Semantic Extraction from List Web Pages

Ismail JELLOULI[1] and Mohammed EL MOHAJIR[2]

**[1] Computer Science Department, Faculty of Science Dhar Mehraz
Fez, Morocco**

**[2] Computer Science Department, Faculty of Science Dhar Mehraz
Fez, Morocco**

Abstract

Extracting structured information from web pages is a problem that has many applications and that gained increased interest in recent years.

We propose an approach that can achieve extraction and semantic description of data contained in a list web page. Our approach is fully automatic and is based on a "seed" ontology that contains minimal information about the domain. It uses an instance-based classifier to characterize the attributes of the ontology. In opposition to existing methods, our approach does not make any assumption on the design of web pages ; it is totally layout independent.

Experimental results obtained from different web pages of different web sites from different domains show that our approach is effective.

Keywords- Web Information Extraction; list web pages, probablistic model, ontology

1. Introduction

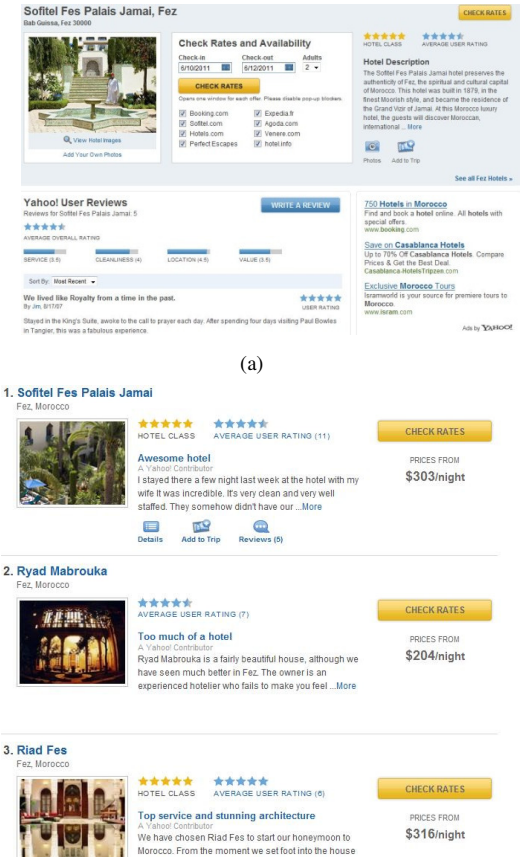
The Web is nowadays an invaluable source of information. This information is usually encoded in HTML for human users. Most of these HTML pages are rich data pages that are endorsed to backend databases and are generated dynamically in response to users' queries. Transforming these web pages into a structured form which may be an SQL like form or an RDF like form is becoming a hot research topic.

Actually, extracting data from the web also called Web Information Extraction (Web IE) has many applications. Web IE can allow software interrogate a web source using the SQL like query languages and hence feed a data integration system. It can also be used to gather information related to a specific topic from different web sites for data warehousing. When Web IE is guided by an ontology, it can be considered as a step towards the semantic web. In this case, IE is a process of semantic enrichment or annotation of web sources.

During the last few years, many web IE systems have been developed. The first ones were manual or semi-supervised while the new systems are in most cases automatic. Some of these systems can accomplish extraction from pages with a single record like the one in figure 1.a. whereas others are applied to pages with list of records (figure 1.b). IE from single record pages is called page level IE while IE from list pages is known as record level IE [5]. Web IE systems can also be categorized according to their output. Some of them produce unlabeled relational tables while others assign labels to the extracted attributes. We give more details on Web IE systems in the related work section.

This paper proposes an ontology-based approach to extract records from list pages. It starts by localizing the data region using an algorithm based on some heuristics, then it uses an instance-based classifier to achieve record and attribute alignment. Our approach is totally automatic and it achieves extraction at a semantic level.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of Web IE related work. Section 3 describes the overall architecture of our system. Section 4 explains our approach to localize the data region. Section 5 presents our instance-based classifying technique to assign labels to data values. Section6 explains our solution to segment the data region into records. Section 7 shows the experiments we conducted to test our approach and the results obtained and section 8 concludes the paper.



(a)

(b)

Figure 1: A sample of Web pages. (a) A segment of a single instance page. (b) A segment of a list of records page.

2. Related work

According to Chang et al. [5], IE systems can be classified into three categories: manually constructed IE systems, semi-supervised IE systems and unsupervised IE systems.

In manual IE systems, the focus is on developing rules oriented languages dedicated to IE. Building a wrapper using these languages is equivalent to the definition of a set of rules that can achieve extraction. These systems have, at least, two limitations. They depend on web pages templates and they require programming skills. Examples of such systems are TSIMMIS [11], Minerva [8], WebOQL [4], XWRAP [14].

Supervised IE systems also known as wrapper induction IE systems require a set of manually labelled training examples to induce extraction rules. Although these systems do not require programming, preparing training examples is tedious and time consuming and wrappers generated are template dependent. STALKER [16], WIEN [13], SoftMelay [12], WHISK [18], NoDoSE [1] are examples of these IE systems.

Unsupervised learning methods (or automatic IE systems) are state-of-the-art IE systems. These methods neither require labelled examples nor human intervention.

DeLa[20], ROADRUNNER [9], IEPAD [6], EXALG[3], TISP [19] and DEPTA [22] are part of such systems.

Most of these systems try to discover the repetitive patterns contained in the web pages to induce the wrappers. They assume that repetition indicates the existence of a list of records. Often, these systems do not assign labels to data extracted unlike to ontology based IE systems.

Ontology-based IE systems [21] can also be categorized as unsupervised. Embley et al. [10] developed one of the first IE systems based on ontologies. They used what they called extraction ontology. This ontology contains objects, relations among these objects and a syntactic description of these objects using regular expressions that were previously manually defined.

3. The architecture of our system

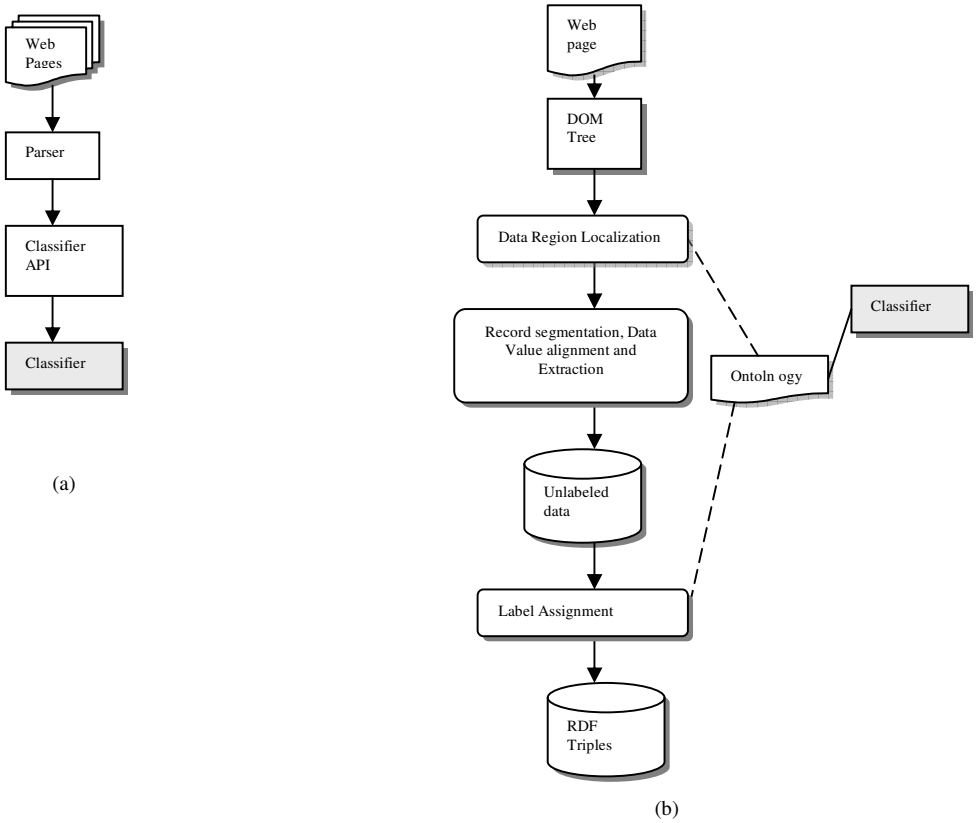


Figure 2: The architecture of our sysyem (a) The Off-line component (b) The On-line component

The input of the online component is web pages dynamically generated from backend databases. We focus on list pages that contain a set of records. Records are considered as instances of a concept of the ontology. The output is a set of RDF triples describing the content of the pages in conformance with the ontology.

The page is first transformed into a DOM tree then the data region inside the page is located. Once the data region is localized, records segmentation, data value alignment and data extraction are achieved using some heuristics and the classifier that was built by the offline component. Data values extracted are then labelled according to the vocabulary of the ontology.

Further details on the components of our system are given in the next three sections.

4. Data region localization

The data region is a zone of the page that contains the list of data to be extracted. Visually, it is situated in the centre and it occupies an important area. Structurally, the data zone is a subtree of the page DOM tree that can be identified by a root node. Hence, localizing the data zone is equivalent to recognizing this node.

For pages with a list of records, Alvarez et al. [2] made the following two observations:

Observation 1. Each record in the DOM tree is composed of a set of consecutive sibling subtrees.

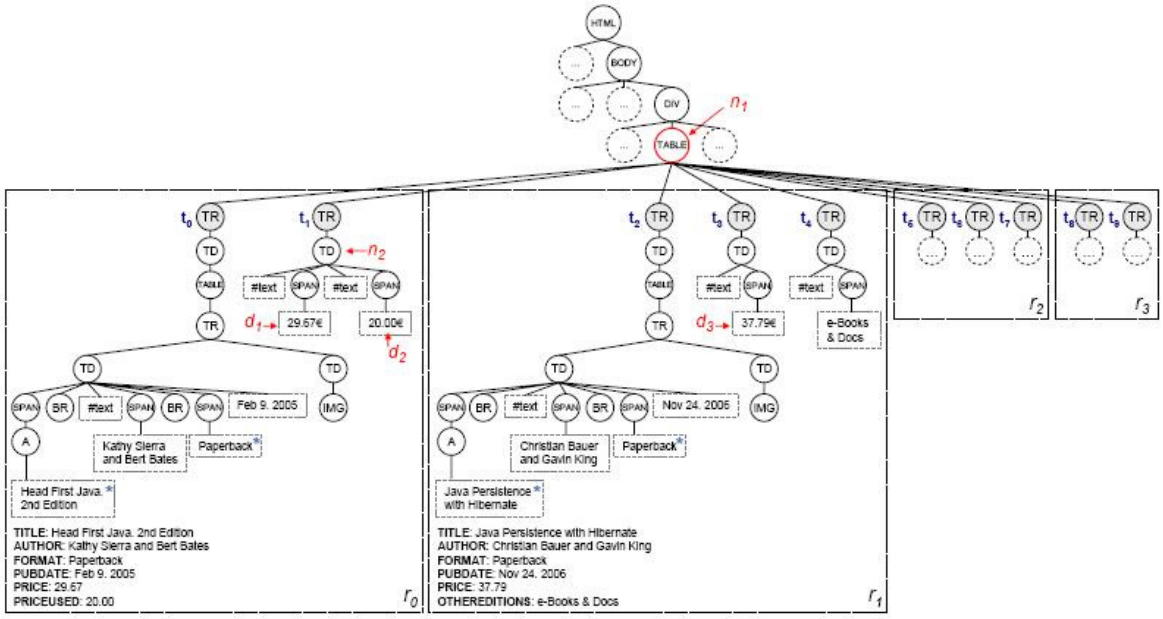


Figure 4: Example of a page DOM tree

Observation 2. The occurrences of each attribute in different data records share the same path from the root of the DOM tree.

Starting from these observations, data region localization becomes equivalent to finding the common parent node of sibling subtrees composing the data records. The algorithm proposed is as follows:

1. ***Let N be the set of all nodes in the DOM tree of the page. To each node n_i, a score s_i is assigned. Initially $s_i = 0$***
2. ***Let T be the set of all text nodes in N***
3. ***Divide T into subsets. Each subset will contain text nodes with the same path from the root.***
4. ***For each pair of text nodes of the same subset, compute n_j as their deepest common ancestor in the DOM tree. Add 1 to s_j***
5. ***n_{max} the node with the higher score is the root of the data rgion. p***

Figure 3: Basic algorithm

The justification of this algorithm is as follows [2]:

1. By observation 1, we can infer that if text nodes are occurrences of attributes in different records then their deepest common ancestor is n_1 as in figure 4. This is the case for nodes d_1 and d_3 for which the deepest common ancestor is n_1. In this case the score of n_1 will be increased.
2. For text nodes that are occurrences of the attributes of the same record, the deepest common ancestor may be deeper than the actual root of the data region. For example, in the figure n_2 is the deepest common ancestor of d_1 and d_2. Even in this case, the algorithm still can recognize the right data region root. In fact, if we consider a pair of text nodes (t_{11}, t_{12}) from the same record. Their deepest ancestor is deeper than the node we are looking for but for all pairs (t_{11}, t_{i1}), $(t_{11},t_{i2}),(t_{12},t_{i1}),(t_{12},t_{i2})$, the deepest common ancestor is the data region root. This leads to a high score for this node.

When the number of records is very small, the algorithm fails to determine the root of the data region. To overcome this limitation, [2] propose to use the information contained in the query web form. For instance, if the query contains the clause "title contains 'java' and format equals 'paperback'" then the only text nodes considered would be those marked with * in the figure.

Differently from [2], we aim not only to extract but also to label the data. Furthermore, we are not necessarily aware of the request forms that generate the list pages since we also consider web pages that were gathered by a crawler via a simple link. However, we have some knowledge about these pages contained in a "seed" ontology that we define more precisely in the following section.

4.1. The ontology

In our case, the ontology is not a complex and well defined ontology like that proposed by Embley [10]. It is rather a "seed" ontology centred on a main concept with few attributes and their aliases. It is derived from a small

number of domain related web sites. Thanks to the "amazon" effect [7] which states that the vocabulary of web databases is usually a small and common vocabulary, such an ontology can efficiently express the content of web pages.

Figure 5 shows an example of the main concept of an ontology used in this paper which describes the concept hotel and its main attributes. A *hotel* has the attributes; *name*, *address*, *city*, *country*, *price*, is rated in a *class*, located in a *city* and offers some *amenities*. In the figure rectangles symbolize attributes and ellipses depict other concepts. The filled arrow indicates that a hotel is a kind of an accommodation.

Figure 6 shows an example of the hotel attributes and their associated aliases.

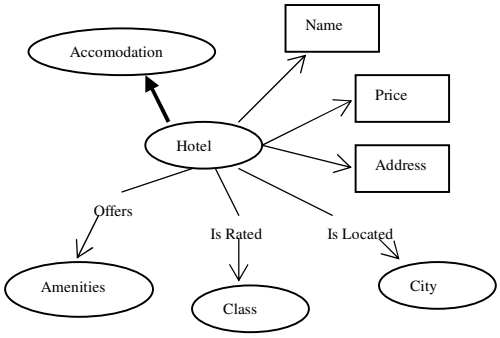


Figure 5: The concept *hotel* of the *travel* ontology

```
name : name,hotel name
descrption : description, hotel description, features
address : address, hotel address, location, locality
city : city, locality
country : country, state
class : class, hotel class, category, hotel category
price : price, price per night, single price
hotel : hotel, guesthouse, auberge
services : facilities, amenities,  room amenities, hotel amenities
service : amenity, facility
```

Figure 6: An example of attributes names and aliases for the concpet hotel

4.2. Our algorithm

Our algorithm for data region localization is based on that of figure 3. But, rather than returning directly the subtree rooted in n_{max}, we compute a correlation measure of that subtree to the ontology. The subtree is returned only if this correlation is higher than a previously fixed threshold. Otherwise, we choose the subtree rooted in the node with the highest score excepting n_{max} and we recalculate the correlation measure. This operation is repeated while the threshold is not attained. The algorithm fails if no subtree is sufficiently correlated to the ontology. Figure 7 shows an extract of our algorithm.

1. ***Let N be the set of all nodes in the DOM trees of the page. To each node n_i , a score s_i is assigned. Initially***
2. ***Let T be the set of all text nodes in N***
3. ***Divide T into subsets. Each subset will contain text nodes with the same path from the root.***
4. ***For each pair of text nodes of the same subset, compute n_j as their deepest common ancestor in the DOM tree. Add 1 to s_j***
5. ***For each node n_i in N sorted in descending order according to their respective scores s_i. If the correlation of the subtree rooted in n_i with the ontology is higher than a threshold T_o, then return n_i.***

Figure 7: Modified algorithm

The correlation is calculated according to the following formula :

$$corr = \frac{F}{\sqrt{m.n}}$$

Where F is the count of attributes and attribute aliases identified in the subtree, *m* is the size of the ontology i.e. the count of attributes and *n* is the size of the subtree i.e. the count of the text nodes. Attribute names and aliases are identified using the dictionary of figure 7 whereas attribute values are recognized using the classifier learned by the offline component. Details on this classifier are given in the following section.

This algorithm, compared to [2] has, at least, two advantages:

1. It reinforces the data region localization thanks to the correlation measure;
2. It is able to recognize data region with few records;

5. Data alignment and extraction

The use of the ontology as a guidance towards the extraction eases considerably the process of extraction and data alignment. In fact, as we have already explained, our system incorporates a classifier component that assigns labels to text nodes. These labels are the names of the attributes of the ontology.

5.1. The classifier

Given a text node in HTML, the problem is to find the class/label of this text node.

The classifier is based on a probabilistic model. In a probabilistic model, the class is a random variable *y* that is valued from an alphabet *Y*. In our case, it is the list of attributes of the ontology. The probabilistic model we use here is Maximum Entropy Model. The aim of Maximum Entropy Model is to estimate the probability of having a value *y* given an observation *x* i.e. $p(y|x)$. This is a conditional probability model. Models that estimate such a probability are called discriminative

models in opposition to generative models that estimate the conjoint probability *p(x,y)*. These last models are called generative because they permit to generate random processes.

Maximum Entropy Model is the discriminative model derived from the Naïve Bayes Classifier which is generative. The Naïve Bayes classifier assumes that observed variables $x=(x_1,x_2,\ldots,x_n)$ are independent given the class *y* i.e.:

$$p(y,X) = p(y)p(X|y) = p(y).\prod_{k=1}^{n} p(x_k|y)$$

This model estimates the conjoint probability and hence requires modelling observations *x*. This is usually intractable. Discriminative models make no assumption about observation and hence are easier to estimate.

Maximum Entropy Model assumes that the conditional probability is log-linear i.e. *log(p(y|x))* is a linear function of *x*. In this case, the conditional probability is expressed as follows:

$$p(y|X) = \frac{1}{Z(X)} \exp\left\{\lambda_y + \sum_{j=1}^{K} \lambda_{y,j} x_j\right\}$$

Where

$$Z(X) = \sum_y \exp\left\{\lambda_y + \sum_{j=1}^{K} \lambda_{y,j} x_j\right\}$$

is a normalizing constant and λ_y is a bias that acts like *log(p(y))* in the Naïve Bayes.

The model can be expressed, in a more compact way as:

$$p(y|X) = \frac{1}{Z(X)} \exp\left\{\sum_{j=1}^{K} \lambda_k f_k(y,X)\right\}$$

Where

$$f_k(y,x) = 1_{\{y'=y\}} x_j$$

f_k is a feature function which is equal to 1 for certain values of *y* and *x*. For example a feature will capture the fact that *y="address"* when x_1*="Avenue"*.

Learning the model is equivalent to finding the values of the parameters

$$\Lambda = \{\lambda_1, \lambda_2, \ldots \lambda_K\}$$

of the model.

Maximum Entropy means the model should preserve as much uncertainty as possible with regard to training examples that are considered as constraints. The entropy was introduced by Shannon's information theory [17]. It measures the randomness of a message or the uncertainty of an event. In our case, entropy is expressed as follows:

$$H(Y|X) = -\sum_{x\in X} p(x) \sum_{y\in Y} p(y|x) \log(p(y|x))$$

So the aim is to find the distribution p* that maximizes H(p) i.e.:

$$p^* = \arg\max_{p\in P} H(p)$$

This is a constrained optimization problem which is resolved by numerical methods like gradient ascent or conjugate gradient. [15].

5.2. Our implementation of the model

In our system, a text node is considered as an observation composed of a set of random variables $X=\{x_1,x_2,\ldots x_n\}$ where *n* is the number of tokens in the text and x_j is the j^{th} token of the text. For example, if the node text is *« 25, Avenue Mohamed V, Fès »* then x_3 = *"Mohamed"*. The model should decide this is an address i.e.:

p(y="address"|x="25, Avenue Mohamed V, Fès ") is greater than all *p(y=y'|x=25, Avenue Mohamed V, Fès")* where y' belongs to the list of attributes (vocabulary) of the ontology (figure 5).

6. Data record segmentation

In most cases, the data region localized should be segmented to records. If it contains *n* text nodes, segmentation possibilities are equivalent to the number of partitions of the set of these text nodes. This number B_n, known as the Bell number, can be estimated as follows:

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

When *n* is quite large, it is out of practice to enumerate all these possibilities.

To overcome this limitation, we utilize the labels associated to text nodes by the offline classifier. In fact, the classifier invoked during the data region localization step has attributed labels to text nodes. The result is a set text nodes labelled with names of the ontology attributes. Ideally, each label should be repeated as many times as the number of records contained in the web page. In practice, some text nodes may be 'mislabelled' and some attributes may be optional in the records. As a result, the frequencies of different labels may vary.

Our solution is based on the following heuristic:
The records of a data region have almost the same structure and start by the same attribute.

Based on this observation, a possible segmentation is equivalent to choosing the starting attribute; each time this attribute is encountered, it is the beginning of a new record. Figure 8(b) shows an example the possibilities of segmentation of the data zone of figure 8(a)

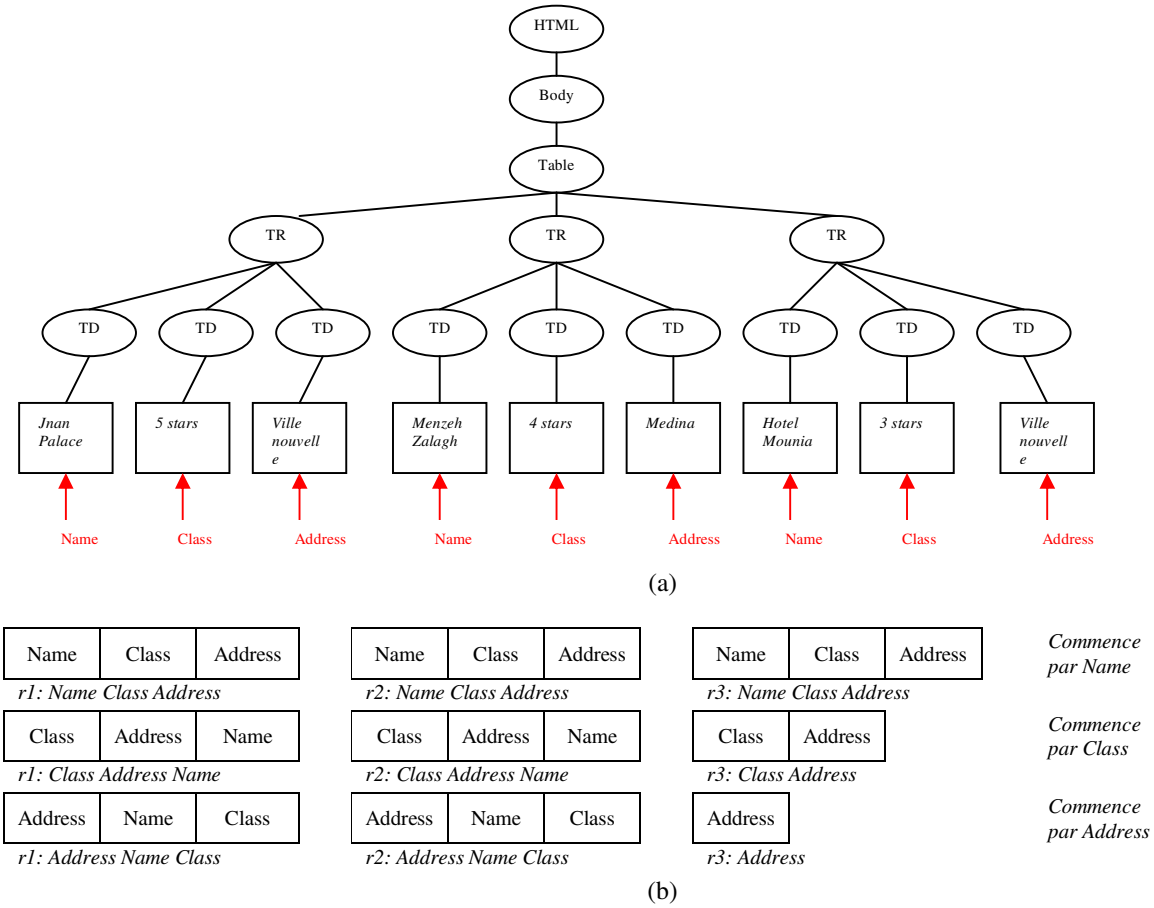


Figure8 : (a) Exemple d'arbre DOM avec nœuds texte étiquetés (b) les segmentations posibles de cet arbre DOM

Once the possibilities enumerated, we choose the best segmentation among them. Since the records are similarly structured, the best segmentation is the one which returns the most similar records. This is measured using an auto-similarity measure expressed as follows:

$$\frac{2\times\sum_{i\neq j} sim(r_i, r_j)}{n\times(n-1)}$$

Where r_i et r_j are strings composed by the labels of the attributes of the record separated by spaces (figure 8(b)). The *sim* function is a Jaccard similarity measure. In the figure, we can see that the first segmentation returns the best similarity measure (equal to 1).

7. Experimentation

To test our solution, we have conducted experiments over its main components: data region localization, segmentation and extraction.

7.1. Datatsets

Our datasets cover three different domains: hotels, books and movies. Data are issued from web pages containing lists of records. Table 1 shows the dispatching of these pages on the three domains:

Table 1: pages of the dataset

Domain	Hotels	Books	Films
# pages	60	60	60

We have extracted relevant information from these pages using wrappers that were adapted to each class of pages. Table 2 gives the number of instances extracted for each domain:

Table 2: Instances of the dataset

Domain	Hotels	Books	Films
# instances	1772	715	1486

The web pages will be exploited to test our solution for data region localization and segmentation whereas instances will be given as input for the learning and testing of the classifier.

7.2. Data region localization

We identified manually the root nodes for the data regions of the dataset pages. This was not achieved for pages one by one since pages issued from the same web site share the same template and hence the process was carried out once for a "class" of pages. Then, we have applied our algorithm of figure 7 to determine root nodes that were compared to those identified manually.

Test procedure

Input :

1. Web pages of the three domains
2. Domain classifiers

Procedure:

1. Identify manually the root node of the data region ;
2. Identify the root node using the algorithm (figure 7) and the ad hoc classifier ;
3. Compare, for each page, nodes manually identified with those identified by the algorithm ;
4. Calculate the recall and precision ratios.

Comments:

Classifier: we suppose that the domain classifier was previously learned. It is invoked by the algorithm in order to assign labels to text nodes of the data region. These labels are then used to calculate the correlation to the ontology.

Manual identification of root nodes: as noticed, pages from the same web site share the same template. Therefore, we can identify root nodes for a class of pages at a time. There is no need to repeat this operation for each page.

Identification of the root node using the algorithm: During the experimentation, the correlation threshold was varied in order to estimate the value that leads to the best results. This value is, of course, domain dependent.

The algorithm identified correctly 175 root nodes over the 180 pages of the dataset meaning a 0.97 recall rate.

7.3. Data region segmentation

Once the data region localized and labels assigned to text nodes, we run the algorithm of data segmentation into records.

Procedure

1. Identify manually the best segmentation ;
2. Run the algorithm to propose a segmentation ;
3. Compare the manual segmentation with the one proposed by the algorithm ;
4. Calculate the precision ratio.

For an auto-similarity threshold of 0.8, our solution segmented correctly 173 pages implying an efficacy of 96%.

7.4. Extraction and labelling

We use the probabilistic model explained in section 5 to learn the classifiers. The datasets were divided into training and testing collections as shown in table 5.

Table 3: Classifier trainnig and testing datasets

	Train	**Test**
	#instances	**#instances**
Hotels	920	852
books	410	305
Films	950	536

Learning procedure

- Identify manually the data region ;
- Extract all text nodes in the data region ;
- Assign labels to text nodes containing ontology instances, mark the other nodes as insignificant for extraction ;
- Generate the input files (train and test) for the Mallet API conforming to the format of figure 9.

```
472 o Price not available
473 o Check Rates
474 rating Hotel Class:
475 address Route de Targa, Lot. Farida, Villa 64 Marrakech 40000
Morocco
477 desc Villa &#39;Dar El Kanoun&#39; is a guest house built in the
traditional Moroccan style. Renovated with passion by its owners and
decorated with Moroccan...
479 o more
480 o Amenities
481 amenities Babysitting, Restaurant in Hotel...
482 o see all
483 o SPONSOR LINKS
484 o Villa Dar El Kanoun Deals
485 o Check Price and Availability
486 o Check-in
487 o Check-out
```

Figure9. Extract from a Mallet input file

The procedure above is applied for both learning and testing web pages. We have learned and tested a classifier for each of the three domains.

Table 6 shows the results of these tests :

Table 6: classifier precision and recall rates

Domain	Precision	Recall
Hotels	0.98	0.96
Books	0.97	0.92
Films	0.96	0.88

The precision and recall rates are calculated as a main value of the corresponding rates of the attributes of each domain respectively.

8. Conclusion

In this paper, we have proposed a solution for semantic extraction from list web pages. We start by localizing the data region using an algorithm that discovers the repetitive patterns in the DOM tree. To reinforce the result returned by this algorithm, we introduced a correlation measure that estimates how the zone returned is related to the ontology. We have also proposed a solution for data segmentation based on the labels assigned by a domain classifier.

In order to evaluate the accuracy of our solution, we conducted a test over three different domains. The results of this test show that our solution is efficient.

As a perspective of this work, we foresee to integrate a reference reconciliation module that will enable us to achieve conjoint extraction from different web pages and the enrichment and/or validation of extracted data by new data from new web pages.

References

[1] Adelberg, B., NoDoSE: A tool for semi-automatically extracting structured and semi-structured data from text documents. SIGMOD Record 27(2): 283-294, 1998.

[2] Manuel Álvarez, Alberto Pan, Juan Raposo, Fernando Bellas, Fidel Cacheda: Extracting lists of data records from semi-structured web pages. Data Knowl. Eng. 64(2): 491-509 (2008)

[3] Arasu, A. and Garcia-Molina, H., Extracting structured data from Web pages. Proceedings of the ACM SIGMOD International Conference on Management of Data, San Diego, California, pp. 337-348, 2003.

[4] Arocena, G. O. and Mendelzon, A. O., WebOQL: Restructuring documents, databases, and Webs. Proceedings of the 14th IEEE International Conference on Data Engineering (ICDE), Orlando, Florida, pp. 24-33, 1998.

[5] C.-H. Chang, M. Kayed, M.R. Girgis, and K.A. Shaalan, "Survey of Web Information Extraction Systems," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 10, pp. 1411-1428, Oct. 2006.J.

[6] Chang, C.-H. and Lui, S.-C., IEPAD: Information extraction based on pattern discovery. Proceedings of the Tenth International Conference on World Wide Web (WWW), Hong-Kong, pp. 223-231, 2001.

[7] Chang, K. C.-C., He, B., Li, C., Patel, M., Zhang, Z. 2004. Structured databases on the Web:Observations and implications. *SIGMOD Record 33*, 3, 61-70.

[8] Crescenzi, V., and Mecca, G., Grammars have exceptions. Information Systems, 23(8): 539-565, 1998.

[9] Crescenzi, V., Mecca, G. and Merialdo, P., RoadRunner: towardsautomatic data extraction from large Web sites. Proceedings of the 26th International Conference on Very Large Database Systems (VLDB), Rome, Italy, pp. 109-118, 2001.

[10] EMBLEY, D.W., CAMPBELL, D.M., JIANG, Y.S., LIDDLE, S.W., LONSDALE, D.W., NG, Y.-K., AND SMITH, R.D. 1999. Conceptual-model-based data extraction from multiple-record web pages. *IEEE Trans. on Data and Knowledge Engineering 31*, 3, 227-251.

[11] Hammer, J., McHugh, J. and Garcia-Molina, Semistructured data: the TSIMMIS experience. In Proceedings of the 1st East-European Symposium on Advances in Databases and Information Systems (ADBIS), St. Petersburg, Rusia, pp. 1-8, 1997.

[12] Hsu, C.-N. and Dung, M., Generating finite-state transducers for semi-structured data extraction from the web. Journal of Information Systems 23(8): 521-538, 1998.

[13] Kushmerick, N., Weld, D., and Doorenbos, R., Wrapper induction for information extraction. Proceedings of the Fifteenth International Conference on Artificial Intelligence (IJCAI), pp. 729-735, 1997.

[14] Liu, L., Pu, C., and Han, W. XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources, Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE), San Diego, California, pp. 611-621, 2000.

[15] Malouf Robert (2002). A comparison of algorithms for maximum entrop parameter estimation. In proceedings of the sixth conference on Natural Language Learning (CoNLL-2002) pp 49-55 San Francesco CA, Morgan Kaufmann.

[16] Muslea, I., Minton, S., and Knoblock, C., A hierarchical approach to wrapper induction. Proceedings of the Third International Conference on Autonomous Agents (AA-99), 1999.

[17] Shannon Claude E (1948). A mathematical theroy of communication. Bell System Technical Journal, 27, 379-423, 623-656.

[18] Soderland, S., Learning information extraction rules for semi-structured and free text. Journal of Machine Learning, 34(1-3): 233-272, 1999.

[19] Tao, C., Embley, D.W. 2007. Automatic hidden-web table interpretation by sibling page comparison. In Conceptual Modeling - ER 2007, Lecture Notes in Computer Science, vol. 4801/2008, Springer Berlin/Heidelberg, 566-581.

[20] Wang, J. and Lochovsky, F. H., Data extraction and label assignment for Web databases, Proceedings of the Twelfth International Conference on World Wide Web (WWW), Budapest, Hungary, pp. 187-196, 2003.

[21] Wimalasuriya Daya C, Dejing Dou: Ontology-based information extraction: An introduction and a surbey of current approaches. Journal of Information Science, 36(3), pp 306-323, 2010

[22] Zhai, Y. and Liu, B. Web Data Extraction Based on Partial Tree Alignment. Proceedings of the 14th International Conference on World Wide Web (WWW), Japan, pp. 76-85, 2005.

Ismail JELLOULI DESA degree in computer science in 2007, currently a Ph D student in computer science. He is an IEEE student member and his research interests include information extraction, semantic web and reference reconciliation.

Mohammed EL MOHAJIR is European Master in Environmental System Modeling (1992) and Doctor of Science (1997). He is Professor at the department of computer sciences at the Faculty of Science Dhar Mahraz. He is the vice-chair of the IEEE Morocco Section. His main research is about conceptual modeling, design and development of decision-support Information Systems, ETL processes for datawarehouse and SOLAP, Distributed and Parallel Processing Systems and semantic web.