Concept Dataset

# Hypernym Recommendation

- Assigning Hypernyms to Clusters
  - For each set produced at the end of clustering, find which entities from the set belong to Hyponym Concept Dataset and collect all concepts they co-occur with.
  - Then these concepts are ranked by number of unique entities in the set it co-occurred with.
  - This ranked list serves as hypernym recommendations for the set.
  - We can assign the topmost hypernym in the rank list as the class label for a cluster.

# Experiments

<b>Religions:</b> Buddhism, Christianity, Islam, Sikhism, Taoism, Zoroastrianism, Jainism, Bahai, Judaism, Hinduism, Confucianism
<b>Government:</b> Monarchy, Limited Democracy, Islamic Republic, Parliamentary Self Governing Territory, Parliamentary Republic, Constitutional Republic, Republic Presidential Multiparty System, Constitutional Democracy, Democratic Republic, Parliamentary Democracy
<b>International Organizations:</b> United Nations Children Fund UNICEF, Southeast European Cooperative Initiative SECI, World Trade Organization WTO, Indian Ocean Commission INOC, Economic and Social Council ECOSOC, Caribbean Community and Common Market CARICOM, Western European Union WEU, Black Sea Economic Cooperation Zone BSEC, Nuclear Energy Agency NEA, World Confederation of Labor WCL
<b>Languages:</b> Hebrew, Portuguese, Danish, Anzanian, Croatian, Phoenician, Brazilian, Surinamese, Burkinabe, Barbadian, Cuban

<b>Instruments:</b> Flute, Tuba , String Orchestra, Chimes, Harmonium, Bassoon, Woodwinds, Glockenspiel, French horn, Timpani, Piano
<b>Intervals:</b> Whole tone, Major sixth, Fifth, Perfect fifth, Seventh, Third, Diminished fifth, Whole step, Fourth, Minor seventh, Major third, Minor third
<b>Genres:</b> Smooth jazz, Gothic, Metal rock, Rock, Pop, Hip hop, Rock n roll, Country, Folk, Punk rock
<b>Audio Equipments:</b> Audio editor , General midi synthesizer , Audio recorder , Multichannel digital audio workstation , Drum sequencer , Mixers , Music engraving system , Audio server , Mastering software , Soundfont sample player