Post-processing Wrapper Generated Tables for Labeling Anonymous Datasets

By
Emdad Ahmed
Department of Computer Science
emdad@wayne.edu
Research Goal

• Our research focuses on issues related to web information retrieval and web information integration. In particular, we are focusing on categorizing and classifying data collections into meaningful groups. This direction of research is essential in the field of autonomous information integration and data aggregation where autonomous systems collect information from Internet resources having no proper labels. In such situations, query systems fail to identify required information and cause the production of wrong responses. Our focus in this research is on generating useful labels for groups of data items to facilitate successful query computation.
Outline of the Presentation

• Abstract
• Problem Description
• Introduction
• Related Work
• Overview of the On going Work
• **Column Name Identification**
• Proposed Work and Possible Future Work
• Conclusions
Column Name Identification

• Google is researching on how to extract the data from the HTML tables and later on how to use this information for searching using keywords.
Google to use tables on web

Posted in SEO Knowledge by SeoTops on August 28th, 2008

World Wide Web Consortium
W3C primarily pursues its mission through the Web, designed to ensure long-term growth for the Web. www.w3.org/ - 33k - Cached - Similar pages

The Web Standards Project
Devoted to adhering to W3C standards in Web

When the earlier search engines used lists to extract information needed for providing results to queries, Google is trying to extract the information from tables. Google is trying to take the information from tables where meaningful data is contained. They will take the data from the tables and will create a separate database for them and will try to relate the data in different tables to each other. They will later allow people to search through these data.

All this is possible because of the peculiar structure of the tables. The tables have columns and there are labels for these columns. This will help Google to extract the information with the corresponding label and will store it in a database. This data stored in the database can be used to help searchers in the later.

The pages in websites can have structured as well as unstructured data. The unstructured information will not be in the form of tables with labels and hence wont have values for the labels. On the other hand, the structured data will be more organized.

Google is researching on how to extract the data from the HTML tables and later on how to use this information for searching using keywords.

Subscribe to our blog to receive new posts and updates by Email

ShareThis
Abstract

- Working toward developing a framework for automated information extraction from Internet documents

- Developing an intelligent Column Name Assignment algorithm, *LADS*

- Current technology does not support assigning meaningful column names to the extracted tables when they are missing

- Used web search engine e.g Google, Yahoo and MSN to execute speculative query

- Our algorithm, *LADS* achieved as high as 93% *probability* of assigning good label to Anonymous Datasets
Introduction

- Web Data Extraction, wrapper, Hidden Web
- Information Extraction and *Labeling/ Annotation* are two separate processes. *Labeling/ Annotation* is a classification process
- Most of the wrapper systems cannot assign field labels to the extracted data records
- Working on Post Processing part of Web Wrapper

*Web Information acquisition process*
Thinking

• In the absence of an existing ontology in a domain, how to come up with terminology that are representative terms of the domain?

• **BioFlow** is a language to solve Web Data Integration in an ad hoc manner

• BioFlow deals with *Web mashups* and *screen scraping* techniques

• We want to give a database abstraction for the web, treat each web page as relation

• Relation has column name, web table may or may **NOT** have column name. Given a web table, how to label the columns with different terminology?
Problem Statement: Sample User Query

• Query 1:
  – Select President, RunningMate
  – From Site S1
  – Where year = 1930;

• Query 2:
  – Select musician, song
  – From Site S2;

• Query 3:
  – Select City, Population
  – From Site S3;

• Query 4:
  – Select Name, Country, Discipline
  – From Site S4;
<table>
<thead>
<tr>
<th>President</th>
<th>Years</th>
<th>Political Party</th>
<th>Vice Presidents</th>
</tr>
</thead>
<tbody>
<tr>
<td>George Washington (1732-1799)</td>
<td>1789-1797</td>
<td>None, Federalist</td>
<td>John Adams</td>
</tr>
<tr>
<td>John Adams (1735-1826)</td>
<td>1797-1801</td>
<td>Federalist</td>
<td>Thomas Jefferson</td>
</tr>
<tr>
<td>Thomas Jefferson (1743-1826)</td>
<td>1801-1809</td>
<td>Democratic-Republican</td>
<td>Aaron Burr, George Clinton</td>
</tr>
<tr>
<td>James Madison (1751-1836)</td>
<td>1809-1817</td>
<td>Democratic-Republican</td>
<td>George Clinton, Elbridge Gerry</td>
</tr>
<tr>
<td>James Monroe (1758-1831)</td>
<td>1817-1825</td>
<td>Democratic-Republican</td>
<td>Daniel Tompkins</td>
</tr>
<tr>
<td>John Quincy Adams (1767-1848)</td>
<td>1825-1829</td>
<td>Democratic-Republican</td>
<td>John Calhoun</td>
</tr>
<tr>
<td>Andrew Jackson (1767-1845)</td>
<td>1829-1837</td>
<td>Democrat</td>
<td>John Calhoun, Martin van Buren</td>
</tr>
<tr>
<td>Martin van Buren (1782-1862)</td>
<td>1837-1841</td>
<td>Democrat</td>
<td>Richard Johnson</td>
</tr>
<tr>
<td>William H. Harrison (1773-1841)</td>
<td>1841</td>
<td>Whig</td>
<td>John Tyler</td>
</tr>
<tr>
<td>John Tyler (1790-1862)</td>
<td>1841-1845</td>
<td>Whig</td>
<td></td>
</tr>
<tr>
<td>James K. Polk (1795-1849)</td>
<td>1845-1849</td>
<td>Democrat</td>
<td>George Dallas</td>
</tr>
<tr>
<td>Zachary Taylor (1784-1850)</td>
<td>1849-1850</td>
<td>Whig</td>
<td>Millard Fillmore</td>
</tr>
<tr>
<td>Millard Fillmore (1800-1874)</td>
<td>1850-1853</td>
<td>Whig</td>
<td></td>
</tr>
<tr>
<td>Franklin Pierce (1804-1869)</td>
<td>1853-1857</td>
<td>Democrat</td>
<td>William King</td>
</tr>
<tr>
<td>James Buchanan (1791-1868)</td>
<td>1857-1861</td>
<td>Democrat</td>
<td>John Breckinridge</td>
</tr>
<tr>
<td>Abraham Lincoln (1809-1865)</td>
<td>1861-1865</td>
<td>Republican</td>
<td>Hannibal Hamlin, Andrew Johnson</td>
</tr>
<tr>
<td>Andrew Johnson (1808-1875)</td>
<td>1865-1869</td>
<td>National Union</td>
<td></td>
</tr>
<tr>
<td>Ulysses S. Grant (1822-1885)</td>
<td>1869-1877</td>
<td>Republican</td>
<td>Schuyler Colfax</td>
</tr>
<tr>
<td>Rutherford Hayes (1822-1893)</td>
<td>1877-1881</td>
<td>Republican</td>
<td>William Wheeler</td>
</tr>
</tbody>
</table>

The Presidents of the USA - EnchantedLearning.com
Screen Shot from Music Domain

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Composer(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>99 Blues</td>
<td>1920</td>
<td>W. Henry McCurdy, M. S. Gordon Saunders</td>
</tr>
<tr>
<td>Sheet Music Book 30 Blues Songs</td>
<td>1765</td>
<td>Arthur Fields, Walter Donovan</td>
</tr>
<tr>
<td>A B C D E F G</td>
<td>1914</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Abyssinian Hymn, The</td>
<td>1544, m. 1804</td>
<td>W. V. von Scheffel, M. Victor Hessler</td>
</tr>
<tr>
<td>Abschied (O Taler weint, o Hohens)</td>
<td>1810</td>
<td>J. L. von Eichendorff, M. F. Mendelssohn-Bartholdy</td>
</tr>
<tr>
<td>Abschied (Ich hab denn wohl, bei stilles Haus!)</td>
<td>1828</td>
<td>W. Ferdinand Ralhum, M. Wenzel Muller</td>
</tr>
<tr>
<td>Abschied vom Drindl</td>
<td>[1856]</td>
<td>German Folksong</td>
</tr>
<tr>
<td>Absence and Return</td>
<td>1890</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Absent</td>
<td>1899</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Abt, Franz</td>
<td>1889-1895</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Accomplished Maid, The</td>
<td>1790</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Ach Du Lieber Augustin</td>
<td>1815</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Acres of Clams</td>
<td></td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Adam Belf</td>
<td>1882</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Adam de la Hale</td>
<td>1240-1285</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Adams and Liberty</td>
<td>1798</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Adeste Fideles</td>
<td>1752</td>
<td>Paul Gerhardt</td>
</tr>
<tr>
<td>Adieu to Dear Cambodia</td>
<td></td>
<td>Traditional Welsh</td>
</tr>
<tr>
<td>Rank</td>
<td>City</td>
<td>Population</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>1</td>
<td>London</td>
<td>7,172,091</td>
</tr>
<tr>
<td>2</td>
<td>Birmingham</td>
<td>970,892</td>
</tr>
<tr>
<td>3</td>
<td>Glasgow</td>
<td>629,501</td>
</tr>
<tr>
<td>4</td>
<td>Liverpool</td>
<td>469,017</td>
</tr>
<tr>
<td>Name</td>
<td>Gender</td>
<td>Date of Birth</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>AALTO Sami</td>
<td>Male</td>
<td>Nov 11 1980</td>
</tr>
<tr>
<td>AAMODT Ragnhild</td>
<td>Female</td>
<td>Sep 09 1980</td>
</tr>
<tr>
<td>AARRASS Jamal</td>
<td>Male</td>
<td>Nov 15 1981</td>
</tr>
<tr>
<td>AAVA Moonika</td>
<td>Female</td>
<td>Jun 19 1979</td>
</tr>
<tr>
<td>ABAJO Jose Luis</td>
<td>Male</td>
<td>Jun 22 1978</td>
</tr>
<tr>
<td>ABAKER Eldoma</td>
<td>Male</td>
<td>Feb 15 1978</td>
</tr>
<tr>
<td>ABAKUMOVA Maria</td>
<td>Female</td>
<td>Jan 15 1986</td>
</tr>
<tr>
<td>ABALDE Tamara</td>
<td>Female</td>
<td>Feb 06 1989</td>
</tr>
<tr>
<td>ABALMASAU Aliaksei</td>
<td>Male</td>
<td>Jun 20 1980</td>
</tr>
<tr>
<td>ABALO Luc</td>
<td>Male</td>
<td>Sep 06 1984</td>
</tr>
<tr>
<td>ABASOVA Tamilla</td>
<td>Female</td>
<td>Dec 09 1982</td>
</tr>
<tr>
<td>ABATE Emanuele</td>
<td>Male</td>
<td>Jul 08 1985</td>
</tr>
<tr>
<td>ABATE Ignazio</td>
<td>Male</td>
<td>Nov 12 1986</td>
</tr>
<tr>
<td>ABATI Joel</td>
<td>Male</td>
<td>Apr 25 1970</td>
</tr>
<tr>
<td>ABBADI Manius</td>
<td>Male</td>
<td>Apr 13 1976</td>
</tr>
</tbody>
</table>
Labeling Use Cases

- *Label is missing* (e.g. search engine result pages, Amazon.com book search results etc)

- Label is there, but *Wrapper can not find them* (e.g. due to lack of TH, THEAD tag) (example in http://www.pdinfo.com/list/a.htm)

- Label is there, but *User view is different* than that of Web page content view (NOC vs Name of Country, http://results.beijing2008.cn/WRM/ENG/BIO/Athlete/A.shtml)
  - The 3rd use cases conform to Schema matching technique (Anupam et al OntoMatch IRI 2009)
  - Our Generic Labeling Algorithm LADS is equally applicable in all 3 use cases, as we consider a subset of the tabular data content to come up with our solution
Motivating Example

- Given a set of homogeneous data values such as
  - \{Toyota, GM, Ford, Honda\}
    - Assign *make* as label
  - \{Apple, Orange, Banana\}
    - Assign *fruit* as label
  - \{India, Pakistan, USA, UK\}
    - Assign *country* as label
  - \{2000, 1999, 1980\}
    - Assign *year* as label

In the context of a given active domain
Automatic Wrapper Induction from Hidden-Web Sources with Domain Knowledge [WIDM 2008]

• There are two parts in extracting valuable data from hidden-Web sources
  • Understanding the structure of a given HTML form and relating its fields to concepts of the domain, and
  • Understanding how resulting records are represented in an HTML result page (column labeling)

• To restrict the scope of this particularly broad problem, we limit ourselves to services related to a given domain of interest, described by domain knowledge. Clearly, with human providing feedback, supervised techniques can go further toward a better understanding of the Web. Result pages are annotated with domain concepts (based on available domain instances that are recognized)
Why contextualize into DOMAIN

• As understanding services of the hidden Web is a very broad, difficult, and undoubtedly AI-complete problem, we limit our interest to a specific application domain, relying on available domain-specific knowledge
Domain Labels

The Domain Labels are just a set of concept names that describe the different concepts that may appear in this specific domain. We shall, for example, consider the following self-describing Labels for the research publication domain:

- Title, Author, Date, Journal, Conference.
- Additional Labels (e.g., Publisher, Pages, Volume) could be added in a similar way.
Deep Web DB Abstraction Model

- User Query (Q), Interface Schema (Web Form, F), Result Schema (Table, T)
Example Candidate Labels for Domains

- **Music**
  - Artist, Title, Album

- **Movie**
  - Actor, Director, DirectedBy, Film, Movie, Title

- **Watch**
  - Band, Brand, Display, Gender, Model, Price

- **Political**
  - President, VP, Party, Senator, Governor

- **Demographic**
  - City, County, Country, Latitude, Longitude, Population, Capital

- **Athlete**
  - Name, Gender, DOB, Country, Discipline
More about Labels

• Each **Domain** will have a pre-determined set of Labels, those will be used to Label the Anonymous Datasets

• We consider the set of potential Labels as controlled vocabulary for the domain of interest

• It is like ontology (**which only has schema, not instances of data**).

• We consider the Labels as a set of lexical token, or terms
More about Labels

• Labels are useful for building the concept taxonomies. (like age which can be infant, adolescent, old etc)

• Our laborious method of column labeling will help build ontology, later can help user to write query

• Labels might also be extracted from Wikipedia taxonomy with some linguistic analysis that would offer a set of words that could be used to identify Labels, although this would be harder

• Labels could be developed by domain experts, learned from user behavior
Conceptualization Level

Ontology

Thesaurus

Vocabulary

Knowledgebase
Why contextualize Queries into Domains?

• Suppose we want to label $Sin$
  – Genome
  – Trigonometry
  – Natural Language
  – Immigration System

• Suppose we want to label $CD$
  – Computer Peripherals
  – Chemistry
  – Music
Why contextualize Queries into Domains?

Because correct Labeling depends on the Sense of a word

- Suppose we want to label *Washington*
  - President
  - University
  - US State
  - Capital

- **Word Sense Disambiguation** problem
- We need to consider one domain at a time
- Handling multiple domain at a time will lead to *spurious result*
- We assume that each value will have one type
Statistical Distribution of isA pattern for Washington

Figure 4.3: Statistical distribution of 'is a' patterns for Washington
More Examples of WSD

• Suppose we want to label *hot*
  – High temperature
  – Fiery
  – Excited
  – Eager
  – Spicy
  – Simply
  – incredibly good-looking

• Suppose we want to label *Java*
  – Programming Language
  – Island
  – Coffee
Label Sense Disambiguation

Label depends on the Sense of a Domain as well:

- Suppose we want to use the label *chair*
  - Furniture
  - Conference
  - Academic

- Suppose we want to use the label *title*
  - Music
  - Vehicle

- Label Sense Disambiguation problem

- We need to consider one domain at a time

- Handling multiple domain at a time will lead to *spurious result*

- We assume that each value will have one type
Semantic Disambiguation

• Try some form of post processing of the Web wrapper to automatically assign column name
• Improve the generalization power of statistical models i.e Naive Baysian.
• Our algorithm LADS achieves fully automatic classification of a set of homogeneous data
• Semi Automatic grouping of a set of labels into a DOMAIN
• Our system is novel in the sense that we accommodate labels from the user query variable as well (more about this in later slides)
Conflicting labels (*most likely synonym*)

- Synonym attributes are negatively correlated
  - synonym attributes are semantically alternatives thus, *rarely co-occur, examples are as follows:*
  - DOB, Date of Birth
  - Present Address, Address
  - Student, Pupil
  - President, Vice President?
  - Artist, singer, performer
  - Doctoral Student, PhD Student
  - Gender, Sex
  - Doctor, Physician
  - Disease, illness
Mining Labels from Web Pages

• Others have proposed methods for augmenting web data extractors to mine labels within the Web Pages

• Problems:
  – Not all pages contain labels and
  – We may not like the labels chosen by the web page authors (User view different from page content view)
The Labeling Problem

Table 1: An anonymous datasets about Computer Network Switch, source: www.pricegrabber.com

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-Link</td>
<td>DGS-2208</td>
<td>8-port</td>
<td>10/100/1000</td>
<td>4.5 stars</td>
<td>$98.79</td>
</tr>
<tr>
<td>Linksys</td>
<td>EG005W</td>
<td>5-Port</td>
<td>10/100/1000</td>
<td>4 stars</td>
<td>$98.99</td>
</tr>
<tr>
<td>Cisco</td>
<td>WSC2960G48TCL</td>
<td>48-port</td>
<td>1000</td>
<td></td>
<td>$9090.00</td>
</tr>
</tbody>
</table>
Steps for Automated Labeling of Anonymous Datasets

• Step 1: Candidate Label Selection
  – Web Form Tag and User Query Variable

• Step 2: Label Assignment
  – Build Speculative Query
  – Execute Speculative Query using Web Search Engine, e.g Google, Yahoo, MSN

• Step 3: Prune and Rank result
  – Find Highest Affinity Labels (statistical fingerprint)
Amazon.com Web Form Interface

Advanced Search

Books Search

Keywords

Author

Title

ISBN(s)

Publisher

Subject

Advanced Search

Browse Subjects

Hot New Releases

Bestsellers

The Ne
User Query Variable

SELECT *\_\_Title, Rating*
FROM amazon.com
WHERE Title like "Discrete Mathematics"
AND Rating=5;
Amazon.com Book search results
Hearst Patterns (by M. Hearst 1992)

- Examples for **hyponymy** (*sub / superconcept relationship*) patterns
  (*lexico-syntactic pattern*):
  - **Vehicles such as cars, trucks and bikes**
    
    \[
    \text{vehicle} \leftarrow \text{car, truck, bike}
    \]
  - **Such fruits as oranges, nectarines or apples,**
    
    \[
    \text{fruit} \leftarrow \text{orange, nectarine, apple}
    \]
  - **Swimming, running and other activities**
    
    \[
    \text{activity} \leftarrow \text{swimming, running}
    \]
  - **Publications, especially papers and books**
    
    \[
    \text{publication} \leftarrow \text{paper, book}
    \]
  - **A seabass is a fish.**
    
    \[
    \text{Fish} \leftarrow \text{seabass}
    \]
  - **Basmati is a rice**
    
    \[
    \text{Rice} \leftarrow \text{Basmati}
    \]
Exhaustive list of Pattern (P)

- **P1**: \(<L> <V>\)
- **H1**: \(<L> \text{ such as } <V | L’}\)
- **H2**: \(\text{such } <L> \text{ as } <V | L’}\)
- **H3**: \(<L> \text{ (especially } | \text{ including) } <V | L’}\)
- **H4**: \(<V | L’> \text{ (and } | \text{ or) other } <L>\)
- **D1**: \(\text{the } <V | L>\)
- **D2**: \(\text{the } <L | V>\)
- **A**: \(<V | L’>, \text{ a } <L>\)
- **C**: \(<V | L’> \text{ is a } <L>\)
Pattern Matching expressions

- artist such as Miles Davis (H1)

- album such as Kind of Blue (H1)

- such albums as A Love Supreme (H2)

- artists, especially Miles Davis (H3)

- John Coltrane and other artists (H4)

- the Kind of Blue album (D1)

- the album Kind of Blue (D2)
Drawing an Analogy

- **Computer Networks**
  - Telephone Network
  - IP Network
  - Again Telephone like Network for QoS

- **Web Database**
  - Raw Text
  - Structured Text
  - Again Raw Text for Labeling
Algorithm LADS

INPUT

- \text{L(1..m)} : A set of labels
- \text{A(1..n)} : A set of anonymous attributes, columns to be labeled
- Precondition \( m > n \), i.e we have more labels \( m \), available to assign to anonymous attributes of size \( n \)
- \text{P (1..k)} : A set of Patterns
- \text{NP (1..l)} : A set of generic numeric patterns (e.g $, year, dd-mm-yyyy)
- \text{V(1..t, 1..n)} : Table Data Value set

VARIABLE

- \text{N} : Number of search engine hits
- \text{H(1..n, 1..m)}: 2D array to store hit count for all anonymous attributes with all labels
- \text{R} \rightarrow 1..t Random indexes between 1..t, generate one time
- Formulate Speculative query: \( \text{L} \times \text{P} \times \text{V} \rightarrow \text{N} \)
Algorithm LADS

**Algorithm 1 LADS: Labeling Anonymous Data Set**

**INPUT**
- \(L(1 \ldots m)\): A set of labels
- \(A(1 \ldots n)\): A set of anonymous attribute
- Precondition \(m > n\)
- \(P(1 \ldots k)\): A set of Pattern
- \(NP(1 \ldots l)\): A set of generic numeric pattern
  (e.g $, year, dd-mm-yyyy$)
- \(V(1 \ldots t, 1 \ldots n)\): Table Data Value set

**VARIABLE**
- \(N\): Number of Google hits
- \(H(1 \ldots n, 1 \ldots m)\): 2D array to store hit count
- \(R \rightarrow 1 \ldots t\) Random indexes between 1 \ldots t
  generate one time

**Formulate Speculative query:** \(L \times P \times V \rightarrow N\)

for \(i = 1\) to \(n\) do
  \(H[i][1..m] = 0\)
  for each \(r \in R\) do
    \(iValue = V_{ri}\)
    for \(j = 1\) to \(m\) do
      \(lValue = L[j]\)
      for \(p = 1\) to \(k\) do
        \(QueryVar = lValue + P[p] + iValue\)
        \(N \leftarrow \text{GoogleExecuteQuery}(QueryVar)\)
        \(H[i][j] += N\)
      end for
    end for
  end for
end for

for \(i = 1\) to \(n\) do
  Let \(z\) be index(1..m) of highest count in \(H[i][1..m]\)
  \(A[i] = L[z]\)
end for
Definition 4.3.1 *(Speculative Query)* A speculative query consists of 3-parts: label, pattern and values. In order to overcome the limitation of keyword based search paradigm, we formulate speculative query to web search engine as exact phrase matching. Example: ”artist Miles Davis” is a speculative query. ◇
Anatomy of a Google Search Results Page
Description of the Legends

- A – Google Navigation Bar
- B – Search Field
- C – Search Button
- D – Advanced Search
- E – Preferences
- F – Search Statistics
- G – Top contextual navigation link
- H – Integrated results
- I – Page title
- J – Text below the title
- K – URL of result
- L – Size
- M – Cached
- N – Similar pages
- O – Indented result
- P – More result
- Q – Plus Box result
- R – Related search terms
Affinity Based Labeling

• We want to estimate the likelihood of a given label $l_i$ being a good descriptor of an anonymous attribute $A_j$
• $P(l_i)$ is the prior basis for the label
• $P(A_j)$ is the normalizing factor, can be ignored
• Then only remaining part is the $P(A_j|l_i)$
• The term will be calculated by hit count from Web Search Engine. (web statistics)
• We want to estimate

$$P(l_i|A_j) = \frac{P(A_j|l_i)P(l_i)}{P(A_j)}$$
Some Observation

- Label $l_i$ and value $v_x$ will appear close to each other in web document.

Figure 1: Flybase Overview Page for Gene sin3
Definition 4.3.5 (Document Count (DC)) The Document Count of a query expression $e$, denoted $DC(e)$, is the number of documents relevant to $e$ according to a given Web search engine.
Definition 4.3.6 (LAA) Given an anonymous relation $R(A_1, \ldots, A_n)$ and a set of candidate labels $L = \{l_1, \ldots, l_m\}$, the Label-Attribute Affinity between $A_j$ and $l_i$, denoted $LAA(A_j, l_i)$, is defined as

$$LAA(A_j, l_i) = P(A_j | l_i) = \frac{1}{|A_j|} \sum_{x=1}^{m} \frac{DC(l_i \land v_x)}{\sum_{y=1}^{m} DC(l_y \land v_x)}$$
Estimating $P(A_j \mid l_i)$

**Document count** of a query expression $e$, denoted $DC(e)$ is the number of documents relevant to $e$ according to a given web search engine (e.g. Google, Yahoo, MSN)

\[
LAA = P(A_j \mid l_i) = \left( \frac{1}{\| A_j \|} \right) \sum_{x=1}^{\| A_j \|} \frac{DC(l_i \land v_x)}{\sum_{y=1}^{m} DC(l_y \land v_x)}
\]
Formulating Speculative Query

• Speculative Query is a statement saying that a given label $l_i$ is a good descriptor for anonymous attribute $A_j$

• Use the number of documents that web search engine classifies as relevant answer to that query to estimate the probabilities

• If label $l_i$ is a better match for $A_j$ than label $l_k$, a Web document containing instance of $A_j$ is more likely to refer to $l_i$ than to $l_k$
The Labeling Problem

- **Input**: Anonymous Datasets, R
- **Output**: Label assignment to $A_1, A_2, \ldots, A_n$
- **Candidate set of Label**: L \{artist, title, album\}

Table 3: An anonymous datasets about music, containing a single relation $R(A_1, A_2, A_3, A_4)$

<table>
<thead>
<tr>
<th>R</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Miles Davis</td>
<td>Kind of Blue</td>
<td>1959</td>
<td>5 stars</td>
</tr>
<tr>
<td></td>
<td>John Coltrane</td>
<td>A Love Supreme</td>
<td>1964</td>
<td>5 stars</td>
</tr>
<tr>
<td></td>
<td>John Coltrane</td>
<td>My Favourite Things</td>
<td>1960</td>
<td>4.5 stars</td>
</tr>
</tbody>
</table>
Formulating Speculative Query

• Example: label = artist; value = Miles Davis

• Exact Phrase match: “artist Miles Davis”

• Why Exact Phrase match?
  - LA is 200 Miles away from California Davis

<table>
<thead>
<tr>
<th>Label</th>
<th>Value</th>
<th>Google</th>
<th>Yahoo</th>
<th>MSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>artist</td>
<td>Miles Davis</td>
<td>53700</td>
<td>78900</td>
<td>27500</td>
</tr>
<tr>
<td>title</td>
<td>Miles Davis</td>
<td>1930</td>
<td>3550</td>
<td>704</td>
</tr>
<tr>
<td>album</td>
<td>Miles Davis</td>
<td>15800</td>
<td>15600</td>
<td>4560</td>
</tr>
<tr>
<td>artist</td>
<td>John Coltrane</td>
<td>28700</td>
<td>44200</td>
<td>10100</td>
</tr>
<tr>
<td>title</td>
<td>John Coltrane</td>
<td>3400</td>
<td>1150</td>
<td>1410</td>
</tr>
<tr>
<td>album</td>
<td>John Coltrane</td>
<td>553</td>
<td>6180</td>
<td>1220</td>
</tr>
<tr>
<td>artist</td>
<td>Kind of Blue</td>
<td>102</td>
<td>104</td>
<td>73</td>
</tr>
<tr>
<td>title</td>
<td>Kind of Blue</td>
<td>715</td>
<td>697</td>
<td>323</td>
</tr>
<tr>
<td>album</td>
<td>Kind of Blue</td>
<td>15400</td>
<td>36000</td>
<td>7660</td>
</tr>
<tr>
<td>artist</td>
<td>A Love Supreme</td>
<td>7</td>
<td>0</td>
<td>44</td>
</tr>
<tr>
<td>title</td>
<td>A Love Supreme</td>
<td>814</td>
<td>465</td>
<td>274</td>
</tr>
<tr>
<td>album</td>
<td>A Love Supreme</td>
<td>10500</td>
<td>1150</td>
<td>3160</td>
</tr>
</tbody>
</table>
Speculative Labeling

• If we populate our probabilistic model above using the tuples in the Anonymous Datasets, we will get as high as 83% probability that $A_1$ is *artist* and 93% probability that $A_2$ is *album*

<table>
<thead>
<tr>
<th>R</th>
<th>$A_1$</th>
<th>$A_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Miles Davis</td>
<td>Kind of Blue</td>
</tr>
<tr>
<td>2</td>
<td>John Coltrane</td>
<td>A Love Supreme</td>
</tr>
<tr>
<td>3</td>
<td>John Coltrane</td>
<td>My Favourite Things</td>
</tr>
</tbody>
</table>

Table 4: LAA based on Different Search Engine result

<table>
<thead>
<tr>
<th></th>
<th>Google</th>
<th>Yahoo</th>
<th>MSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(A_1 \mid$ artist)</td>
<td>0.8153</td>
<td>0.8311</td>
<td>0.8163</td>
</tr>
<tr>
<td>$P(A_1 \mid$ title)</td>
<td>0.0655</td>
<td>0.029</td>
<td>0.0658</td>
</tr>
<tr>
<td>$P(A_1 \mid$ album)</td>
<td>0.0194</td>
<td>0.1395</td>
<td>0.1174</td>
</tr>
<tr>
<td>$P(A_2 \mid$ artist)</td>
<td>0.003</td>
<td>0.001</td>
<td>0.0108</td>
</tr>
<tr>
<td>$P(A_2 \mid$ title)</td>
<td>0.0579</td>
<td>0.1534</td>
<td>0.0593</td>
</tr>
<tr>
<td>$P(A_2 \mid$ album)</td>
<td>0.9358</td>
<td>0.8451</td>
<td>0.9296</td>
</tr>
</tbody>
</table>
Rephrase of the Labeling problem

- Maximum weight matching of a complete bipartite graph $G(V, E)$
- Vertices are the columns in $R$ and $L$
- Weight of each edge $(A_j, L_i)$ indicates how well $L_i$ describes $A_j$
- Use a Greedy Labeling Strategy
  - Find a label for each attribute in isolation
  - Performance improvement of $n(n-1)/2$, where $n$ is the number of anonymous attributes
Sample Anonymous Datasets from Movie and Watch Domain

Table 7: Sample Anonymous Datasets from Movie Domain, Labels: movie, film, rating, director, genre

<table>
<thead>
<tr>
<th>A₁</th>
<th>A₂</th>
<th>A₃</th>
<th>A₄</th>
<th>A₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>the spy who loved me</td>
<td>Lewis Gilbert</td>
<td>3.5 stars</td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td>The World is not enough</td>
<td>Michael Apted</td>
<td>3 stars</td>
<td>Spy film</td>
<td></td>
</tr>
<tr>
<td>Saving Private Ryan</td>
<td>Steven Spielberg</td>
<td>5 stars</td>
<td>War</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A₁</th>
<th>A₂</th>
<th>A₃</th>
<th>A₄</th>
<th>A₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armani</td>
<td>AR5447</td>
<td>Ladies</td>
<td>Stainless steel bracelet</td>
<td>$195.00</td>
</tr>
<tr>
<td>Seiko</td>
<td>SDWG32</td>
<td>Men</td>
<td>Stainless steel Butterfly clasp</td>
<td>$400.00</td>
</tr>
<tr>
<td>Longines</td>
<td>L51580966</td>
<td>Petite</td>
<td>Stainless steel bracelet</td>
<td>$1,700.00</td>
</tr>
<tr>
<td>Casio</td>
<td>DW5600E1V</td>
<td>Men</td>
<td>Plastic, black</td>
<td>$70.00</td>
</tr>
</tbody>
</table>
Statistical Fingerprints

• What remains?
  – Matching patterns in a corpus
  – Obtaining the most relevant result, \((L,V)\) pair for the instance

• World Wide Web is the biggest corpus available

• Using Google API, ‘Exact Phrase Match’
  – Not downloading web pages for further processing
  – Just taking the number of web pages in which pattern appears

• Instantiate all the patterns with each concept from the given ontology
Statistical Fingerprints

• Counting
  – For certain instances or concepts, generates all possible patterns for each concepts in the given ontology
  – Entity, $e \in E$, Concept, $c \in C$, pattern $p \in \{\text{Hearst1, ..., Copula}\}$ $\text{count}_r(e, c, p)$ returns the number of Google hits of instantiated pattern
  – Finding the top hit pair with maximal counts
    • Principle of disambiguation by maximal evidence
Cases of Statistical Fingerprints

\[ count_r(e, c) = \sum_{p \in P_r} \text{count}(e, c, p) \]

\[ SF(e, r, C) := \{(c, n) \mid c \in C \land n = count_r(e, c)\} \]

\[ SF_{\max}(e, r, C) := \{(c, n) \mid c := \arg\max_{c' \in C} count_r(e, c') \land n = count_r(e, c)\} \]

\[ SF_m(e, r, C) := \{(c, n) \mid C = \{c_1, c_2, ..., c|C|\} \land count_r(e, c_1) \leq ... \leq count_r(e, c|C|) \land c \in \{c_1, ..., c_m\} \land n = count_r(e, c)\} \]

\[ SF_\theta(e, r, C) := \{(c, n) \mid count_r(e, c) \geq \theta \land n = count_r(e, c)\} \]

\[ SF_{m, \theta}(e, r, C) = SF_m(e, r, C) \cap SF_\theta(e, r, C) \]

Sum of every pattern matching result of certain concept

(Concept,Count) pair for each concept in ontology

(Concept,Count) pair having maximal count

(Concept,Count) pairs having maximal m-top count

(Concept,Count) pairs having count bigger than Threshold

(Concept,Count) pairs having maximal m-top count with Threshold
Statistical Fingerprints for Armani (Watch Domain)
Statistical Fingerprints for Longines
Implementation Details

• Suppose there are \( n \) anonymous attribute \( \{A_1, A_2, \ldots, A_n\} \) and \( t \) tuples. We will randomly generate \( r \) indices in the range \( \{1..t\} \). Then we will formulate speculative query taking the instances of those \( t \) indices.

• **Data Pre processing**
  – For each anonymous attribute \( A_1, A_2, \ldots A_n \)
    • Determine its content type to be either Numeric or Alphabetic
  – **Subsequent processing only considers alphabetic values**

• We have used all the rules, patterns consistently

• Used Google, Yahoo and MSN Wrapper

• Our column labeling is more robust as it takes into account column content
Some observation

- Most recent events have more hits than historic past events. We got more hits for “President George W. Bush” (8,600,000) than for “President George Washington” (297,000)
- Hits count depend on popularity of the label as well as value
- Query with or without quote produces different number of hits, but the decision of the algorithm never changes
- Our label would be pair wise disjoint, type/ concept wise. Example: Address is a concept. In the label field it should have one instance as addr. If multiple instances are there, the algorithm will NOT be able to distinguish between those
Possible Extension of our Work

- Can we label a domain, not just a single table?
3 Layers of Data Abstraction

• **Concept Layer** (i.e. DOMAIN)
• **Terminology Layer** (defines a mapping between terms and concepts)
• **Data Layer**

• Concept Layer is the highest one (e.g. Music domain).
• Terminology Layer is immediate below the Concept Layer (e.g. terms like musician, song, album)
• Data Layer is the lowest one, hold the actual data i.e. musician name, title of album (e.g. Miles Davis, A Love supreme)
## Layer, Responsibility, Hints

<table>
<thead>
<tr>
<th>Layer</th>
<th>Responsibility</th>
<th>Hints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept layer (i.e Domain)</td>
<td>USER</td>
<td>Select musician, song from site S1 in DOMAIN Music</td>
</tr>
<tr>
<td>Terminology</td>
<td>USER + SYSTEM</td>
<td>SQL query, Form tag, what else ?</td>
</tr>
<tr>
<td>Data Layer</td>
<td>Wrapper</td>
<td>FastWrap <em>(Shafkat et al IRI 2009)</em></td>
</tr>
</tbody>
</table>
Key Aspects

• The key aspect of our perspective is smart focusing on the user and his expectations when searching information on the web. To be able to do this we need the user to share his expectation with us.
• Predefined lexicon-pattern mapping function
• instances are mainly organized in three kinds of patterns
  • Concept Instance Pattern
  • Juxtaposition Instance Pattern
  • Relation-word pattern
Example from Political DOMAIN

• Suppose a user want to query a site in DOMAIN political

• The DOMAIN has a set of terms like \{President, VP, Senator, Governor, Major, Party\}

• Consider the following user query:
  Select President, Vice President
  From site S1 in DOMAIN political
  Where year =1930;
screen shot of the site
Subsequent Query

• In the screen shot, there are 4 columns, whereas user want result from 2 columns

• The user may issue query later to ask for the party, may be as follows:

  Select President, Party
  From site S1 in DOMAIN political
  Where year =1930;

• We may consider a session of queries from a user to the same site
2 Approaches to Column Labeling

• Annotate all the available columns of a table first time in one shot

• Annotate column every time incrementally (See in 2\textsuperscript{nd} query, \textit{party} information was not asked in first query)
How to populate Label in DOMAIN

• Step 1:
  Compute all the affinity of the user’s query variable to table and report result as ranked order list of labels for the anonymous datasets.

• Step 2:
  If user was happy with the result:
    It is an indication that the query was successful.
    Check the DOMAIN’s list of labels.
    For each label in User Query Variable DO
      – If the user supplied query labels are there in DOMAIN, do nothing
      – If the user query labels are NOT there in the DOMAIN, add the user supplied labels to the DOMAIN
    Else
    Do nothing
Benefit of the 2 approaches

• By populating the DOMAIN’s set of label, we will be able to help user write query for a domain
• That is the system will be able to help guide user to write query
• we are achieving two objectives in one shot:
  – *classifying* a set of homogeneous objects into a label (fully automatic)
  – *Grouping* a set of terms in a DOMAIN (semi automatic)
Thinking

• Again, if the system want to materialize the web response in a relational table (Materialized View) so that later query can be served locally instead from remote site, the whole table need to be labeled irrespective of the user query. Then it makes sense to annotate the whole table from all the possible set of Labels of a domain.

• Our position is that, the set of labels for a domain will evolve incrementally as the system learns the user query [we are Refuting WIDM 2008 paper, it says the DOMAIN need to be configured with Labels only one time]
Test Run

import java.io.*;
import java.util.*;
import java.util.StringTokenizer;
import java.util.Random;
import java.util.regex.*;

public class Test1 {

/* To change this template, choose Tools | Templates
and open the template in the editor. */

/**
 * @author ended, last modified April 20, 2009
 */

deleteComma(String message): String
main(String[] args)

album such as Kind of Blue
<b>&lt;/b&gt; for &lt;b&gt;igu
s
album specially Kind of Blue
album including Kind of Blue
artist
album
BUILD SUCCESSFUL (total time: 30 seconds)
More Test Dataset (www.allmovie.com)

- Manually selected Candidate Set of Labels
  - film, movie, Director, Rating, Genre

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>The spy who loved me</td>
<td>Lewis Gilbert</td>
<td>3.5 stars</td>
<td>Action</td>
</tr>
<tr>
<td>The World Is Not Enough</td>
<td>Michael Apted</td>
<td>3 stars</td>
<td>Spy film</td>
</tr>
<tr>
<td>Saving Private Ryan</td>
<td>Steven Spielberg</td>
<td>5 stars</td>
<td>War</td>
</tr>
</tbody>
</table>
Test Run

```java
int [][] X = new int [4][6];  // Rows: number of array elements
int randomTupleIndex;

String L[] = {"artist", "title", "album"};
String L[] = {"actor", "director", "genre", "rating", "title", "film"};
String L[] = {"title", "film", "movie", "director", "rating", "genre"};


```

Integration Informatics Laboratory, Department of Computer Science, Wayne State University
Complexity Analysis of LADS

Algorithm 1 LADS: Labeling Anonymous Data Set

\[
\text{for } i = 1 \text{ to } n \text{ do} \\
\quad H[i][1..m] = 0 \\
\quad \text{for each } r \in R \text{ do} \\
\quad \quad \text{instanceValue} = V_{ri} \\
\quad \quad \text{for } j = 1 \text{ to } m \text{ do} \\
\quad \quad \quad \text{labelValue} = L[j] \\
\quad \quad \quad \text{for } p = 1 \text{ to } k \text{ do} \\
\quad \quad \quad \quad QueryVar = \text{labelValue} + P[p] + \text{instanceValue} \\
\quad \quad \quad \quad N \leftarrow \text{GoogleExecuteQuery}(QueryVar) \\
\quad \quad \quad \quad H[i][j]+ = N \\
\quad \quad \text{end for} \\
\quad \text{end for} \\
\quad \text{end for} \\
\text{end for} \\
\text{for } i = 1 \text{ to } n \text{ do} \\
\quad \text{Let } z \text{ be index}(1..m) \text{ of highest count in } H[i][1..m] \\
\quad A[i] = L[z] \\
\text{end for}
\]
Complexity Analysis of LADS

- $m =$ Number of candidate labels, $L_1, \ldots, L_m$
- $n =$ Number of Anonymous attributes, $A_1, \ldots, A_n$
- $k =$ Number of patterns, $P_1, \ldots, P_k$
- $r =$ Number of random tuples $t_1, \ldots, t_r$
- Complexity $\Theta(mnkr)$

- **Two variant of the LADS:**
  - Naïve LADS (N-LADS)
    - Complexity $\Theta(mn)$
  - Greedy LADS (G-LADS)
    - Cost of operation $mn - n(n-1)/2$
    - Performance improvement of $n(n-1)/2$
Real Life Scenarios

• The number of pattern can be at most 10. We proved that “L V” i.e label followed by value pattern alone is sufficient to disambiguate column labeling. So $k$ is small constant, then the complexity reduces to: $θ (mnr)$.

• Experiment shows that the number of random tuples required for LADS is only $3 ~ 9$. So $r$ can be replaced with a small constant. Then the complexity become: $θ (mn)$.

• According to literature review, the number of anonymous attribute in a web table is the range of $6 ~ 10$. So $n$ can be thought of as small constant, then the complexity of LADS reduces to $θ (m)$, i.e the complexity of the algorithm is $linear$ in the number of supplied candidate set of labels.

• Our experimental result shows that it takes a couple of minute to label the column of a some reasonable size of real life application table.

• The cost is actually the number of queries submitted to Google. It cost represents the network latency for all the queries.

• For the above analysis, maximum number of queries submitted to Google is $m * 10 * 10 * 9 = 900m$
Failure Case Analysis

• Annotated value of 5 stars, 4.5 stars etc as movie
• It should be labeled with rating or rank.
• Explanation:
  – in most of the web pages the rating field is represented as star ** ** ** image, textual label is absent in the most of the cases
Experiment with Synthetic Dataset

- **Candidate Set of Labels**: {Furniture, Toy, Clothing, Electronics}

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
</tr>
</thead>
<tbody>
<tr>
<td>chair</td>
<td>baseball</td>
<td>pants</td>
</tr>
<tr>
<td>desk</td>
<td>doll</td>
<td>coat</td>
</tr>
<tr>
<td>bed</td>
<td>teddy bear</td>
<td>hat</td>
</tr>
<tr>
<td>couch</td>
<td>skipping rope</td>
<td>dress</td>
</tr>
<tr>
<td>table</td>
<td>jigsaw puzzle</td>
<td>scarf</td>
</tr>
</tbody>
</table>
# Experiment with Political Domain

- **Candidate Set of Labels**: \{president, Governor, Senator, vice president, party\}

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
</tr>
</thead>
<tbody>
<tr>
<td>James (Jimmy) Earl Carter, Jr</td>
<td>Democrat</td>
<td>Walter Mondale</td>
</tr>
<tr>
<td>Ronald Wilson Reagan</td>
<td>Republican</td>
<td>George H. W. Bush</td>
</tr>
<tr>
<td>George H. W. Bush</td>
<td>Republican</td>
<td>James Danforth (Dan) Quayle</td>
</tr>
<tr>
<td>William (Bill) Jefferson Clinton</td>
<td>Democrat</td>
<td>Al Gore</td>
</tr>
<tr>
<td>George W. Bush</td>
<td>Republican</td>
<td>Richard Cheney</td>
</tr>
</tbody>
</table>
Test Run

```java
(int l[] = {"president", "Governor", "Senator", "vice president", "party

\text{\{"Jimmy Carter", "Democrat", "Walter Mondale"},
\text{Wilson Reagan", "Republican", "George H. W. Bush"},
\text{N. W. Bush", "Republican", "James Dan Quayle"},
\text{Clinton", "Democrat", "Al Gore"},
\text{ge W. Bush", "Republican", "Richard Cheney"},
\text{ack Obama", "Democrat", "Joseph Biden"

\text{1} = {"", "such as", "Specially", "including"};
\text{1} = new String[8];
```
BAMM Datasets

• http://metaquerier.cs.uiuc.edu/repository/datasets/bamm/browsable.html

• BAMM Dataset, Candidate Set of Label Book Domain (compiled from 55 sources)
  – Title
  – Author, First Name, Last Name
  – Subject
  – Price
  – Publish, Publisher
  – Category
  – Format, binding
  – Publication Date
  – ISBN
  – Keyword
BAMM Datasets

- BAMM Dataset, Candidate Set of Label for Automobile Domain (compiled from 55 sources)
  - Make
  - Model
  - Price
  - Year
  - Mileage
  - Color
  - Class, Type, Style, Category
  - Zip Code, Area, State
BAMM Datasets

• BAMM Dataset, Candidate Set of Label for Movie Domain (compiled from 52 sources)
  – Title
  – Actor, Artist
  – Director
  – Format
  – Type, Genre, Category
  – Star, Rating
  – Cast/Crew, People
  – Price
  – Studio
  – Keyword
BAMM Datasets

- BAMM Dataset, Candidate Set of Label for Music Domain (compiled from 49 sources)
  - Artist
  - Album
  - Song, Title
  - Style
  - Soundtrack
  - Band
  - Genre
  - Label
  - Catalog #
  - Category
  - Keyword
  - Format
Table 4.15: Subjective Evaluation of Labeling

<table>
<thead>
<tr>
<th>Sl no</th>
<th>Domain</th>
<th>No. of anonymous attr</th>
<th>Matches</th>
<th>Mismatches</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Movie</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Music</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Political</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Synthetic</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
Proposed Work and Possible Future Work

Diagram:
- Textual Interface
- Search Interface
- Intelligent Wrapper
- Local Data Repository
- WWW

Integration Informatics Laboratory, Department of Computer Science, Wayne State University
Possible Future Work

• Possible future work includes:
  – Label Column Name based on Google snippets only
  – Label Column Name based on combination of Google hits and Dictionary Extraction
  – Label Column Name based on combination of General Knowledge Source and Domain Specific Knowledge Source (e.g. CooksRecipe.com)
  – We have tested our hypothesis on 5-10 different domains. We wish to test these on broader aspects like 50-100 different domains

• Google snippets
  – Short excerpts from the web page that show a bit of the context of the query term
  – Cached Metadata, shows 10 record at a time
  – Example Query: “Camry is a”
Conclusion

- **Contribution:** A fully automatic, generic method for labeling Anonymous Datasets based on Multiple Search Engine’s Recommendation

- We used the *principle of disambiguation by maximal evidence*. We showed that the proposed algorithm fairly works in different domain.

- *Hits count can be inaccurate*, but have found to be useful in practice

- A novel approach toward the Self-annotating Web

- No laborious and unsupervised manual annotation

- Self-annotating web
  - *The web pages are used to annotate web-pages*
  - *Most powerful because there is a huge amount of implicit knowledge in the Web*
Conclusion (contd)

• Semantics ≈ syntax + statistics
  – Semantics can be obtained by (syntax + statistics)
  – Annotation by maximal statistical result
  – Also based on the collective knowledge system rather than individual system

• new paradigm to overcome the annotation problem
• unsupervised instance categorization
• difficult task: open domain, many categories
• Labeling/ Semantic Annotation still has a long way to go, but it will go a long way as the demand is immense
References

BIBLIOGRAPHY


Acknowledgement

• PhD Dissertation Committee:
  – Dr. Hasan M. Jamil (Advisor)
  – Dr. Farshad Fotouhi (Committee)
  – Dr. Shiyong Lu (Committee)
  – Dr. Geoffry S. Nathan (External Faculty)

• IIL Group PhD Students:
  – Shazzad Hosain, Munirul Islam, Saied Haidarian, Anupam Bhattacharjee, Shafkat Amin, Shahriyar Hossain, Aminul Islam, Zakia Sultana, Saikat Dey

• Other PhD Students:
  – Salahuddin Aziz, Sharrukh Zaman, Yousof Gawasme, Shafi Al Alam, Mohammad Golam Saklayen
Questions ?