Journal of Computer Science 5 (8): 596-607, 2009
ISSN 1549-3636

Information Extraction from Hypertext Mark-Up Language Web Pages

Mahmoud Shaker, Hamidah Ibrahim, Aida Mustapha and Lili Nurliyana Abdullah
Department of Computer Science, Faculty of Computer Science and Information Technology,
University Putra Malaysia, 43400 Serdang, Malaysia

Abstract: Problems statement: Nowadays, many users use web search engines to find and gather information. User faces an increasing amount of various HTML information sources. The issue of correlating, integrating and presenting related information to users becomes important. When a user uses a search engine such as Yahoo and Google to seek specific information, the results are not only information about the availability of the desired information, but also information about other pages on which the desired information is mentioned. The number of selected pages is enormous. Therefore, the performance capabilities, the overlap among results for the same queries and limitations of web search engines are an important and large area of research. Extracting information from the web pages also becomes very important because the massive and increasing amount of diverse HTML information sources in the internet that are available to users and the variety of web pages making the process of information extraction from web a challenging problem. **Approach:** This study proposed an approach for extracting information from HTML web pages which was able to extract relevant information from different web pages based on standard classifications. **Results:** Proposed approach was evaluated by conducting experiments on a number of web pages from different domains and achieved increment in precision and F measure as well as decrement in recall. **Conclusion:** Experiments demonstrated that our approach extracted the attributes besides the sub attributes that described the extracted attributes and values of the sub attributes from various web pages. Proposed approach was able to extract the attributes that appear in different names in some of the web pages.

Key words: HTML web pages, information extraction

INTRODUCTION

At the present time, the Internet is general and many people use the Internet to find information. A variety of web pages and the frequently changing of information in web pages make searching and extracting information very difficult. When Internet users want to get information, they first visit search engines such as Yahoo and Google and then visit all web sites suggested by the search engine.

Many researchers such as[7,10,16,17] research on extraction of information from web pages in different domains (traveling, products, business intelligence) but these researches deal with limited web pages and the user still need to use the search engines such as Yahoo and Google to collect more information.

Many of the web pages that the corporations used to announce their products (Internet shops) consist of attributes, sub attributes and values of sub attributes. The sub attributes and values of sub attributes represent the relevant information that the user needs. Products in a single group (web pages) in a single store (Internet shop) tend to have the same attributes, while products in different groups (web pages) have different sets of attributes, for instance:

- One Internet shop presents the attributes, the other does not
- The same attribute is identified differently
- The same attribute contains different kinds of data (sub attributes)

We have proposed a framework for extracting and classifying web pages which consists of three main components: (i) Query Interface (QI) which is used for accepting user's queries and searching web pages based on the user's queries through search engine, (ii) Information Extraction (IE) extracts the relevant information from various web pages obtained from QI and (iii) Relevant Information Analyzer (RIA) analyses the extracted information and removes the repeated information of the same product.

Corresponding Author: Mahmoud Shaker, Department of Computer Science, Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 Serdang, Malaysia

Related works: Many researchers proposed approaches for extracting information from HTML web pages as discussed below.

The Information Systems Universal Data Browser (IS UDB)[7] which has been proposed by Guntis Arnicans and Girts Karnitis is used for searching, extracting, analyzing, classifying, translating, storing, integrating and browsing HTML data. The IS UDB deals with limited HTML data sources (web pages), thus user needs to use search engines such as Yahoo and Google to get the required information.

Another stream of researcher works on extraction of information with agent. Jung *et al*.[17] proposed an Intelligent Traveler Support System (ITSS) for helping traveler to get information about traveling that allows traveler to find important information more easily and effectively. The system deals with limited web pages which are related to destinations and weather. Thus, travelers need to search through the numerous web pages to gather all the necessary information by using search engines such as Yahoo and Google.

Tina Eliassi-Rad and Jude Shavlik[18] have proposed a Wisconsin Adaptive Web Assistant (WAWA) system. They have presented a system for rapidly and easily building instructable and self-adaptive software agents that retrieve and extract information. WAWA interacts with the user and an on-line (textual) environment (e.g., the Web) to build an intelligent agent for retrieving and extracting information. The proposed system needs to embed into a major existing Web browser, thereby minimizing new interface features that users must learn in order to interact with this system as well as develop methods whereby WAWA can automatically infer reasonable training examples by observing users' normal use of their browsers.

Lam *et al*.[14] proposed a system which used different methodologies to extract the information. The extraction task is only individual page based. It means that all the fields for the same record are supposed to be contained in the same page. However, in many other situations, the fields may be located in different relevant pages, such as several linked web pages. Therefore this system needs to handle multi-page extractions.

Fatima Ashraf *et al*.[4] have employed clustering techniques for automatic information extraction from HTML documents containing HTML data. They proposed a system which is called ClusTex. They extend the work in Fatima Ashraf and Reda Alhajj[3] by testing their proposed system in different domains such as Cell phone sales and Marathon schedule. If the tokens of one kind differ from each other in format, then this leads to an incorrect clustering of some tokens.

Saggion *et al*.[10] proposed the MUSING project (Multi-industry, Semantic-based next generation business intelligence). The MUSING project needs to cover many semantic categories including locations, organizations and specific business events to help companies that want to take their business overseas and concerned in knowing the best place to exploit.

Jansen *et al*.[1] proposed a model to improve web search engines by classifying user search based on intention in terms of the type of content specified and operationalize these classifications with defining characteristics. The limitation of this study is that they assigned each query to one and only one category.

Vadrevu *et al*.[16] have focused on information extraction from web pages using presentation regularities and domain knowledge. They argued that there is a need to divide a web page into information blocks or several segments before organizing the content into hierarchical groups and during this process (partition a web page) some of the attribute labels of values may be missing.

Fei *et al*.[5] proposed an information extraction system that aims to automate the tedious process of extracting large collections of facts from large-scale, domain-independent and scalable manner. This system depends on existing search engines creates its own set of challenges. The biggest of these challenges from the fact that search engines only make a small fraction of their results accessible to users.

Zhao *et al*.[9] proposed a new technique to extract the precise search result records template for any search engine automatically. The statistical-based solution does have an inherent weakness in dealing with attributes that have the same or nearly the same values (data units) in all search result records. These data units will be mistakenly recognized as template texts.

Paul Viola and Mukund Narasimhand[15], present a classification algorithm based on discriminatively trained Context Free Grammar (CFG) to extract information from HTML text. The challenge is in converting the HTML information of customer (which is already available in an unstructured form on web sites and in email) into the regularized or schematized form required by a database system.

Utku Irmak and Torsten Suel[19], proposed a complete system for semi-automatic wrapper generation that can be trained on different data sources in a simple interactive manner. This method typically requires the labeling of a single tuple, followed by a selection of a tuple set from a ranked list where the desired set is usually among the first few, plus the labeling of another tuple in the rare case when the desired set is not found in the list.

Gilles Nachouki[6], proposed a new method for extracting information from the web page by using wrappers. The description of the relation to extract is given in the form of a set of example instances.

The structure of the Standard Classifications (SC) and a Web Page (WP): The structure of the standard classifications consists of an attribute, a sub attribute and group of the sub attributes. The following explains the structure of the standard classifications[7,12]:

- Attribute describes the properties of a product. Each product usually has a description of its properties and various aspects of its use. For example the attributes which are used for describing the properties of Nokia product are Size, Display, Memory, Data
- Sub attribute describes the properties of an attribute. For example: Width, Height, Weight, describe the attribute Size
- Group of sub attributes, the sub attributes that belong to the same attribute are grouped together in a group. For example, Width, Height and Weight that belong to the attribute Size are grouped in the same group

We use Attr (SC), Sub_Attr (SC) and G_Sub (SC) to denote the attributes of SC, the sub attributes of SC and group of sub attributes, respectively.

A web page has similar structure as the SC that are attributes, sub attributes and group of sub attributes with additional element, value which describes the value of a sub attribute. For example, class32 and 123 kbps are the values of GPRS which is one of the sub attributes that describes the attribute Data.

The symbol Attr (WP), Sub_Attr (WP) and G_Sub (WP) denote the attributes of WP, the sub attributes of WP and group of the sub attributes, respectively.

We have analyzed several web pages that corporations used to announce their products such as www.gsmarena.com, www.letsgomobile.org, www.esato.com and www.buy.com. We observed the following cases:

The same attribute is presented differently: Figure 1 shows example of a web page that is used to announce Nokia product which consists of attributes, sub attributes and values of the sub attributes. For example, the attribute GENERAL consists of the sub attributes 2GNetwork, 3GNetwork, Announced and Status. Each sub attribute has a value. For example the value of the sub attribute Weight is 110 g.

Figure 2 shows another example of a web page with similar structure as the web page in Fig. 1.

Fig. 1: Example of attributes, sub attributes and values of the sub attributes

Fig. 2: Example of attributes, sub attributes and values of the sub attributes

If we compare the attributes of Fig. 1 and 2, it is found that the attributes have different names and the same attribute may contain different kinds of sub attributes. For example the attribute Memory in Fig. 1 consists of the sub attributes Phonebook, Call records and Card slot while in Fig. 2, the same attribute consists of the sub attributes Internal memory, External memory, Memory slots and Storage types.

The sub attributes appear as attributes: The structure of the web page in Fig. 3 consists of sub attributes and values of the sub attributes. The sub attributes appear as attributes. For example, the sub attributes Height and Width which belong to the attribute Size appear as attributes in Fig. 3.

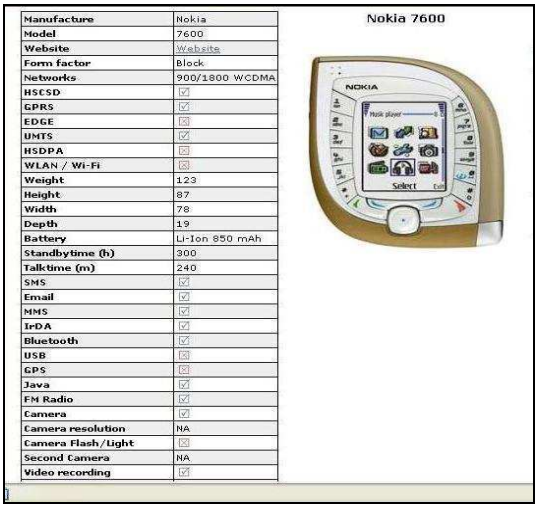


Fig. 3: Example of sub attributes appear as attributes

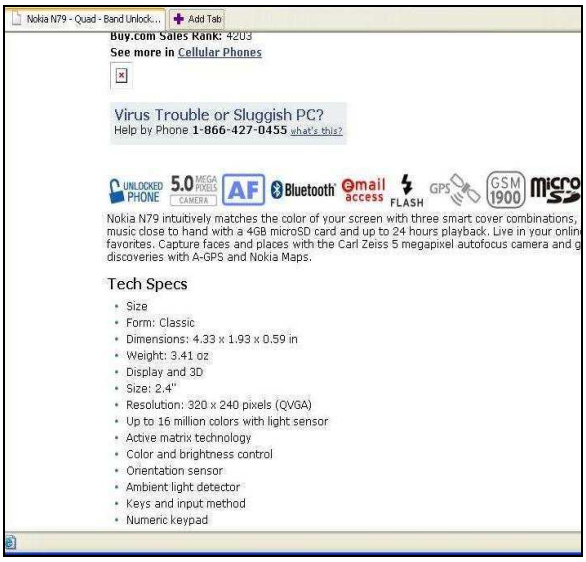


Fig. 4: Example of attributes, sub attributes and values of the sub attributes

The sub attributes appear in different form: Figure 4 shows another example of a web page where the sub attribute and value of the sub attribute appear in different form such as Weight: 3.41oz which describes the attribute Size.

MATERIALS AND METHODS

The steps of the IE: IE extracts and classifies the web pages that are received from QI. Two processes need to be considered, namely: (i) Extraction and (ii) Classification.

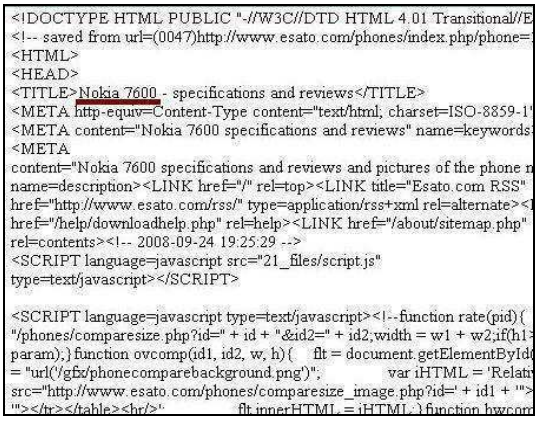


Fig. 5: Example of source code (title of a web page)

Extraction: The process of extracting information consists of three steps, namely: (i) Determine relevant web page by analyzing the title of a web page, (ii) Extract attribute and (iii) Extract sub attribute and value of the sub attribute.

Determine relevant web page: Two tasks are performed in this step, namely: (i) Check the title of a web page and (ii) Save the tokens which are found between the tag <TABLE> and </TABLE> in an array.

Check the title of a web page: Not all of the web pages that are received from QI are related to user's desire. Therefore, IE determines relevant web page by analyzing the title of a web page. IE checks the title of each web page by comparing the tokens which are found between the tag <TITLE> and </TITLE> with a table consisting of a list of product names.

Input: HTM files where HTM = {HTM1, HTM2, … , HTMn}; table (html code) which consists of href, src, DIV, BODY, so on; table (name of products)
Output: Relevant web page
BEGIN
  For each HTMi in HTM do
  BEGIN
    Title_array = { }
    For each token between <TITLE> and </TITLE> do
      If token ∉ html code then Title_array ← Token
    If Title_array ∩ name of products = ∅ then
      Ignore this web page
  END
END

Figure 5 shows an example of source code consisting of title of a web page that is matched with the table consisting of a list of Nokia products.

Save tokens in an array: After IE checks the title of a web page, IE saves the tokens which are found between the tag <TABLE> and </TABLE> in an array for matching them with SC. The tag <TR> denotes the row of <TABLE> and the tag <TD> denotes the field of <TR>. If there is more than one tag <TD> then IE saves the tokens and prefix it with the symbol "-" which denotes a sub attribute (WP) and symbol ":" which denotes the value of a sub attribute (WP). If there is only one <TD> in one of <TR> then IE saves the tokens with prefix "*" which denotes an attribute (WP).

```
Input: Relevant web page
Output: List of tokens
BEGIN
  Table_array = ""
  For each token between <TABLE> and </TABLE>
  do
  BEGIN
    TR_array = ""
    Count_TD = 0
    For each token between <TR> and </TR> do
    BEGIN
      If token = <TD> then Count_TD = Count_TD + 1
      TR_array ← token
    END
  If Count_TD > 1 then
  BEGIN
    Selected_Sub_attr = 0
    For each element in TR_array do
      If token ∉ html code then
      BEGIN
        If Selected_Sub_attr = 0 then
        BEGIN
          Selected_Sub_attr = 1
          Table_array ← token as Sub_Attr (WP)
          with the symbol "-"
        ELSE
          Table_array ← token as value of Sub_Attr
          (WP) with the symbol ":"
        END
      END
  ELSE
    For each element in TR_array do
      If token ∉ html code then
        Table_array ← Attr (WP) with the symbol "*"
    END
  END
END
```

Figure 6 shows an example of a source code (WP) with the tags <TABLE>, <TR> and <TD>. Figure 7 shows the sub attributes and values of the sub attributes found in Fig. 6 saved with the symbols "-" and ":" in an array. For example, the sub attribute Brand saved with the symbol "-" which denotes a sub attribute (WP) and the value Nokia with the symbol ":" which denotes the value of a sub attribute (WP).

```
<DIV id=pricerunner>
<TABLE style="TEXT-ALIGN: left" cellSpacing=0 cellPadding=0>
 <TBODY>
 <TR>
  <TD class=spec_item>Brand </TD>
  <TD>Nokia </TD></TR>
 <TR>
  <TD class=spec_item>Type </TD>
  <TD>6212 classic </TD></TR>
 <TR>
  <TD class=spec_item>Form factor </TD>
  <TD>Candybar </TD></TR>
 <TR>
  <TD class=spec_item>Color </TD>
  <TD>Black </TD></TR>
 <TR>
```

Fig. 6: Example of a source code (WP) with the tags <TABLE>, <TR> and <TD>

```
-Brand
:Nokia

-Type
:6212 classic

-Form factor
:Candybar

-Color
:Black
```

Fig. 7: The sub attributes (WP) and values of sub attributes (WP) shown in Fig. 6 saved in an array

Extract attribute: IE matches the tokens which are saved in an array with Attr (SC). If there is a match then IE extracts the Attr (WP), Sub_Attr (WP) and value of Sub_Atrr (WP).

```
Input: List of tokens
Output: The extracted attribute, sub attributes and
values of the sub attributes
BEGIN
  Matched_Attr = 0
  For each token in Table_array do
  BEGIN
    If token prefixed with the symbol "*" and
    token = Attr (SC) then
    BEGIN
      Matched_Attr = 1
      Extract Attr (WP)
    ELSE
      If Matched_Attr = 1 then
```

```
        BEGIN
          If token prefixed with the symbol "-" then
          BEGIN
            Extract Sub_Attr (WP)
            Correct_Match = Correct_Match + 1
          ELSE
            Extract value of Sub_Attr (WP)
          END
        END
      END
    END
END
```

Figure 8 shows example of extracted attribute and sub attributes. The attribute Size (WP) matched with the attribute Size (SC), therefore IE extracts the attribute, the sub attributes that are Width, Height and Depth that describe the extracted attribute and values of the sub attributes.

If there is no match among a token saved in an array and Attr (SC) then IE matches the token with Sub_Attr (SC) as shown in the next step.

Extract sub attribute and value of the sub attribute: In this step, there are two types of matching, namely: (i) match token with Sub_Attr (SC) and (ii) match G_Sub (WP) with each G_Sub (SC).

Match token with Sub_Attr (SC): In some of the web pages, the sub attribute appears as attribute. Therefore, IE matches the token with Sub_Attr (SC). If there is a match then IE extracts the token and saves it in a text file as a sub attribute together with its value.

Input: Tokens
Output: The extracted sub attribute and value of the sub attribute

```
BEGIN
  For each token do
  BEGIN
    If token prefixed with the symbol "-" and token =
    Sub_Attr (SC) then
    BEGIN
      Extract token as a sub attribute
      Correct_Match = Correct_Match + 1
    ELSE
      Extract value of Sub_Attr (WP)
    END
  END
END
```

Figure 9 illustrates an example of the extracted sub attributes. The attribute Width (WP) matched with the sub attribute Width (SC) which describes the attribute Size. Therefore, IE extracts the attribute Width (WP) as sub attribute.

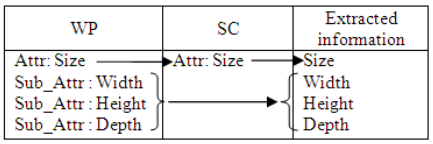


Fig. 8: Example of matching Attr (WP) with Attr (SC)

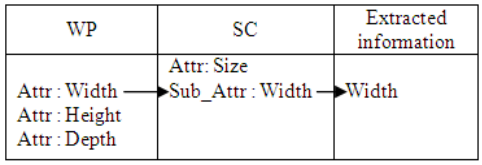


Fig. 9: Example of matching Attr (WP) with Sub_Attr (SC)

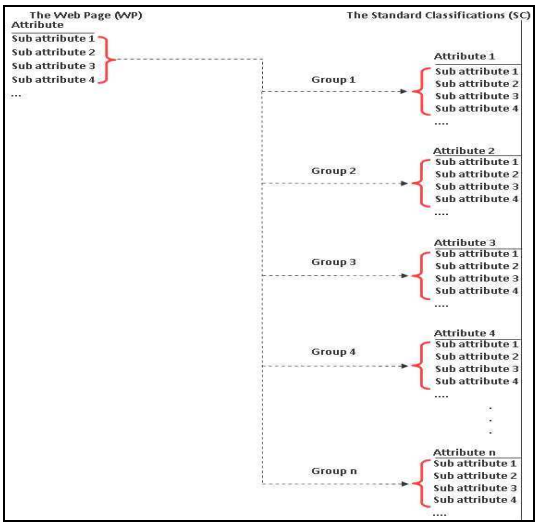


Fig. 10: IE matches a group of the sub attributes (WP) with each group of the sub attributes (SC)

Match G_Sub (WP) with each G_Sub (SC): Sometimes an attribute (WP) appears in different names which are not found in the standard classifications (SC), therefore IE matches G_Sub (WP) that describes the Attr (WP) which appears in different name with each G_Sub (SC). IE saves the number of sub attributes from each G_Sub (SC) that matched with G_Sub (WP) in an array. IE selects the G_Sub (SC) with the maximum number of matched sub attributes and extracts Attr (WP), G_Sub (WP) and values of the sub attributes as shown in Fig. 10.

Input: List of tokens
Output: The extracted attribute, sub attributes and values of the sub attributes

```
BEGIN
  Maximum_array = 0
  For each G_Sub (SC) do
    If G_Sub (WP) ⊆ G_Sub (SC) then
      Maximum_array ← The number of sub
      attributes (SC) that matched
  If number of elements in Maximum_array > 0
  BEGIN
    Select the Attr (WP), G_Sub (WP) and values of
    the sub attributes with the maximum number of
    matched sub attributes from Maximum_array
    Correct_Match = Correct_Match + Number of
    matched sub attributes
  ELSE
    InCorrect_Match = InCorrect_Match + Number
    of unmatched sub attributes
  END
END
```

Figure 11 shows example of extracted attribute and sub attributes. The attribute Dimensions (WP) is not found in the SC, therefore IE matches the group of the sub attributes that describes the attribute Dimensions (WP) with each group of the sub attributes (SC).

Classification: IE classifies the extracted information. Two steps must be considered, namely: (i) Identify the index number of Attr (SC) that matched and (ii) Group the extracted attributes and sub attributes based on the index number.

Identify the index number of Attr (SC) that matched: IE saves the Attr (WP), Sub_Attr (WP) and value of Sub_Attr (WP) in a text file with the index of Attr (SC) that matched. Figure 12 shows the example of the attributes that are saved in database with the index number Index_no.

Figure 13 shows the example of the sub attributes and values of the sub attributes, where each line begins with the index of Attr (SC) that is matched. For example, IE saves the sub attribute weight with the index of the attribute Size.

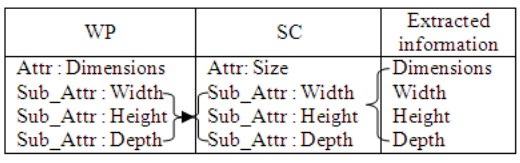


Fig. 11: Example of matching G_Sub (WP) with G_Sub (SC)

Group the extracted attributes and sub attributes based on the index number: The matched attributes and sub attributes are then grouped based on the index number. For example, the lines with the index 6 are grouped together as attribute DATA, as shown in Fig. 14 which illustrates the example of the extracted attributes and sub attributes that are shown in Fig. 13 after grouping them based on the index number.

In Fig. 14, the symbol "*" denotes Attr (WP), the symbol "-" denotes Sub_Attr (WP) and the lines without the symbols "*" and "-" represent the value of Sub_Attr (WP).

IE saves the extracted information in a text file. Figure 15 shows an example of a text file.

Attribute	Index_no
General	1
Size	2
Display	3
Ringtones	4
Memory	5
Data	6
Features	7
Battery	8

Fig. 12: Attr (SC) saved in database, Index_no denotes the index of Attr (SC)

```
6- umts
6: Yes

6- hsdpa
6: No


2- weight
2: 123

2- height
2: 87

2- width
2: 78

2- depth
2: 19


8- standbytime(h)
8: 300

8- talktime(m)
8: 240

7- sms
7: Yes

7- email
7: Yes

7- mms
7: Yes


6- bluetooth
6: Yes

6- usb
6: No
```

Fig. 13: Attr (WP) in a text file with index number of Attr (SC)

Fig. 14: Attr (WP), Sub_Attr (WP) and value of Sub_Attr (WP) in a text file after grouping

Fig. 15: Example of a text file with the extracted attributes, sub attributes and values of sub attributes from a web page

Next, the name of the text file, path of the text file, name of product, number of matched sub attributes-values extracted (WP) and number of unmatched sub attributes-values extracted (WP) are saved in a table (Structured Information). Figure 16 shows an example of the structured information.

	Name of Text	Path of Text	Product	No of correct values extracted	No of incorrect values extracted	Total number of possible correct values
▶	Text 1	F:\my project2\WP in Text after classify\1.txt	Nokia N79	52	2	54
	Text 2	F:\my project2\WP in Text after classify\2.txt	Nokia 7600	53	1	54
	Text 3	F:\my project2\WP in Text after classify\3.txt	Nokia 6212 classic	49	0	49
	Text 4	F:\my project2\WP in Text after classify\4.txt	Nokia 6600 fold	28	0	28
	Text 5	F:\my project2\WP in Text after classify\5.txt	Nokia 7310 Supernova	27	0	27
	Text 6	F:\my project2\WP in Text after classify\6.txt	nokia 7600	14	10	24
	Text 7	F:\my project2\WP in Text after classify\7.txt	Nokia 5800	32	0	32
*						

Fig. 16: Example of the structured information

The steps of Relevant Information Analyzer (RIA): RIA analyzes the relevant information extracted from Information Extraction (IE). RIA identifies the attributes and sub attributes that belong to the same product which are extracted repetitively and compares among them to remove the repetitive attributes and sub attributes. RIA comprises of two main steps for analyzing the relevant information extracted from IE.

Group the records with the same name of a product in a table: RIA groups the records in the Structured Information based on the name of the product. Those records with the same product name are saved in the same table (Similar Table).

For example, there are two text files in Fig. 16 that are Text 2 consisting of 53 extracted sub attributes and Text 6 consisting of 14 extracted sub attributes for the same product Nokia 7600. Text 2 and 6 are then saved in the same table by RIA.

Compare the extracted sub attributes that belong to the same product: RIA compares the extracted sub attributes that belong to the same product and removes the attributes and sub attributes that are duplicates.

y = number of records in Similar Table
Array [ ] = “”
Name_text [ ] = “”
Matched_array = “” /* used for storing the name of the text file that is matched
For x = 1 to y – 1 do
Begin
  Array [x] ← Attr (WP) and G_Sub (WP) which are saved in a text file
  Name_text [x] ← Name of the text file saved in Similar Table
    For z = x + 1 to y do
    Begin
      Array [z] ← Attr (WP) and G_Sub (WP) which are saved in a text file
      Name_text [z] ← Name of the text file saved in Similar Table
      If Name_text [x] and Name_text [z] ∉ Matched_array then

```
        Begin
          If Array [x] ⊆ Array [z] then
          Begin
            Matched_array ← Name_text [x]
          Else
            If Array [z] ⊆ Array [x] then
              Matched_array ← Name_text [z]
          End
      End
    End
End
```

For example, refer to Text 2 and 6 shown in Fig. 16. RIA compares the sub attributes of Text 2 and 6. Text 2 consists of 53 extracted sub attributes while Text 6 consists of 14 extracted sub attributes which are found to be part of the extracted attributes of Text 2. Therefore, RIA removes Text 6. Figure 17 shows example of the extracted information.

RESULTS AND DISCUSSION

In, results we present details of the experiments followed by discussion and comparison with those reported in the literature. To evaluate our approach, the following three domains were selected: (1) Nokia products, (2) office materials and (3) Kodak single use cameras.

Evaluation: The parameters used to evaluate our approach are precision, recall and the geometrical average of these two, the F value. The F measure can be defined to have a metric that can be used to compare various IE systems by only one value[13]. Researchers in the IE field commonly report their result by using these metrics:

Precision (P) = C / (C+I)
Recall (R) = C / T

Where:
C = Number of correct sub attributes-values extracted
I = Number of incorrect sub attributes-values extracted
T = Total number of possible correct sub attributes-values

$$f = \frac{\left(\beta^2 + 1\right) * P * R}{\beta^2 * P + R}$$

where, β2 is the weight of R over P, a value of β2 = 1 means that recall and precision are weighted equally. Fatima Ashraf *et al.*[4] reported the F value where β2 is taken to be 1.

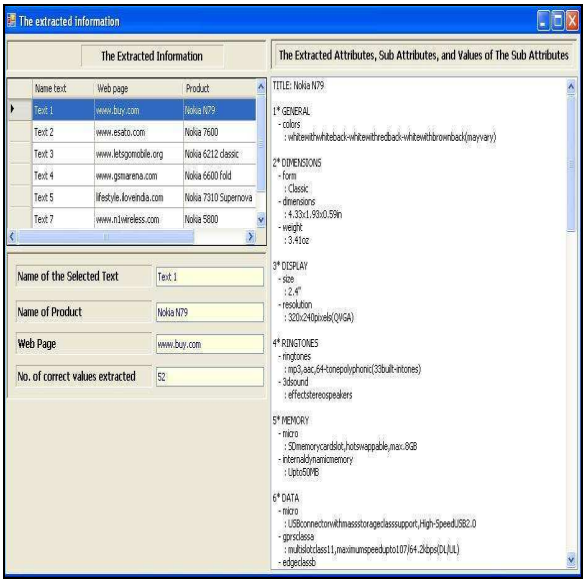


Fig. 17: Example of the extracted information

Experiments and results: Nokia products: we have used the standard classification which has been proposed by Guntis Arnicans and Girts Karnitis[7] to evaluate the proposed approach and compare the results with previous approach. To eavluate our approach, the following web sites were selected that are www.buy.com "Cell Phones and Services" which is used by[3], www.gsmarena.com, www.esato.com, www.letsgomobile.org and lifestyle.iloveindia.com which are used to announce the products of Nokia mobile phone.

Fatima Ashraf *et al.*[4] tested their approach on www.buy.com "Cell Phones and Services" and they reported P = 94.55%, R = 100% and F = 97.19%. They analyzed the test results on a web page from www.buy.com. This web page contains of the Manufacturer, the Cell Phone Model and the Price. In their work, if the tokens of one kind differ from each other in format, then this would lead to an incorrect clustering of some tokens. Our approach extracts the attributes which are Size, Display, Ringtones, Memory, Data, Features and Battery from the web site www.buy.com besides the sub attributes that describe the attributes and values of the sub attributes. While the same attributes, sub attributes and values of the sub attributes in addition to the attribute General are extracted from the web sites www.gsmarena.com, www.esato.com, www.letsgomobile.org and lifestyle.iloveindia.com. We reported P = 99.07%, R = 99.07% and F = 99.07% as shown in Table 1.

Table 1: Extraction results from our approach compared to Fatima Ashraf *et al*.[4]

	Precision (%)	Recall (%)	F (%)
The proposed approach	99.07	99.07	99.07
Previous approach[4]	94.55	100.00	97.19

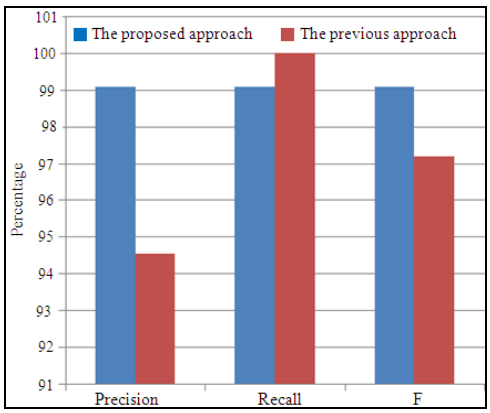
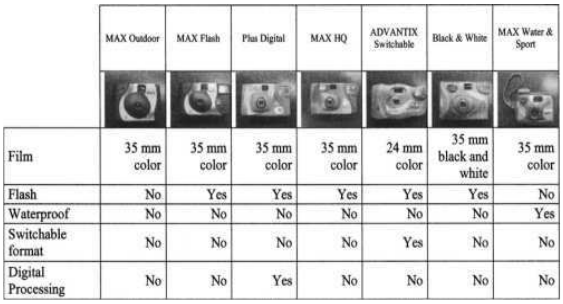


Fig. 18: Extraction results from our approach compared to Fatima Ashraf *et al*.[4]

Figure 18 shows the increment in precision and F measure that is achieved in our approach and decrement in recall. The ratio of increment in precision is 4.52%, the ratio of decrement in recall is 0.93% and the ratio of increment in F is 1.88%. Kaiser and Miksch[13] explained that if a system optimized for high precision the feasibility of not detecting all relevant information improves while if recall is optimized it is possible that the system classifies irrelevant information as relevant.

Office materials: We used the standard classification which has been proposed by[2]. The following web sites were selected that are www.ebay.com "Office Materials Domain" which is used by[2] to create their standard classification, www.commerce.com.tw and www.tootoomart.com which are used to announce the office material products. We reported P = 100%, R = 100% and F = 100%.

Kodak single use cameras: We used the standard classification which is called Kodak single use cameras domain that consists of seven cameras manufactured by Kodak that are readily available in the market with functions, namely: Flash, digital processing, waterproof, black and white and advanced photo system with switchable format. Figure 19 shows the seven cameras which have been used by many researchers to create a standard classification of the products such as[8,11]. They described the major attributes of each camera which are listed in Fig. 19.



Film	35 mm color	35 mm color	35 mm color	35 mm color	24 mm color	35 mm black and white	35 mm color
Flash	No	Yes	Yes	Yes	Yes	Yes	No
Waterproof	No	No	No	No	No	No	Yes
Switchable format	No	No	No	No	Yes	No	No
Digital Processing	No	No	Yes	No	No	No	No

Fig. 19: Summary of one-time-use cameras family

Fig. 20: Herb information (drug)

Table 2: Overall extraction results from different domains

Domain	P (%)	R (%)	F (%)
Office materials	100.00	100.00	100.00
Nokia products	99.07	99.07	99.07
Herbs	94.88	94.88	94.88
Kodak single use cameras	83.35	83.35	83.35

The following web sites were selected that are shopping.msn.com, shopping.yahoo.com, www.dealtime.com and www.epinions.com which are used to announce the Kodak camera products. We selected the web pages that announce the Max Flash camera, Plus Digital camera, Max HQ camera and Max Water and Sport camera shown in Fig. 19 as an example to test our approach. We reported P = 83.35%, R = 83.35% and F = 83.35.

To evaluate our approach without using standard classification, we analyze further the test results on herbs web pages from www.holisticonline.com, www.gardenexpress.com.au, www.naturehills.com and www.ces.ncsu.edu. Those web pages contain herbs information that relate to drug as shown in Fig. 20, herb's tree and herb's flower. The attributes that describe the herbs are saved in database. We reported P = 94.88%, R = 94.88% and F = 94.88%.

Table 2 and Fig. 21 show the overall results from the four domains that were tested.

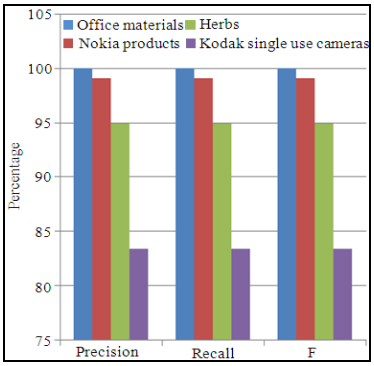


Fig. 21: Overall extraction results from different domains

CONCLUSION

In this study, we proposed an approach for extracting relevant information from various web pages. Experiments demonstrated that our approach extracts the attributes besides the sub attributes that describe the extracted attributes and values of the sub attributes from various web pages. Besides, the proposed approach is able to extract the attributes that appear in different names in some of the web pages.

There are a number of suggestions to extend this study. One direction is to link the presented research to various search engines such as Msn, Yahoo and Google, to search relevant information based on the user's queries for extracting information from various web pages obtained from different search engines. Besides, a high ranking for a specific keywords in one search engine does not automatically mean that the obtained web pages will rank highly for the same keywords in another search engine. Another direction is to add an approach for parsing the web pages which are not based on the English language.

REFERENCES

1. Jansen, B.J., D.L. Booth and A. Spink, 2007. Determining the user intent of web search engine queries. Proceedings of the 16th International Conference on World Wide Web, May 8-12, ACM Press, Canada, pp: 1149-1150. http://portal.acm.org/citation.cfm?id=1242739
2. Beneventano, D. and S. Magnani, 2004. A framework for the classification and the reclassification of electronic catalogs. Proceedings of the 2004 ACM Symposium on Applied Computing, Mar. 14-17, ACM Press, Cyprus, pp: 784-788. http://portal.acm.org/citation.cfm?doid=967900.968062
3. Ashraf, F. and R. Alhajj, 2007. ClusTex: Information extraction from HTML pages. Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, May 21-23, IEEE Xplore Press, Niagara Falls, Ont., pp: 355-360. DOI: 10.1109/AINAW.2007.119
4. Fatima Ashraf, Tansel Ozyer and Reda Alhajj, 2008. Employing clustering techniques for automatic information extraction from HTML documents. J. IEEE Trans. Syst., 38: 660-673. DOI: 10.1109/TSMCC.2008.923882
5. Fei, Hong, Zhuang and Zhao, 2006. Information extraction system in large-scale web. J. Commun. Inform. Technol., 2: 809-812. DOI: 10.1109/ISCIT.2005.1566990
6. Nachouki, G., 2006. A method for information extraction from the web. J. Inform. Commun. Technol., 1: 517-521. DOI: 10.1109/ICTTA.2006.1684424
7. Arnicans G. and G. Karnitis, 2006. Intelligent integration of information from semi-structured web data sources on the base of ontology and meta-models. Proceedings of the 7th International Baltic Conference, (IBC'06), IEEE Xplore Press, Vilnius, pp: 177-186. DOI: 10.1109/DBIS.2006.1678494
8. Thevenot, H.J., F. Alizon, T.W. Simpson1 and S.B. Shooter, 2007. An index-based method to manage the tradeoff between diversity and commonality during product family design. J. Concurr. Eng. Res. Appli., 15: 127-139. DOI: 10.1177/1063293X07079318
9. Zhao, H., W. Meng and C. Yu, 2007. Mining templates from search result records of search engines. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 12-15, ACM Press, San Jose, California, USA., pp: 884-893. DOI: 10.1145/1281192.1281286
10. Saggion, H., A. Funk, D. Maynard and K. Bontcheva, 2007. Ontology-based information extraction for business intelligence. Proceedings of the 6th. International Semantic Web Conference and the 2nd Asian Semantic Web Conference, United Kingdom, pp: 843-856. http://iswc2007.semanticweb.org/papers/837.pdf
11. Nanda, J., H.J. Thevenot, T.W. Simpson, R.B. Stone, M.T. Bohm and S.B. Shooter, 2007. Product family design knowledge representation, aggregation, reuse and analysis. J. Artifi. Intell. Eng. Des. Anal. Manufactur., 21: 173-192. DOI: 10.1017/S0890060407070217

12. Nanda, J., T.W. Simpson, S.R.T. Kumara and S.B. Shooter, 2006. A methodology for product family ontology development using formal concept analysis and web ontology language. J. Comput. Inform. Sci. Eng., 6: 1-11. DOI: 10.1115/1.2190237
13. Kaiser, K. and S. Miksch, 2007. Modeling treatment processes using information extraction. Stud. Computat. Intell., 84: 189-224. DOI: 10.1007/978-3-540-47527-9
14. Lam, M.I., Z. Gong and M. Muyeba, 2008. A Method for web information extraction. Lecture Notes Comput. Sci., 4976: 383-394. DOI: 10.1007/978-3-540-78849-2
15. Viola, P. and M. Narasimhand, 2005. Learning to extract information from semi-structured text using a discriminative context free grammar. Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information, Aug. 15-19, ACM Press, Retrieval, Brazil, pp: 330-337. http://portal.acm.org/citation.cfm?id=1076091
16. Vadrevu, S., F. Gelgi and H. Davulcu, 2007. Information extraction from web pages using presentation regularities and domain knowledge. J. World Wide Web, 10: 157-179. DOI: 10.1007/s11280-007-0021-1
17. Jung, S.W., K.H. Sung, T.W. Park and H.C. Kwon, 2001. Intelligent integration of information on the internet for travelers on demand. Proceedings of ISIE IEEE International Symposium, June 12-16, Pusin, Korea, pp: 338-342. DOI: 10.1109/ISIE.2001.931810
18. Eliassi-Rad, T. and J. Shavlik, 2003. A system for building intelligent agents that learn to retrieve and extract information. J. User Model. User-Adapted Interact. USA., 13: 58-88. DOI: 10.1023/A:1024009718142
19. Irmak, U. and T. Suel, 2006. Interactive wrapper generation with minimal user effort. Proceedings of the 15th International Conference on World Wide Web, May 23-26, ACM Press, Scotland, pp: 553-563. http://portal.acm.org/citation.cfm?id=1135859