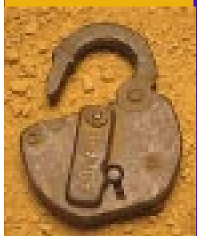


Semi-structured data extraction and schema knowledge mining

Chen Enhong Wang Xufa
Department of Computer science
University of science and Technology of China
Hefei Anhui 230027 P.R.China

INKYO IN

29/11/2000

Contents

Contents

Introduction

- WWW has become a huge information resource
- Vast information is stored in a static HTML format
- Semi-structured data
- Frequent itemset discovery of association rule mining method

Contents

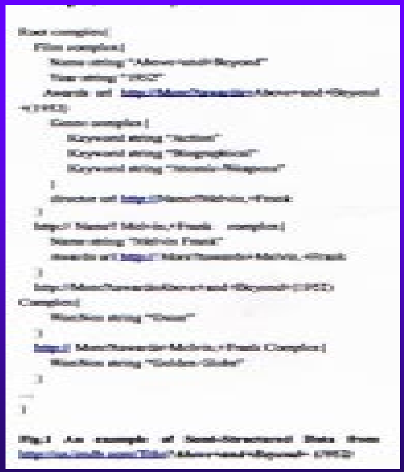
- Introduction
- Semi-structured data representation
- The implementation of semi-structured data extraction
- Schema knowledge discovery for semi-structured data
- Conclusion

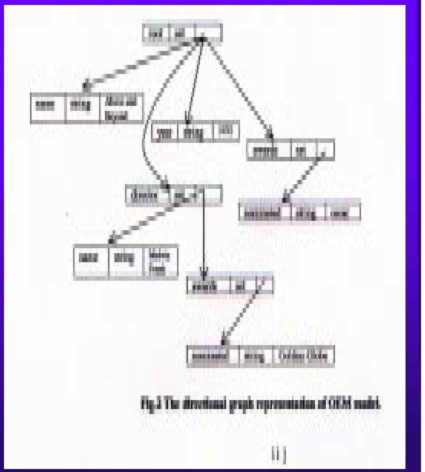
Semi-structured data representation model

- In OEM, each object contains an object identifier and a value(atomic/complex)
- Atomice values : intergers, real, strings, images, program.
- A complex value is a collection of 0 or more OEM sub-objects.

Example

(semi-structured model)





Contents

- Introduction
- Semi-structured data representation
- The implementation of semi-structured data extraction
- Schema knowledge discovery for semi-structured data
- Conclusion

The implementation of semi-structured data extraction

- Procedure

  - provide an initial http address to semi-structured data extractor
  - extractor starts to get the needed HTML file from corresponding remote web server
  - store it in OEM model
  - if useful hyperlinks are detected, be inserted in a Queue
  - After extraction, the semi-structured data can be used for schema knowledge discovery

Example

(The file used to extract information on film pages on web)

1[
2Extract <TITLE>* /*the match pattern*/
3Add label: Name /*the label to be added*/
4Num of Value: 1 /*the number of value*/
5]
……..
6[
7Extract: HREF="/More?tawards+*
8Add label: Award
9Num of Value: 1
10Page type: 1 /*a hyperlink*/
11]
……..

Algorithm

```
Algorithm 1: extract_info( )
Input:    Q: Queue to store the http address;
Output:   Semi-structured Data Represented by OEM;
{
  Match ← True;
  While (Q<>empty) do {
    addr ← first entry in Q;
    get an HTML document Doc(addr) from remote
    web server;
      read the corresponding tag file Tag(Doc(addr));
      repeat {
    if  (Match == True or Cur_tag == NULL) then
      S ← the starting position of next string in
            Doc(addr);
      Cur_tag ← Current tag in Tag(Doc(addr));
      if  (Cur_tag is the prefix pointed by S)   then
          Match = True;
          P(S, Tag(Doc(addr)));
      else advance the pointer in Tag(Doc(addr));
           Match = False;
      endif
     }Until EOF(Doc(addr)) or EOF(Tag(Doc(addr)))
  } end while
}
```

P(S, Tag(f)) performs a particular data extraction task

o. First case

- The information V followed cur_tag needs to be extracted. If V is atomic, then add<label, V> to OEM database
- If V is a hyperlink pointed to another page, then append V and the specification file number for extracting corresponding web pages to the tail of queue Q.

o. Second case

- The algorithm has detected cur_tag. This means that the contents following cur_tag in the file have no more values for the current attribute.

Contents

- Introduction
- Semi-structured data representation
- The implementation of semi-structured data extraction
- Schema knowledge discovery for semi-structured data
- Conclusion

Schema knowledge discovery for semi-structured data

- Definition 1

  Extension : If object O has n outgoing edges, with l_i labeled on each edge and ending object O_i

  $Ext(O)=\{<l_1, O_1>, \ldots, <l_n, O_n>\}$ ➔ a direct extension of O

  $Ext(O_{ij})=\{<l_{i1}, O_{i1}>, \ldots, <l_{im}, Ext(O_{im})>\}$ ➔ an extension of O_{ij}

- Definition 2

  Transaction : If no any object includes Ext(T) as an element in its extension, then we call $Ext(T)=\{<l_1, T_1>, \ldots, <l_n, T_n>\}$ a transaction.

- Definition 3

  Frequent K-schema : A K-schema is a generalized extension with K atomic object, I.e. each object has no extension.

The directional graph representation

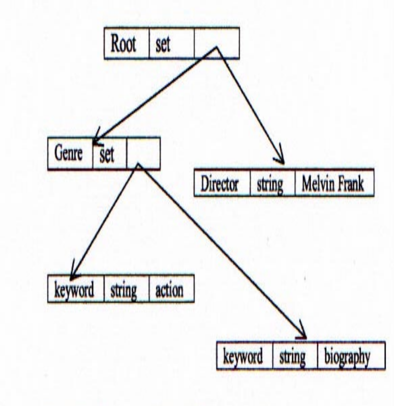


Fig.4 The Directional Graph Representation

- A transaction : T={<genre, keyset>, …,<Director, Melvin Frank>}
- A extension : Keyset is a complex object whose extension is Ext(keyset)={<Keyword, action>, <Keyword, biography>}
- Frequent K-schema : generalized extension is {<Genre, {<Keyword, Action>, <Keyword, Biographical>}>, <Dirctor, Melvin Frank>}

Schema supported by transaction

- Two 1-schema

  $PT_1 = \{l_1, <l_3, O_4>\}$

  $PT_2 = \{l_1, l_4, <l_5, O_7>\}$

- A 2-schema(Based on PT_1, PT_2)

  $PT_1PT_2 = \{l_1, \{\{l_3, O_4\}, \{l_4, <l_5, O_7>\}\}\}$

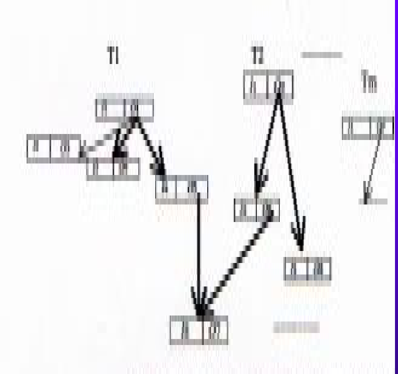
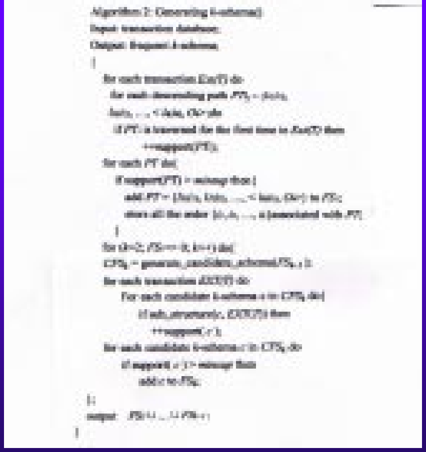


Fig.5 The 2-schema supported by transaction T1 and T2, those with thick black arrows and the connected nodes

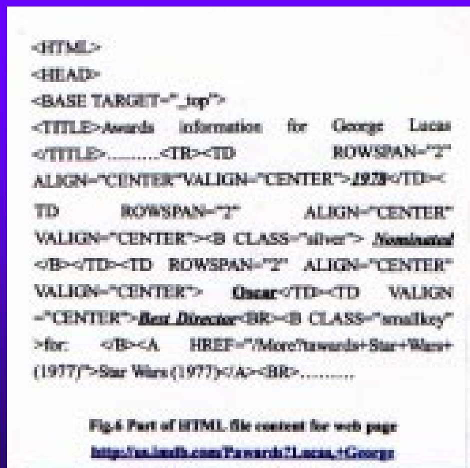
Algorithm

(Generating K-schema)



Algorithm 2: Generating k-schema()
Input: transaction database;
Output: frequent k-schema;

HTML file



```
<HTML>
<HEAD>
<BASE TARGET="_top">
<TITLE>Awards information for George Lucas
</TITLE>.........<TR><TD ROWSPAN="2"
ALIGN="CENTER"VALIGN="CENTER">1978</TD><
TD ROWSPAN="2" ALIGN="CENTER"
VALIGN="CENTER"><B CLASS="silver"> Nominated
</B></TD><TD ROWSPAN="2" ALIGN="CENTER"
VALIGN="CENTER"> Oscar</TD><TD VALIGN
="CENTER">Best Director<BR><B CLASS="smallkey"
>for: </B><A HREF="/More?awards+Star+Wars+
(1977)">Star Wars (1977)</A><BR>.........
```

Fig.6 Part of HTML file content for web page http://us.imdb.com/Pawards?Lucas,+George

Structure association schema

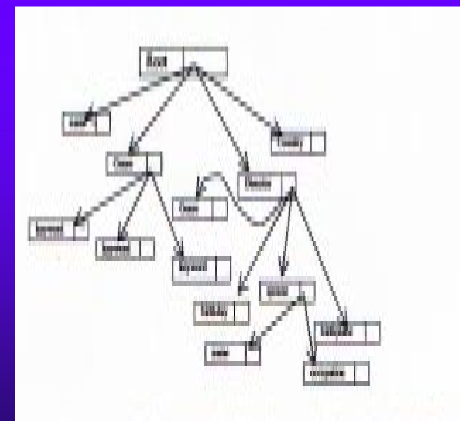


Fig7. An example of the schema mined from http://us.imdb.com

Contents

- Introduction
- Semi-structured data representation
- The implementation of semi-structured data extraction
- Schema knowledge discovery for semi-structured data
- Conclusion

Conclusion

- With the rapid growth of WWW, the semi-structured data will be richer and richer.
- Two directions will be introduced in the future.
  - Machine learning method to the recognition of tag information in extraction
  - The clustering method in semi-structured data knowledge discovery