

---

# From Web Content Mining to Natural Language Processing

---

Bing Liu  
Department of Computer Science  
University of Illinois at Chicago  
<http://www.cs.uic.edu/~liub>

---

## Introduction

- The Web is perhaps the single largest open data source in the world.
- Web mining aims to develop new techniques to extract/mine useful knowledge from the Web.
- A multidisciplinary field:
  - data mining, machine learning, natural language processing, statistics, databases, information retrieval, multimedia, etc.
- Due to the heterogeneity and lack of structure of Web data, automated mining of targeted or unexpected information is a challenging task.

# The Web: Opportunities & Challenges

- Web offers an unprecedented opportunity and challenges to data mining and NLP,
  - **Huge amount of information**, but easily accessible.
  - **Wide and diverse coverage**: One can find information about almost anything.
  - **Information/data of almost all types**, e.g., structured tables, **texts**, multimedia data, etc.
  - **Semi-structured** due to the nested structure of HTML code.
  - **Linked**, hyperlinks among pages within a site, and across different sites.
  - **Redundant**, the same piece of information or its variants appearing in multiple pages.

- **Noisy**. A Web page typically contains a mixture of many kinds of information, e.g., main contents, advertisements, navigation panels, copyright notices, etc.
- **Surface Web and deep Web**.
  - **Surface Web**: pages that can be browsed using a Web browser.
  - **Deep Web**: databases that can only be accessed through parameterized query interfaces.
- **Services**. Many Web sites enable people to perform operations with input parameters, i.e., they provide services.
- **Dynamic**. Information on the Web changes constantly.
- **Virtual society**. It is not only about data, information and services, but also about interactions among people, organizations and automated systems.

---

## Web mining (Liu, Web Data Mining book 2007)

- Web mining generally consists of:
    - **Web usage mining**: the discovery of user access patterns from Web usage logs.
    - **Web structure mining**: the discovery of useful knowledge from the structure of hyperlinks.
    - **Web content mining**: extraction/mining of useful information or knowledge from Web page contents.
  - This tutorial focuses on **Web content mining** and its strong connection with **NLP**.
- 

---

## Why NLP is so important for Web mining?

- **Everything on the Web is for human consumption** (for people to view) rather than for computer systems to process.
    - Thus all the information (except multi-media content) is expressed in natural language.
  - From mining structured data, semi-structured data to mining unstructured text, NLP is everywhere.
  - Most existing techniques are based on data mining, machine learning and IR methods.
  - But NLP techniques are becoming increasingly necessary.
  - There are applications everywhere.
-

---

## Tutorial topics

- Web content mining is still a large field.
- This tutorial introduces the following topics:
  - Structured data extraction
  - Information integration
  - Information synthesis
  - Opinion mining and summarization
- All those topics have immediate applications.

---

## 1. Structured Data Extraction

---

Wrapper induction  
Automatic data extraction

# Introduction

- A large amount of information on the Web is contained in **regularly structured data objects**.
  - often data records retrieved from databases.
- Such Web data records are important: **lists of products and services**.
- **Applications**: Gather data to provide valued added services
  - comparative shopping, object search (rather than page search), etc.
- **Two types of pages with structured data**:
  - **List pages**, and **detail pages**

# List Page – two lists of products

Two lists



The screenshot shows a web browser window with the title "CompUSA.com - Product Results - Microsoft Internet Explorer". The address bar contains the URL "http://www.compUSA.com/products/products.asp?N=200049&cm\_re=A\_-\_HPF\_-\_Flat+Panel+%28LCD%29". The search engine used is Google. The page content is divided into two main sections:

- Top Sellers:** A horizontal row of four product cards. Each card includes a product image, name, price, and an "Add To Cart" button. The products are:
  - EN7410 17-inch LCD Monitor, Black/Dark Charcoal: \$299.99
  - 17-inch LCD Monitor: \$249.99
  - AL1714ch 17-inch LCD Monitor, Black: \$269.99
  - SyncMaster 712n 17-inch LCD Monitor, Black: \$299.99 (with a "SAVE \$70" offer)
- Main Product List:** A vertical list of three products with detailed information:
  - EN7410 17-inch LCD Monitor, Black/Dark Charcoal: Product Number: 318020, Mfr. Part #: EN7410, Brand: E-Viewsonic, Price: \$299.99
  - 17-inch LCD Monitor: Product Number: 316328, Mfr. Part #: 130611, Brand: Norwood Micro, Price: \$249.99
  - AL1714ch 17-inch LCD Monitor, Black: Product Number: 317993, Mfr. Part #: ET L1809 031, Brand: Acer, Price: \$269.99

# Detail Page – detailed description

View Cart | My Account | Order Status | Help

**COMPUSA**


Consumer Business Services Auctions Locations Gift Cards Free Shipping\*

Computers & Peripherals | Upgrades | Software | Accessories | Electronics | Games & Movies | Office Supplies | See All >

Search:  **GO!** [Check Out Our Interactive Ad](#)

[CompUSA.com](#) » [Categories](#) » [Electronics](#) » [Digital Photography](#) » [Digital Cameras](#)

### Kodak EasyShare Z730 Digital Camera, 5.0 Megapixels



Brand: Kodak [« Visit their Showcase](#)  
Mfg Part #: 8857963  
SKU: 336442

Delivery Only Special - pricing not available in store  
[See product info from Kodak](#)

Customer Rating: ★★★★★ 4.6 out of 5  
[Read all 57 reviews](#) [Rate this product](#)

**Delivery**

Was: \$249.99  
**\$179.99** (28% Off)  
**SAVE \$70** after:  
FREE shipping \$70.00 instant savings

Usually Ships In:  
2 - 4 Weeks  
[Estimate Arrival Time](#)

**In-Store**

**\$249.99**

Ready for Pick-Up In:  
15 Minutes  
[Check Store Availability](#)


**Add to Cart**  
[Protect this product \(learn how\)](#)

[Overview](#) [Tech Specs](#) [Add-Ons](#) [Rebate Info](#) [Ratings / Reviews](#) [Add to Wishlist](#) | [Print](#) | [E-Mail](#) | [Compare](#)

Overview for Kodak EasyShare Z730 Digital Camera, 5.0 Megapixels  
(Based on manufacturer's information)

Imagine. Invent. Inspire.

# Extraction Task: an illustration




**Cabinet Organizers by Copco**

9-in. [Round Turntable: White](#) ★★★★★ \$4.95 **BUY**

12-in. [Round Turntable: White](#) ★★★★★ \$7.95 **BUY**


---



**Cabinet Organizers**

14.75x9 [Cabinet Organizer \(Non-skid\): White](#) ★★★★★ \$7.95 **BUY**

---



**Cabinet Organizers**

22x6 [Cookware Lid Rack](#) ★★★★★ \$19.95 **BUY**

**nesting**

image 1	<a href="#">Cabinet Organizers by Copco</a>	9-in.	Round Turntable: White	*****	\$4.95
image 1	<a href="#">Cabinet Organizers by Copco</a>	12-in.	Round Turntable: White	*****	\$7.95
image 2	Cabinet Organizers	14.75x9	Cabinet Organizer (Non-skid): White	*****	\$7.95
image 2	Cabinet Organizers	22x6	Cookware Lid Rack	*****	\$19.95

# Data Model and Solution

## Web data model: Nested relations

- See formal definitions in (Grumbach and Mecca, ICDT-99; Liu, Web Data Mining book 2006)

## Solve the problem

- Two main types of techniques
  - Wrapper induction – supervised
  - Automatic extraction – unsupervised
- Information that can be exploited
  - Source files (e.g., Web pages in HTML)
    - Represented as strings or trees
  - Visual information (e.g., rendering information)

# Tree and Visual information

1. **Buy new: \$1,194.00**  
Usually ships in 1 to 2 days

Customer Rating:

Best use: (what's this?)	Business:	Portability:	Desktop Replacement:	Entertainment:
--------------------------	-----------	--------------	----------------------	----------------

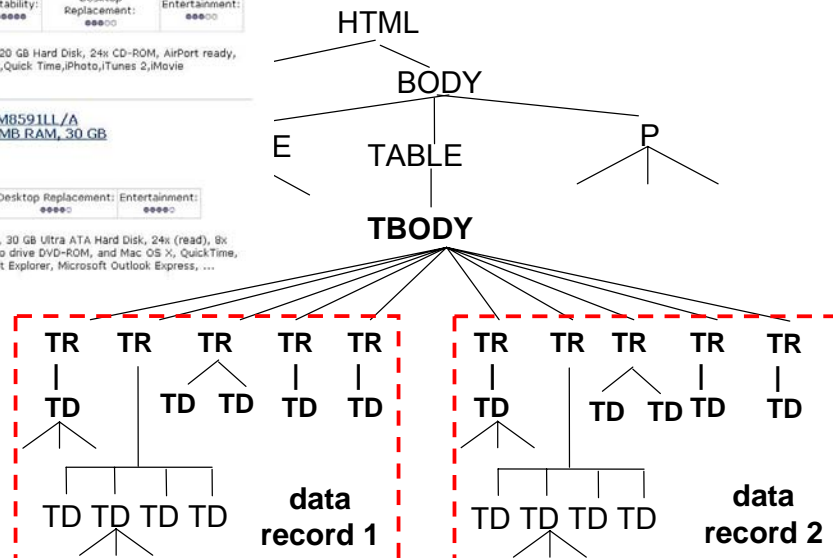
600 MHz PowerPC G3, 128 MB SDRAM, 20 GB Hard Disk, 24x CD-ROM, AirPort ready, and Mac OS X, Mac OS X, Mac OS 9.2, Quick Time, Photo, iTunes 2, iMovie 2, AppleWorks, Microsoft IE

2. **Buy new: \$2,399.99**

Customer Rating:

Best use: (what's this?)	Portability:	Desktop Replacement:	Entertainment:
--------------------------	--------------	----------------------	----------------

667 MHz PowerPC G4, 256 MB SDRAM, 30 GB Ultra ATA Hard Disk, 24x (read), 8x (write) CD-RW, 8x; included via combo drive DVD-ROM, and Mac OS X, QuickTime, iMovie 2, iTunes(6), Microsoft Internet Explorer, Microsoft Outlook Express, ...



---

## Road map

- ➔ ■ **Wrapper Induction (supervised)**
    - Given a set of manually labeled pages, a machine learning method is applied to learn extraction rules or patterns.
  - **Automatic data extraction (unsupervised)**
    - Given only a single page with multiple data records, generate extraction patterns.
    - Given a set of positive pages, generate extraction patterns.
  - **NLP connection**
- 

---

## Wrapper induction

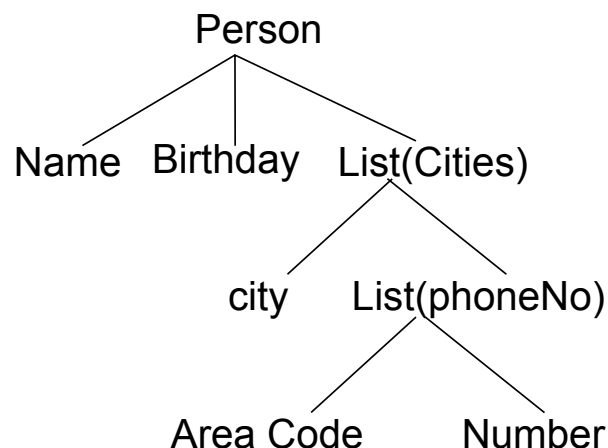
- Using machine learning to generate extraction rules.
    - The user marks/labels the target items in a few training pages.
    - The system learns extraction rules from these pages.
    - The rules are applied to extract target items from other pages.
  - Many wrapper induction systems, e.g.,
    - WIEN (Kushmerick et al, IJCAI-97),
    - Softmealy (Hsu and Dung, 1998),
    - Stalker (Muslea et al. Agents-99),
    - BWI (Freitag and Kushmerick, AAAI-00),
    - WL<sup>2</sup> (Cohen et al. WWW-02),
    - IDE (Zhai and Liu. WISE-05).
  - We will only focus on **Stalker**, which also has a commercial version called **Fetch**.
-

## Stalker: A hierarchical wrapper induction system (Muslea et al. Agents-99)

- Hierarchical wrapper learning
  - Extraction is isolated at different levels of hierarchy
  - This is suitable for nested data records (embedded list)
- Each item is extracted independent of others.
- Manual labeling is needed for each level.
- Each target item is extracted using two rules
  - A **start rule** for detecting the beginning of the target item.
  - A **end rule** for detecting the ending of the target item.

## Hierarchical extraction based on tree

Name: John Smith  
Birthday: Oct 5, 1950  
Cities:  
Chicago:  
    (312) 378 3350  
    (312) 755 1987  
New York:  
    (212) 399 1987



- To extract each target item (a node), the wrapper needs a rule that extracts the item from its parent.

# Wrapper Induction (Muslea et al., Agents-99)

- Using machine learning to generate extraction rules.
  - The user marks the target items in a few training pages.
  - The system learns extraction rules from these pages.
  - The rules are applied to extract items from other pages.

## Training Examples

E1: 513 Pico, <b>Venice</b>, Phone 1-<b>800</b>-555-1515

E2: 90 Colfax, <b>Palms</b>, Phone (800) 508-1570

E3: 523 1<sup>st</sup> St., <b>LA</b>, Phone 1-<b>800</b>-578-2293

E4: 403 La Tijera, <b>Watts</b>, Phone: (310) 798-0008

## Output Extraction Rules

- |                       |                   |
|-----------------------|-------------------|
| ■ <b>Start rules:</b> | <b>End rules:</b> |
| R1: SkipTo(())        | SkipTo(())        |
| R2: SkipTo(<-b>)      | SkipTo(</b>)      |

# Learning extraction rules

- Stalker uses sequential covering to learn extraction rules for each target item.
  - In each iteration, it learns a perfect rule that covers as many positive examples as possible without covering any negative example.
  - Once a positive example is covered by a rule, it is removed.
  - The algorithm ends when all the positive examples are covered. The result is an ordered list of all learned rules

---

## Some other issues in wrapper learning

- **Active learning**
    - How to automatically choose examples for the user to label (Muslea et al, AAAI-00)
    - IDE (Zhai & Liu, WISE-05), which uses instance-based learning, and it is automatically active.
  - **Wrapper verification**
    - Check whether the current wrapper still work properly (Kushmerick, 2003)
  - **Wrapper maintenance**
    - If the wrapper no longer works properly, is it possible to re-label automatically (Kushmerick AAAI-99; Lerman et al, JAIR-03)
- 

---

## Limitations of Supervised Learning

- **Manual Labeling is labor intensive and time consuming, especially if one wants to extract data from a huge number of sites.**
  - **Wrapper maintenance is very costly:**
    - If Web sites change frequently.
    - It is necessary to detect when a wrapper stops to work properly.
    - Any change may make existing extraction rules invalid.
    - Re-learning is needed, and most likely manual re-labeling as well.
-

---

## Road map

- Wrapper Induction (supervised)
    - Given a set of manually labeled pages, a machine learning method is applied to learn extraction rules or patterns.
  - Automatic data extraction (unsupervised)
    - □ Given only a single page with multiple data records, generate extraction patterns.
    - Given a set of positive pages, generate extraction patterns.
  - NLP connection
- 

---

## Automatic Extraction

**There are two main problem formulations:**

**Problem 1:** Extraction based on a single list page (Liu et al. KDD-03; Liu, Web Data Mining book 2007)

**Problem 2:** Extraction based on multiple input pages of the same type (list pages or detail pages) (Grumbach and Mecca, ICDT-99).

- Problem 1 is more general: Algorithms for solving Problem 1 can solve Problem 2.
    - Thus, we only discuss Problem 1.
-

# Automatic Extraction: Problem 1

The screenshot shows a Microsoft Internet Explorer window displaying the CompUSA.com product results page. The browser's address bar shows the URL: `http://www.compUSA.com/products/products.asp?N=200049&cm_re=A_HPF_-_Flat+Panel+LCD%29`. The page content is annotated with dashed green boxes and arrows:

- Data region1**: A dashed green box encloses the top section of the product grid, which includes the 'Top Sellers' header and the first four product cards.
- Data records**: Three arrows point to individual product cards within the 'Data region1' box, indicating the specific data records being extracted.
- Data region2**: A dashed green box encloses the lower section of the product grid, starting from the 'Sort by' dropdown and including the detailed product information for three items.

The product cards in the top section include:

- EN7410 17-inch LCD Monitor, Black/Dark Charcoal: \$299.99
- 17-inch LCD Monitor: \$249.99
- AL1714cb 17-inch LCD Monitor, Black: \$269.99
- SyncMaster 712n 17-inch LCD Monitor, Black: Was: \$369.99, Now: \$299.99 (SAVE \$70 after \$70.00 mail-in rebate(s))

The detailed product information in the lower section includes:

- EN7410 17-inch LCD Monitor, Black/Dark Charcoal: Product Number: 318020, Mr. Part # EN7410, Brand: Eubason, Price: \$299.99
- 17-inch LCD Monitor: Product Number: 318528, Mr. Part # Y30617, Brand: Norwood Micro, Price: \$249.99
- AL1714cb 17-inch LCD Monitor, Black: Product Number: 317993, Mr. Part # ET L1809 031, Brand: Acer, Price: \$269.99

ACL-07 tutorial, by Bing Liu, UIC

25

## Solution Techniques

- Identify data regions and data records
  - By finding repeated patterns
    - string matching (treat HTML source as a string)
    - tree matching (treat HTML source as a tree)
- Align data items: Multiple alignment
  - Many multiple alignment algorithms exist, however, they
    - tend to make unnecessary commitments in early (can be wrong) alignments.
    - inefficient.
  - An new algorithm, called **Partial Tree Alignment**, was proposed to deal with the problems (Zhai and Liu, WWW-05)

## String edit distance (definition)

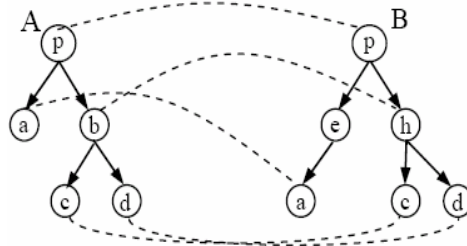
Assume we are given two strings  $s_1$  and  $s_2$ . The following recurrence relations define the edit distance,  $d(s_1, s_2)$ , of two strings  $s_1$  and  $s_2$ :

$$\begin{aligned}d(\varepsilon, \varepsilon) &= 0 && // \varepsilon \text{ represents an empty string} \\d(s, \varepsilon) = d(\varepsilon, s) &= |s| && // |s| \text{ is the length of string } s \\d(s_1+ch_1, s_2+ch_2) &= \min(d(s_1, s_2) + r(ch_1, ch_2), d(s_1+ch_1, s_2) + 1, \\& \quad d(s_1, s_2+ch_2) + 1)\end{aligned}$$

where  $ch_1$  and  $ch_2$  are the last characters of  $s_1$  and  $s_2$  respectively, and  $r(ch_1, ch_2) = 0$  if  $ch_1 = ch_2$ ;  $r(ch_1, ch_2) = 1$ , otherwise.

## Tree matching

- There are many definitions of tree matching and tree edit distances. E.g.,



- Here we only briefly discuss a **restricted tree matching algorithm**, called **Simple Tree Matching** (Yang, 1991), which is quite effective for data extraction
  - No node replacement and no level crossing are allowed.
  - Dynamic programming solution

## Simple tree matching

(Yang 1991; Liu, Web Data Mining book 2007)

- Let  $A = RA:\langle A_1, \dots, A_k \rangle$  and  $B = RB:\langle B_1, \dots, B_n \rangle$  be two trees,
  - where  $RA$  and  $RB$  are the roots of  $A$  and  $B$ , and  $A_i$  and  $B_j$  are their  $i$ -th and  $j$ -th first-level subtrees
- Let  $W(A, B)$  be the number of pairs in the maximum matching of trees  $A$  and  $B$ .
- If  $RA$  and  $RB$  contain identical symbols, then
$$W(A, B) = m(\langle A_1, \dots, A_k \rangle, \langle B_1, \dots, B_n \rangle) + 1,$$
  - where  $m(\langle A_1, \dots, A_k \rangle, \langle B_1, \dots, B_n \rangle)$  is the number of pairs in the maximum matching of  $\langle A_1, \dots, A_k \rangle$  and  $\langle B_1, \dots, B_n \rangle$ .
- If  $RA \neq RB$ ,  $W(A, B) = 0$ .

## Simple tree match formulation

(Liu, Web Data Mining Book 2007)

- A dynamic programming formulation

$$W(A, B) = \begin{cases} 0 & \text{if } R_A \neq R_B \\ m(\langle A_1, \dots, A_k \rangle, \langle B_1, \dots, B_n \rangle) + 1 & \text{otherwise} \end{cases}$$

$$m(\langle \rangle, \langle \rangle) = 0$$

//  $\langle \rangle$  represents an empty sub-tree list.

$$m(s, \langle \rangle) = m(\langle \rangle, s) = 0$$

//  $s$  matches any non-empty sub-tree list

$$m(\langle A_1, \dots, A_k \rangle, \langle B_1, \dots, B_n \rangle) = \max(m(\langle A_1, \dots, A_{k-1} \rangle, \langle B_1, \dots, B_{n-1} \rangle) + W(A_k, B_n),$$

$$m(\langle A_1, \dots, A_k \rangle, \langle B_1, \dots, B_{n-1} \rangle),$$

$$m(\langle A_1, \dots, A_{k-1} \rangle, \langle B_1, \dots, B_n \rangle)).$$

---

## Multiple alignment

- Pairwise alignment/matching is not sufficient because a web page usually contain more than one data record.
  - We need multiple alignment.
  - Optimal alignment/matching is exponential.
  - We discuss two techniques
    - Center Star method
    - Partial tree alignment.
- 

---

## Center star method

- A simple & classic technique. Often used for multiple string alignments, but can be adopted for trees.
- Let the set of strings to be aligned be  $S$ . In the method, a string  $s_c$  that minimizes,

$$\sum_{s_i \in S} d(s_c, s_i)$$

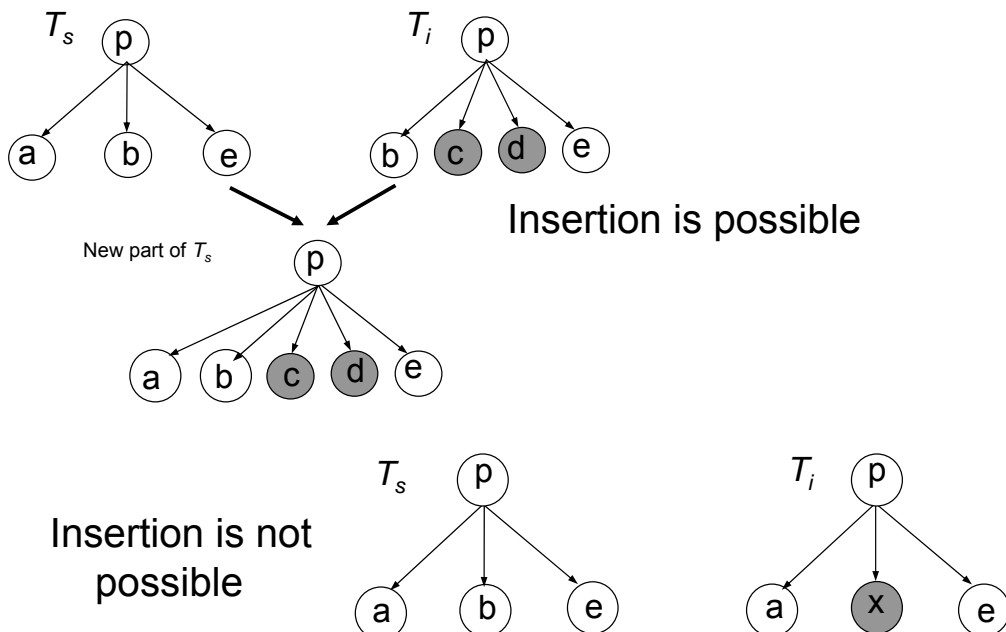
is first selected as the **center string**.  $d(s_c, s_i)$  is the distance of two strings.  $O(k^2n^2)$  pair-wise matches.

- The algorithm then iteratively computes the alignment of rest of the strings with  $s_c$ .
-

## Partial tree alignment (Zhai and Liu, WWW-05)

- **Choose a seed tree:** A seed tree, denoted by  $T_s$ , is picked with the maximum number of data items.
  - The seed tree is similar to center string, but without the  $O(k^2n^2)$  pair-wise tree matching to choose it.
  - **Tree matching:**
    - For each unmatched tree  $T_i$  ( $i \neq s$ ),
      - match  $T_s$  and  $T_i$ .
      - Each pair of matched nodes are linked (aligned).
      - For each unmatched node  $n_j$  in  $T_i$  do
        - expand  $T_s$  by inserting  $n_j$  into  $T_s$  if a position for insertion can be uniquely determined in  $T_s$ .
- The expanded seed tree  $T_s$  is then used in subsequent matching.

## Partial tree alignment of two trees




---

## More information ...

- More information,
  - IEPAD (Chang and Lui, WWW-01)
  - MDR (Liu et al. KDD-03)
  - DeLa (Wang and Lochovsky, WWW-03)
  - Lerman et al (SIGMOD-04)
  - DEPTA (Zhai and Liu, WWW-2005)
  - NET (Liu and Zhai WISE-2005)
  - (Zhao et al, WWW-05)
- (Liu, [Web Data Mining book, 2007](#)) contains formulations, algorithms and more ...

---

## Road map

- Wrapper Induction (supervised)
  - Given a set of manually labeled pages, a machine learning method is applied to learn extraction rules or patterns.
- **Automatic data extraction (unsupervised)**
  - Given only a single page with multiple data records, generate extraction patterns.
  -  □ **Given a set of positive pages, generate extraction patterns.**
- NLP connection

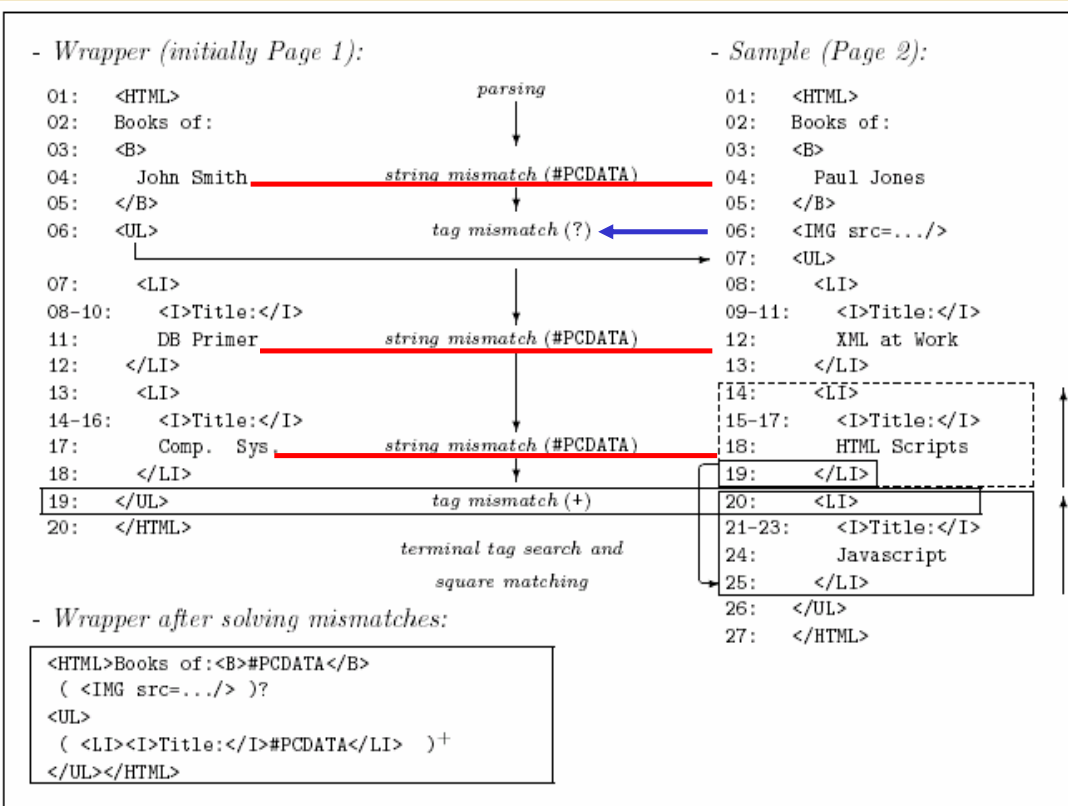
# The RoadRunner System

(Crescenzi et al. VLDB-01)

- Given a set of positive examples (multiple sample pages). Each contains one or more data records.
- From these pages, generate a wrapper as a union-free regular expression (i.e., no disjunction).

## The approach

- To start, a sample page is taken as the wrapper.
- The wrapper is then refined by solving mismatches between the wrapper and each sample page, which generalizes the wrapper.
  - A mismatch occurs when some token in the sample does not match the grammar of the wrapper.



---

## The EXALG System

(Arasu and Garcia-Molina, SIGMOD-03)

- The same setting as for RoadRunner: need multiple input pages of the same template.

### The approach:

Step 1: find sets of tokens (called equivalence classes) having the same frequency of occurrence in every page.

Step 2: expand the sets by differentiating “roles” of tokens using contexts. Same token in different contexts are treated as different tokens.

Step 3: build the page template using the equivalence classes based on what is in between two consecutive tokens, empty, data or list.

---

---

## Road map

- Wrapper Induction (supervised)
  - Given a set of manually labeled pages, a machine learning method is applied to learn extraction rules or patterns.
- Automatic data extraction (unsupervised)
  - Given only a single page with multiple data records, generate extraction patterns.
  - Given a set of positive pages, generate extraction patterns.

 ■ **NLP connection**

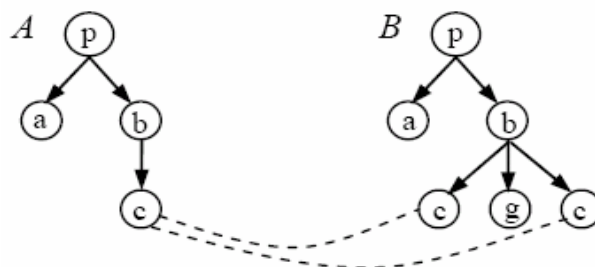
---

# Where is NLP?

- **Two places**
  - Item matching based on text.
  - Integration of data from multiple sites.
- Regarding structured data, the key difference between those on the Web and those in relational databases is that
  - Web uses natural language: It is for general public to view. Few abbreviations or acronyms .
  - Databases may not: Abbreviations or acronyms are widely used.

# Content item matching

- **In tree matching, it is possible that multiple matches can give the same maximum score: Which is right?**
- For example, node *c* in tree *A* can match either the first or the last node *c* in tree *B*.



- **Current method: Word match**, but not enough
  - “Ship in 24 hours” =? “Overnight delivery”

---

## Integration of data from multiple sites

- In a real application, one needs to extract data from multiple sources and put them into databases. This introduces three problems.
  - **Column match**: match columns in different data tables that contain the same type of information (e.g., product names)
  - **Value match**: match values that are semantically identical but represented differently in different Web sites. E.g.,
    - “Coke” and “Coca Cola”, “authors” and “writers”).
  - **Labeling**: Give a natural language label or attribute name to each column.
- **Very hard problems! This leads to our next topic.**

---

## 2. Information Integration

---

---

## Introduction

- At the end of last topic, we identified the problem of integrating extracted data:
    - column match and instance value match.
  - Unfortunately, limited research has been done in this specific context.
  - Much of the Web information integration research has been focused on the integration of **Web query interfaces**.
  - In this part, we introduce
    - some basic integration techniques, and
    - Web query interface integration
- 

---

## Road map

- **Basic integration techniques**
  - ➔ □ **Schema matching problem**
    - Different approaches
  - Web query interface integration
    - The problem
    - Some techniques
  - NLP connection
-

---

## Database integration (Rahm and Bernstein, 2001)

- Information integration started with database integration, which has been studied in the database community since the early 1980s.
- **Fundamental problem:** **schema matching**, which takes two (or more) database schemas to produce a mapping between **elements** (or **attributes**) of the two (or more) schemas that correspond semantically to each other.
- **Objective:** merge the schemas into a single global schema.

---

## Integrating two schemas

- Consider two schemas, S1 and S2, representing two customer relations, Cust and Customer.

S1	S2
<b>Cust</b>	<b>Customer</b>
CNo	CustID
CompName	Company
FirstName	Contact
LastName	Phone

## Integrating two schemas (contd)

- Represent the mapping with a similarity relation,  $\cong$ , over the power sets of  $S_1$  and  $S_2$ , where each pair in  $\cong$  represents one element of the mapping. E.g.,

Cust.CNo  $\cong$  Customer.CustID

Cust.CompName  $\cong$  Customer.Company

{Cust.FirstName, Cust.LastName}  $\cong$   
Customer.Contact

## Different types of matching

- **Schema-level only matching**: only schema information is considered.
- **Domain and instance-level only matching**: some instance data (data records) and possibly the domain of each attribute are used. This case is quite common on the Web.
- **Integrated matching of schema, domain and instance data**: Both schema and instance data (possibly domain information) are available.

---

## Road map

- **Basic integration techniques**
  - Schema matching problem
  - ➔ □ **Different approaches**
- **Web query interface integration**
  - The problem
  - Some techniques
- **NLP connection**

---

## Pre-processing for integration (He and Chang SIGMOG-03, Madhavan et al. VLDB-01, Wu et al. SIGMOD-04)

- **Tokenization**: break an item into atomic words using a dictionary, e.g.,
  - Break “fromCity” into “from” and “city”
  - Break “first-name” into “first” and “name”
- **Expansion**: expand abbreviations and acronyms to their full words, e.g.,
  - From “dept” to “departure”
- **Stopword removal and stemming**
- **Standardization of words**: Irregular words are standardized to a single form, e.g.,
  - From “colour” to “color”

## Schema-level matching (Rahm and Bernstein, 2001)

- Schema level matching relies on information such as name, description, data type, relationship type (e.g., part-of, is-a, etc), constraints, etc.
- **Match cardinality:**
  - **1:1 match:** one element in one schema matches one element of another schema.
  - **1:m match:** one element in one schema matches m elements of another schema.
  - **m:n match:** m elements in one schema matches n elements of another schema.

## An example

S <sub>1</sub>	S <sub>2</sub>
Cust	Customer
CustomID	CustID
Name	FirstName
Phone	LastName

We can find the following 1:1 and 1:m matches:

1:1	CustomID	CustID
1:m	Name	FirstName, LastName

- m:1 match is similar to 1:m match. m:n match is complex, and there is little work on it.

---

## Linguistic approaches (See (Liu, Web Data Mining book 2007) for many references)

- They are used to derive match candidates based on names, comments or descriptions of schema elements:
  - **Name match:**
    - Equality of names
    - Synonyms
    - Equality of hypernyms: A is a hypernym of B is B is a kind-of A.
    - Common sub-strings
    - Cosine similarity
    - User-provided name match: usually a domain dependent match dictionary
- 

---

## Linguistic approaches (contd)

- **Description match:** in many databases, there are comments to schema elements, e.g.,

```
 $S_1$ : CNo           // customer unique number  
 $S_2$ : CustID       // id number of a customer
```

- Cosine similarity from information retrieval (IR) can be used to compare comments after stemming and stopword removal.
-

---

## Constraint based approaches (See (Liu, Web Data Mining book 2007) for references)

- **Constraints** such as data types, value ranges, uniqueness, relationship types, etc.
- An **equivalent or compatibility table** for data types and keys can be provided. E.g.,
  - string  $\cong$  varchar, and (primary key)  $\cong$  unique
- For **structured schemas**, hierarchical relationships such as
  - is-a and part-ofmay be utilized to help matching.
- **Note:** On the Web, the constraint information is often not available, but some can be inferred based on the domain and instance data.

---

## Domain and instance-level matching

(See (Liu, Web Data Mining book 2007) for references)

- In many applications, some data instances or attribute domains may be available.
- Value characteristics are used in matching.
- Two different types of domains
  - **Simple domain:** each value in the domain has only a single component (the value cannot be decomposed).
  - **Composite domain:** each value in the domain contains more than one component.

---

## Match of simple domains

- A simple domain can be of any type.
- If the **data type** information is not available (this is often the case on the Web), the instance values can often be used to infer types, e.g.,
  - **Words** may be considered as **strings**
  - **Phone numbers** can have a **regular expression** pattern.
- **Data type patterns** (in regular expressions) can be learnt automatically or defined manually.
  - E.g., used to identify such types as integer, real, string, month, weekday, date, time, zip code, phone numbers, etc.

---

## Match of simple domains (contd)

- **Matching methods:**
  - Data types are used as constraints.
  - For numeric data, value ranges, averages, variances can be computed and utilized.
  - For categorical data: compare domain values.
  - **For textual data: cosine similarity.**
  - Schema element names as values: A set of values in a schema match a set of attribute names of another schema.  
E.g.,
    - In one schema, the attribute **color** has the domain {**yellow, red, blue**}, but in another schema, it has the element or attribute names called **yellow, red** and **blue** (values are yes and no).

## Handling composite domains

- A composite domain is usually indicated by its values containing delimiters, e.g.,
  - punctuation marks (e.g., “-”, “/”, “\_”)
  - White spaces
  - Etc.
- To detect a composite domain, these delimiters can be used. They are also used to split a composite value into simple values.
- Match methods for simple domains can then be applied.

## Combining similarities

- Similarities from many match indicators can be combined to find the most accurate candidates.
- Given the set of similarity values,  $sim_1(u, v)$ ,  $sim_2(u, v)$ , ...,  $sim_n(u, v)$ , from comparing two schema elements  $u$  (from  $S_1$ ) and  $v$  (from  $S_2$ ), many combination methods can be used:
  - Max:  $CSim(u, v) = \max\{sim_1(u, v), sim_2(u, v), \dots, sim_n(u, v)\}$
  - Weighted sum:  $CSim(u, v) = \lambda_1 * sim_1(u, v) + \lambda_2 sim_2(u, v) + \dots + \lambda_n * sim_n(u, v)$
  - Weighted average:  $CSim(u, v) = \frac{\lambda_1 Sim_1(u, v) + \lambda_2 Sim_2(u, v) + \dots + \lambda_n Sim_n(u, v)}{n}$
  - Machine learning: E.g., each similarity as a feature.
  - Others.

---

## 1:m match: two types

- **Part-of type**: each relevant schema element on the many side is a part of the element on the one side. E.g.,
  - “Street”, “city”, and “state” in a schema are parts of “address” in another schema.
- **Is-a type**: each relevant element on the many side is a specialization of the schema element on the one side. E.g.,
  - “Adults” and “Children” in one schema are specializations of “Passengers” in another schema.
- Special methods are needed to identify these types (Wu et al. SIGMOD-04).

---

## Some other issues (Rahm and Bernstein, 2001)

- **Reuse of previous match results**: when matching many schemas, earlier results may be used in later matching.
  - **Transitive property**: if X in schema S1 matches Y in S2, and Y also matches Z in S3, then we conclude X matches Z.
- **When matching a large number of schemas**, **statistical approaches** such as data mining can be used, rather than only doing pair-wise match.
- **Schema match results can be expressed in various ways**: Top N candidates, MaxDelta, Threshold, etc.
- **User interaction**: to pick and to correct matches.

---

## Road map

- Basic integration techniques
  - Schema matching problem
  - Different approaches
- Web query interface integration
  - □ **The problem**
  - Some techniques
- NLP connection

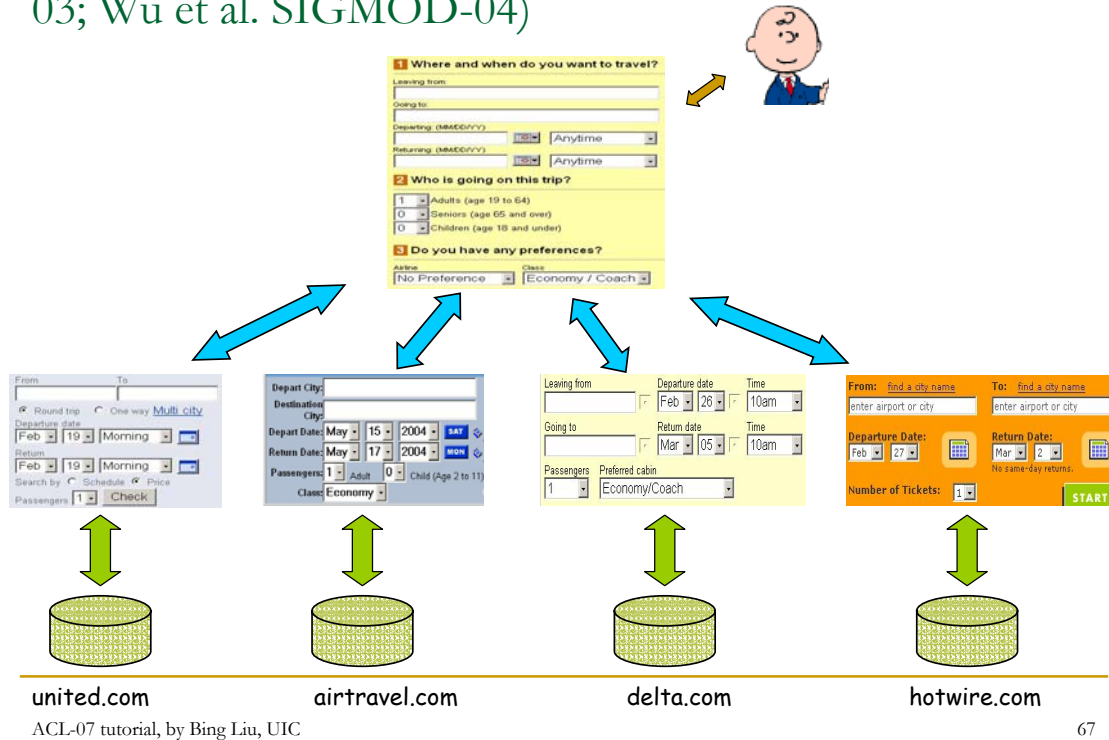
---

## Web information integration

(See (Liu, Web Data Mining book 2007) for references)

- Many integration tasks,
  - Integrating Web query interfaces (search forms)
  - Integrating ontologies (taxonomy)
  - Integrating extracted data
  - ...
- We only introduce **query interface integration** as it has been studied extensively.
  - Many web sites provide **forms** (called **query interfaces**) to query their underlying databases (often called the **deep web** as opposed to the **surface Web** that can be browsed).
  - Applications: meta-search and meta-query

# Global Query Interface (He and Chang, SIGMOD-03; Wu et al. SIGMOD-04)



ACL-07 tutorial, by Bing Liu, UIC

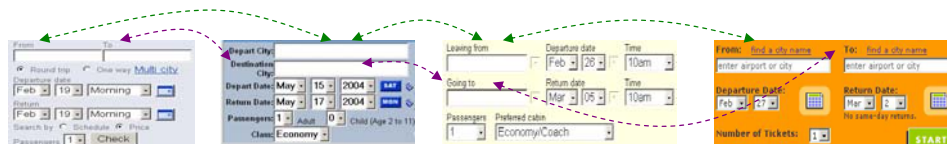
67

## Building global query interface (QI)

- A unified query interface:
  - **Conciseness** - Combine semantically similar fields over source interfaces
  - **Completeness** - Retain source-specific fields
  - **User-friendliness** – Highly related fields are close together
- Two-phrased integration
  - **Interface Matching** – Identify semantically similar fields

The screenshot shows the unified query interface with the following fields:
 

- Section 1: Leaving from, Going to, Departing (MMDD/YY), Returning (MMDD/YY).
- Section 2: 1 Adults (age 19 to 64), 0 Seniors (age 65 and over), 0 Children (age 18 and under).
- Section 3: Airline, Class (No Preference, Economy / Coach).



- **Interface Integration** – Merge the source query interfaces

ACL-07 tutorial, by Bing Liu, UIC

68

## Schema model of query interfaces

(He and Chang, SIGMOD-03)

- In each domain, there is a set of essential concepts  $C = \{c_1, c_2, \dots, c_n\}$ , used in query interfaces to enable the user to restrict the search.
- A query interface uses a subset of the concepts  $S \subseteq C$ . A concept  $i$  in  $S$  may be represented in the interface with a set of attributes (or fields)  $f_{i1}, f_{i2}, \dots, f_{ik}$ .
- Each concept is often represented with a single attribute.
  - Each attribute is labeled with a word or phrase, called the **label** of the attribute, which is visible to the user.
  - Each attribute may also have a set of possible values, its **domain**.

## Schema model of query interfaces (contd)

- All the attributes with their labels in a query interface are called the **schema** of the query interface.
- Each attribute also has a **name** in the HTML code. The name is attached to a TEXTBOX (which takes the user input). However,
  - this name is not visible to the user.
  - It is attached to the input value of the attribute and returned to the server as the attribute of the input value.
- For practical schema integration, we are not concerned with the set of concepts but only the **label** and **name** of each attribute and its domain.

# Interface matching $\approx$ schema matching

The image shows two screenshots of flight booking interfaces. The left interface (yellow background) has fields for 'Leaving from', 'Departure date' (Feb 26), 'Time' (10am), 'Going to', 'Return date' (Mar 05), 'Time' (10am), 'Passengers' (1), and 'Preferred cabin' (Economy/Coach). The right interface (orange background) has fields for 'From: find a city name' (enter airport or city), 'To: find a city name' (enter airport or city), 'Departure Date' (Feb 27), 'Return Date' (Mar 2), 'Number of Tickets' (1), and a 'START' button. It also includes a note 'No same-day returns.' and calendar icons for date selection.



Interface 1 ( $S_1$ )

Leaving from  
Going to  
Departure date  
Return date  
Passengers:  
Time  
Preferred cabin

Interface 2 ( $S_2$ )

From  
To  
Departure date  
Return date  
Number of tickets

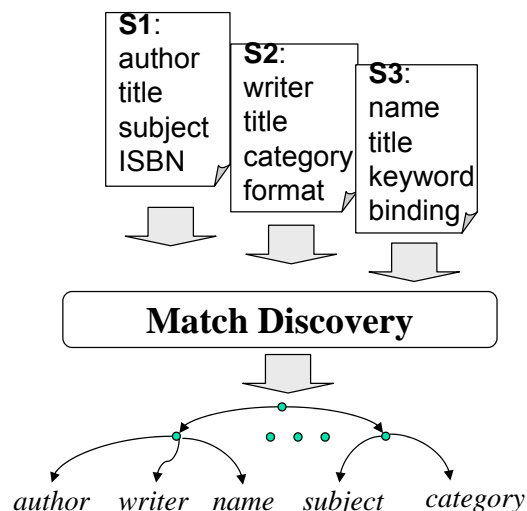
# Web is different from databases

(He and Chang, SIGMOD-03)

- **Limited use of acronyms and abbreviations on the Web:** but **natural language words and phrases**, for general public to understand.
  - Databases use acronyms and abbreviations extensively.
- **Limited vocabulary:** for easy understanding
- **A large number of similar databases:** a large number of sites offer the same services or selling the same products. Data mining is applicable!
- **Additional structures:** the information is usually organized in some meaningful way in the interface. E.g.,
  - Related attributes are together.
  - Hierarchical organization.

## The interface integration problem

- Identifying synonym attributes in an application domain.  
E.g. in the book domain: Author–Writer, Subject–Category



## Road map

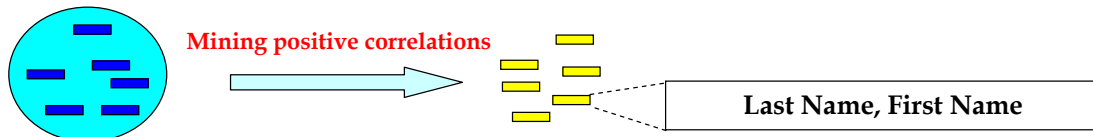
- Basic integration techniques
  - Schema matching problem
  - Different approaches
- Web query interface integration
  - The problem
  - **Some techniques**
- NLP connection

# Schema matching as correlation mining

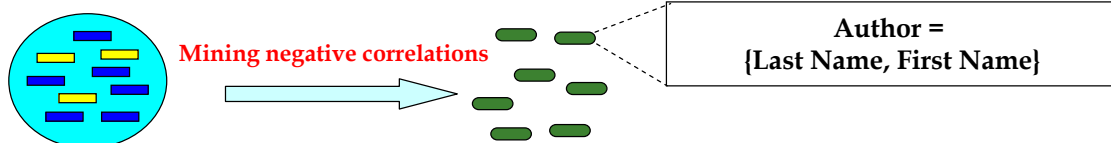
(He and Chang, KDD-04)

- It needs a large number of input query interfaces.
  - Synonym attributes are **negatively correlated**
    - They are semantically alternatives.
    - thus, *rarely co-occur* in query interfaces
  - Grouping attributes (they form a bigger concept together) are **positively correlation**
    - grouping attributes semantically complement
    - They *often co-occur* in query interfaces
- A data mining problem.

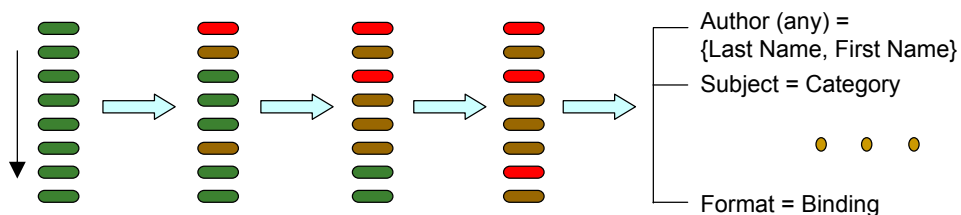
## 1. Positive correlation mining as potential groups



## 2. Negative correlation mining as potential matchings



## 3. Match selection as model construction



## Correlation measures

- It was found that many existing correlation measures were not suitable.

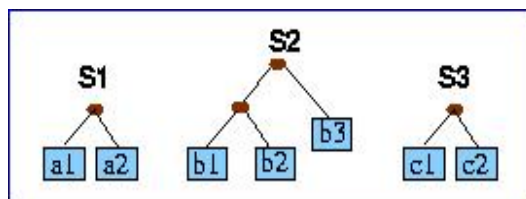
	$A_p$	$\neg A_p$	
$A_q$	$f_{11}$	$f_{10}$	$f_{1+}$
$\neg A_q$	$f_{01}$	$f_{00}$	$f_{0+}$
	$f_{+1}$	$f_{+0}$	$f_{++}$

- Negative correlation:  $corr_n(A_p, A_q) = H(A_p, A_q) = \frac{f_{01}f_{10}}{f_{+1}f_{1+}}$
- Positive correlation:  $corr_p(A_p, A_q) = \begin{cases} 1 - H(A_p, A_q) & \frac{f_{11}}{f_{++}} < \tau_d \\ 0 & \text{otherwise.} \end{cases}$

## A clustering approach (Wu et al., SIGMOD-04)

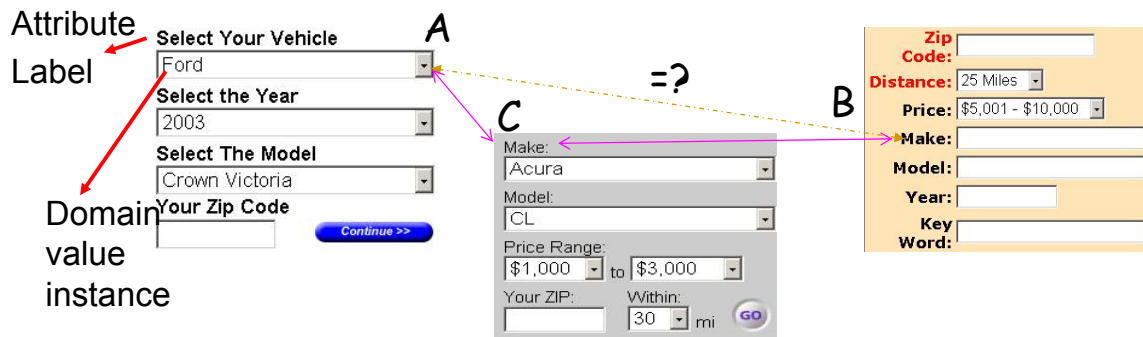
- 1:1 match using clustering.
- Clustering algorithm:** Agglomerative hierarchical clustering.
- Each cluster contains a set of candidate matches. E.g., final clusters:  $\{\{a1, b1, c1\}, \{b2, c2\}, \{a2\}, \{b3\}\}$

Interfaces:



- Similarity measures**
  - linguistic similarity
  - domain similarity

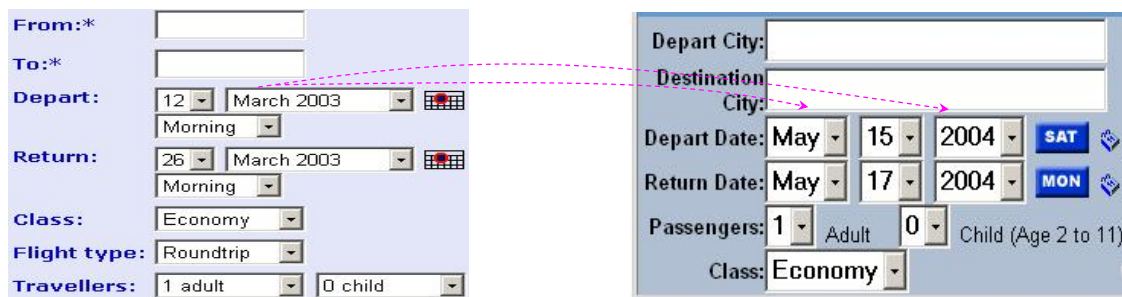
# Using the transitive property



## Observations:

- It is difficult to match “Select your vehicle” field, **A**, with “make” field, **B**
- But **A**’s instances are similar to **C**’s, and **C**’s label is similar to **B**’s
- Thus, **C** can serve as a “bridge” to connect **A** and **B**!

# Complex Mappings



**Part-of type** – contents of fields on the **many** side are part of the content of field on the **one** side

- **Commonalities** – (1) field proximity, (2) parent label similarity, and (3) value characteristics

## Complex Mappings (Cont'd)

The image shows two screenshots of a flight booking interface. The left screenshot is a yellow form with fields for 'Leaving from', 'Departure date', 'Time', 'Going to', 'Return date', 'Time', 'Passengers', and 'Preferred cabin'. The right screenshot is a blue form with fields for 'Depart City', 'Destination City', 'Depart Date', 'Return Date', 'Passengers', and 'Class'. Red dashed lines connect the 'Passengers' field in the yellow form to the 'Passengers' field in the blue form, and the 'Preferred cabin' field in the yellow form to the 'Class' field in the blue form.

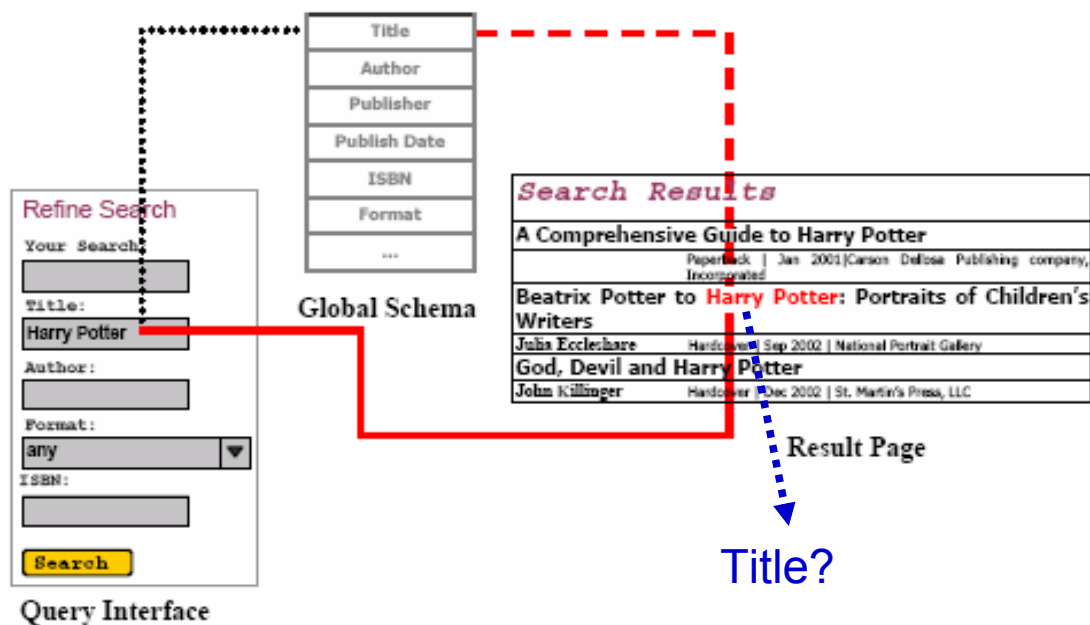
**Is-a type** – contents of fields on the **many** side are sum/union of the content of field on the **one** side.

- **Commonalities** – (1) field proximity, (2) parent label similarity, and (3) value characteristics

## Instance-based matching via query probing (Wang et al. VLDB-04)

- Both query interfaces and returned results (called instances) are considered in matching.
  - Assume a global schema (GS) is given and a set of instances are also given.
  - The method uses each instance value (IV) of every attribute in GS to probe the underlying database to obtain the count of IV appeared in the returned results.
  - These counts are used to help matching.
- It performs matches of
  - interface schema and global schema,
  - result schema and global schema, and
  - interface schema and results schema.

# Query Interface and Result Page

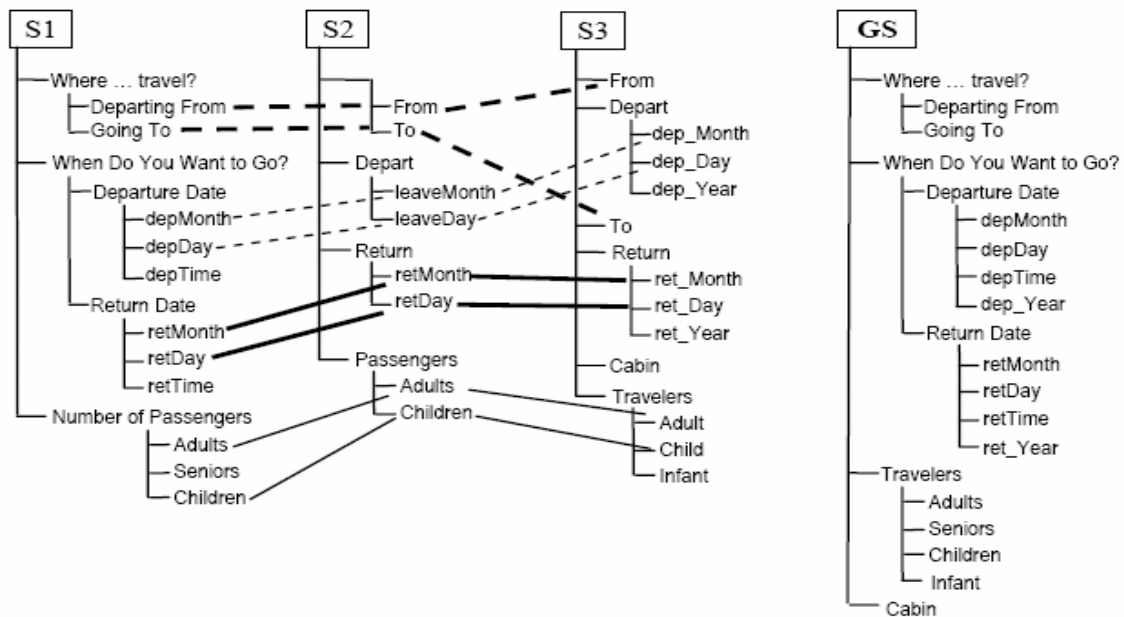


## Constructing a global query interface

(Dragut et al. VLDB-06)

- Once a set of query interfaces in the same domain is matched, we want to automatically construct a *well-designed global query interface*.
- Considerations:
  - **Structural appropriateness**: group attributes appropriately and produce a hierarchical structure.
  - **Lexical appropriateness**: choose the right label for each attribute or element.
  - **Instance appropriateness**: choose the right domain values.

## An example



## Road map

- Basic integration techniques
  - Schema matching problem
  - Different approaches
- Web query interface integration
  - The problem
  - Some techniques
- ➔ ■ **NLP connection**

---

## NLP connection

- **Everywhere!**
  - Current techniques are mainly based on heuristics related to **text (linguistic) similarity**, **structural information** and **patterns** discovered from a large number of interfaces.
  - **The focus on NLP is at the word and phrase level**, although there are also some sentences, e.g., “*where do you want to go?*”
  - Key: **identify synonyms and hypernyms relationships**.
- 

---

## Summary

- Information integration is an active research area.
  - Industrial activities are vibrant.
  - We only introduced some basic integration methods and Web query interface integration.
  - Another area of research is Web ontology matching See (Noy and Musen, AAAI-00; Agrawal and Srikant, WWW-01; Doan et al. WWW-02; Zhang and Lee, WWW-04).
  - Finally, database schema matching is a prominent research area in the database community as well. See (Doan and Halevy, AI Magazine 2005) for a short survey.
-

---

## 3. Information Synthesis

---

According to Dictionary.com,

**Synthesis:** The combining of separate elements or substances to form a coherent whole

---

### Web Search

- **Web search paradigm:**
    - Given a query, a few words
    - A search engine returns a ranked list of pages.
    - The user then browses and reads the pages to find what s/he wants.
  - **Sufficient** if one is looking for a specific piece of information, e.g., homepage of a person, a paper – *navigational search*
  - **Not sufficient** for open-ended research, learning or exploration, for which more can be done – *information search*
-

# Search results clustering

- The aim is to produce a taxonomy to provide navigational and browsing help by
  - organizing search results (**snippets**) into a small number of hierarchical clusters.
- Several researchers have worked on it.
  - E.g., Hearst & Pedersen, SIGIR-96; Zamir & Etzioni, WWW-1998; Vaithyanathan & Dom, ICML-1999; Leuski & Allan, RIAO-00; Zeng et al. SIGIR-04; Kummamuru et al. WWW-04.
- Some search engines already provide categorized results, e.g., vivisimo.com, northernlight.com
- Note: Ontology learning also uses clustering to build ontologies (e.g., Maedche and Staab, 2001).

## Vivisimo.com results for “web mining”

The screenshot shows a Microsoft Internet Explorer browser window displaying the Vivisimo search engine results for the query "Web mining". The browser's address bar shows the URL: <http://vivisimo.com/search?query=Web+mining&v%3Aproject=vivisimo-com&v%3Asources=Web>. The search results are clustered into categories on the left, including Web mining (240), Data Mining (68), Industry (30), Resources (23), Gold (18), Marketing (15), Semantic Web (8), Mining Association (9), Design (9), Mining Equipment (9), and Mining Engineering (9). The main content area displays the top 240 results of at least 4,545,000 retrieved for the query "Web mining". The results include sponsored links for "Web Mining Software" (QL2) and "Data Mining" (ScrapeGoat), and organic results for "KDNuggets Directory", "National Mining Association", and "Megaputer Intelligence". The browser's taskbar at the bottom shows several open applications, including Eudora, knowledge-synt..., Vivisimo - Cluster..., DBLP: Alexander..., and ontology.

---

## Issues with clustering

- Search results clustering is well known and is in commercial systems.
    - Clusters provide browsing help so that the user can focus on what he/she really wants.
    - **But rarely used** (- inaccurate and/or not needed?)
  - **To some extent, clustering is problematic.**
    - A Web page may not be homogeneous, but contain multiple topics, which make:
      - accurate clustering very difficult, and
      - page-level clustering problematic.
- 

---

## Informational search

- If we search for information about a topic, ideally we want the returned results to be
    - **Authoritative**
    - **Complete (reasonably)**
    - **Unbiased**
  - **But few pages on the Web provide**
    - complete and unbiased information about a topic, e.g., “data mining”,
    - because a page owner often describes what he/she does.
  - A typical Web page is “**opinionated/subjective**”, “**biased**”, and “**incomplete**”.
-

---

## Information synthesis

- **Information Synthesis:** Can a system provide “unbiased” and “complete” information on a search topic? E.g.,
  - **Topic hierarchy:** Find and combine related bits and pieces from a large number of pages to produce a topic hierarchy.
  - **Pointers to content pages/page segments:** Each topic or sub-topic points to pages/page segments that contain related information.

---

## Information synthesis: a case study

(Liu, Chee and Ng, WWW-03)

- Traditionally, when one wants to learn about a topic,
  - one reads a book or a survey paper.
  - **With the Web, the habit is changing!**
- Learning in-depth knowledge of a topic from the Web is becoming increasingly popular.
  - Web’s convenience, richness of information, diversity, etc
  - For emerging topics, it may be essential - **no book.**
- **Can we mine “a book” from the Web on a topic?**
  - **Knowledge in a book is well organized:** the authors have painstakingly synthesize and organize the knowledge about the topic and present it in a coherent manner.

## An example

- Given the topic “data mining”, can the system produce the following, a concept hierarchy?
  - **Classification**
    - **Decision trees**
      - ... (Web pages containing the descriptions of the topic)
    - **Naïve bayes**
      - ...
    - ...
  - **Clustering**
    - **Hierarchical**
    - **Partitioning**
    - **K-means**
    - ....
  - **Association rules**
  - **Sequential patterns**
  - ...

## The Approach: Exploiting information redundancy

- **Web information redundancy**: many Web pages contain similar information.
  - to produce unbiased and complete information
- **Observation 1**: If some **phrases** are mentioned in a number of pages, they are likely to be important concepts or sub-topics of the given topic.
- This means that we can use data mining to find concepts and sub-topics:
  - What are candidate words or phrases that may represent concepts of sub-topics?

---

## Each Web page is already organized

- **Observation 2:** The content of almost every Web page is already organized.
    - Different levels of headings
    - Emphasized words and phrases
  - They are indicated by various HTML emphasizing tags, e.g., <H1>, <H2>, <H3>, <B>, <I>, etc.
  - Can we utilize existing page organizations to find a globally “unbiased” and “complete” organization of the topic?
    - Cannot rely on only one page: incomplete, and biased.
- 

---

## Linguistic patterns: find sub-topics

- Certain syntactic language patterns express some relationship of concepts.
  - The following patterns represent hierarchical relationships, concepts and sub-concepts:
    - Such as
    - For example (e.g.,)
    - Including
    - E.g., “There are many *clustering techniques* (e.g., *hierarchical, partitioning, k-means, k-medoids*).”
-

## Data Mining

Clustering  
Classification  
Data Warehouses  
Databases  
Knowledge Discovery  
Web Mining  
Information Discovery  
Association Rules  
Machine Learning  
Sequential Patterns

## Web Mining

Web Usage Mining  
Web Content Mining  
Data Mining  
Webminers  
Text Mining  
Personalization  
Information Extraction  
Semantic Web Mining  
XML  
Mining Web Data

## Some concepts extraction results

### Classification

Neural networks  
Trees  
Naive bayes  
Decision trees  
K nearest neighbor  
Regression  
Neural net  
Sliq algorithm  
Parallel algorithms  
Classification rule learning  
ID3 algorithm  
C4.5 algorithm  
Probabilistic models

### Clustering

Hierarchical  
K means  
Density based  
Partitioning  
K medoids  
Distance based methods  
Mixture models  
Graphical techniques  
Intelligent miner  
Agglomerative  
Graph based algorithms

## More on syntactic language patterns

- As we discussed earlier, syntactic language patterns do convey some semantic relationships.
- Earlier work by Hearst (Hearst, SIGIR-92) used patterns to find concepts/sub-concepts relations.
- WWW-04 has two papers on this issue (Cimiano, Handschuh and Staab 2004) and (Etzioni et al 2004).
  - apply lexicon-syntactic patterns such as those discussed 5 slides ago and more
  - Use a search engine to find concepts and sub-concepts (class/instance) relationships.

---

## PANKOW (Cimiano, Handschuh and Staab WWW-04)

- The linguistic patterns used are (the first 4 are from (Hearst SIGIR-92)):
  - 1: <concept>s such as <instance>
  - 2: such <concepts>s as <instance>
  - 3: <concepts>s, (especially|including)<instance>
  - 4: <instance> (and|or) other <concept>s
  - 5: the <instance> <concept>
  - 6: the <concept> <instance>
  - 7: <instance>, a <concept>
  - 8: <instance> is a <concept>

---

## The steps

- PANKOW categorizes instances into given concept classes, e.g., is “Japan” a “country” or a “hotel”?
- Given a proper noun (instance), it is introduced together with given ontology concepts into the linguistic patterns to form hypothesis phrases, e.g.,
  - Proper noun: Japan
  - Given concepts: country, hotel.
  - ⇒ “Japan is a country”, “Japan is a hotel” ....
- All the hypothesis phrases are sent to Google.
- Counts from Google are collected

---

## Categorization step

- The system sums up the counts for each instance and concept pair (*i*:instance, *c*:concept, *p*:pattern).

$$count(i, c) = \sum_{p \in P} count(i, c, p)$$

- The candidate proper noun (instance) is given to the highest ranked concept(s):

$$R = \{(i, c_i) \mid i \in I, c_i = \arg \max_{c \in C} count(i, c)\}$$

- *I*: instances, *C*: concepts
  - **Result:** Categorization was reasonably accurate, but concept or sub-concept extraction was not.
- 

---

## KnowItAll (Etzioni et al WWW-04 and AAAI-04)

- Basically use the same approach of linguistic patterns and Web search to find concept/sub-concept (also called class/instance) relationships.
  - KnowItAll has more sophisticated mechanisms to assess the probability of every extraction, using Naïve Bayesian classifiers.
  - It thus does better in class/instance extraction.
-

## Syntactic patterns used in KnowItAll

NP1 {“, ”} “such as” NPList2

NP1 {“, ”} “and other” NP2

NP1 {“, ”} “including” NPList2

NP1 {“, ”} “is a” NP2

NP1 {“, ”} “is the” NP2 “of” NP3

“the” NP1 “of” NP2 “is” NP3

...

## Main Modules of KnowItAll

- **Extractor**: generate a set of extraction rules for each class and relation from the language patterns. E.g.,
  - “NP1 such as NPList2” indicates that each NP in NPList1 is a instance of class NP1. “He visited cities such as Tokyo, Paris, and Chicago”.
  - KnowItAll will extract three instances of class CITY.
- **Search engine interface**: a search query is automatically formed for each extraction rule. E.g., “cities such as”. KnowItAll will
  - search with a number search engines
  - Download the returned pages
  - Apply extraction rule to appropriate sentences.
- **Assessor**: Each extracted candidate is assessed to check its likelihood for being correct. Here it uses Point-Mutual Information and a Bayesian classifier.

---

## NLP connection

- Information synthesis is becoming important as we move up the information food chain.
  - **The questions is:** Can a system provide unbiased and complete information about a search topic rather than only bits and pieces?
  - **Key:** Exploiting information redundancy
  - A lot of NLP techniques needed.
  - Also very useful to perform information synthesis with a set of normal text documents?
- 

---

## 4. Opinion Mining and Summarization

---

---

## Introduction – facts and opinions

- Two main types of textual information on the Web.
    - **Facts and Opinions**
  - Current search engines search for facts (assume they are true)
    - Facts can be expressed with topic keywords.
  - Search engines do not search for opinions
    - Opinions are hard to express with a few keywords
      - How do people think of Motorola Cell phones?
    - Current search ranking strategy is not appropriate for opinion retrieval/search.
- 

---

## Introduction – user generated content

- **Word-of-mouth on the Web**
    - One can express personal experiences and opinions on almost anything, at review sites, forums, discussion groups, blogs ... (called the user generated content.)
    - They contain valuable information
    - **Web/global scale:** No longer – one's circle of friends
  - **Our interest:** to mine opinions expressed in the user-generated content
    - An intellectually very challenging problem.
    - Practically very useful.
-

---

## Introduction – Applications

- **Businesses and organizations:** product and service benchmarking. Market intelligence.
    - Business spends a huge amount of money to find consumer sentiments and opinions.
      - Consultants, surveys and focused groups, etc
  - **Individuals:** interested in other's opinions when
    - Purchasing a product or using a service,
    - Finding opinions on political topics,
  - **Ads placements:** Placing ads in the user-generated content
    - Place an ad when one praises a product.
    - Place an ad from a competitor if one criticizes a product.
  - **Opinion retrieval/search:** providing general search for opinions.
- 

---

## Two types of evaluation

- **Direct Opinions:** sentiment expressions on some objects, e.g., products, events, topics, persons.
    - E.g., “the picture quality of this camera is great”
    - Subjective
  - **Comparisons:** relations expressing similarities or differences of more than one object. Usually expressing an ordering.
    - E.g., “car x is cheaper than car y.”
    - Objective or subjective.
-

---

## Opinion search (Liu, Web Data Mining book, 2007)

- Can you search for opinions as conveniently as general Web search?
- Whenever you need to make a decision, you may want some opinions from others,
  - Wouldn't it be nice? you can find them on a search system instantly, by issuing queries such as
    - Opinions: "Motorola cell phones"
    - Comparisons: "Motorola vs. Nokia"
- Cannot be done yet! (but could be soon ...)

---

## Typical opinion search queries

- Find the opinion of a person or organization (opinion holder) on a particular object or a feature of the object.
  - E.g., what is Bill Clinton's opinion on abortion?
- Find positive and/or negative opinions on a particular object (or some features of the object), e.g.,
  - customer opinions on a digital camera.
  - public opinions on a political topic.
- Find how opinions on an object change over time.
- How object A compares with Object B?
  - Gmail vs. Hotmail

---

## Find the opinion of a person on X

- In some cases, the general search engine can handle it, i.e., using suitable keywords.
    - Bill Clinton's opinion on abortion
  - Reason:
    - One person or organization usually has only one opinion on a particular topic.
    - The opinion is likely contained in a single document.
    - Thus, a good keyword query may be sufficient.
- 

---

## Find opinions on an object

### We use product reviews as an example:

- Searching for opinions in product reviews is different from general Web search.
    - E.g., search for opinions on “Motorola RAZR V3”
  - General Web search (for a fact): rank pages according to some authority and relevance scores.
    - The user views the first page (if the search is perfect).
    - **One fact = Multiple facts**
  - Opinion search: rank is desirable, however
    - reading only the review ranked at the top is not appropriate because it is only the opinion of one person.
    - **One opinion ≠ Multiple opinions**
-

---

## Search opinions (contd)

- **Ranking:**
    - produce two rankings
      - Positive opinions and negative opinions
      - Some kind of summary of both, e.g., # of each
    - Or, one ranking but
      - The top (say 30) reviews should reflect the natural distribution of all reviews (assume that there is no spam), i.e., with the right balance of positive and negative reviews.
  - **Questions:**
    - Should the user reads all the top reviews? OR
    - Should the system prepare a summary of the reviews?
- 

---

## Reviews are similar to surveys

- **Reviews can be regarded as traditional surveys.**
    - In traditional survey, returned survey forms are treated as raw data.
    - Analysis is performed to summarize the survey results.
      - E.g., % against or for a particular issue, etc.
  - In opinion search,
    - Can a summary be produced?
    - What should the summary be?
-

---

# Roadmap

- ➔ ■ **Opinion mining – the abstraction**
  - Document level sentiment classification
  - Sentence level sentiment analysis
  - Feature-based opinion mining and summarization
  - Comparative sentence and relation extraction
  - Summary

---

# Opinion mining – the **abstraction**

(Hu and Liu, KDD-04; Liu, Web Data Mining book 2007)

- **Basic components of an opinion**
  - **Opinion holder**: The person or organization that holds a specific opinion on a particular object.
  - **Object**: on which an opinion is expressed
  - **Opinion**: a view, attitude, or appraisal on an object from an opinion holder.
- **Objectives of opinion mining**: many ...
- **Let us abstract the problem**
  - **put existing research into a common framework**
- We use **consumer reviews of products** to develop the ideas. Other opinionated contexts are similar.

## Object/entity

- **Definition (object):** An **object**  $O$  is an entity which can be a product, person, event, organization, or topic.  $O$  is represented as
  - a hierarchy of **components**, **sub-components**, and so on.
  - Each node represents a component and is associated with a set of **attributes** of the component.
  - $O$  is the root node (which also has a set of attributes)
- **An opinion can be expressed on any node or attribute of the node.**
- To simplify our discussion, we use “**features**” to represent both components and attributes.
  - The term “feature” should be understood in a **broad sense**,
    - Product feature, topic or sub-topic, event or sub-event, etc
- **Note:** the object  $O$  itself is also a feature.

## Model of a review

- An object  $O$  is represented with a finite set of features,  $F = \{f_1, f_2, \dots, f_n\}$ .
  - Each feature  $f_i$  in  $F$  can be expressed with a finite set of words or phrases  $W_i$ , which are **synonyms**.

**That is to say:** we have a set of corresponding synonym sets  $W = \{W_1, W_2, \dots, W_n\}$  for the features.
- **Model of a review:** An **opinion holder**  $j$  comments on a subset of the **features**  $S_j \subseteq F$  of object  $O$ .
  - For each feature  $f_k \in S_j$  that  $j$  comments on, he/she
    - chooses a word or phrase from  $W_k$  to describe the feature, and
    - expresses a positive, negative or neutral **opinion** on  $f_k$ .

---

## Opinion mining tasks

- At the document (or review) level:
    - Task:** sentiment classification of reviews
      - **Classes:** positive, negative, and neutral
      - **Assumption:** each document (or review) focuses on a single object (not true in many discussion posts) and contains opinion from a single opinion holder.
  - At the sentence level:
    - Task 1:** identifying subjective/opinionated sentences
      - **Classes:** objective and subjective (opinionated)
    - Task 2:** sentiment classification of sentences
      - **Classes:** positive, negative and neutral.
      - **Assumption:** a sentence contains only one opinion
        - not true in many cases.
      - Then we can also consider clauses or phrases.
- 

---

## Opinion mining tasks (contd)

- At the feature level:
    - Task 1:** Identify and extract object features that have been commented on by an opinion holder (e.g., a reviewer).
    - Task 2:** Determine whether the opinions on the features are positive, negative or neutral.
    - Task 3:** Group feature synonyms.
      - Produce a feature-based opinion summary of multiple reviews (**more on this later**).
  - **Opinion holders:** identify holders is also useful, e.g., in news articles, etc, but they are usually known in the user generated content, i.e., authors of the posts.
-

---

## More at the feature level

- **Problem 1:** Both  $F$  and  $W$  are unknown.
  - We need to perform all three tasks:
- **Problem 2:**  $F$  is known but  $W$  is unknown.
  - All three tasks are still needed. Task 3 is easier. It becomes the problem of matching the discovered features with the set of given features  $F$ .
- **Problem 3:**  $W$  is known ( $F$  is known too).
  - Only task 2 is needed.

**F:** the set of features

**W:** synonyms of each feature

---

---

## Roadmap

- Opinion mining – the abstraction
- ➔ ■ **Document level sentiment classification**
- Sentence level sentiment analysis
- Feature-based opinion mining and summarization
- Comparative sentence and relation extraction
- Summary

---

## Sentiment classification

- Classify documents (e.g., reviews) based on the overall sentiments expressed by opinion holders (authors),
    - Positive, negative, and (possibly) neutral
    - Since in our model **an object  $O$  itself is also a feature**, then **sentiment classification** essentially determines the opinion expressed on  $O$  in each document (e.g., review).
  - Similar but different from topic-based text classification.
    - In topic-based text classification, topic words are important.
    - In sentiment classification, sentiment words are more important, e.g., great, excellent, horrible, bad, worst, etc.
- 

---

## Unsupervised review classification

(Turney, ACL-02)

- Data: reviews from epinions.com on automobiles, banks, movies, and travel destinations.
  - The approach: Three steps
  - Step 1:
    - Part-of-speech tagging
    - Extracting two consecutive words (**two-word phrases**) from reviews if their tags conform to some given patterns, e.g., (1) JJ, (2) NN.
-

---

- Step 2: Estimate the semantic orientation (SO) of the extracted phrases

- Use Pointwise mutual information

$$PMI(word_1, word_2) = \log_2 \left( \frac{P(word_1 \wedge word_2)}{P(word_1)P(word_2)} \right)$$

- Semantic orientation (SO):

$$SO(\text{phrase}) = PMI(\text{phrase}, \text{"excellent"}) \\ - PMI(\text{phrase}, \text{"poor"})$$

- Using AltaVista near operator to do search to find the number of hits to compute PMI and SO.
- 

---

- Step 3: Compute the average SO of all phrases

- classify the review as **recommended** if average SO is positive, **not recommended** otherwise.

- Final classification accuracy:

- automobiles - 84%
  - banks - 80%
  - movies - 65.83
  - travel destinations - 70.53%
-

## Sentiment classification using machine learning methods (Pang et al, EMNLP-02)

- This paper directly applied several machine learning techniques to classify movie reviews into positive and negative.
- Three classification techniques were tried:
  - Naïve Bayes
  - Maximum entropy
  - Support vector machine
- Pre-processing settings: negation tag, unigram (single words), bigram, POS tag, position.
- SVM: the best accuracy 83% (unigram)

## Review classification by scoring features

(Dave, Lawrence and Pennock, WWW-03)

- It first selects a set of features  $F = f_1, f_2, \dots$ 
  - Note: machine learning features, but product features.

- Score the features
  - C and C' are classes

$$\text{score}(f_i) = \frac{P(f_i | C) - P(f_i | C')}{P(f_i | C) + P(f_i | C')}$$

- Classification of a review  $d_j$  (using sign):

$$\text{class}(d_j) = \begin{cases} C & \text{eval}(d_j) > 0 \\ C' & \text{eval}(d_j) < 0 \end{cases}$$

$$\text{eval}(d_j) = \sum_i \text{score}(f_i)$$

- Accuracy of 84-88%.

---

## Other related works

- Using PMI, syntactic relations and other attributes with SVM (Mullen and Collier, EMNLP-04).
  - Sentiment classification considering rating scales (Pang and Lee, ACL-05).
  - Comparing supervised and unsupervised methods (Chaovalit and Zhou, HICSS-05)
  - Using semi-supervised learning (Goldberg and Zhu, Workshop on TextGraphs, at HLT-NAAL-06).
  - Review identification and sentiment classification of reviews (Ng, Dasgupta and Arifin, ACL-06).
  - Sentiment classification on customer feedback data (Gamon, Coling-04).
  - Comparative experiments (Cui et al. AAAI-06)
- 

---

## Roadmap

- Opinion mining – the abstraction
  - Document level sentiment classification
  - ➔ ■ **Sentence level sentiment analysis**
  - Feature-based opinion mining and summarization
  - Comparative sentence and relation extraction
  - Summary
-

---

## Sentence-level sentiment analysis

- Document-level sentiment classification is too coarse for most applications.
  - Let us move to the sentence level.
  - Much of the work on sentence level sentiment analysis focuses on identifying subjective sentences in news articles.
    - Classification: objective and subjective.
    - All techniques use some forms of machine learning.
    - E.g., using a naïve Bayesian classifier with a set of data features/attributes extracted from training sentences (Wiebe et al. ACL-99).
- 

---

## Using learnt patterns (Riloff and Wiebe, EMNLP-03)

- A bootstrapping approach.
    - A high precision classifier is first used to automatically identify some subjective and objective sentences.
      - Two high precision (but low recall) classifiers are used,
        - a high precision subjective classifier
        - A high precision objective classifier
        - Based on manually collected lexical items, single words and n-grams, which are good subjective clues.
    - A set of patterns are then learned from these identified subjective and objective sentences.
      - Syntactic templates are provided to restrict the kinds of patterns to be discovered, e.g., <subj> passive-verb.
    - The learned patterns are then used to extract more subject and objective sentences (the process can be repeated).
-

## Subjectivity and polarity (orientation)

(Yu and Hazivassiloglou, EMNLP-03)

- For subjective or opinion sentence identification, three methods are tried:
  - Sentence similarity.
  - Naïve Bayesian classification.
  - Multiple naïve Bayesian (NB) classifiers.
- For opinion orientation (positive, negative or neutral) (also called polarity) classification, it uses a similar method to (Turney, ACL-02), but
  - with more seed words (rather than two) and based on log-likelihood ratio (LLR).
  - For classification of each word, it takes the average of LLR scores of words in the sentence and use cutoffs to decide positive, negative or neutral.

## Other related work

- Consider gradable adjectives (Hatzivassiloglou and Wiebe, Coling-00)
- Semi-supervised learning with the initial training set identified by some strong patterns and then applying NB or self-training (Wiebe and Riloff, CILing-05).
- Finding strength of opinions at the clause level (Wilson et al. AAAI-04).
- Sum up orientations of opinion words in a sentence (or within some word window) (Kim and Hovy, COLING-04).
- Find clause or phrase polarities based on priori opinion words and classification (Wilson et al. EMNLP-05)
- Semi-supervised learning to classify sentences in reviews (Gamon et al. IDA-05).
- Sentiment sentence retrieval (Eguchi and Lavrendo, EMNLP-06)

---

## Let us go further?

- Sentiment classification at both document and sentence (or clause) levels are useful, **but**
    - They do not find what the opinion holder like and dislike.
  - An negative sentiment on an object
    - does not mean that the opinion holder dislikes everything about the object.
  - A positive sentiment on an object
    - does not mean that the opinion holder likes everything about the object.
  - **We need to go to the feature level.**
- 

---

## But before we go further

- Let us discuss **Opinion Words or Phrases** (also called polar words, opinion bearing words, etc). E.g.,
    - **Positive**: beautiful, wonderful, good, amazing,
    - **Negative**: bad, poor, terrible, cost someone an arm and a leg (idiom).
  - They are instrumental for opinion mining (obviously)
  - Three main ways to compile such a list:
    - **Manual approach**: not a bad idea, only an one-time effort
    - **Corpus-based approaches**
    - **Dictionary-based approaches**
  - **Important to note:**
    - **Some opinion words are context independent (e.g., good).**
    - **Some are context dependent (e.g., long).**
-

## Corpus-based approaches

- **Rely on syntactic or co-occurrence patterns in large corpora.** (Hazivassiloglou and McKeown, ACL-97; Turney, ACL-02; Yu and Hazivassiloglou, EMNLP-03; Kanayama and Nasukawa, EMNLP-06; Ding and Liu 2007)
  - Can find domain (not context!) dependent orientations (positive, negative, or neutral).
- (Turney, ACL-02) and (Yu and Hazivassiloglou, EMNLP-03) are similar.
  - Assign opinion orientations (polarities) to words/phrases.
  - (Yu and Hazivassiloglou, EMNLP-03) is different from (Turney, ACL-02)
    - use more seed words (rather than two) and use log-likelihood ratio (rather than PMI).

## Corpus-based approaches (contd)

- **Use constraints (or conventions) on connectives** to identify opinion words (Hazivassiloglou and McKeown, ACL-97; Kanayama and Nasukawa, EMNLP-06; Ding and Liu, 2007). E.g.,
- **Conjunction:** conjoined adjectives usually have the same orientation (Hazivassiloglou and McKeown, ACL-97).
  - E.g., “This car is *beautiful and spacious*.” (conjunction)
  - AND, OR, BUT, EITHER-OR, and NEITHER-NOR have similar constraints.
  - **Learning using**
    - **log-linear model:** determine if two conjoined adjectives are of the same or different orientations.
    - **Clustering:** produce two sets of words: positive and negative
  - **Corpus:** 21 million word 1987 Wall Street Journal corpus.

## Corpus-based approaches (contd)

- (Kanayama and Nasukawa, EMNLP-06) takes a similar approach to (Hazivassiloglou and McKeown, ACL-97) but for Japanese words:
  - Instead of using learning, it uses two criteria to determine whether to add a word to positive or negative lexicon.
  - Have an initial seed lexicon of positive and negative words.
- (Ding and Liu, 2007) also exploits constraints on connectives, but with two differences
  - It uses them to assign opinion orientations to product features (more on this later).
    - One word may indicate different opinions in the same domain.
      - “The battery life is *long*” (+) and “It takes a *long* time to focus” (-).
    - **Find domain opinion words is insufficient.**
  - It can be used without a large corpus.

## Dictionary-based approaches

- **Typically use WordNet’s synsets and hierarchies to acquire opinion words**
  - Start with a small seed set of opinion words.
  - Use the set to search for synonyms and antonyms in WordNet (Hu and Liu, KDD-04; Kim and Hovy, COLING-04).
  - Manual inspection may be used afterward.
- Use additional information (e.g., glosses) from WordNet (Andreevskaia and Bergler, EACL-06) and learning (Esuti and Sebastiani, CIKM-05).
- **Weakness of the approach:** Do not find context dependent opinion words, e.g., small, long, fast.

---

# Roadmap

- Opinion mining – the abstraction
- Document level sentiment classification
- Sentence level sentiment analysis
- ➔ ■ **Feature-based opinion mining and summarization**
- Comparative sentence and relation extraction
- Summary

---

# Feature-based opinion mining and summarization

(Hu and Liu, KDD-04)

- **Again focus on reviews (easier to work in a concrete domain!)**
- **Objective:** find what reviewers (opinion holders) liked and disliked
  - Product features and opinions on the features
- Since the number of reviews on an object can be large, an **opinion summary** should be produced.
  - Desirable to be a **structured summary**.
  - Easy to visualize and to compare.
  - **Analogous to but different from multi-document summarization.**

---

## The tasks

- Recall the three tasks in our model.
  - Task 1*: Extract object features that have been commented on in each review.
  - Task 2*: Determine whether the opinions on the features are positive, negative or neutral.
  - Task 3*: Group feature synonyms.
    - Produce a summary
- Task 2 may not be needed depending on the format of reviews.

---

## Different review format

**Format 1 - Pros, Cons and detailed review**: The reviewer is asked to describe Pros and Cons separately and also write a detailed review.  
[Epinions.com](#) uses this format.

**Format 2 - Pros and Cons**: The reviewer is asked to describe Pros and Cons separately.  
[Cnet.com](#) used to use this format.

**Format 3 - free format**: The reviewer can write freely, i.e., no separation of Pros and Cons.  
[Amazon.com](#) uses this format.

---

## Format 1

### My SLR is on the shelf

by [camerafun4](#), Aug 09 '04

**Pros:** Great photos, easy to use, very small  
**Cons:** Battery usage; included memory is stingy.

I had never used a digital camera prior to purchasing  
have always used a SLR ... [Read the full review](#)

## Format 3

GREAT Camera., Jun 3, 2004

Reviewer: [jprice174](#) from Atlanta, Ga.

I did a lot of research last year before I bought this camera... It kinda hurt to leave behind my beloved nikon 35mm SLR, but I was going to Italy, and I needed something smaller, and digital.

The **pictures** coming out of this camera are amazing. The 'auto' feature takes great pictures most of the time. And with digital, you're not wasting film if the picture doesn't come out.

## Format 2

User  
rating  
Perfect  
**10**

out of 10

"It is a great digitbal still camera for this century"

September 1, 2004

### Pros:

It's small in size, and the rotatable lens is great. It's very easy to use, and has fast response from the shutter. The LCD has increased from 1.5 in to 1.8, which gives bigger view. It has lots of modes to choose from in order to take better pictures.

### Cons:

It almost has no cons, it would be better if the LCD is bigger and it's going to be best if the model is designed to a smaller size.

## Feature-based opinion summary (Hu and Liu, KDD-04)

GREAT Camera., Jun 3, 2004

Reviewer: [jprice174](#) from Atlanta, Ga.

I did a lot of research last year before I bought this camera... It kinda hurt to leave behind my beloved nikon 35mm SLR, but I was going to Italy, and I needed something smaller, and digital.

The **pictures** coming out of this camera are amazing. The 'auto' feature takes great pictures most of the time. And with digital, you're not wasting film if the picture doesn't come out. ...

### Feature Based Summary:

#### Feature1: **picture**

Positive: 12

- The **pictures** coming out of this camera are amazing.
- Overall this is a good camera with a really good **picture** clarity.

...

Negative: 2

- The **pictures** come out hazy if your hands shake even for a moment during the entire process of taking a picture.
- Focusing on a display rack about 20 feet away in a brightly lit room during day time, **pictures** produced by this camera were blurry and in a shade of orange.

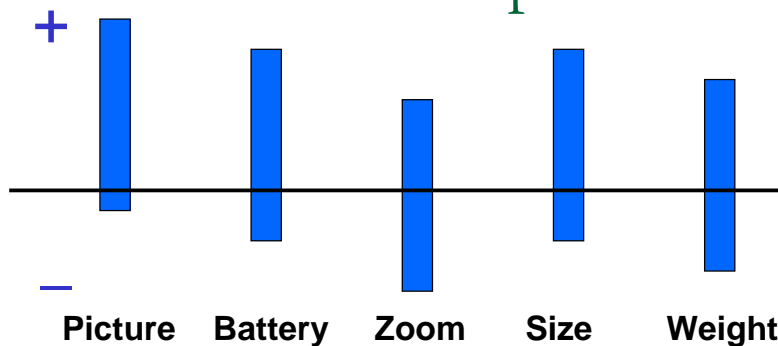
#### Feature2: **battery life**

...

## Visual summarization & comparison

- Summary of reviews of

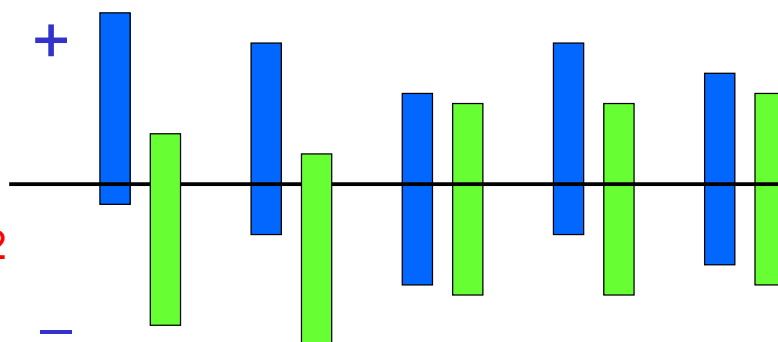
**Digital camera 1**



- Comparison of reviews of

**Digital camera 1**

**Digital camera 2**



## Feature extraction from Pros and Cons of Format 1 (Liu et al WWW-03; Hu and Liu, AAAI-CAAW-05)

- Observation:** Each sentence segment in Pros or Cons contains only one feature. Sentence segments can be separated by commas, periods, semi-colons, hyphens, '&'s, 'and's, 'but's, etc.
- Pros in Example 1 can be separated into 3 segments:**

great photos	<photo>
easy to use	<use>
very small	<small> ⇒ <size>
- Cons can be separated into 2 segments:**

battery usage	<battery>
included memory is stingy	<memory>

## Extraction using label sequential rules

- Label sequential rules (LSR) are a special kind of sequential patterns, discovered from sequences.
- LSR Mining is supervised (Liu's Web mining book 2006).
- The training data set is a set of sequences, e.g.,

*"Included memory is stingy"*

is turned into a sequence with POS tags.

$\langle \{included, VB\} \{memory, NN\} \{is, VB\} \{stingy, JJ\} \rangle$

then turned into

$\langle \{included, VB\} \{\$feature, NN\} \{is, VB\} \{stingy, JJ\} \rangle$

## Using LSRs for extraction

- Based on a set of training sequences, we can mine label sequential rules, e.g.,

$\langle \{easy, JJ\} \{to\} \{*, VB\} \rangle \rightarrow \langle \{easy, JJ\} \{to\} \{\$feature, VB\} \rangle$   
[sup = 10%, conf = 95%]

### Feature Extraction

- Only the right hand side of each rule is needed.
- The word in the sentence segment of a new review that matches **\$feature** is extracted.
- We need to deal with conflict resolution also (multiple rules are applicable).

---

## Extraction of features of formats 2 and 3

- Reviews of these formats are usually complete sentences  
e.g., “the pictures are very clear.”
    - Explicit feature: **picture**
  - “It is small enough to fit easily in a coat pocket or purse.”
    - Implicit feature: **size**
  - Extraction: Frequency based approach
    - Frequent features
    - Infrequent features
- 

---

## Frequency based approach

(Hu and Liu, KDD-04; Liu, Web Data Mining book 2007)

- **Frequent features**: those features that have been talked about by many reviewers.
  - Use sequential pattern mining
  - **Why the frequency based approach?**
    - Different reviewers tell different stories (irrelevant)
    - When product features are discussed, the words that they use converge.
    - They are main features.
  - Sequential pattern mining finds **frequent phrases**.
  - **Froogle has an implementation of the approach (no POS restriction).**
-

## Using part-of relationship and the Web

(Popescu and Etzioni, EMNLP-05)

- Improved (Hu and Liu, KDD-04) by removing those frequent noun phrases that may not be features: better precision (a small drop in recall).
- It identifies **part-of** relationship
  - Each noun phrase is given a pointwise mutual information score between the phrase and **part discriminators** associated with the product class, e.g., a scanner class.
  - The part discriminators for the scanner class are, “of scanner”, “scanner has”, “scanner comes with”, etc, which are used to find components or parts of scanners by searching on the Web: the KnowItAll approach, (Etzioni et al, WWW-04).

## Infrequent features extraction

- How to find the infrequent features?
- Observation: the same opinion word can be used to describe different features and objects.
  - “The pictures are absolutely **amazing**.”
  - “The software that comes with it is **amazing**.”

■ Frequent features

■ Infrequent features



■ Opinion words



---

## Identify feature synonyms

- Liu et al (WWW-05) made an attempt using only WordNet.
  - Carenini et al (K-CAP-05) proposed a more sophisticated method based on several similarity metrics, but it requires a taxonomy of features to be given.
    - The system merges each discovered feature to a feature node in the taxonomy.
    - The similarity metrics are defined based on string similarity, synonyms and other distances measured using WordNet.
    - Experimental results based on digital camera and DVD reviews show promising results.
  - Many ideas in [information integration](#) are applicable.
- 

---

## Identify opinion orientation on feature

- For each feature, we identify the sentiment or opinion orientation expressed by a reviewer.
  - We work based on sentences, but also consider,
    - A sentence can contain multiple features.
    - Different features may have different opinions.
    - E.g., The [battery life](#) and [picture quality](#) are *great* (+), but the [view finder](#) is *small* (-).
  - Almost all approaches make use of **opinion words and phrases**. But notice again:
    - Some opinion words have context independent orientations, e.g., “great”.
    - Some other opinion words have context dependent orientations, e.g., “small”
  - Many ways to use them.
-

# Aggregation of opinion words

(Hu and Liu, KDD-04; Ding and Liu, 2007)

- **Input:** a pair  $(f, s)$ , where  $f$  is a product feature and  $s$  is a sentence that contains  $f$ .
- **Output:** whether the opinion on  $f$  in  $s$  is positive, negative, or neutral.
- Two steps:
  - Step 1: split the sentence if needed based on BUT words (but, except that, etc).
  - Step 2: work on the segment  $s_f$  containing  $f$ . Let the set of opinion words in  $s_f$  be  $w_1, \dots, w_n$ . Sum up their orientations (1, -1, 0), and assign the orientation to  $(f, s)$  accordingly.
- In (Ding and Liu, 2007), step 2 is changed to 
$$\sum_{i=1}^n \frac{w_i \cdot o}{d(w_i, f)}$$
with better results.  $w_i \cdot o$  is the opinion orientation of  $w_i$ .  $d(w_i, f)$  is the distance from  $f$  to  $w_i$ .

# Context dependent opinions

- Popescu and Etzioni (EMNLP-05) used
  - constraints of connectives in (Hazivassiloglou and McKeown, ACL-97), and some additional constraints, e.g., morphological relationships, synonymy and antonymy, and
  - relaxation labeling to propagate opinion orientations to words and features.
- Ding and Liu (2007) used
  - constraints of connectives both at intra-sentence and inter-sentence levels, and
  - additional constraints of, e.g., TOO, BUT, NEGATION, .... to directly assign opinions to  $(f, s)$  with good results (> 0.85 of F-score).

---

## Some other related work

- Morinaga et al. (KDD-02).
- Yi et al. (ICDM-03)
- Kobayashi et al. (AAAI-CAAW-05)
- Ku et al. (AAAI-CAAW-05)
- Carenini et al (EACL-06)
- Kim and Hovy (ACL-06a)
- Kim and Hovy (ACL-06b)
- Eguchi and Lavrendo (EMNLP-06)
- Zhuang et al (CIKM-06)
- Many more

---

## Roadmap

- Opinion mining – the abstraction
- Document level sentiment classification
- Sentence level sentiment analysis
- Feature-based opinion mining and summarization
- ➔ ■ **Comparative sentence and relation extraction**
- Summary

---

## Extraction of Comparatives

(Jinal and Liu, SIGIR-06, AAAI-06; Liu's Web Data Mining book)

- Recall: Two types of evaluation
  - Direct opinions: "This car is bad"
  - Comparisons: "Car X is not as good as car Y"
- They use different language constructs.
- Direct expression of sentiments are good. Comparison may be better.
  - Good or bad, compared to what?
- Comparative Sentence Mining
  - Identify comparative sentences, and
  - extract comparative relations from them.

---

## Linguistic Perspective

- Comparative sentences use morphemes like
  - *more/most, -er/-est, less/least* and *as*.
- *than* and *as* are used to make a 'standard' against which an entity is compared.

### Limitations

- Limited coverage
  - Ex: "In market capital, Intel is way ahead of Amd"
- Non-comparatives with comparative words
  - Ex1: "In the context of speed, faster means better"
- For human consumption; no computational methods

## Types of Comparatives: Gradable

- **Gradable**
  - **Non-Equal Gradable**: Relations of the type *greater or less than*
    - Keywords like *better, ahead, beats, etc*
    - Ex: “optics of camera A is better than that of camera B”
  - **Equative**: Relations of the type *equal to*
    - Keywords and phrases like *equal to, same as, both, all*
    - Ex: “camera A and camera B both come in 7MP”
  - **Superlative**: Relations of the type *greater or less than all others*
    - Keywords and phrases like *best, most, better than all*
    - Ex: “camera A is the cheapest camera available in market”

## Types of comparatives: non-gradable

- **Non-Gradable**: Sentences that compare features of two or more objects, but do not grade them. Sentences which imply:
  - Object A is similar to or different from Object B with regard to some features.
  - Object A has feature  $F_1$ , Object B has feature  $F_2$  ( $F_1$  and  $F_2$  are usually substitutable).
  - Object A has feature  $F$ , but object B does not have.

## Comparative Relation: gradable

- **Definition:** A **gradable comparative relation** captures the essence of a gradable comparative sentence and is represented with the following:

(**relationWord**, **features**, **entityS1**, **entityS2**, **type**)

- **relationWord**: The keyword used to express a comparative relation in a sentence.
- **features**: a set of features being compared.
- **entityS1** and **entityS2**: Sets of entities being compared.
- **type**: *non-equal gradable, equative or superlative.*

## Examples: Comparative relations

- Ex1: “*car X has better controls than car Y*”  
(**relationWord** = better, **features** = controls, **entityS1** = car X, **entityS2** = car Y, **type** = non-equal-gradable)
- Ex2: “*car X and car Y have equal mileage*”  
(**relationWord** = equal, **features** = mileage, **entityS1** = car X, **entityS2** = car Y, **type** = equative)
- Ex3: “*Car X is cheaper than both car Y and car Z*”  
(**relationWord** = cheaper, **features** = null, **entityS1** = car X, **entityS2** = {car Y, car Z}, **type** = non-equal-gradable )
- Ex4: “*company X produces a variety of cars, but still best cars come from company Y*”  
(**relationWord** = best, **features** = cars, **entityS1** = company Y, **entityS2** = null, **type** = superlative)

---

## Tasks

Given a collection of evaluative texts

**Task 1:** Identify comparative sentences.

**Task 2:** Categorize different types of comparative sentences.

**Task 2:** Extract comparative relations from the sentences.

---

## Identify comparative sentences

(Jinal and Liu, SIGIR-06)

### Keyword strategy

- **An observation:** It is easy to find a small set of keywords that covers almost all comparative sentences, i.e., with a very high **recall** and a reasonable **precision**
- We have compiled a list of **83 keywords** used in comparative sentences, which includes:
  - Words with POS tags of JJR, JJS, RBR, RBS
    - POS tags are used as keyword instead of individual words.
    - Exceptions: more, less, most and least
  - Other indicative words like beat, exceed, ahead, etc
  - Phrases like in the lead, on par with, etc

---

## 2-step learning strategy

- **Step1**: Extract sentences which contain at least a keyword (**recall = 98%**, **precision = 32%** on our data set for gradables)
- **Step2**: Use the naïve Bayes (NB) classifier to classify sentences into two classes
  - **comparative** and **non-comparative**.
  - **Attributes**: **class sequential rules** (CSRs) generated from sentences in step1, e.g.,  
 $\langle \{1\}\{3\}\{7, 8\} \rangle \rightarrow \text{class}_i [\text{sup} = 2/5, \text{conf} = 3/4]$

---

### 1. Sequence data preparation

- Use words within radius  $r$  of a keyword to form a sequence (words are replaced with POS tags)
- ....

### 2. CSR generation

- Use different minimum supports for different keywords (multiple minimum supports)
- 13 manual rules, which were hard to generate automatically.

### 3. Learning using a NB classifier

- Use CSRs and manual rules as attributes to build a final classifier.

---

## Classify different types of comparatives

- Classify comparative sentences into three types: **non-equal gradable, equative, and superlative**
  - SVM learner gave the best result.
  - Attribute set is the set of keywords.
  - If the sentence has a particular keyword in the attribute set, the corresponding value is 1, and 0 otherwise.

---

## Extraction of comparative relations

(Jindal and Liu, AAAI-06; Liu's Web mining book 2006)

### Assumptions

- There is only one relation in a sentence.
- Entities and features are nouns (includes nouns, plural nouns and proper nouns) and pronouns.
  - **Adjectival comparatives**
  - **Does not deal with adverbial comparatives**

### 3 steps

- Sequence data generation
- Label sequential rule (LSR) generation
- Build a sequential cover/extractor from LSRs

## Sequence data generation

- **Label Set** = {\$entityS1, \$entityS2, \$feature}
- Three labels are used as **pivots** to generate sequences.
  - Radius of 4 for optimal results
- Following words are also added
  - **Distance words** = {l1, l2, l3, l4, r1, r2, r3, r4}, where “l” means distance of *i* to the left of the pivot.  
“r” means the distance of *i* to the right of pivot.
  - Special words **#start** and **#end** are used to mark the start and the end of a sentence.

## Sequence data generation example

The comparative sentence

“Canon/NNP has/VBZ better/JJR optics/NNS” has  
\$entityS1 “Canon” and \$feature “optics”.

**Sequences are:**

- <{#start}{l1}{**\$entityS1, NNP**}{r1}{has, VBZ }{r2 }  
{better, JJR}{r3}{**\$Feature, NNS**}{r4}{#end}>
- <{#start}{l4}{**\$entityS1, NNP**}{l3}{has, VBZ}{l2 }  
{better, JJR}{l1}{**\$Feature, NNS**}{r1}{#end}>

---

## Build a sequential cover from LSRs

LSR:  $\langle \{*, NN\}\{VBZ\} \rangle \rightarrow \langle \{\$entityS1, NN\}\{VBZ\} \rangle$

- ❑ Select the LSR rule with the highest confidence. Replace the matched elements in the sentences that satisfy the rule with the labels in the rule.
- ❑ Recalculate the confidence of each remaining rule based on the modified data from step 1.
- ❑ Repeat step 1 and 2 until no rule left with confidence higher than the *minconf* value (we used 90%).

(Details skipped)

---

---

## Experimental results (Jindal and Liu, AAAI-06)

- **Identifying Gradable Comparative Sentences**
  - ❑ precision = 82% and recall = 81%.
- **Classification into three gradable types**
  - ❑ SVM gave accuracy of 96%
- **Extraction of comparative relations**
  - ❑ LSR (label sequential rules): F-score = 72%

---

## Roadmap

- Opinion mining – the abstraction
- Document level sentiment classification
- Sentence level sentiment analysis
- Feature-based opinion mining and summarization
- Comparative sentence and relation extraction
- ➔ ■ **Summary**

---

## Summary

Two types of evaluations have been discussed

- **Direct opinions**
  - Document level, sentence level and feature level
  - Structured summary of multiple reviews
- **Comparisons**
  - Identification of comparative sentences
  - Extraction of comparative relations
- **Very challenging problems**
  - Current techniques are still primitive
- **Industrial applications are coming soon...**

---

## Tutorial conclusion

- Almost all the information on the Web (except multi-media information) is expressed in natural language,
  - because everything is for people to read.
  - NLP is thus needed everywhere.
- This tutorial covered only a few topics among many in Web content mining.
- **The Web offers a golden opportunity for NLP researchers** in terms of everything, i.e.,
  - science, technology and practical applications

---

## Thank you!

---