Web Mining
(網路探勘)

Information Integration
(資訊整合)

1011WM10
TLMXM1A
Wed 8,9 (15:10-17:00) U705

Min-Yuh Day
戴敏育
Assistant Professor
專任助理教授
Dept. of Information Management, Tamkang University
淡江大學 資訊管理學系

http://mail. tku.edu.tw/myday/

2012-12-05

課程大綱 (Syllabus)

週次	日期	內容（Subject/Topics）
1	101/09/12	Introduction to Web Mining (網路探勘導論)
2	101/09/19	Association Rules and Sequential Patterns (關聯規則和序列模式)
3	101/09/26	Supervised Learning (監督式學習)
4	101/10/03	Unsupervised Learning (非監督式學習)
5	101/10/10	國慶紀念日(放假一天)
6	101/10/17	Paper Reading and Discussion (論文研讀與討論)
7	101/10/24	Partially Supervised Learning (部分監督式學習)
8	101/10/31	Information Retrieval and Web Search (資訊檢索與網路搜尋)
9	101/11/07	Social Network Analysis (社會網路分析)

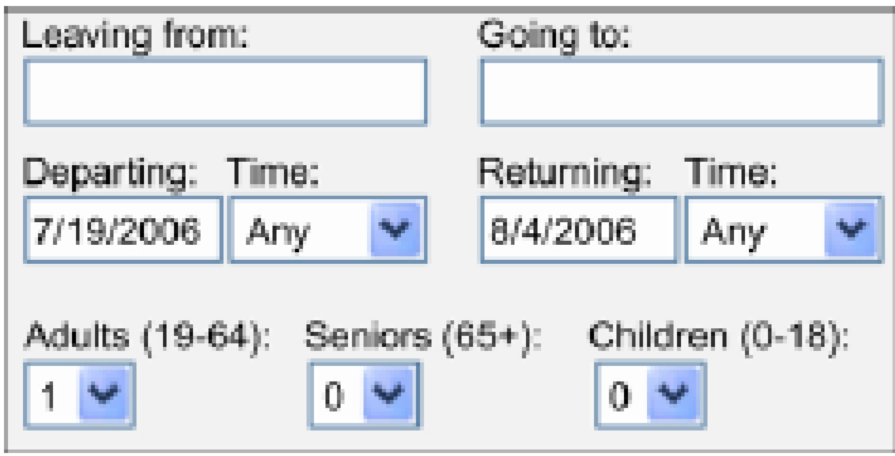
課程大綱 (Syllabus)

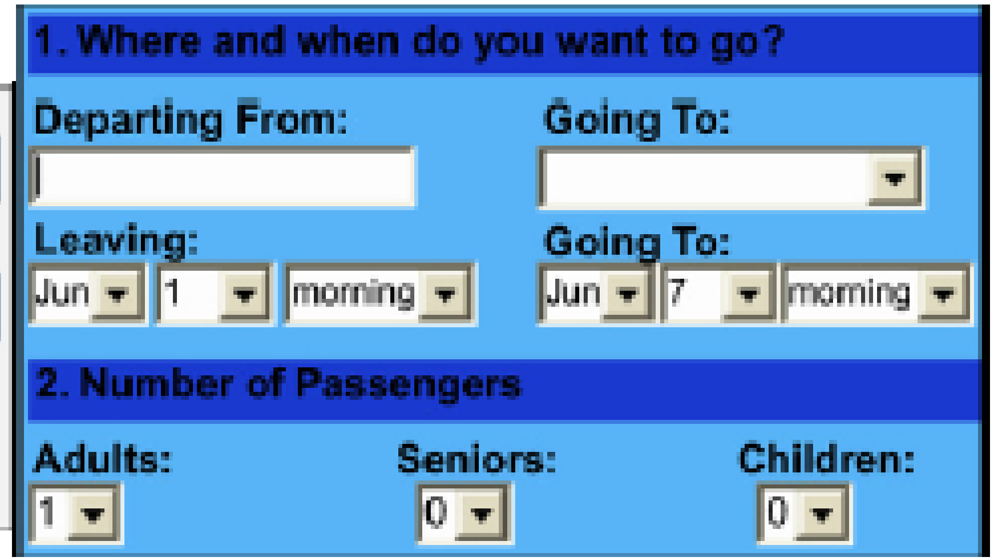
週次	日期	內容（Subject/Topics）
10	101/11/14	Midterm Presentation (期中報告)
11	101/11/21	Web Crawling (網路爬行)
12	101/11/28	Structured Data Extraction (結構化資料擷取)
13	101/12/05	Information Integration (資訊整合)
14	101/12/12	Opinion Mining and Sentiment Analysis (意見探勘與情感分析)
15	101/12/19	Paper Reading and Discussion (論文研讀與討論)
16	101/12/26	Web Usage Mining (網路使用挖掘)
17	102/01/02	Project Presentation 1 (期末報告1)
18	102/01/09	Project Presentation 2 (期末報告2)

Outline

- Information Integration
- Database Integration
  - Schema matching
- **Web query interface integration**
  - Integration of Web Query Interfaces

Two examples of Web query interfaces





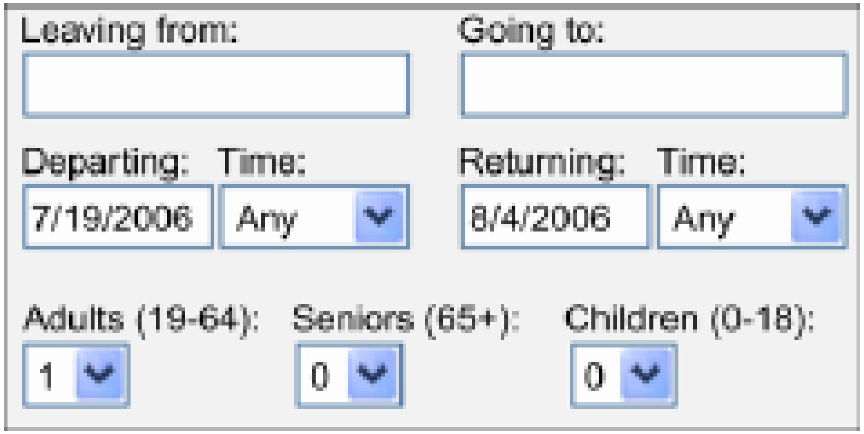
- Web query interfaces are used to formulate queries to retrieve needed data from **Web databases** **(called the deep Web).**

Introduction

- Integrating extracted data
  - column match
  - instance value match.
- Basic integration techniques
- Web information integration research
  - Integration of **Web query interfaces**
  - **Web query interface integration**

Web

- Surface Web
  - The surface Web can be browsed using any Web browser
- Deep Web
  - Deep Web consists of databases that can only be accessed through parameterized query interfaces



Source: Bing Liu (2011) , Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data

Database integration
(Rahm and Berstein 2001)

- Information integration
  - started with database integration
  - database community (since the early 1980s).
- Fundamental problem:
  - schema matching
    - 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, *S*1 and *S*2, representing two customer relations, Cust and Customer.

S1	S2
Cust	**Customer**
CNo	CustID
CompName	Company
FirstName	Contact
LastName	Phone

Source: Bing Liu (2011) , Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data

Integrating two schemas

- Consider two schemas, *S*1 and *S*2, representing two customer relations, Cust and Customer.

S1

Cust

CNo

CompName

FirstName
LastName

S2

Customer

CustID

Company

Contact
Phone

Integrating two schemas

- Represent the mapping with a similarity relation, ≅, over the power sets of $S1$ and $S2$, where each pair in ≅ represents one element of the mapping. E.g.,

  Cust.CNo ≅ Customer.CustID

  Cust.CompName ≅ Customer.Company

  {Cust.FirstName, Cust.LastName} ≅ 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.

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 Berstein 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_1	S_2
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

- 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 (cont.)

- 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

- 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 ≅ varchar, and (primiary key) ≅ unique
- For structured schemas, hierarchical relationships such as
  – is-a and part-of

  may 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

- 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 (cont.)

- 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: $CSim(u, v) = \max\{sim_1(u, v), sim_2(u, v), \ldots, sim_n(u, v)\}$
  - Weighted sum: $CSim(u, v) = \lambda_1 * sim_1(u, v) + \lambda_2 sim_2(u, v) + \ldots + \lambda_n * sim_n(u, v)$
  - Weighted average: $CSim(u,v) = \frac{\lambda_1 Sim_1(u,v) + \lambda_2 Sim_2(u,v) + \ldots + \lambda_n Sim_n(u,v)}{n}$
  - Machine learning: E.g., each similarity as a feature.
  - Many 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

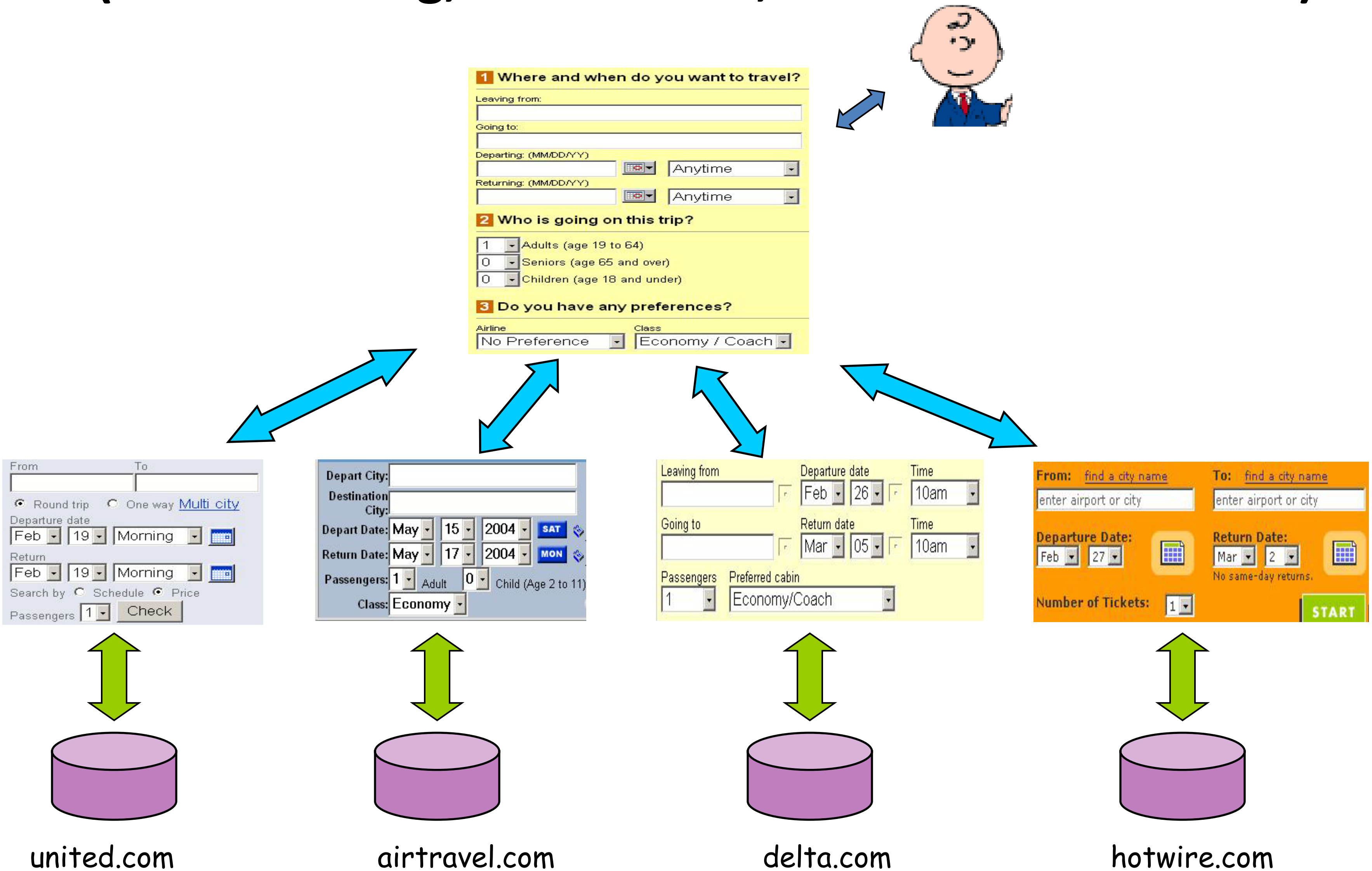
(Rahm and Berstein 2001)

- 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.

Web information integration

- Many integration tasks,
  - Integrating Web query interfaces (search forms)
  - Integrating ontologies (taxonomy)
  - Integrating extracted data
  - …
- Query interface integration
  - 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

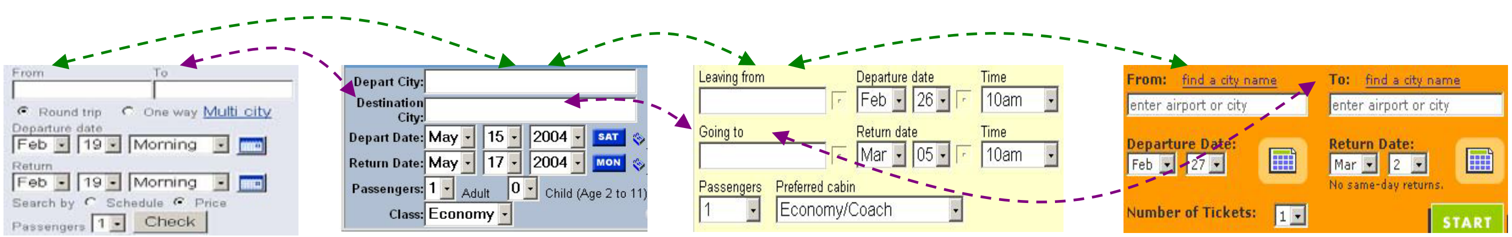
Global Query Interface
(He and Chang, SIGMOD-03; Wu et al. SIGMOD-04)



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



  - Interface Integration – Merge the source query interfaces

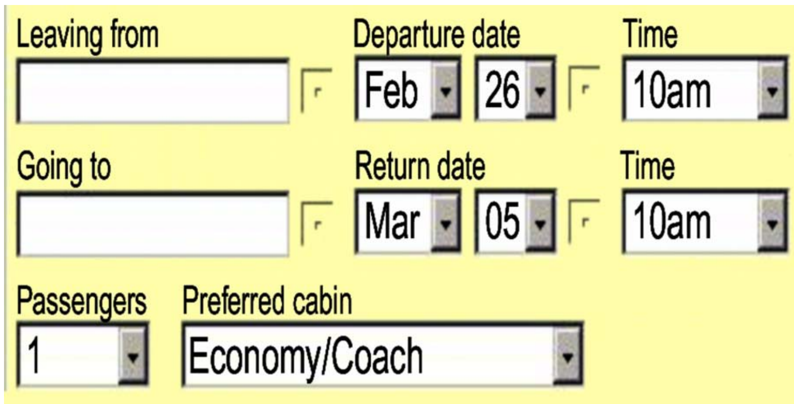
Schema model of query interfaces
(He and Chang, SIGMOD-03)

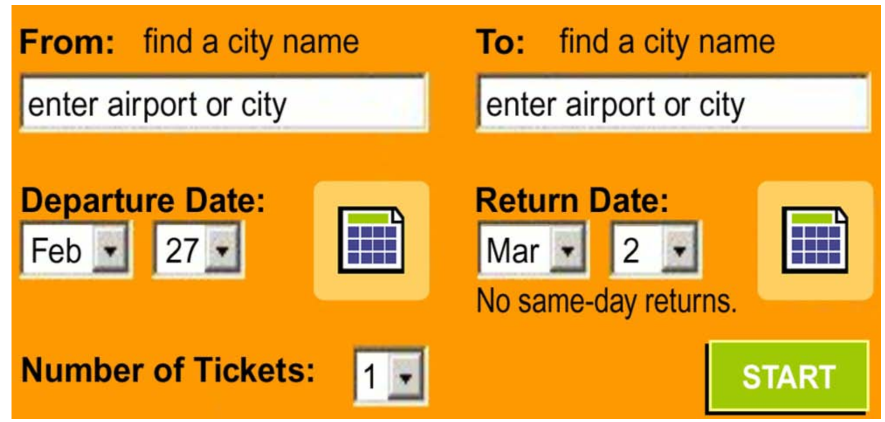
- In each domain, there is a set of essential concepts $C = \{c_1, c_2, \ldots, c_n\}$, used in query interfaces to enable the user to restrict the search.
- A query interface uses a subset of the concepts $S \subseteq C$. A concept i in S may be represented in the interface with a set of attributes (or fields) $f_{i1}, f_{i2}, \ldots, f_{ik}$.
- 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 (cont.)

- 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 ≈ schema matching





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

Source: Bing Liu (2011) , Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data

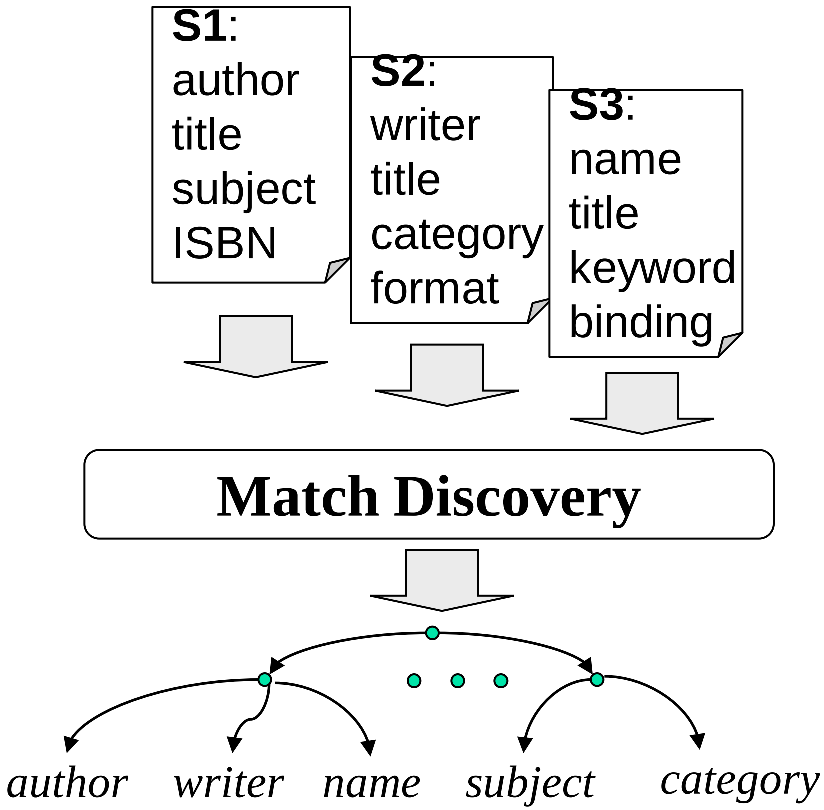
Web is different from databases

(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

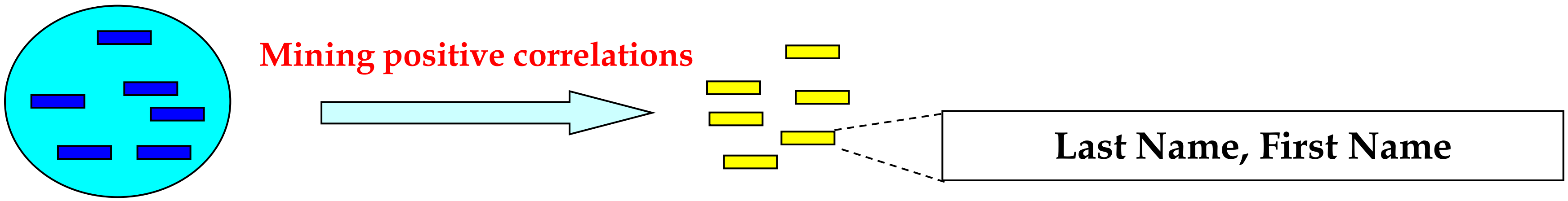


Source: Bing Liu (2011) , Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data

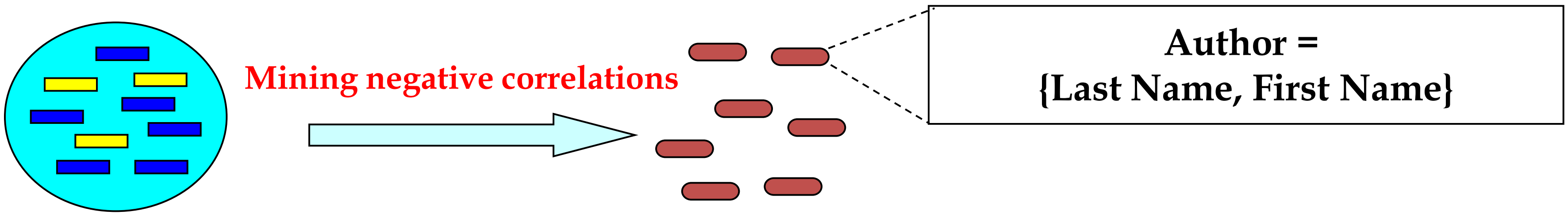
Schema matching as correlation mining
(He and Chang, KDD-04)

- 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.

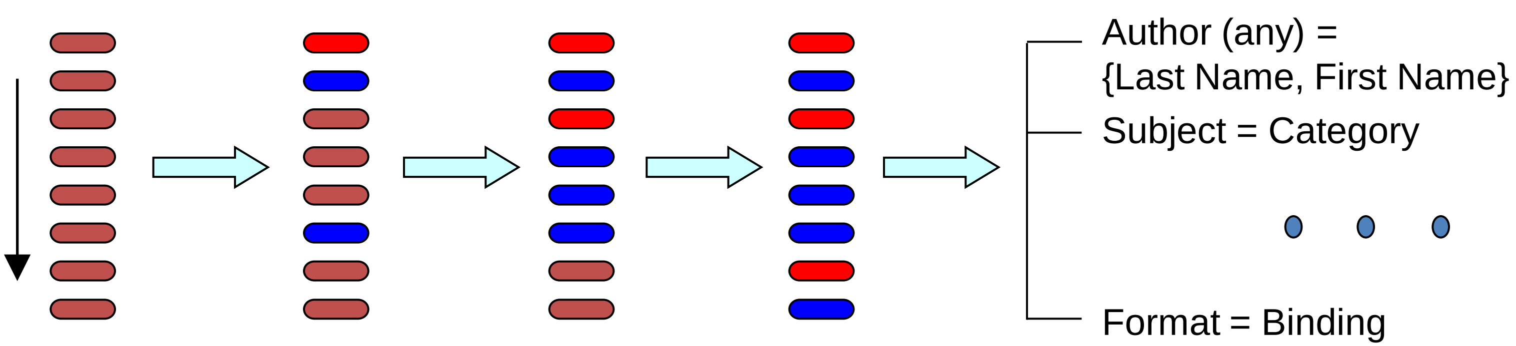
1. Positive correlation mining as potential groups



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

Source: Bing Liu (2011) , Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data

A clustering approach

(Wu et al., SIGMOD-04)

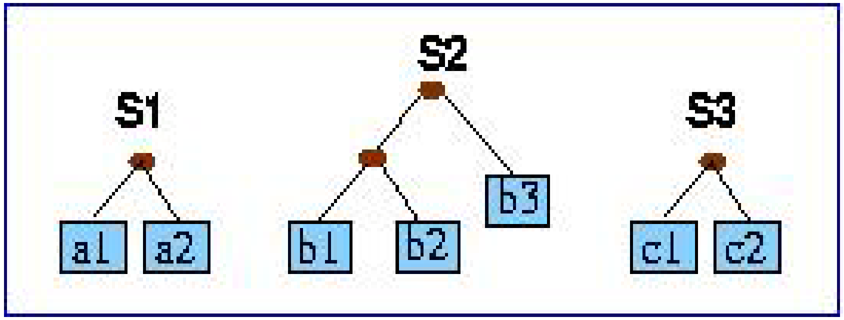
1:1 match using clustering.

Clustering algorithm: Agglomerative hierarchical clustering.

Each cluster contains a set of candidate matches. E.g.,

final clusters: {{a1,b1,c1}, {b2,c2},{a2},{b3}}

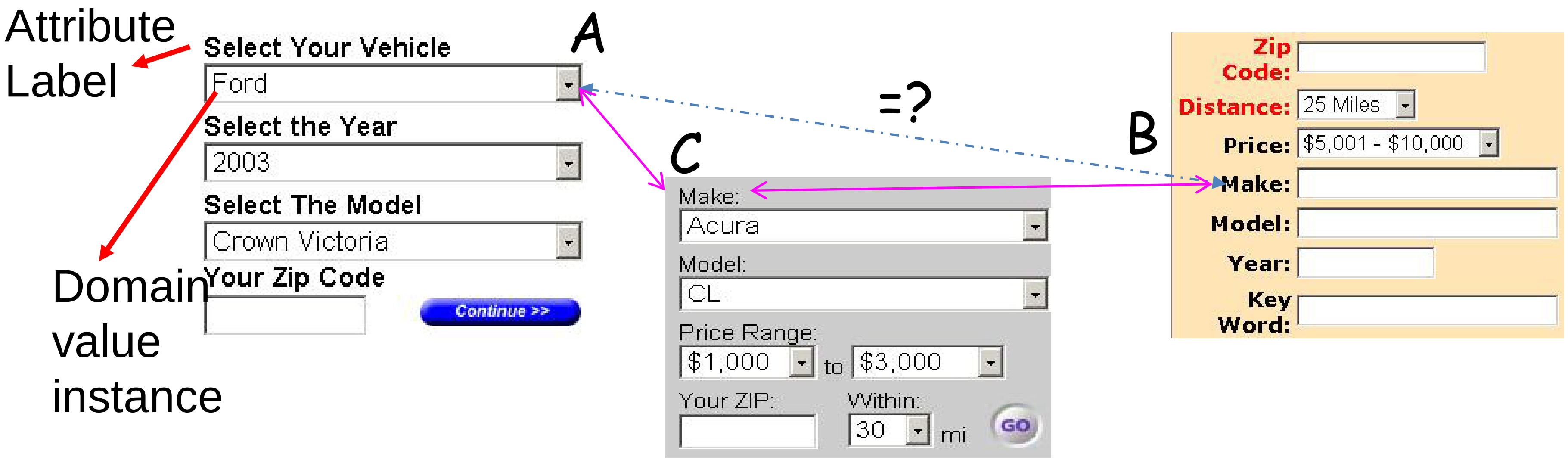
Interfaces:



Similarity measures

- linguistic similarity
- domain similarity

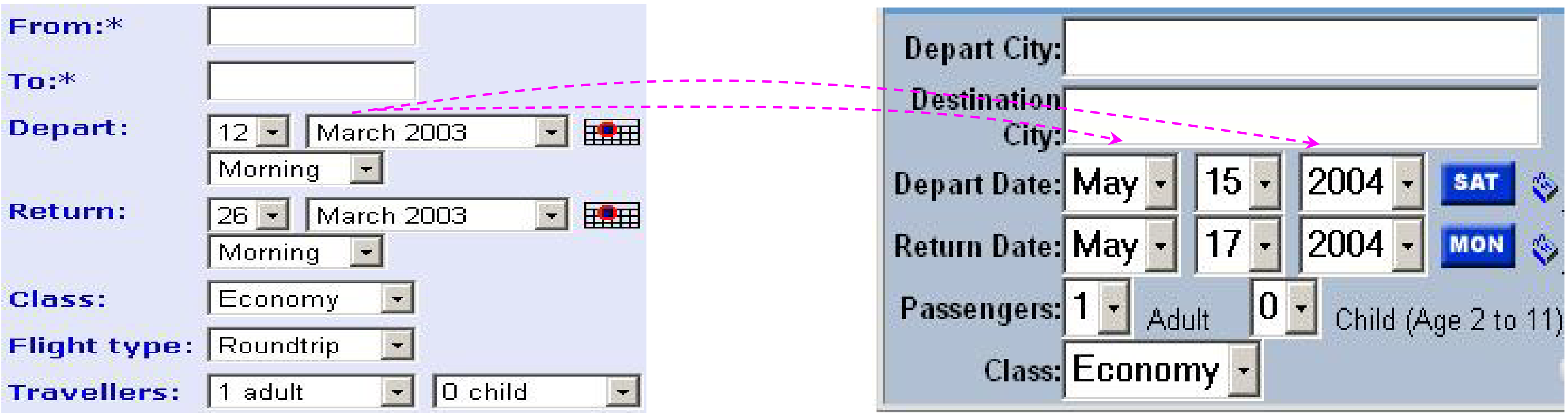
Using the transitive property



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**!

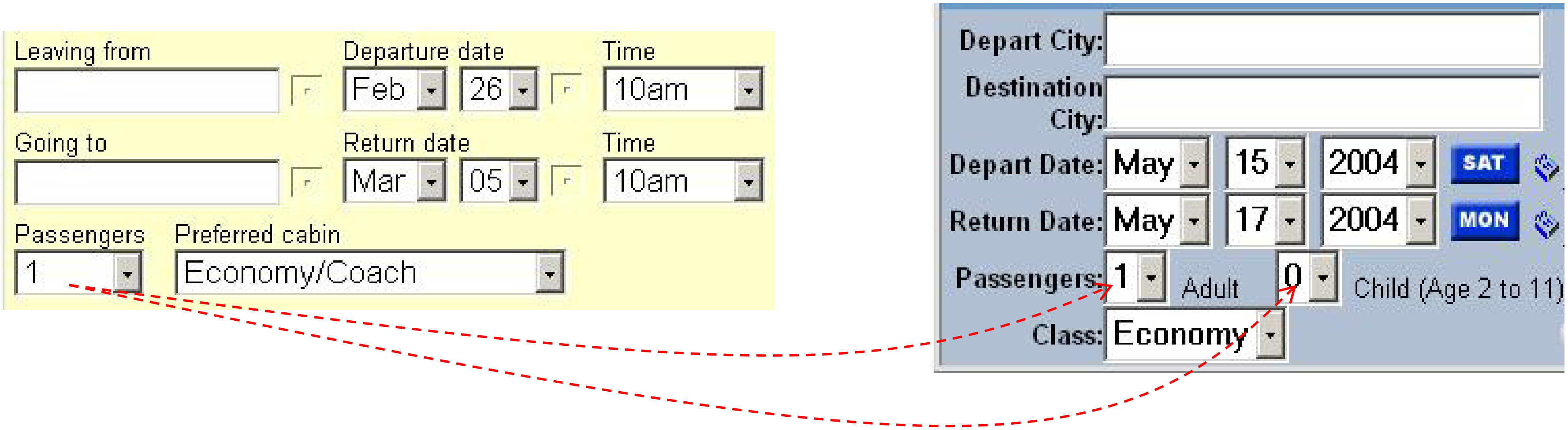
Complex Mappings



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.)



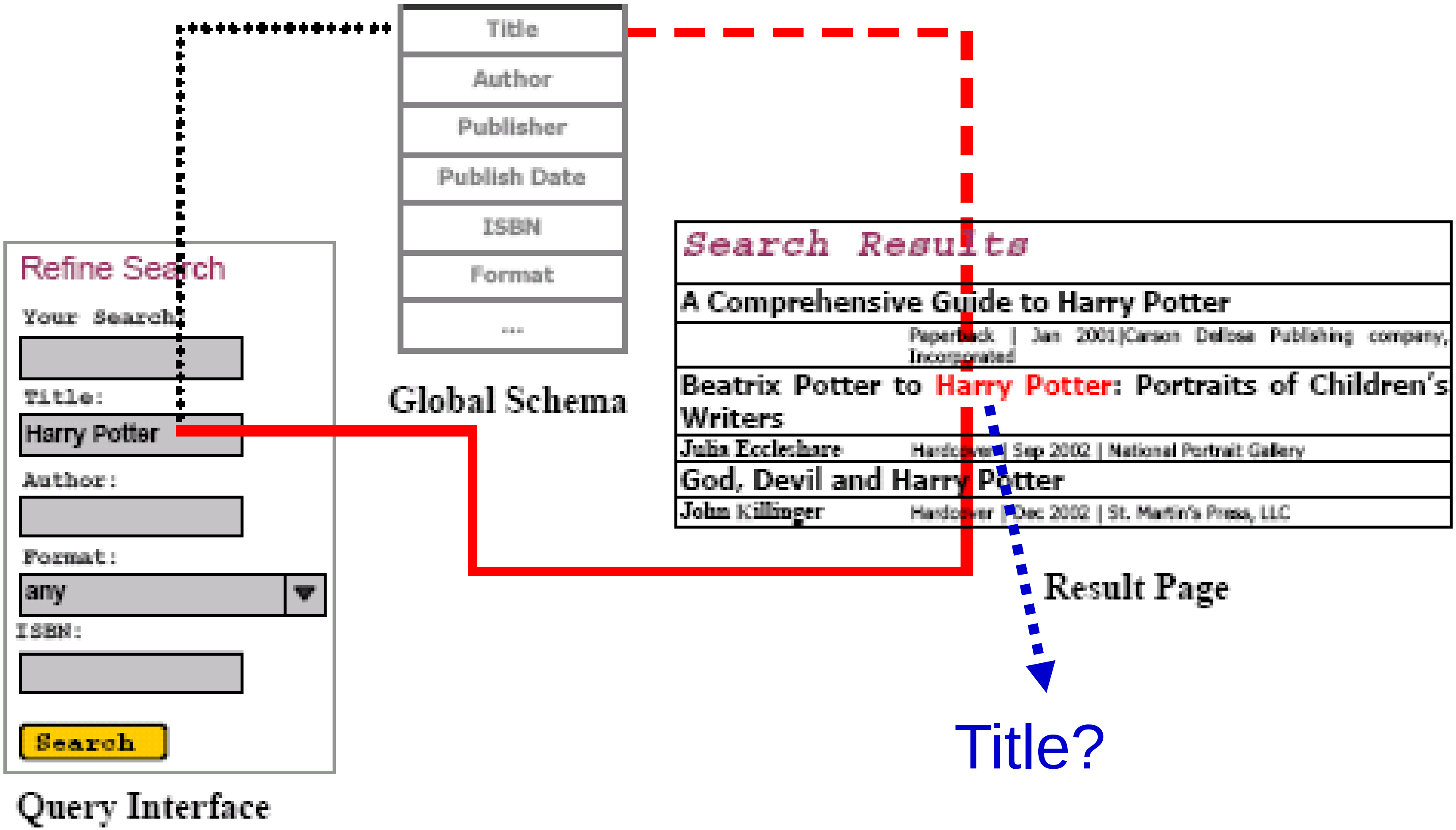
Is-a type – contents of fields on the many side are sum/union of the content of field on the one side.

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.

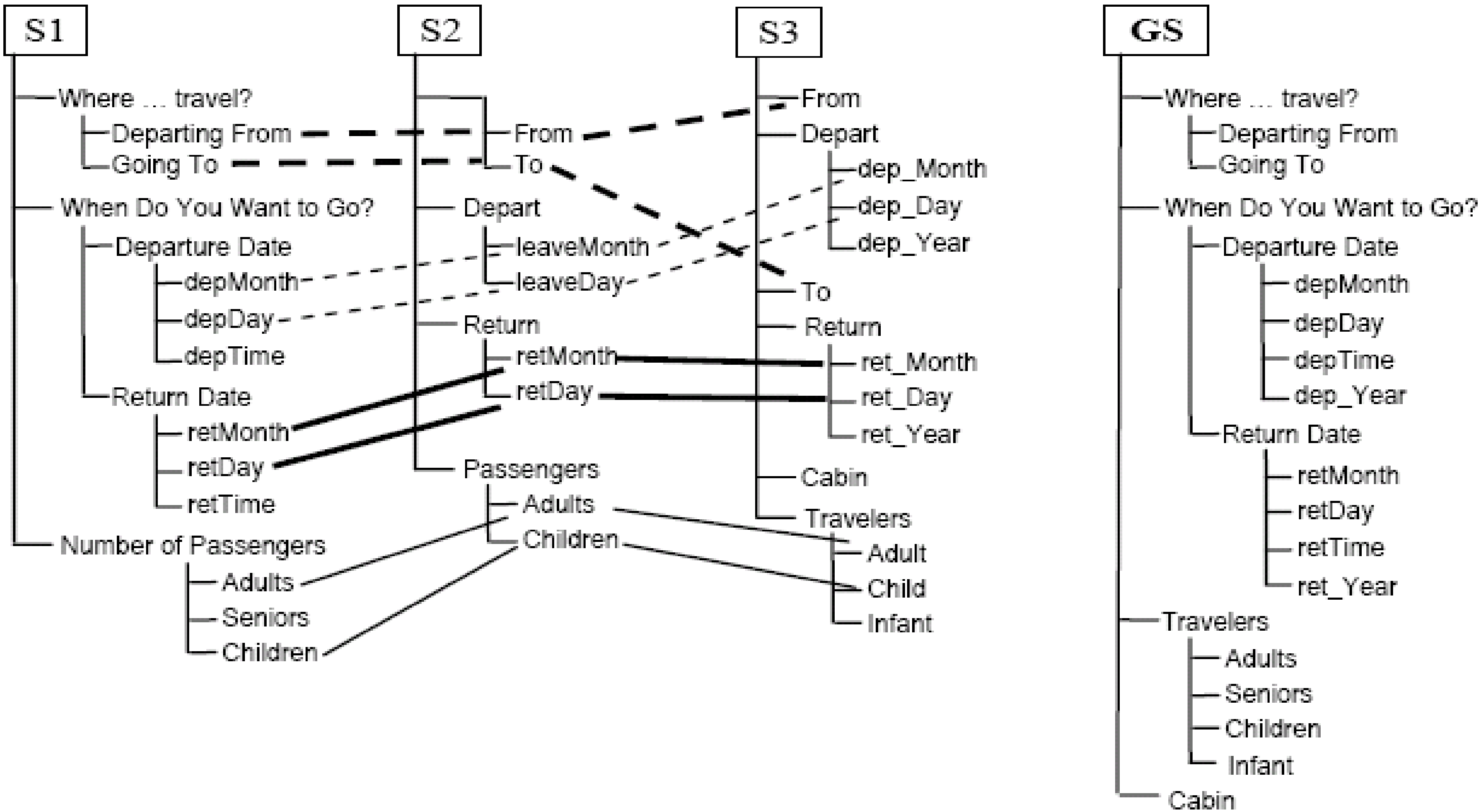
Query Interface and Result Page



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



Source: Bing Liu (2011) , Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data

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.
- Basic integration methods
- 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).
- Database schema matching is a prominent research area in the database community
  - See (Doan and Halevy, AI Magazine 2005) for a short survey.

References

- Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” 2nd Edition, Springer. http://www.cs.uic.edu/~liub/WebMiningBook.html