Università degli Studi dell'Insubria

Facoltà di Scienze MM.FF.NN.

Dottorato di Ricerca in Informatica

Web Content Mining with Multi-Source Machine Learning for Intelligent Web Agents

Advisors:

Prof. Elisabetta Binaghi - *Università degli Studi dell'Insubria*

Prof. Fabio Crestani - *Università della Svizzera Italiana*

Candidate:

Moreno Carullo - Matr. 608371

To my family

Abstract

The web is recognized as the largest data source in the world. The nature of such data is characterized by partial or no structure, and even worse there exist no standard data schema for the even low-volumed structured data. Web Mining aims to extract useful knowledge from the Web by using a variety of techniques that have to cope with the heterogeneity and lack of a unique and fixed way of representing information.

An important aspect in Web Mining is played by the automation of extraction rules with proper algorithms. Machine Learning techniques have been successfully applied to Web Mining and Information Extraction tasks thanks to the generalization and adaptation capabilities that are a key requirement on general content, heterogeneous web pages.

The World Wide Web is a graph, more precisely a *directed labeled graph* where the nodes are represented by the pages and the edges are represented by links between them. Recent works propose the exploitation of the web structure (Link Analysis) for content extraction, for example one can leverage the content category of neighbor pages to categorize the contents of difficult web pages where word-frequency-based techniques are not robust enough.

In this thesis we propose an automated method suitable for a wide range of domains based on Machine Learning and Link Analysis. In particular we propose an inductive model able to recognize *content pages* where structured information is located after being trained with proper input data. In order to keep the recognition speed high enough for real-world applications an additional algorithm is proposed which lets the approach to boost both in speed and quality. The

proposed method has been tested with controlled dataset in a classic train-and-test scenario and in a real-world web crawling system.

Acknowledgements

Ideas and motivations behind this thesis have emerged after research works conducted together with the research group I am part of, namely Prof. Elisabetta Binaghi and Ignazio Gallo. Earlier works on closely related areas, such as Document Clustering [Frakes and Baeza-Yates, 1992], Named Entity Recognition [Nadeau and Sekine, 2007] and Text Categorization [Sebastiani, 2002] have contributed to the development of the proposed approach. Huge thanks to Prof. Fabio Crestani for his contribution to the development of the thesis.

This research project has been funded by and contributed by 7Pixel, a company that owns and runs leader price comparison services in Europe (`http://www.trovaprezzi.it` and `http://www.shoppydoo.com` are the two major brands).

The development of the real-world web crawling system would not have been possible without the continued effort of the 7Pixel R&D Team, in particular Alessandro and Roberto. Thank you.

Varese, December 2010

Moreno Carullo

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The web is recognized as the largest data source in the world. The nature of such data is characterized by partial or no structure, and even worse there exist no standard data schema for the even low-volumed structured data. Web Mining [Kosala and Blockeel, 2000] aims to extract useful knowledge from the Web by using a variety of techniques that have to cope with the heterogeneity and lack of a unique and fixed way of representing information. According to [Kuhlins and Tredwell, 2003] about 80% of websites found on the web are backed by databases: unfortunately the presentation phase of the web application front ends hides these structured sources. Web (Content) Mining can be seen as the reverse of this process.

The need of data extraction techniques suitable for focused topics is crucial for the building and maintenance of intelligent crawlers and web agents oriented to advanced web-based user services like general search engines, price comparison services etc. Structuring and understanding the web of data is the key for the development of the Semantic Web [Berners-Lee et al., 2001].

Web Mining is a vivid research area closely related to Information Extraction (IE). In fact one can view the Web Mining as a part of (Web) IE and vice versa, and following the considerations in [Kosala and Blockeel, 2000] Web Mining has to be considered a part of IE. In this thesis in particular we consider Web Content Mining and Web Information Extraction as synonyms as the relevant works on these topics are closely related. IE has been heavily funded by the U.S. government, beginning with the Message Understanding Conference (MUC) [NIST, b]

Conferences and continuing with the Automatic Content Extraction (ACE) [NIST, a] Evaluation project. Input for classic IE tasks can be unstructured documents like free text written in natural language or semi-structured documents like web pages.

An important aspect in Web Content Mining is played by the automation of extraction rules with proper algorithms. Machine Learning (ML) techniques have been successfully applied to Web Mining [Kosala and Blockeel, 2000] and Information Extraction tasks thanks to the generalization and adaptation capabilities that are a key requirement on general content, heterogeneous web pages. Supervised, unsupervised [Duda et al., 2000] and semi-supervised [Zhu, 2005] approaches to ML are largely used depending on the context and the availability of supervised data.

1.1 Real World Scenarios

The proposed method is targeted to a wide range of semantic data extraction from web sources, though in this research work two of them are considered: news harvesting and commercial offers. Moreover an extensive in-field evaluation and test will be developed for the latter by employing a crawler with data coming from the price comparison website ShoppyDoo [1].

On-line newspapers and news portals provide a huge amount of semi-structured information: title, abstract, body and main image are the most frequent data found on such websites. Once again there are thousands of sites with very different formats and no standard mean to aggregate contents, apart of feed formats like RSS [2] that are often implemented exposing partial information, and moreover do not provide a standardized way of retrieving past entries. For these reasons news harvesting is still an interesting and challenging task. Automatic techniques for news harvesting have been investigated in recent works [McKeown et al., 2002, Reis et al., 2004] with satisfactory performance, though with domain-oriented approaches.

The commercial offers domain deals with retrieving data from e-commerce websites for automatic product and price comparison services. In particular a given e-commerce website is viewed as a collection of product offers each with product name, relevant image, price, descrip-

[1] `http://www.shoppydoo.com`

[2] `http://validator.w3.org/feed/docs/rss2.html`

tion, technical details, etc. As with the news harvesting domain, target websites have different structures and provide no standardized mean of collecting data.

1.2 The Proposed Approach

In this work we propose a method for Web Content Mining suitable for a wide range of websites within a given topic. In particular we address the Web Content Mining problem by designing a model able to extract data from any website of a given domain by mimicking the visual recognition process performed by humans when browsing the web. This approach is enabled by the employment of Machine Learning models and the combined use of Web Content Mining and Web Structure Mining techniques.

The visual recognition ability is permitted by the use of a web rendering technology. In contrast to other known approaches in literature where only the page source is downloaded, in this way every additional resource of the page (images, scripts, etc) is needed and thus the overall download and recognition time increases significantly. Since this is absolutely required for the algorithm to reconstruct all the visual cues, an additional second phase is added to the process where a static wrapper can be built for each considered website allowing fast and accurate data extraction.

The proposed solution is then experimentally evaluated considering the two aforementioned application domains. A real-world case study has also been performed in the e-commerce scenario through the development of distributed set of intelligent web crawling agents. The case study has been used both for adjustments of the approach and the experimental evaluation of critical performance aspects.

1.3 Structure

The remainder of this thesis is structured as follows:

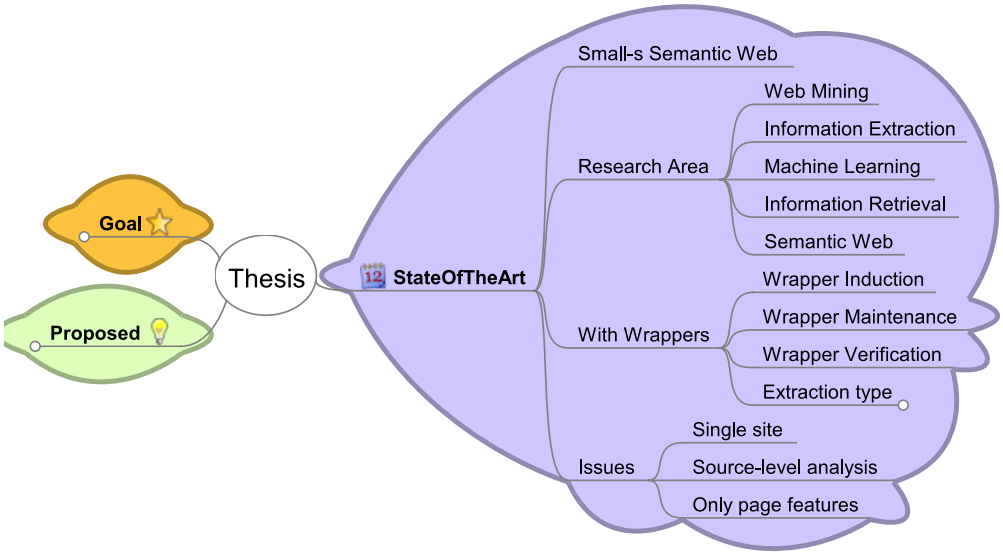
- chapter 2 proposes a state of the art of Web Content Mining techniques together with a coverage of relevant Machine Learning literature tailored for such problems. We focus on

the isolation of successful techniques and potential improvement areas.

- chapter 3 details the proposed Machine Learning-based Web Content Mining solution.
- chapter 4 shows the experimental assessment of the proposed approach. Both quantitative and qualitative considerations are drawn looking at controlled datasets results and a real-world case study.
- chapter 5 overviews conclusions and identifies future works.

Chapter 2

State of the Art



In this section we present a state of the art of Web Content Mining (WCM) techniques. Such techniques stems from the necessity to automatically extract huge amounts of data from the web by circumventing the main problem of the World Wide Web: it has been developed for human beings and does not provide a unified method to access the underlying semantic structure. A fundamental objective of WCM is thus the mapping of a set of webpages to a structured data table (figure 2.1). Strong and different efforts in defining common formats for the exchange of machine-understandable data have been proposed, however the standardization and adoption of such techniques has still a long way ahead. Moreover, there will always exist website owners

Figure 2.1 – *A fundamental objective of Web Content Mining is to map a set of web pages to a structured data table.*

not willing to distribute their structured data for free on the World Wide Web. Further details on structured data formats are reported in section 2.1.

Data extraction programs that collect information from various Internet sources are a necessity in many domains. Newswires aggregators need to collect items from various structured and unstructured sources, group information and show it to the user automatically by checking online sources. Price comparison services need to automatically seek and update price offerings of millions of products from the web - this would not be possible without an automatic data collection method. Automatic online monitoring of competitors is another interesting task that can be performed with the help of intelligent data extraction agents [Kuhlins and Tredwell, 2003].

2.1 Structured Data Formats for the Web

The evolution of the World Wide Web from a loosely structured human understandable data source to a structured and machine understandable data source is at the basis of the Semantic Web [Berners-Lee et al., 2001]. The first step in the addition of semantics to web resources is to define a way to represent data items defined in the resources their self. This first phase has also been named small-s Semantic Web. Further developments include the definition of and employment of ontologies to relate data.

```
[...]
<div id="produit-etendu" class="hproduct">
<div id="produit-etendu-89" class="item">
   <table width="100%"><tbody>
      <tr>
         <td style="vertical-align:top;">
            <div id="produit-etendu-image">
                                 <img src="produits-foie-gras/roti-de-magret-G.jpg"
                                    class="photo" border="0"
                                    alt="R&#244;ti de magret du Sud-ouest"><br>
            </div>
         </td>
         <td style="vertical-align:top;">
               <div id="produit-etendu-caracteristiques">
               <h1 id="produit-etendu-titre" class="fn">
                  Rôti de magret du Sud-ouest
               </h1>
               <div id="produit-etendu-description" class="description">
                  <h2>au piment d'Espelette</h2>
<P>Ce rôti de magret de canard déjà assaisonné d'un mélange d'épices est prêt à cuire :
pratique pour ne plus se poser la question sur <strong>comment cuisiner le roti de
magret de canard</strong>. Dans un four chaud thermostat 7 (210°C), laisser le rôti
de magret pendant 35 minutes.</p>
</DIV>
</div>
               <div id="produit-etendu-prix-achat">
                  <table  border="0" cellspacing="0" cellpadding="0" >
                     <tr>
                        <td width="44" height="23" class="red prix-red">
<span class="price"><span class="value-title" title="19.65 EUR"><font size="4">19</font>
<font size="3"><sup>&euro;65</sup></font></span></span>
                        </td>
                        <td style="vertical-align:middle;" valign="middle">
[...]
```

Figure 2.2 – *An example of microformats (hproduct) annotation of an e-commerce web page. Cascading Style Sheet classes are used to locate columns of a data record into an HTML page.*

The first phase of the evolution on a large scale has already been started. Several formats have been defined: RDFa [W3C., 2008], microformats, eRDF, and HTML5 microdata. These data description methods have different peculiarities [Adida, 2008], however the main idea is to add nodes or attributes to the XHTML [W3C., 2010] or HTML [W3C., 1999a] tree to describe data in a structured way. The most complete and extensible method is RDFa, although microformats - an example annotation of XHTML is reported in figure 2.2 - are deployed in a wider range of websites.

The adoption of microformats is expanding on the web: “hundreds of millions” of pages are

said to be available on the net according to [Adida, 2008]. However the definition of different formats for different application domains and the subsequent adoption of such formats from publishers is a long process. In particular, in order to collect up-to date statistics, we have analyzed the adoption of the *hproduct* microformat on 864 European websites at august 2010, with the result that only two websites (0.2% of the total) supported the microformat. Though structured data interchange is the solution for the future, there is plenty of information on the web that still needs intelligent and adaptive solutions able to fill the gap between scarcely structured data and the need to process such information automatically.

2.2 Wrappers and Wrapper Induction

Information Extraction is the research area devoted to the process of automatically extracting structured data from unstructured data, e.g. structured elements from free text. Web Mining is a vivid research area closely related to Information Extraction (IE). In fact Web Mining can be seen as a part of IE or vice versa, and following the considerations in [Kosala and Blockeel, 2000] Web Mining has to be considered a part of IE. In this proposal in particular we consider Web Content Mining and Web Information Extraction as synonyms. The application of Information Extraction techniques to web pages requires the building of dedicated software named Wrappers that allow to consider data of interest within a website in the form of a structured table.

Wrappers can be generated with various degrees of automation with Wrapper Inductors (WI) [Kushmerick, 1997, Kuhlins and Tredwell, 2003] that induce the generality of extraction rules by analyzing web pages of a given web resource.

Two main surveys can be found in literature on this broad topic, the first [Laender et al., 2002] classifies wrapper methods into categories depending on how they represent a web page and how they recognize the data of interest. The second [Chang et al., 2006] in particular focuses on how the wrapper induction works, namely with total supervision, no supervision or partial supervision by an expert of the domain. Other similar surveys are available in literature [Fiumara, 2007].

Considering these surveys of recent works, Wrappers and Wrapper Inductors have been classified in many ways in literature. We summarize the most relevant aspects that have been

taken into account by the majority of researchers:

- the *nature* of target data. Desired data can be structured items to be discovered across the site, relations between items or both. The extraction of structured items is a first required step for more complex inferences: in this thesis we focus on this phase.

- the *representation* of documents, that is whether or not web documents are considered as plain text, formatted text, or whatever. The exploitation of the whole set of information provided in the web page can be critical in the building of complex recognition models. In section 2.2.1 further details have been analyzed.

- the *scope* of data extraction, that is which part(s) of the website are considered in the process. According to [Sarawagi, 2002] Wrappers can be conceived at page level, where each record of the target data table is contained within a *content page* of a website, at record level, where each page can contain multiple records and at site level, where all pages within a website contribute to a single record of the target data table.

- the *level of automation*, that is the amount of manual work needed in the extraction process given a target website. This aspect can be further expanded considering whether or not the approach requires programming skills.

- the *induction techniques* used to build (induce) the wrapper. Machine Learning, Logic Induction, statistical approaches or handcrafted rules are known approaches in literature.

- the amount of *domain knowledge* required to perform the data extraction task. In section 2.5 we analyze some domain-tailored approaches that solve a specific task by employing a set of rules and heuristics. State-of-the art techniques are defined in a general and domain-agnostic way. However such general approaches need to be instanced with specific features to obtain satisfactory results in practice.

One important issue with Wrappers and Wrapper Inductors is that they are commonly built to induce an extraction rule with a given web resource and not on multiple-site scenarios. However there are few work that focus on the automatic extraction of contents from multiple websites: in the next sections some examples will be provided.

2.2.1 Wrappers and Page Representation

Web Content extraction methods use different techniques to represent web pages and consequently to obtain the extraction of data. Two main groups of approaches can be found, the former being plain text approaches and a structured approach where all available information is exploited. The page representation technique strongly affects the extraction quality and limitations. Moreover a representation technique robust to evolutions of the web should be abstract enough to describe its content and structure without specific details of the underlying standards.

The first works analyzed directly the HTML [W3C., 1999a] source, by defining regular expressions or similar language rules and constraints. The pioneristic TSIMMIS system proposed in [Hammer et al., 1997] for example is one of the first approaches providing a framework for manual building of wrappers. In that system each wrapper is defined by a specification file that states where the interesting data is located in the page. Extraction rules were conceived to be developed by programmers with specific technical skills.

The ever-changing presentation details of the HTML language and related technologies like Cascading Style Sheets [W3C., 2009] and Javascript [Ecma International, 2009] make techniques that directly analyze the page source difficult to maintain. Moreover, the continued effort in the separation of data (XHTML) and its presentation (CSS) requires to consider both elements to full exploitation of all visual semantic cues.

Another class of approaches is based on the Document Object Model [W3C., 1997] (DOM) for HTML. The DOM models each web page with a tree where HTML tags are internal nodes while leaf nodes are represented by text, images and hyperlinks. The DOM for HTML is used in web browser to represent the rendered page and expose a common API for its modification. For this reason each element in the tree that is displayed also has layout properties, such as position and size. A collection of works based on this are detailed in [Cai et al., 2003].

A limitation of the DOM is within its nature, that is a lack of correlation between blocks that are distant in the tree but are close when presented to the user. More complex analysis of the page structure are possible to obtain a semantic segmentation of the page and its contents. The approach proposed by Cai et al. in [Cai et al., 2003] and named VIPS combines the DOM tree and heuristic rules to define semantic blocks of contents by making use of the visual layout

structure and thus mimicking the user understanding of web contents. The approach is assessed both in the segmentation task and in a Information Retrieval task where the page segmentation is used to improve the term correlation at block-level instead of page-level.

The XPath standard [W3C., 1999b] is a useful tool developed to locate elements within hierarchical documents that can be modeled with the DOM, such as HTML and XML. The ability to locate specific elements permits to easily build a wrapper by defining an XPath query for each desired output field. A large amount of works in literature have been developed with this technology, though in its raw form it requires detailed technical skills in composing proper queries for Information Extraction.

2.2.2 Wrapper Induction

The wrapper induction method proposed by Kushmerick in his Ph.D. thesis [Kushmerick, 1997] has defined a formal framework for automatically generating wrappers given a supervised truth. The solution he proposed was the first work where an automatic method was used to automatically generate those wrappers requiring tedious manual work in the past.

The aim of the method he proposed is to build a wrapper belonging to the $HLRT$ class, that is the acronym of the head-left-right-tail string delimiters used by the wrapper to identify data of interest to extract. The $HLRT$ class is designed to extract data of interest from a tabular structure, in fact the "head" string delimiter is used to identify the top of the table, the "tail" delimiter is used for the bottom part and the "left" and "right" strings are placed at the side of a data field of interest.

Considering for example the HTML source reported in figure 2.3, the head delimiter is $< P >$, the left delimiter for the country is $< B >$, the right delimiter is $< /B >$ and the tail string is $< HR >$. The $HLRT$ class permits to define wrappers to extract K attributes by defining K pairs (l_i, r_i) of attribute delimiters, thus allowing the definition of a tuple for the extraction of both Country and its telephone code in the example above.

Through induction the $HLRT$ wrapper can be generated, given a proper supervised truth. The $HLRT$ class of wrapper is not ensured to be learnable, because there are certain HTML structures that are not tabular and do not obey to the head-left-right-top pattern. In order to em-

```
<HTML><TITLE>Some Country Codes</TITLE><BODY>
<B>Some Country Codes</B><P>
<B>Congo</B> <I>242</I><BR>
<B>Egypt</B> <I>20</I><BR>
<B>Belize</B> <I>501</I><BR>
<B>Spain</B> <I>34</I><BR>
<HR><B>End</B></BODY></HTML>
```

Figure 2.3 – *A fictious HTML source providing telephone country codes. This example was first proposed in Kushmerick's thesis [Kushmerick, 1997].*

brace a larger set of websites the LR, OCLR, HOCLRT, N-LR and N-HLRT were defined considering more complex layouts for example hierarchical structures. The overall set of wrappers was extensively experimented stating that a wrapper could be constructed for 70% of websites.

Another contribution contained in Kushmerick's thesis is what he named *corroboration*. Corroboration is a formal framework for automatic labeling of pages in a given restricted domain. The approach he describes is at the basis of the elimination of manual and specialized (e.g. with programming skills) human work when developing and maintaining web data extraction systems.

Although the great variety of the approaches introduces specific problems, there are two main issues with Wrapper Inductors techniques, namely wrapper verification [Kushmerick, 2000] and wrapper maintenance [Kushmerick, 1999]. The former is about checking that a given instance of a generated wrapper still works properly, while the latter is the problem of updating a given wrapper instance automatically or with minimal human intervention.

These two known issues of wrapper induction approaches have to be considered carefully when building real-world systems voted to stability, scalability and ease of maintenance. The wrapper induction approach proposed in this work has been designed from the ground with these aspects in mind and consequently to maximize the correctness of collected results while requiring minimal human intervention.

2.3 Web Structure and Link Analysis

The World Wide Web is a graph, more precisely a *directed labeled graph* where the nodes are represented by the pages and the edges are represented by links between them. In addition each edge is labeled, since a page linking to another brings some text or more generally a piece of multimedia data (an image, a video) to describe the connection.

According to [Kosala and Blockeel, 2000] three main areas of research can be distinguished: Web Content Mining (WCM) - the application of data mining techniques to Web documents, Web Usage Mining - the analysis of interactions between the user and the Web and Web Structure Mining and/or Link Analysis in which the structure of hyperlinks is used to solve a problem in a graph-structured domain.

These WM tasks can be combined in an unique application, reinforcing the mining process by the allied analysis of different Web characteristics. Recent works propose in particular the integration of Web Structure and Text Mining tasks. The hyper link information, and in particular the anchor text - the text appearing in the predecessor page and pointing to the target, has been used in Information Retrieval tasks and classification tasks [Spertus, 1997, Fürnkranz, 2002].

Page classification in particular can benefit from Link Analysis since it permits to determine the topic of each page in a more robust way, considering in the classification process the predicted category of neighbors [Chakrabarti et al., 1998, Oh et al., 2000, Joachims et al., 2001]. The approaches described in those works permit to provide additional information in particular when the web page to be classified has no text or it is not enough to provide a meaningful prediction of its contents.

Another interesting work where the web structure is exploited to classify contents is THESUS [Halkidi et al., 2003]. THESUS proposes a system for web document organization and querying where the semantic content considered for a web page includes also keywords from all pages' incoming links. The collection of considered web pages is clustered into thematic subsets based on their semantics by a novel similarity measure that considers the page contents and its neighbors. The proposed technique is then experimentally evaluated in terms of semantic clustering quality and retrieval effectiveness.

2.4 Machine Learning

Since the year 2000 the application of learning strategies to WM tasks was intensively studied showing advantages in terms of both effectiveness and portability over conventional and earlier strategies based on knowledge engineering approach [Kosala and Blockeel, 2000]. In supervised learning training examples consist of input/output pairs $(\vec{x}, y)$ and the goal of the learning algorithm is to predict the output values $\hat{y}$ of never seen input values $\vec{x}'$. In unsupervised learning [Jain, 1999] training examples are constituted only by input patterns $\vec{x}$; the learning algorithm is able to generalize from input patterns to discover similarities among data [Michalski et al., 1983, Mitchell, 1997].

Multi-Layer Perceptron (MLP), Support Vector Machines (SVM) [Burges, 1998] and Radial Basis Function Network (RBFN) [Moody and Darken, 1989] are among the most common supervised learning techniques. Supervised learning models interact with input patterns $\vec{x}$ directly, through activation/transfer functions [Duda et al., 2000] that connect each layer of the network to the others or through distance metrics (or kernels) $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ that permit to map the space of input pattern $\vec{x}$ to another where the learning problem is easier. Such functions are used to solve non-linear, complex learning hyperplanes that can be found in real-world problems.

In addition to these two well-known settings, other recent learning paradigms that allow additional flexibility can be considered in the definition of a novel Web Content Mining approach: semi-supervised [Zhu, 2005] and multi-instance [Dietterich and Lathrop, 1997] learning.

Semi-supervised learning [Zhu, 2005] is a paradigm where the supervised set TrS of pairs $(\vec{x}, y)$ is enriched with a set $Unsup$ of elements $\vec{x}$ where no supervision has been given. Several semi-supervised learning approaches have been studied and discovered in the past, for instance in self-training a classifier is trained with the TrS set and in a second phase the predictions of the classifier on the $Unsup$ set are added to the training elements if some conditions are met.

Multiple-instance Learning (MIL) has been introduced in the context of the drug activity prediction problem [Dietterich and Lathrop, 1997]. The drug activity prediction was about saying whether or not a new molecule was qualified to make some drug, given a collection of known molecules. For each molecule there are many alternative low-energy shapes. The main issue

is that biochemists only know if a molecule is qualified to make a drug or not, however they do not know which of its low-energy shapes is responsible for the qualification. The solution proposed in that paper is to model each supervised molecule as a bag $(\{x_1, \ldots, x_n\}, y)$ with a binary label y (positive or negative) and a set of *instances* $\{x_1, \ldots, x_n\}$, one for each low-energy alternative shape. The MIL paradigm has then been applied to Content-Based Image Retrieval [Maron and Ratan, 1998], Text Classification [Andrews et al., 2003] and Web Mining [Zhou et al., 2005].

In [Zhou and Xu, 2007] the semi-supervised and multi-instance learning paradigms have been demonstrated to be in relation, since in multi-instance learning the user labels only one instance per bag, giving the algorithm a similar amount of information found in a semi-supervised setting. Multi-instance learning is a special case of semi-supervised learning.

These paradigms have been successfully applied in the context of Web Mining. A suitable ML strategy able to cope with heterogeneous, multi-source, redundant or missing data is fundamental in the development of a robust and effective WCM strategy.

2.4.1 Radial Basis Function Networks

In order to make this work self-contained, in this section we analyze the Radial Basis Function Network (RBFN) in detail. The solution proposed in section 3 involves a revised RBFN with extended distance metrics.

RFBN belong to the multi-level neural networks family, where hidden layer units model non-linear functions of input features for example through a non-linear distance metric. Activation of hidden units is therefore computed according to the distance between the input vector and a reference vector. RBFN have been introduced by Moody and Darken [Moody and Darken, 1989] as an alternative neural network model with a fast and effective learning algorithm. They have been demonstrated [Hartman et al., 1990, Park and Sandberg, 1993] to have the universal approximation property and therefore allow to solve from a theoretic point of view all classification problems.

Radial Basis Functions (RBF) techniques have their origin in exact interpolation methods [Powell, 1987] for multi-dimensional spaces. The exact interpolation problem requires that

each input vector must be mapped to a given target vector. Consider the $g : \mathbb{X}^d \rightarrow \mathbb{R}$ mapping and a dataset $\{(\vec{x}_1, t_1), \ldots, (\vec{x}_N, t_N)\}$ with $\vec{x} \in \mathbb{X}^d$ and $t_i \in \mathbb{R}$. Given these elements the aim is to build a function $h(\cdot)$ such that $h(\vec{x}_i) = t_i,\ i = 1, \ldots, N$. The RBF solution employes a set of N non-linear *basis functions* $\phi(\cdot)$, one for each input vector $\vec{x}_i$, defined as:

$$h(\vec{x}) = \sum_{i}^{N} w_i\ \phi(\|\vec{x} - \vec{x}_i\|) \tag{2.1}$$

where the i-esim basis function is built on top of the $\|\vec{x} - \vec{x}_i\|$ distance between the i-esim input vector and the $\vec{x}$ input parameter of the h function, and where the w_i variables can be found solving the linear system $\mathbf{\Phi}\vec{w} = \vec{t}$ where $\vec{t} = (t_1, \ldots, t_N)$, $\vec{w} = (w_1, \ldots, w_N)$ and the matrix $\mathbf{\Phi}$ defined by $\mathbf{\Phi}_{i,j} = \phi(\|\vec{x} - \vec{x}_i\|)$. If the inverse of $\mathbf{\Phi}$ is well-defined the solution is given by $\vec{w} = \mathbf{\Phi}^{-1}\vec{t}$.

A well-known function used for ϕ is the Gaussian $\phi(x) = \exp\left(-x^2/2^2\right)$. The functions used for ϕ have positive activation values in a small portion of the domain, for this reason they are said to be "radial". The exact-interpolation framework we have illustrated can be easily extended to k-dimensional targets by considering a set of k parallel non-linear systems of equations.

RBFN can be defined starting from the exact interpolation framework we have described so far. The key differences [Broomhead and Lowe, 1988, Moody and Darken, 1989] between exact interpolation with RBF and RBFN are 1) the number M of basis functions must be chosen considering the complexity of the problem instead of considering $M = N$, and $M \leq N$ for the generalization property to hold true 2) RBF centroids are free variables $\vec{\mu}_i$ that must be set by the learning phase 3) there is one σ_i parameter for each of the M basis functions and its value is determined in the learning phase 4) a bias $w_{k,0}$ variable is added in the linear sum to handle situations where input and output spaces have different scales.

The RBFN formula for the j-th output dimension of the target space is thus

$$f_j(\vec{x}) \quad = \quad \sum_{i=1}^{M} w_{j,i}\ \phi_i(\vec{x}) + w_{j,0} \tag{2.2}$$

The original RBFN learning strategy is two-phased: in the former the first-level basis function parameters $\vec{\mu}_i$ and σ_i are computed while in the latter the second-level weight matrix of elements $w_{j,i}$ is optimized. A fundamental difference between the two phases is that the former

is unsupervised and thus target optimization values are not considered, while the latter is completely supervised. The separation of phases also permits to solve the optimization problem with linear techniques instead of more complex non-linear methods. This learning approach requires the user to set the number M of RBF. The M value must be chosen carefully since it influences greatly the speed and generalization power of the trained model. Cross-validation techniques can be used to set the M parameter (v. [Bishop et al., 1975], pag.372-375).

In the original RBFN learning strategy the first unsupervised phase uses *K-means* clustering to discover K representative centroids with $K = M$ [Jain, 1999]. When the $\vec{\mu}_i$ values are computed with this algorithm the σ_i values can computed considering variance within data belonging to a given cluster.

In the second phase the $w_{k,i}$ variables are set. Since the number $M < N$ it is not in general feasible to find the exact solution. For this reason a cost function is used to minimize the sum-of-squares error:

$$E = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=0}^{k} [f_j(\vec{x}_i) - t_{j,i}]^2 \tag{2.3}$$

The corresponding system of linear equations for $E(\cdot)$ can be solved using the pseudoinverse technique, the Widrow-Hoff LMS algorithm and other techniques.

The original RBFN learning strategy is designed for fast computational times, however the additional information of target values can improve the optimization in the first level of the network. The Reformulated Radial Basis Neural Networks proposed in [Karayiannis, 1999] proposes an alternative learning approach where the $\vec{\mu}_i$, σ_i and $w_{j,i}$ parameters $i = 1, \ldots, M,\ j = 1, \ldots, k$ are all set in a supervised setting.

2.5 Domain-specific Approaches

The Columbia Newsblaster system [McKeown et al., 2002], also known as Google News, employs a set of handcrafted rules to extract the image of an article – the authors comment on the set of rules as perfect precision and high recall. Another related work in the web image identification area is the one by Maekawa et al. [Maekawa et al., 2006] where a general-content

approach for image classification is presented. The scope of that work is to classify the images found in web pages in layout classes, each defining a role in a given web page.

Within the field of web news harvesting the work proposed in [Reis et al., 2004] employs tree edit distance to discover similarities between pages containing semantic content (title, article body, image, etc). That system crawls the web for news websites and clusters automatically extracted news entries by exploiting domain-specific knowledge. They use a restricted tree edit distance algorithm on clusters of pages to generate a pattern (that is, a Wrapper Induction). With the help of domain-specific heuristics clusters of pages and their wrappers are enriched with labels for the title and body of each news article.

A remarking characteristic of recent works on content mining is the use of domain-specific features or rules. A clear limitation of such setting is the possibility to re-use models and tools to a wide range of scenarios. Moreover much works are limited in the generalization power by the page representation, locking the learned wrapper or extractor to a given web resource.

2.6 Summary

Analyzing recent works in WCM literature we now summarize the most important aspects that either need to be clearly considered or need to improved with respect to the current state of the art:

1. the growth of Semantic Web technologies will enable to gather structured data from any websites without the need of intelligent and adaptive methods. The very low current usage of such technologies however prohibits to depend solely on the availability of such data in websites.

2. the Wrapper Verification and Maintenance problems need to be challenged in order to permit the building of a large-scale automatic method of WCM that does not need much work by specialized experts.

3. the single-source limit of wrappers inhibits to generalize a WCM on unseen sources without an analysis and some sort of ground truth necessary to permit the Wrapper Induction.

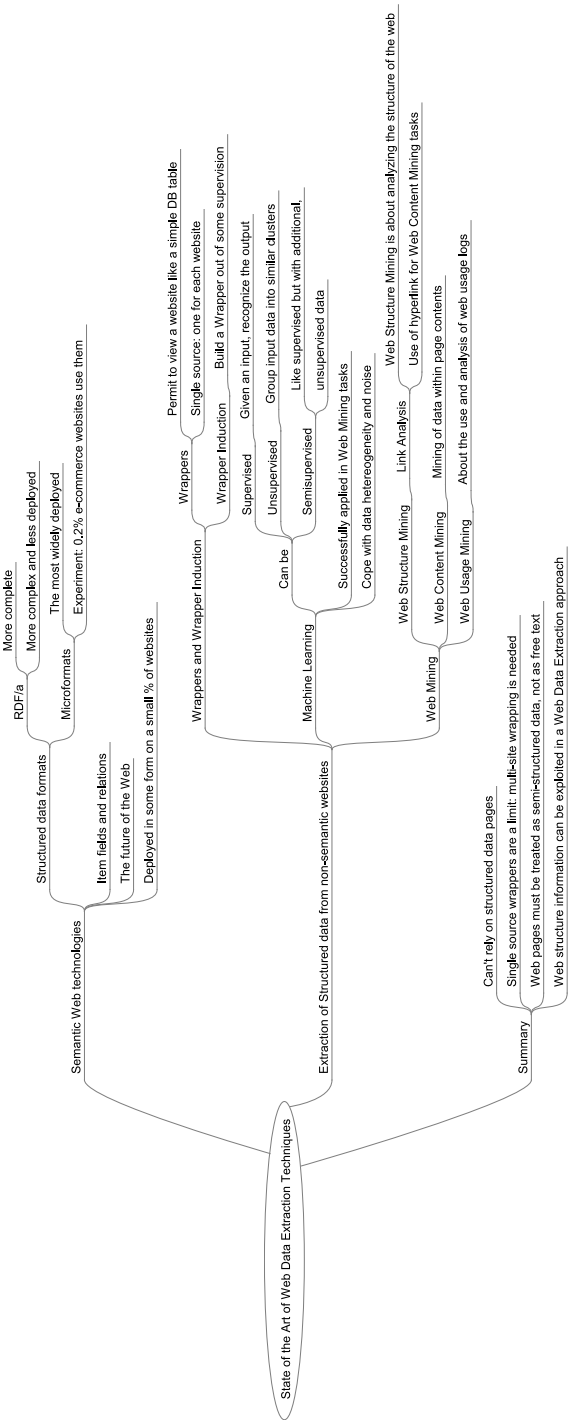


Figure 2.4 – *Mind map of the state of the art analysis.*

Although some domain-specific works are able to work on a multitude of websites, there are no well-known general content extraction approaches in literature.

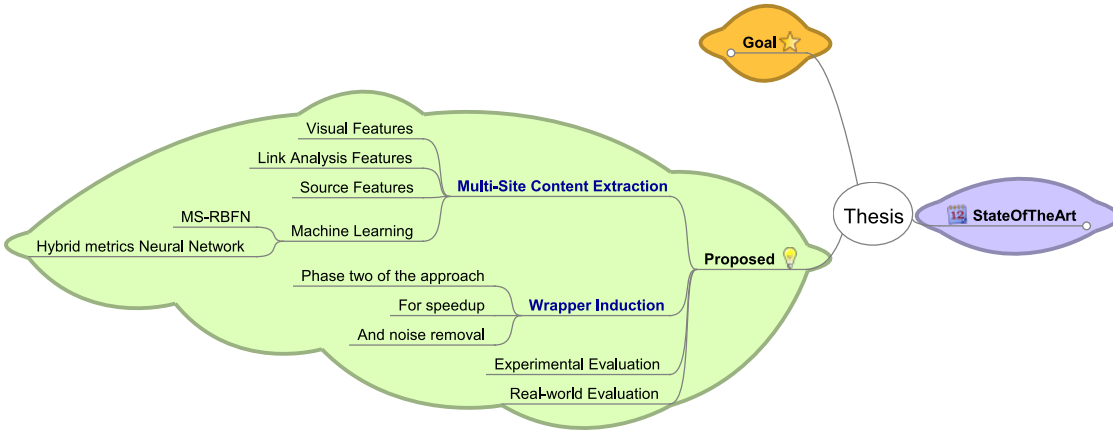
4. wrapper induction methods that rely on delimiters at source level, or other plain-text feature are throwing away all that information that is crucial when presenting web pages to the user. Moreover the evolution of web standards towards a separation of data (XHTML) and its presentation (CSS) prevents to obtain such information directly from the "page source" - a smarter approach is needed by seeing exactly what the user sees.

5. the web structure information can be be considered to improve the effectiveness of WCM strategies. Web Content and Web Structure information are of heterogeneous nature and may require proper Machine Learning techniques to integrate them in an effective way.

A summary mind map of this chapter is proposed in figure 2.4.

Chapter 3

The Proposed Approach



Starting from the study of strengths and weaknesses of approaches in literature, we propose to apply a two-phased strategy in the building of a topic-oriented Web Content Mining (WCM) method. In the next sections we refer to *content page* as the page were the data is to be extracted, that is pages that contain a single data record of interest, *data of interest* as the complete set of required information and *field of interest* as a specific data element required to build a complete data record. The proposed solution thus follows the *page level* approach to data extraction detailed in section 2.2.

Figure 3.1 – *An example Web Content Mining problem where different* field of interest *are put in evidence.*

3.1 The proposed approach

At the basis of our approach is the fact that web pages of a given topic have *content pages* with similar aspects, that is to say that invariant properties can be captured in Web page layouts in such a way that the *field of interests* classes of elements are properly defined. To this purpose, we proceed from the assumption that Web usability guidelines are more and more applied in the design of Web sites allowing users to waste no time reading all items they see in a web page, and let them localize interesting information immediately [Nielsen, 2001]. A sample web news *content page* is reported in figure 3.1 with *field of interest* circled in red.

In the context of e-commerce web sites for example, the meaning of web-usability is narrowed down to efficiency: triggering sales and/or performing other transactions is valuable to the business [Gehrke and Turban, 1999]. Even if absolute normative rules are not possible, as every situation requires creativity, compliance with the general principle ensures a high level of consistency and regularity among web sites at least belonging to the same typology.

Our approach to WCM is composed of two main phases:

1. in the first phase elements to extract from web pages of a given topic are recognized with a general model that has been previously trained with proper ground truth. This phase uses visual features and a page representation that mimics what is shown to the user to learn how a *content page* is layed out and how elements of interest are presented (position in the page, size, etc). This phase is detailed in section 3.2

2. in the second phase and after enough data has been extracted for each website, a site-specific wrapper is built starting from previously collected data. This phase permits both to speed-up the recognition of data in each web page and to make the system more robust in detection of false positives, since a clustering strategy is adopted to decide the final structure of a website wrapper. This phase is detailed in section 3.3

The two phases deal with different aspects and issues of WCM. The first permits to challenge massive data extraction by building a single model for a given topic, and to maintain already-built wrappers for specific websites. That is possible by exploiting visual features and structural features of websites and pages. The second phase permits to boost performance both in terms of quality and speed.

An interesting aspect of our method is that it is developed for the extraction of both text contents and image contents. Further developments could extend the proposed solution to other multimedia sources found on the web.

3.2 A Multi-site Web Content Mining Algorithm

Phase one of the proposed approach – that we name LEWECOM (Learnable Web Content Mining approach) – needs to build a general model to recognize a given *data of interest* set. In the

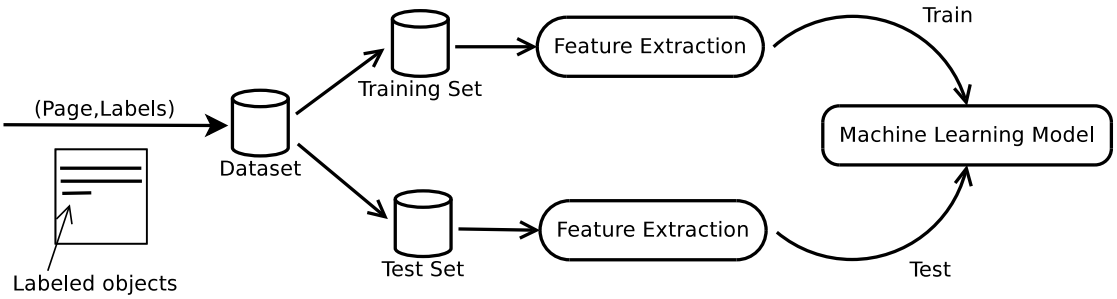


Figure 3.2 – *Outline of the Machine-Learning based scheme.*

following we refer to *fields of interest* with $\pi_i \in \Pi$ where Π is the schema composed of desired data fields π_i.

The LEWECOM phase is based on a Machine Learning (ML) model in a supervised setting (figure 3.2), thus requiring a ground truth built by a pool of domain experts. The required supervised set is a collection D of web pages where position of elements in Π have been annotated. In order for the inductive model to generalize well on unseen websites, the dimension $|D|$ has to be big enough to consider a scalable annotation method for the experts. More on this will be presented in chapter 4.

The LEWECOM approach has three main parts. The first is the definition of how the web page is represented, that is the structure and characteristics of a web page. The second is a set of representative features that work on the defined web page structure. The third is the definition of a proper ML model able to induce from the heterogeneous data in D.

Our idea is to keep in pair with current and future standard and technologies by approaching web content mining with the help of a web rendering engine. We basically follow a DOM-aware strategy as discussed in section 2.2 with the strong requirement of using an HTML DOM tree extracted from a rendering of the page. Each web page p is thus a semi-structured source of information where $B = \{b_1, \ldots, b_m\},\ B \subset \mathbb{B}$ is the set of *web objects* in $p \in \mathbb{P}$, where $\mathbb{B}$ is the domain of all web objects and $\mathbb{P}$ is the domain of pages. With the term "web object" we refer either to a string of text that appears in p as a the inner text of a HTML DOM [W3C., 2003] node or to an image appearing in the tree. The set B is obtained through the $extract\_blocks : \mathbb{P} \rightarrow \mathbb{B}$ function, further explained in Sec. 3.2.6.

Figure 3.3 – *A troublesome situation where visual information is obtained directly from the source of the page. A robust method has to deal with all presentation-level details (e.g. Cascading Style Sheets) in order to correctly compute the layout of web elements.*

We define three feature types:

1. *Visual Features* that describe how web objects are layed out in the page: position, size, text formatting, etc.

2. *Link Analysis Features* features that exploit the structure of websites to expand the available information needed to detect *fields of interest*.

3. *Source Features* - features of web object that can be derived directly from the source of the web page e.g. textual features.

In the following sections motivation and details behind the three kinds of features will be given. Section 3.2.5 describes requirements of the ML model to work properly with the pro-

posed set of features. Section 3.2.4 details the complete learning and generalization phases and section 3.4 describes a proposed ML model for the LEWECOM approach.

3.2.1 Visual Features

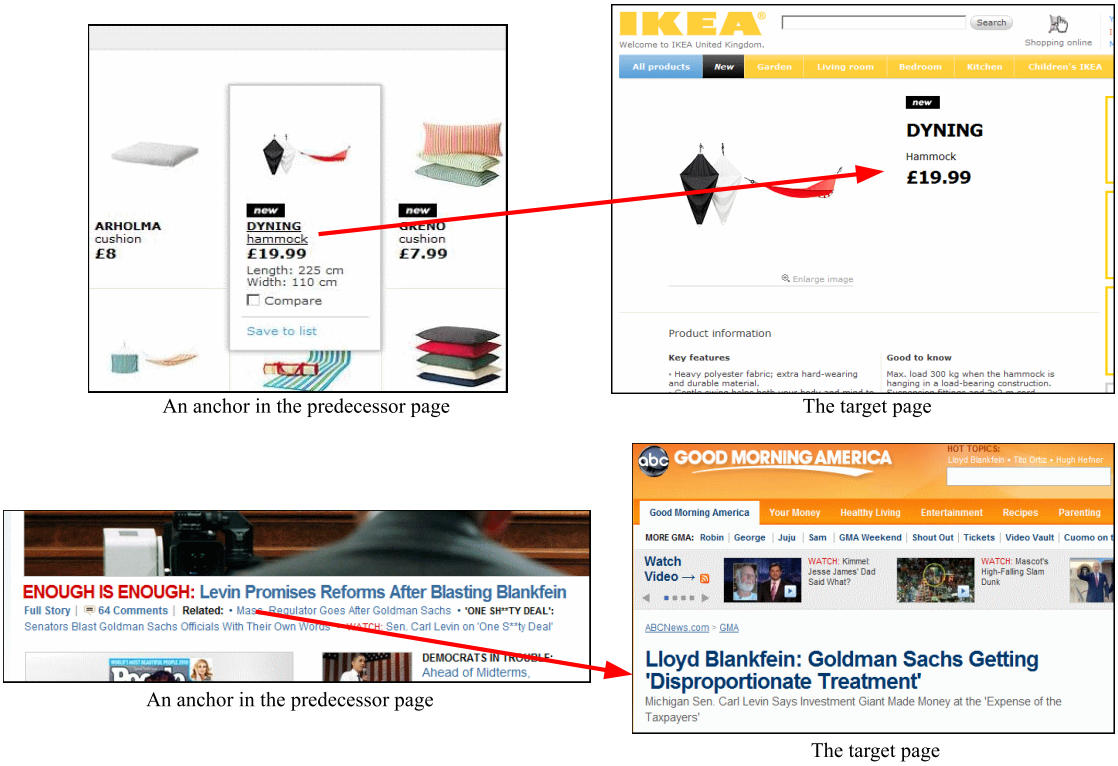
Features coming from the rendering of a web page are likely to be highly informative when identifying a given entity. When recognizing text entities of a given semantic class the formatting and position of objects have a great discriminating power, consider for example the title of an article or the price of a product: they are likely to be on the top of the page with a big font and some form of emphasis like bolding or underlining. Robust exploitation of visual features heavily rely on page representation and rendering: computation of visual features at source-level may result inaccurate because of the thousands of ways to change font formatting and layout information of page elements. In figure 3.3 a troublesome situation is reported when visual features are computed using source-level information only.

The rendered page text is properly annotated with structures that suggest the formatting of the text (font weight, font size) and the presence of images or other media objects. From this rendering view one can derive the position of each object (text, images) in the web page, and by analyzing each resource additional metadata can be obtained: object size, width, URL, file name, usage count in the page, etc.

The segmentation of the text found in the web page should be relevant to the content, and techniques such as [Cai et al., 2003] can be used. Further details on text segmentation are given in section 3.2.6.

3.2.2 Link Analysis

Every and each page in the web has a definite topic that can be explicitly declared in the page itself with a short text – a kind of *field of interest* – we name *main entity*. Consider for instance online blogs, the *main entity* of each post's page is the title of the post. An online newspaper's page will have the new's title as the *main entity*, and an e-commerce website the product's name. In Figure 3.4 two examples of websites with a *main entity* are reported, together with one of the incoming hyperlinks to the target page.



An anchor in the predecessor page

The target page

An anchor in the predecessor page

The target page

Figure 3.4 – *Examples from an e-commerce website and an online newspaper. In both cases the anchortext in the predecessor page and the* main entity *in the target page are linked through the red arrow.*

There is a lot of information given by link analysis that has proven to be effective in Information Retrieval and web analysis by the PageRank [Page et al., 1999] algorithm used in the Google search engine [Brin and Page, 1998] and the HITS [Kleinberg, 1999] algorithm and related works. The PageRank algorithm proves that the anchor text t_i of each link l_i to a page p_j has a strong weight when determining whether t_i is relevant in p_j or in other words that p_j is relevant when the user is searching for t_i. We believe that the presence of a link l_i with text t_i can be used as a highly discriminating feature when recognizing the *main entity*.

These considerations enable us to extend our page representation. In addition to the set B we know the set of incoming links to p and in particular for each link we know the *anchor text*, that is the text in the *predecessor page* pointing to p. The set of anchor texts is $A = \{a_1, \ldots, a_n\}$.

Both text links and images can be considered to build the A set, where the anchor text and the alternative text of the image can be used respectively.

3.2.3 Feature Extraction

In the *field of interest* recognition problem the object to classify is a web object $b \in B$ given its page p and the A set for p. We therefore partition the features in two subsets: web object features and anchortext features. The elements in the former are functions $f \in \mathbb{F}$ in the form $f : \mathbb{B} \times \mathbb{P} \rightarrow \mathbb{R}$ while in the latter we have functions $g \in \mathbb{G}$ in the form $g : \mathbb{A} \times \mathbb{B} \rightarrow \mathbb{R}$.

Web object features describe low-level characteristics of elements $b \in B$ found in the page and their relation with the page itself, such as the title, page dimensions etc. In the following we refer to the text of the web object with $b.text$. The text can be the inner text for textual web objects and the alternative text for image web objects. The first feature we describe are *visual* web object features:

- f_{fsize} : expresses the font size of the text $b.text$ normalized w.r.t. a maximum value defined a priori. The idea behind this feature is that some kinds of *field of interest* are often presented with a big font-size to be easily identified by the user.

- f_{fbold} : expresses whether or not the considered text $b.text$ is displayed with a bold font. The idea behind this feature is closely related to the f_{fsize} feature and is driven by the fact that some *field of interest* are often presented with a bold font to let web users find them quickly.

- f_{fstroked} : is a boolean feature that is true if the text $b.text$ is displayed with a stroke on it. This feature captures the fact that the importance of the text b_i has been outdated by a block b_j with newer information. Recall for example e-commerce websites that leave an old price with a stroked font and a newer, discounted one.

- f_{h} : is the real-valued normalized height of the web object, where the original height has been scaled with 768 as maximum value, where (eventually) larger web objects are considered to have a 768 pixel height. This feature has been mainly introduced to describe

images, that cannot be described with font formatting information. The idea behind this feature and the f_{W} feature described below is that the *field of interest* class for images is likely to have a proper size with respect to the page ratio. Think about web news articles, where story images are likely to be near the title or the abstract.

- f_{W} : is the real-valued normalized width of the web object, considering maximum value of 1024 and an approach for larger web objects similar to the one used for f_{h}.

- $f_{\text{dstfromentity}}$: is a real-valued feature expressing the distance of web object b from a given web object b_i previously recognized as the field of interest π_j. This feature stems from the observation that interesting data carrying semantics is layed out in specific areas of pages, and different fields of interest can be placed near one another. For example when recognizing the position of a field of interest π_{image} one can use $f_{\text{dstfromentity}_{date}}$ to express that most of the times the field π_{image} is nearby the π_{date} field.

- $f_{\text{equaltextto}}$: is a real-valued feature expressing the similarity between the text $b.text$ and the text $b_i.text$ of the web object b_i previously recognized as the field of interest π_j. The driving reason behind the definition of this feature is the observation that most of the times the image describing the main item in the page has a describing text that is very similar to the name of the main item in the page. The similarity between the $b.text$ and $b_i.text$ strings is expressed by the *Dice coefficient* [Frakes and Baeza-Yates, 1992], computed as:

$$dice(a, b) = \frac{2C}{|a| + |b|}$$

where C is the number of common terms between a and b, $|a|$ and $|b|$ are the number of terms of a and b, respectively.

there are other non-visual web object features that can be derived directly from the source of the page we named *Source Features* before:

- f_{intitle} : expresses the percentage of matching between the terms of the text within web object b and the terms in the title t of the page, following the web usability guideline stating that the main object of a web page should be named in the title. For example if

the web page p title is t="The quick brown fox jumps over the lazy dog" and $b.text =$ "The quick brown fox" then $f_{\text{intitle}}(b, p) = 16/35$, where 16 is the total length of the common terms and 35 is the length of t.

- $f_{\text{smartintitle}}$: works similarly to f_{intitle} except that it tries to remove the website name from the title. Suppose the URL of page p is `http://www.eshop.com` and the title t="Eshop - Product XYZ" and $b.text$="Product XYZ" then $f_{\text{smartintitle}}(b, p) = 1$ since the first level domain of the URL is considered as a potential website name, and is removed from the title when comparing with textblock contents.

- f_{type} is a binary feature vector equal to:

$$f_{\text{type}} = (1, 0, 0, 0) \quad \text{if the web object is a JPEG image}$$
$$f_{\text{type}} = (0, 1, 0, 0) \quad \text{if the web object is a GIF image}$$
$$f_{\text{type}} = (0, 0, 1, 0) \quad \text{if the web object is a PNG image}$$
$$f_{\text{type}} = (0, 0, 0, 1) \quad \text{otherwise}$$

  and thus gives information on the format of the image. This feature is led by the observation that candidate field of interest elements in the web page are in most cases photographs and thus available in JPEG or PNG format. The image type information is derived from the MIME type.

- f_{name} is a real-valued number that expresses the fraction of numeric chars in the file name of the web object, e.g. if the URL $\mathcal{U}_|$ of image I_j is "`http://www.some.url.com/path/to/file/image-10.jpg`", then the file name without extension is "image-10" and the feature value
  $f_{\text{name}}(I_j) = 1/4$ since the string "image-10" has length equal to 8 and "10" has length equal to 2. The intuition behind this feature is given by the database-generated nature of web contents, where a numerical identifier of the story or object of interest is likely to appear in the URL of the page.

- f_{regex} is a boolean feature that is equal to 1 when the text $b.text$ matches a given regular expression. This feature can be used to develop domain-oriented features by adding strong

and safe information to the recognition process. For example one can introduce a price regular expression, a date regular expression, etc.

Anchortext features take into account anchors and web objects to allow the definition of low-level relationships between the two. Here we define a simple, low level feature of this kind:

- g_{dice} : expresses the similarity between the text $b.text$ and the anchortext a. The employed similarity measure is the *Dice coefficient* [Frakes and Baeza-Yates, 1992] as previously described for the $f_{\text{equaltextto}}$ feature.
- g_{regex} is a boolean feature defined similarly to f_{regex}, except that the considered text is the anchortext a.

In the g_{dice} and f_{intitle} features a tokenizer is required to obtain a set of terms from a text block or an anchor text or from the title of the page. The tokenizer has to be selected depending on the domain of the problem.

Given a set $F \subset \mathbb{F}$ and a set $G \subset \mathbb{G}$ and a (p, b, A) tuple we define the $\Theta \subset \mathbb{R}^n,\ n = |F|$ web object feature space and the Γ anchortext feature space. An element γ from the (b, A) pair is in the form $\gamma = \{\vec{x}_1, \ldots, \vec{x}_m\}$ where each instance $\vec{x}_i$ is computed by the set of features defined from a (a, b) pair $\forall a \in A$. The cardinality m of a γ element is not fixed a priori since the number $|A|$ of anchortexts pointing to a given page may vary.

3.2.4 Learning And Generalization

Let us now consider the training phase of the algorithm and the step-by-step phases for the recognition of a given *field of interest* $\pi_i \in \Pi$. The composition of learning and generalization for each π_i gives the complete algorithm able all the *data of interest*.

Given the supervised dataset D, the textblock features F, the anchortext features G and the a properly configured ML model M, the first step is to train the algorithm with data extracted from D. Details on the selected model M are given in section 3.2.5.

1. Split the set D in D_{TrS} and D_{TeS} with some rule, e.g. $2/3$ in D_{TrS} and $D_{TeS} = D \setminus D_{TrS}$.

2. Initialize the training set $TrS = \{\}$.

3. $\forall (p, e, A) \in D$:

   (a) Apply $extract\_blocks$ to obtain B.

   (b) $\forall b \in B$

      i. Compute $\hat{\theta}$ with features defined in F.

      ii. Compute $\hat{\gamma}$ with features defined in G.

      iii. Set the label y to ω_1 (*field of interest*) if $b = e$ and to ω_2 (not *field of interest*) otherwise.

      iv. Add the $((\hat{\theta}, \hat{\gamma}), y)$ element to TrS.

4. Train the machine learning model M with TrS.

The model can then be used recognize a given *field of interest* using the generalization phase. Be (p, A) an input pair, and H the output set of recognized *fields of interest*:

1. Initialize the output set $H = \{\}$

2. Apply $extract\_blocks$ to obtain B.

3. $\forall b \in B$:

   (a) Compute θ with features defined in F.

   (b) Compute γ with features defined in G.

   (c) Recognize the label $\hat{y} = h((\theta, \gamma))$

   (d) If $\hat{y} = w_1$, add b to H

The performance can then be evaluated using the generalization phase over all elements in D_{TeS}.

3.2.5 Machine Learning Model

For each field of interest $\pi_i \in \Pi$ the proposed approach requires a proper model M able to learn from the training set TrS of elements $((\hat{\theta}, \hat{\gamma}), y)$. Classic ML approaches work with a vector of features $\vec{x} \in \mathbb{R}^n$ whereas elements in TrS are composed objects $(\hat{\theta}, \hat{\gamma}) \in \Theta \times \Gamma$ with $\Theta \subset \mathbb{R}^n$ and Γ is the feature space for variable-length distributions of features.

If the selected configuration for field of interest π_i has $G = \emptyset$ then a classic ML model 2.4 can be employed to solve the learning and generalization problem. Otherwise a proper solution able to deal with the non-euclidean input space is required. A proposed solution is presented in section 3.4.

3.2.6 Procedural Details

In this section further details are given to understand how the proposed approach works. In particular the text block extraction process $extract\_blocks$ is described and details on how the page is seen by the algorithm are further explained.

Page Rendering

The rendering engine program is based on XULRunner [1] and permits to obtain visual layout information for elements $b \in B$. This procedure is configured to mimic a 1024x768 screen, as one of the most widely used setups on the web.

Text Block Extraction Process

The blocks $b_i \in B$ set should as much as possible be consistent with the semantics of the page. A naïve approach is to consider leaf DOM nodes only, however due to the complex and heterogeneous real-world DOM structures it is necessary to consider also some non-leaf nodes since the resulting b_i elements should be as much as possible aligned with semantics. In [Cai et al., 2003] Deng et al. show how a good DOM segmentation algorithm can contribute to obtain a semantically meaningful aggregation of a page's contents. Inspired from that work

[1] `https://developer.mozilla.org/en/XULRunner`

we consider all leaf DOM nodes, plus nodes resulting from the merge of sub-nodes with name equal to: “b”,“i”,“u”,“span”,“em”. Considering for example the DOM subtree:

```
<div><b>This <em>is</em> the
<span class="entity">ENTITY</span>
<u>of</u> interest</b></div>
```

the resulting blocks are:

- This
- is
- the
- ENTITY
- of
- interest
- This is the ENTITY of interest

where the latter stems from the merge of the sub-nodes with the “em”, “span” and “u” names.

3.3 Wrapper Induction with the Proposed Approach

The use of a web browser rendering engine in the LEWECOM algorithm implies the fact that each page must be processed completely, including the download of images, Cascading Style Sheets and client-side scripts, and more. This added task is absolutely required to let the algorithm work in a general setting, but can be avoided in some scenarios to improve both precision and speed. An average web page is composed by dozens of images, stylesheets and rich-medias that have to be loaded, taking up to 30 seconds or more in the worst case (see section 4.6 for more details).

In particular when the final aim is the extraction of data from a website, we can take advantage of redundant information found in each *content page* to improve the precision, recall and

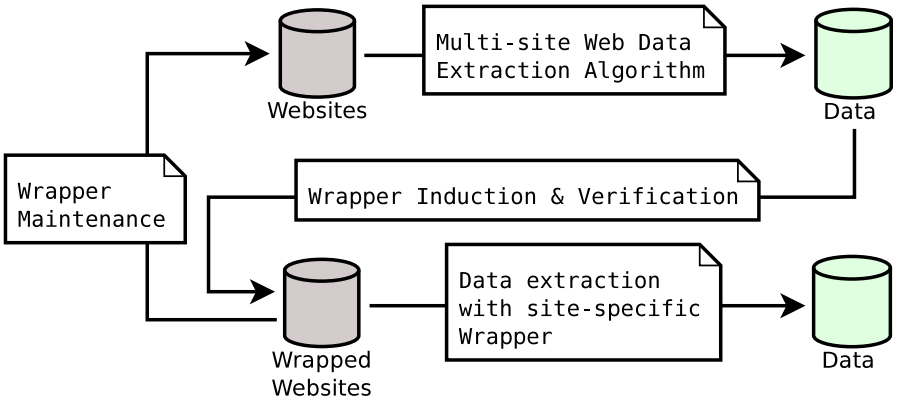


Figure 3.5 – *The overall schema integrating both the multi-site extraction algorithm and the Wrapper Induction technique described in this section.*

speed. Every and each of these parameters is of great impact in large-scale web-crawling processes where automatic and robust recognition of data is at the basis of complex web services. The integration of the LEWECOM multi-site data extraction algorithm in the complete process is sketched in figure 3.5.

For each web page w_j of web site W the LEWECOM algorithm produces a set of elements $e_j = \{n_1, \ldots, n_{|\Pi|}\}$ where n_i is the DOM node for the schema field π_i. A given web site W thus has a set of extracted data $E_W = \{e_1, \ldots, e_{|W|}\}$. The n_i elements can be also undefined nodes when LEWECOM decides that e_j is not a content page.

The aim of the LEWECOM Wrapper Induction (LWI) algorithm is to build a wrapper $wrapper_W$ s.t. $wrapper_W(w_j) \simeq e_j$, $\forall w_j \in W$ where $\simeq$ says that the equality relation holds true only in real *content pages*. In the following we refer to the XPath of an XML/DOM node n with $n.xpath$, and its text contents with $n.textContent$. The algorithm requires that all pages are in the XHTML standard, supporting older formats by converting them on the fly. The LWI algorithm works as follows:

1. The $e_j = \{n_1, \ldots, n_{|\pi|}\}$ element of each page is converted to $\tilde{e}_j = \{\chi_1, \ldots, \chi_{|\pi|}\}$ where $\chi_i = n.xpath$. The converted set is $\tilde{E}_W = \{\tilde{e}_1, \ldots, \tilde{e}_{|\pi|}\}$.

2. Our aim is to apply an XPath directly on the XHTML source to prevent a complete rendering, consequently since the HTML DOM2 is in principle different from the DOM

of the XHTML (e.g. new dynamic nodes can be added at runtime) it is necessary to find the XPath at source level for a given DOM node by finding the best matching node n' s.t. its distance xpath-ed$(n.xpath, n'.xpath)$ is minimal and $n.textContent == n'.textContent$. So $\hat{\chi}_i = n'_i.xpath \; \forall n_i \in e_j$. The distance xpath-ed is defined considering the amount of node names that have to be changed to make $n.xpath = n'.xpath$.

The aforementioned method defines the set $\hat{E}_W = \{\hat{e}_1, \ldots, \hat{e}_{|\pi|}\}$. This conversion however is not guaranteed to succeed, in particular when the node n has been added dynamically by external resources (e.g. Javascript). Consequently for each $e_j \in E_W$ we have $\hat{e}_j = \{\hat{\chi}_1, \ldots, \hat{\chi}_{|\pi|}\}$ if the conversion succeeds or $\hat{e}_j = \emptyset$ otherwise.

3. Given the quotient sets $\tilde{E}_W/ =$ and $\hat{E}_W/ =$, we find the most common XPath sets at HTML DOM level and at source level:

$$\tilde{e}_W = \tilde{e}_j, \; j = \text{argmax}_j |[\tilde{e}_j]|, \; \tilde{e}_j \neq \emptyset \tag{3.1}$$

$$\hat{e}_W = \hat{e}_j, \; j = \text{argmax}_j |[\hat{e}_j]|, \; \hat{e}_j \neq \emptyset \tag{3.2}$$

4. The $wrapper_W(w)$ is then built considering either $\tilde{e}_W$ or $\hat{e}_W$ to extract data. If $[\tilde{e}_j]| > |[\hat{e}_j]|$ then $\tilde{e}_W$ is considered to extract data from a web-rendered DOM. Otherwise $\hat{e}_W$ is considered to extract data from the XHTML source.

The LWI algorithm improves LEWECOM in both efficiency and effectiveness.

The efficiency is improved by reducing execution speed and running software complexity. The improved speed comes from the ability to recognize the data of interest in a given *content page* by simply analyzing the HTML source of the page. The running software complexity is orders of magnitude smaller since the wrapper runner algorithm is composed by an HTML-to-XML engine and an XPath evaluator. The simplicity and maintainability of software deployed in large-scale web agents is of great value in real-world scenarios where the speed of response to changes is a crucial issue.

The effectiveness is improved both in Precision and Recall [Frakes and Baeza-Yates, 1992]. The precision improvement over the LEWECOM standalone algorithm is given by the fact that

false-positive *content pages* where a non-empty data set was extracted are easily filtered out as noise by the grouping algorithm.

Since the LEWECOM considers as *content pages* all pages that have valid data for each field of interest, false positives can be found by the algorithm. The grouping LWI algorithm filters these false positives out. The improvement in Recall comes from cases when some *content pages* are not recognized as such by the LEWECOM standalone algorithm.

3.4 Multi Source Radial Basis Function Network

In this section we describe the ML model used to learn and recognize general *fields of interest* when anchortext features G are required. In particular, we adopted the Radial Basis Function Network (RBFN) model introduced by [Moody and Darken, 1989] for its proven training speed and robustness on classification and regression tasks. These capabilities are especially suitable for the inherent complexity in the WCM context. A detailed overview of RBFN is given in section 2.4.1.

In our context the entity to be classified is the web object b, with the additional context of its page p and the anchortext set A. As introduced in section 3.2.3, we define two main kinds of features that give rise to two separate feature spaces Θ and Γ. Since $\Theta \subset \mathbb{R}^n$ with $n = |F|$, the euclidean L_2 distance can be used in this space. The elements in the Γ space however are in the form $\gamma = \{\vec{x}_1, \ldots, \vec{x}_{|\gamma|}\}$ where $|\gamma|$ is not fixed. A suitable distance for such variable-length object is the Earth Mover's Distance (EMD) [Rubner et al., 2000], that has been developed for variable-length distributions. The γ object can be seen as a distribution modeling the relation of a textblock with respect to its anchors.

We describe the learning object (p, b, A) with $((\theta, \gamma), y)$ where θ is obtained from (b, p) with features $f \in F$ and γ is obtained from (A, b) with features $g \in G$ and $y \in \Omega,\ \Omega = \{\omega_1, \omega_2\}$ is the supervised label that defines whether $(y = \omega_1)$ or not $(y = \omega_2)$ a texblock b is a *main entity*.

RBFNs are a general purpose ML solution and its application in a specific problem domain implies the definition of a proper distance metric to learn from the feature space. For instance a RBFN has been adapted to Multi-Instance Learning problems by using the Haus-

dorff distance in [Zhang and Zhou, 2006], and in [Carullo et al., 2009] a content-based image soft-categorization algorithm has been defined by integrating the EMD [Rubner et al., 2000] in a RBFN.

It is non-trivial to define a true distance metric over Θ and Γ since they are heterogeneous and defined over two different spaces, with different distance metrics. Our idea is to circumvent the distance metric definition problem by letting the ML model to learn adaptively from the data how the two different feature spaces should be combined together. We thus define a novel ML model we name MS-RBFN (Multi Source Radial Basis Function Network) as a non-linear function $h : \Theta \times \Gamma \rightarrow \Omega$ that maps the problem space to the categories space as a result of the learning phase on the training set $TrS = \{((\theta_1, \gamma_1), y_1), \ldots, ((\theta_N, \gamma_N), y_N)\}$.

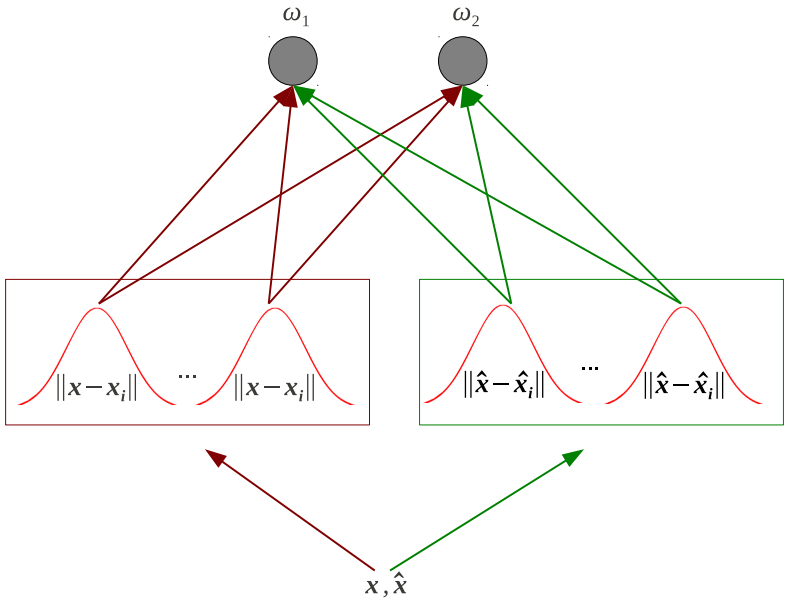


Figure 3.6 – *A schema of the Multi Source Radial Basis Function Network.*

A schematization of the network architecture is proposed in figure 3.6. The network is structured as follows:

1. a hybrid first level of:

   (a) M_1 units $\phi_i : \Theta \rightarrow \mathbb{R}$ to map the textblock feature space to the distance space w.r.t centroids μ_{θ_i}.

(b) M_2 units $\psi_j : \Gamma \to \mathbb{R}$ to map the anchortext feature space to the distance space w.r.t centroids μ_{γ_j}.

with:

$$\phi_i(\theta) = \exp(-\|\theta - \mu_{\theta_i}\|/\sigma_{\theta_i})$$
$$\psi_j(\gamma) = \exp(-\text{emd}(\gamma, \mu_{\gamma_j})/\sigma_{\gamma_j})$$

where μ_{θ_i} is the the i-th centroid in the Θ space, μ_{γ_j} is the j-th centroid in Γ space and σ_{θ_i} and σ_{γ_j} are the spreads of the basis functions for the Θ and Γ space, respectively. The $\|\cdot\|$ distance in ϕ_i is the euclidean distance and $\text{emd}(\cdot,\cdot)$ is the EMD distance.

2. a second level of linear weights:

$$\vec{w}_k = \{w_{k,1}, \ldots, w_{k,|\Gamma|}\}, k = 1, \ldots, M_1 + M_2$$

that connects each first level unit with each output unit.

3. the two levels are then linearly combined to build the model function f:

$$o_c(\gamma, \theta) = \sum_{i=1}^{M_1} \phi_i(\gamma) \cdot w_{i,c} + \sum_{j=1}^{M_2} \psi_j(\theta) \cdot w_{(j+M_1),c} \tag{3.3}$$
$$f(\gamma, \theta) = \text{argmax}_{c=1,\ldots,|\Omega|}\, o_c(\gamma, \theta) \tag{3.4}$$

The training scheme is two-phased as in the original RBFN [Moody and Darken, 1989]: one is unsupervised and selects $\mu_{\theta_i},\ i = 1, \ldots, M_1$ and $\mu_{\gamma_j},\ j = 1, \ldots, M_2$ while the other solves a linear problem to find values for $\vec{w}_k,\ k = 1, \ldots, M_1 + M_2$.

1. the first phase finds suitable centroids $\mu_{\theta_i},\ i = 1, \ldots, M_1$ by running the K-Means clustering algorithm with $K = M_1$. Then the p-means heuristic [Moody and Darken, 1989] is applied to compute the processing unit spreads $\sigma_{\theta_i},\ i = 1, \ldots, M_1$. Similarly the $\mu_{\gamma_j},\ j = 1, \ldots, M_2$ centroids are selected with an EMD-based K-Means algorithm with $K = M_2$ and the $\sigma_{\gamma_j},\ j = 1, \ldots, M_2$ values are set with the p-means heuristic.

2. the second phase is supervised and computes $\vec{w}_k,\ k = 1, \ldots, M_1 + M_2$ by minimizing the difference between predicted output and truth by Least Mean Squares:

(a) Φ is a $N \times (M_1 + M_2)$ matrix where $\Phi_{n,i} = \phi_i(\theta_n),\ i = 1, \ldots, M_1$ and $\Phi_{n,(M_1+j)} = \psi_j(\gamma_n),\ j = 1, \ldots, M_2$ with $n = 1, \ldots, N$.

(b) W is a $(M_1 + M_2) \times |\Omega|$ matrix where $W_{i,j} = w_{i,j}$.

(c) T is a $N \times |\Omega|$ matrix where $T_i = \hat{\mathbf{y}}_i$.

the minimization problem to solve is $\Phi W = T$ and thus $W = \Phi^{\dagger} T$, where $\Phi^{\dagger}$ is the pseudoinverse.

The model has therefore three user parameters:

1. the M_1 value for first level local processing units θ_i
2. the M_2 value for first level local processing units γ_j
3. the value π of the p-means heuristic, used to determine the spread of first level processing units.

3.5 Summary: Mind Map

The proposed approach has been summarized in the mind map of figure 3.7.

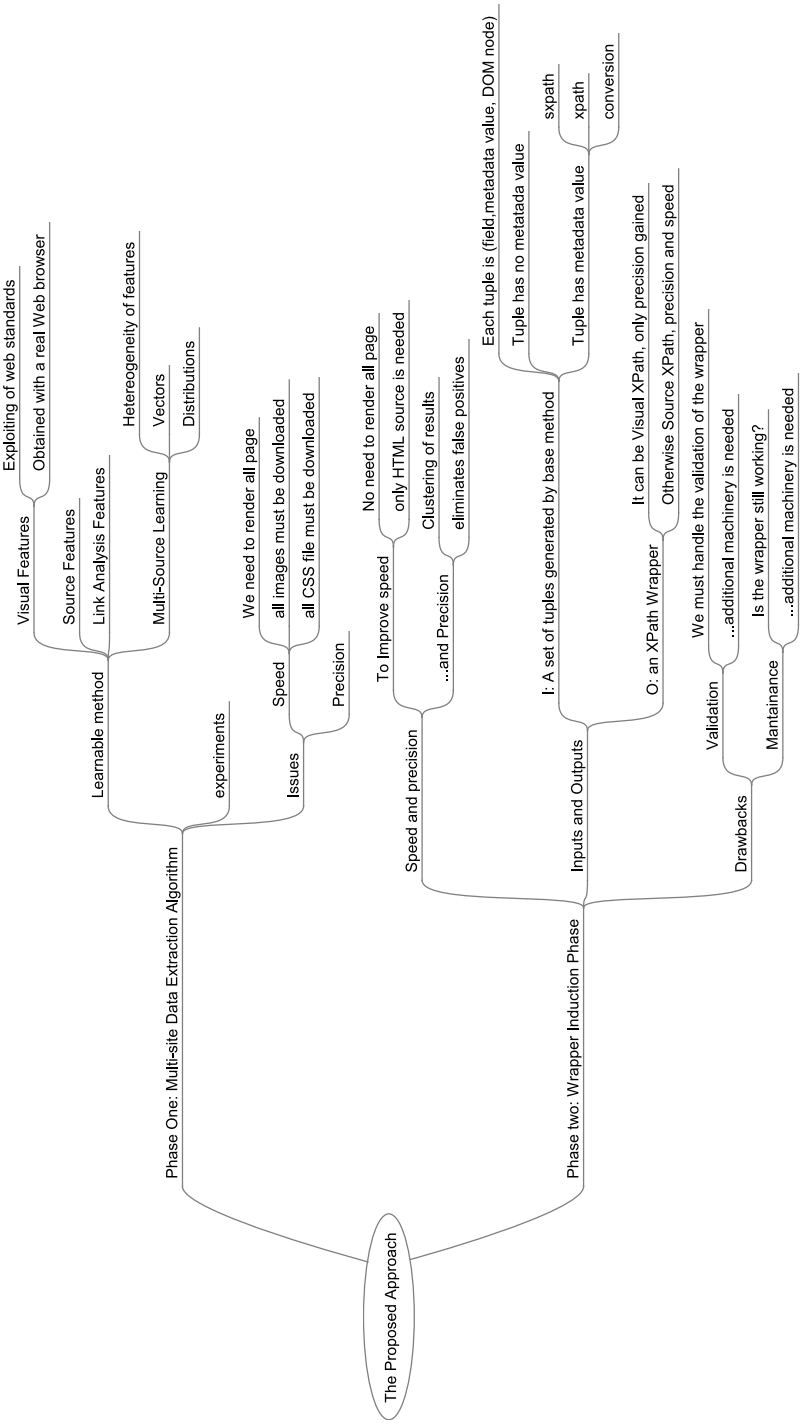


Figure 3.7 – *Mind map of the proposed approach.*

Chapter 4

Experimental Evaluation

In this section we report the experimental evaluation of the effectiveness of the proposed approach as an automated Web Content Mining (WCM) system able to operate on real case workload. In particular the experiments address these questions:

1. evaluate the fitness of multi-site WCM when generalizing common layouts of proper *content pages*.

2. quantify whether or not the combined use of anchortext and web object features is able to improve the recognition process. This step also validates the effectiveness of the proposed Machine-Learning model.

3. isolate the contribution of non-conventional visual features.

4. evaluate the contribution of the proposed Wrapper Induction method in the improvement of the overall strategy.

4.1 Evaluation Metrics

Usually standard evaluation metrics such as error matrix and derived measures [Congalton, 1991] can be adopted to evaluate classification results. However, such metrics are not able to directly evaluate how well the system is able to recognize a given *field of interest* in a given web page.

This can be assessed considering the well-known Information Retrieval metrics Precision (P), Recall (R) and F-Measure (F_β) [Frakes and Baeza-Yates, 1992]:

$$P = \frac{correct}{actual} \tag{4.1}$$

$$R = \frac{correct}{possible} \tag{4.2}$$

$$F_\beta = \frac{(\beta^2 + 1.0) \times P \times R}{\beta^2 \times P + R} \tag{4.3}$$

where *correct* is the number of *field of interest* instances correctly recognized by the system, *actual* is the total number of web objects recognized as *field of interest* by the system, and *possible* is the total number of *field of interest* objects we expected from system. The metrics were evaluated with a macro-average approach, and the F-Measure F_β was used with equal weight for P and R ($\beta = 1$).

4.2 Datasets

The experimental assessment of the proposed approach is based considering three different datasets. The first two datasets we name COMMOFF-1 and WEBNEWS-1 respectively, have been built to analyze WCM when no hyperlink information is available. The third dataset we name COMMOFF-LA is from the web news domain.

We decided to collect and publish datasets used in the experiments since to our knowledge there are no public domain resources on the WCM problem that apply to our paradigm. Each dataset is provided as a plain text file with the URI of each web page and the the set of *fields of interest*. Web site pages are complex and dynamic: advertisements, images, dynamic contents can appear in very different ways in each rendering of each page. Due to this a snapshot of each URI has been taken for each dataset that allows the reproduction of all the HTTP requests as performed during our experiments.

The WEBNEWS-1 dataset was collected on 29 daily news websites with a total dataset dimension of 310 pages. The training set TrS_{news} has 207 pages. The total number of images in the dataset is 13192. The truth set W was built with the help of a script that properly manipulates the RSS feeds of the selected websites.

Figure 4.1 – *Sample pages from the WEBNEWS-1 dataset. The chosen news websites have been selected looking for mostly different page layouts in order to make the learned model more general.*

Figure 4.2 – *Sample pages from the COMMOFF-1 dataset. The 600 selected e-commerce websites have been selected trying to maximize the heterogeneity of layouts. Many websites present difficult situations e.g. where advertised similar products can be mistakenly recognized as the main offering of the page.*

Table 4.1 – *Sample documents from the e-commerce dataset*

Expected Main Entity (e)	Anchor Set (A)
"IKEA STOCKHOLM"	"IKEA STOCKHOLM rahi"
	"IKEA STOCKHOLM rahi 199"
"Sweet Time SY.6231M/26 Chrono Man"	"Put in your basket"
	"Sweet Time SY.6231M/26 Chrono Man"
"Marioneta enanito 35 cm"	"MARIONETA ANÃO 35 CM"

The COMMOFF-1 dataset was collected on 600 e-commerce websites with a total dataset dimension of 1200 pages. The training set TrS_{offers} has 800 pages. Each web page of the dataset has been annotated considering a schema $\Pi = \{\pi_{image}, \pi_{productname}, \pi_{price}\}$ with the first being the main product image. The total number of images in the dataset is 61692. In figure 4.2 two pages of the dataset are reported to exemplify the heterogeneity of considered websites.

The COMMOFF-LA dataset is aimed at evaluating the combined use of visual features and link analysis together with the MS-RBFN Machine Learning model introduced in 3.4. Since to the best of our knowledge no suitable datasets are available for this problem, we have built and published a dataset for the e-commerce domain. The dataset was collected from 51 European e-commerce websites and is composed of 822 web pages, with a total textblock count of 172937 and a total anchor count of 1676. It can be obtained from our website [1]. Each web page has a mean of 2 incoming links and one supervised main-entity. The set A was built considering a complete crawl of the website and thus downward, upward and crosswise hyperlinks were considered [Spertus, 1997]. Both text links and images were considered to build the A set, where the anchor text and the alternative text of the image were used respectively. In Table 4.1 we report some data from the (p, e, A) tuples.

[1] `http://www.dicom.uninsubria.it/~moreno.carullo/thesis/la/`

4.3 Experimental Configuration

In our experiments we used a letter-digit tokenizer for the features, defined as a simple automata splitting contiguous sequences of alphabetic or numerical characters. For example the string "this is the 1st" is splitted as "this","is","the","1","st". Further details on how the tokenizer deals with the feature extraction process, see section 3.2.3.

The Machine Learning model used in the experiments without hyperlink information is the Support Vector Machine. The Support Vector Machine with RBF kernel has two main parameters [Burges, 1998]: the error penalty parameter C and the kernel parameter γ that regulates the spread of the basis functions. The two parameters are selected by cross validation on the train part of the dataset when using the full set of features $F = \{f_{\text{x}}, f_{\text{y}}, f_{\text{w}}, f_{\text{h}}, f_{\text{type}}, f_{\text{name}}, f_{\text{url}}\}$. The resulting parameters used with the WEBNEWS-1 dataset are $C = 2, \gamma = 2$ while with the COMMOFF-1 dataset are $C = 2, \gamma = 8$.

4.4 Web Content Mining without Link Analysis

The effectiveness of our method with no hyperlink information can be measured on the two distinct kinds of *web object* recognized by the approach: text and images. We name *Image of Interest* the former kind of *field of interest* and *text of interest* the latter.

4.4.1 Image Of Interest Recognition

In this section we evaluate the performance of the proposed approach in the task of identifying semantically meaningful images within web pages, i.e. images that are directly related to page content and that we name *Image of Interest* (IOI). News harvesting for example is a challenging task, with the need to collect the title, abstract (or full content) and image for each article. Finding a content related image within the page for a given article exemplifies what we intend for IOI identification. A robust and systematic approach to the identification of the Image of Interest in the domain of web news is of great importance because alternative domain specific solutions suffer of some limitations; in particular RSS [W3C., 2000] are often implemented exposing partial information, and do not provide a standardized way of retrieving past entries.

Table 4.2 – *Web News: feature analysis results*

Id	Features	P	R	F_1
wn_1	x,y,w,h,type,name,intitle	0.97	0.94	0.96
wn_2	x,y,w,h,type	0.96	0.93	0.95
wn_3	x,y,w,h,type,intitle	0.96	0.92	0.94
wn_4	x,y,w,h,type,name	0.97	0.95	0.96
wn_5	y,w,h,type	0.92	0.86	0.89
wn_6	y,w,h,type,name	0.94	0.85	0.89
wn_7	y,w,h,type,name,intitle	0.94	0.85	0.89
wn_8	x,y,w,h	0.97	0.95	0.96
wn_9	w,h,type	0.85	0.68	0.76
wn_{10}	w,h,type,name	0.93	0.81	0.86
wn_{11}	w,h,type,intitle	0.85	0.68	0.76
wn_{12}	w,h,type,name,intitle	0.92	0.79	0.85
wn_{13}	type,name,intitle	0.92	0.12	0.21

Table 4.3 – *Commercial Offers: feature analysis results*

Id	Features	P	R	F_1
co_1	x,y,w,h,type,name,intitle	0.91	0.83	0.87
co_2	x,y,w,h,type	0.86	0.63	0.73
co_3	x,y,w,h,type,intitle	0.88	0.78	0.83
co_4	x,y,w,h,type,name	0.90	0.80	0.85
co_5	y,w,h,type	0.81	0.57	0.67
co_6	y,w,h,type,name	0.87	0.69	0.77
co_7	y,w,h,type,name,intitle	0.89	0.75	0.82
co_8	x,y,w,h	0.85	0.73	0.79
co_9	w,h,type	0.76	0.20	0.31
co_{10}	w,h,type,name	0.84	0.30	0.44
co_{11}	w,h,type,intitle	0.82	0.42	0.56
co_{12}	w,h,type,name,intitle	0.88	0.48	0.62
co_{13}	type,name,intitle	0.83	0.26	0.40

In the e-commerce domain, images together with offer names and price, are the relevant items to be identified within a huge number of diversified product offering web pages.

The feature analysis results for the COMMOFF-1 and WEBNEWS-1 datasets are proposed in tables 4.2 and 4.3. Each experiment is configured with a different feature set and is identified by an Id for the sake of convenience. Only the most significant experiments are reported. The contribution of the f_{X} and f_{Y} layout features is clearly isolated in the result tables: rows related to experiments $wn_i,\ co_i,\ i = 1, \ldots, 8$ report results obtained by using layout features while results in rows $wn_i,\ co_i,\ i = 9, \ldots, 13$ refer to experiments without their use.

The improvement given by the Fl features in the WEBNEWS-1 dataset permits the Recall to be boosted from $\sim$ .81 to $\sim$ .95 (see experiments wn_8 and wn_{10} from table 4.2) and the Precision to further improve its value. In the *Commercial Offers* dataset the Fl features improve precision in all experiments ($co_i,\ i = 1, \ldots, 8$ in table 4.3) from $\sim$ .48 up to $\sim$ .83 in the best case.

The difficulty of the two problems is different: the e-commerce domain requires a larger set of features to obtain satisfactory results, whereas the web news domain can be approached with fewer features. Consequently the boost given by the layout features Fl in the first domain yields

satisfactory results for real-world Web Content Mining applications, while in the second domain the improvement enhances Precision and Recall on already high scores.

The features f_{x} and f_{y} have similar discriminant power, and give their best when used in a pair. The image dimension features f_{w} and f_{h} probably contribute the most in the discrimination of the IOI class (see experiments wn_{12} - wn_{13} and co_{12} - co_{13}).

The f_{type}, f_{name} and f_{intitle} feature as expected improve both precision and recall, in particular in the Commercial Offers dataset.

The best feature set in both domains is $F_{best} = \{f_{\mathrm{x}}, f_{\mathrm{y}}, f_{\mathrm{w}}, f_{\mathrm{h}}, f_{\mathrm{type}}, f_{\mathrm{name}}, f_{\mathrm{intitle}}\}$ when considering the measure F_1, even though in the Web News dataset a smaller set of features brings to similar figures. A difference between the Commercial Offers and Web News dataset is that in the former domain there can be many product images with size and position similar to the IOI. The f_{intitle} feature can be useful to discriminate non-relevant images.

The validity of the feature set F_{best} can be considered an experimental evidence of the general web usability assumption introduced in section 3.1 and a suggestion on the kind of features required to solve the IOI problem in different domains. It is interesting to note how a relatively small set of features combined with robust machine learning techniques are able to solve the image identification problem. The well-known *curse of dimensionality* and overfitting problems [Duda et al., 2000] are likely to arise when the feature set is too large.

In order to compare our approach with solutions employed in literature we have developed the extraction strategy described in [McKeown et al., 2002] and tested on the WEBNEWS-1 dataset obtaining a .8 precision and recall. These experimental results are to be considered carefully since the extraction algorithm described by McKeown et al. is not detailed enough to build a real-world implementation. However, all recent works in literature confirm that rule-based systems for domain-specific extraction have questionable performance when the low-level heuristic assumptions change through time.

Table 4.4 – *Results*

Id	Features	P	R	F_1
$productname127$	fsize,fbold,intitle,smartintitle,x,y,equaltextto(img)	0.9201	0.7465	0.8243
$productname122$	fbold,smartintitle,x,y,equaltextto(img)	0.9659	0.7183	0.8239
$productname249$	fsize,smartintitle,x,y,equaltextto(img),dstfromentity(img)	0.9138	0.7465	0.8217
$productname123$	fsize,fbold,smartintitle,x,y,equaltextto(img)	0.9353	0.7324	0.8215
$productname223$	fsize,fbold,intitle,smartintitle,x,equaltextto(img),dstfromentity(img)	0.9405	0.7127	0.8109
$productname125$	fsize,intitle,smartintitle,x,y,equaltextto(img)	0.9211	0.7239	0.8107
$productname245$	fsize,intitle,x,y,equaltextto(img),dstfromentity(img)	0.8969	0.7352	0.8080
$productname247$	fsize,fbold,intitle,x,y,equaltextto(img),dstfromentity(img)	0.8969	0.7352	0.8080
$productname250$	fbold,smartintitle,x,y,equaltextto(img),dstfromentity(img)	0.9504	0.7014	0.8071
$productname254$	fbold,intitle,smartintitle,x,y,equaltextto(img),dstfromentity(img)	0.9401	0.7070	0.8071
$productname126$	fbold,intitle,smartintitle,x,y,equaltextto(img)	0.9366	0.7070	0.8058
$productname251$	fsize,fbold,smartintitle,x,y,equaltextto(img),dstfromentity(img)	0.8904	0.7324	0.8037
$productname219$	fsize,fbold,smartintitle,x,equaltextto(img),dstfromentity(img)	0.9500	0.6958	0.8033
$productname252$	intitle,smartintitle,x,y,equaltextto(img),dstfromentity(img)	0.9396	0.7014	0.8032
$productname221$	fsize,intitle,smartintitle,x,equaltextto(img),dstfromentity(img)	0.9570	0.6901	0.8020
$productname118$	fbold,intitle,x,y,equaltextto(img)	0.9533	0.6901	0.8007
$productname119$	fsize,fbold,intitle,x,y,equaltextto(img)	0.9164	0.7099	0.8000
$productname246$	fbold,intitle,x,y,equaltextto(img),dstfromentity(img)	0.9358	0.6986	0.8000
$productname235$	fsize,fbold,smartintitle,y,equaltextto(img),dstfromentity(img)	0.9291	0.7014	0.7994
$productname124$	intitle,smartintitle,x,y,equaltextto(img)	0.9459	0.6901	0.7980

4.4.2 Text of Interest Recognition

In this section we analyze the performance of the proposed approach on text data. In particular we consider the COMMOFF-1 dataset and the $\pi_{productname}$ and π_{price} fields of interest. In order to evaluate the usefulness of each feature combined with any other one, an exhaustive experimentation has been conduced with thousands of significative configurations involved. For the sake of readability only top-performing configurations are presented in table 4.4 and table 4.5.

The conducted experiments show that the combined use of source features and visual features allow to challenge the field of interest recognition with satisfactory performance. In table 4.6 and 4.7 we summarize the usefulness of each feature expressed as the number of experiments where the addition of the feature resulted in greater experiments and as average Precision, Recall and F-Measure gain. The importance of each feature is dependent on the considered field of interest: visual and source feature types contribute differently on the two datasets.

A first observation contributes to corroborate the hypothesis about the importance of visual features, in particular the user-perceived visual cues. The performance obtained considering only source-level features is far below ($P < 0.5$ and $R < 0.5$) the one obtained by the worst-performing configuration with visual features enabled. Paired with similar conclusions drawn in section 4.4.1 this experimental configuration confirms the validity of the browser-based rendering approach behind the proposed visual features set.

A second interesting observation can be drawn about the $f_{\text{dstfromentity}}$ and $f_{\text{equaltextto}}$ features. These features are able to introduce further context by allowing to bind the recognition of a field other(s). In particular there can be certain fields that are less difficult to recognize and learn, and one can add to the model these additional hooks to help learning. In our experiments on the COMMOFF-1 dataset both the π_{price} and $\pi_{productname}$ field recognition has shown notable performance gains by considering the distance from the π_{img} field (see also tables 4.6 and 4.7).

A third observation is about the π_{price} experiments. The use of the highly domain-specific f_{regex} feature was fundamental in the learning of an effective model. However the use of small, isolated domain-specific clues do not change the validity of the approach for general Web Content Mining extraction tasks since they can be isolated and controlled with an higher level

Table 4.5 – *Results*

Id	Features	P	R	F_1
$price63$	fsize,fbold,x,y,intitle,dstfromentity(img),isprice,fstroked	0.9133	0.7718	0.8366
$price47$	fsize,fbold,x,y,dstfromentity(img),isprice,fstroked	0.9040	0.7690	0.8311
$price46$	fbold,x,y,dstfromentity(img),isprice,fstroked	0.8867	0.7718	0.8253
$price60$	x,y,intitle,dstfromentity(img),isprice,fstroked	0.8889	0.7662	0.8230
$price62$	fbold,x,y,intitle,dstfromentity(img),isprice,fstroked	0.8885	0.7634	0.8212
$price55$	fsize,fbold,x,intitle,dstfromentity(img),isprice,fstroked	0.9250	0.7296	0.8157
$price15$	fsize,fbold,x,y,isprice,fstroked	0.8837	0.7493	0.8110
$price31$	fsize,fbold,x,y,intitle,isprice,fstroked	0.8837	0.7493	0.8110
$price44$	x,y,dstfromentity(img),isprice,fstroked	0.8758	0.7549	0.8109
$price39$	fsize,fbold,x,dstfromentity(img),isprice,fstroked	0.9242	0.7211	0.8101
$price43$	fsize,fbold,y,dstfromentity(img),isprice,fstroked	0.8938	0.7352	0.8068
$price45$	fsize,x,y,dstfromentity(img),isprice,fstroked	0.8993	0.7296	0.8056
$price59$	fsize,fbold,y,intitle,dstfromentity(img),isprice,fstroked	0.8904	0.7324	0.8037
$price30$	fbold,x,y,intitle,isprice,fstroked	0.8847	0.7352	0.8031
$price61$	fsize,x,y,intitle,dstfromentity(img),isprice,fstroked	0.8982	0.7211	0.8000
$price29$	fsize,x,y,intitle,isprice,fstroked	0.9004	0.7127	0.7956
$price28$	x,y,intitle,isprice,fstroked	0.8858	0.7211	0.7950
$price27$	fsize,fbold,y,intitle,isprice,fstroked	0.8801	0.7239	0.7944
$price13$	fsize,x,y,isprice,fstroked	0.8940	0.7127	0.7931
$price12$	x,y,isprice,fstroked	0.8793	0.7183	0.7907

Table 4.6 – *Feature usefulness for field of interest $\pi_{productname}$. The average gain in Precision, Recall and F-Measure are reported.*

Feature	δP	δR	δF_1	Useful %
equaltextto	0.074	0.200	0.205	84/84
smartintitle	0.023	0.161	0.164	75/84
intitle	0.018	0.153	0.156	69/84
fsize	-0.013	0.082	0.073	38/47
x	-0.013	0.079	0.066	74/77
y	-0.033	0.074	0.054	57/61
dstfromentity	-0.015	0.031	0.026	62/77
fbold	-0.027	0.029	0.022	69/85

Table 4.7 – *Feature usefulness for field of interest π_{price}. The average gain in Precision, Recall and F-Measure are reported.*

Feature	δP	δR	δF_1	Useful %
y	-0.010	0.152	0.105	28/28
x	0.016	0.106	0.082	28/28
dstfromentity	0.012	0.078	0.063	27/28
fbold	0.010	0.051	0.039	23/30
fsize	0.029	0.015	0.021	19/28
intitle	0.002	0.004	0.004	18/30

of control with respect to completely rule-based, handcrafted set of rules.

4.4.3 Minimal Training Analysis

This section evaluates the effect of the dimension $|D|$ of the available supervised training set D since the amount of manual work needed to setup and tune the proposed approach is important when considering it against simple rules-based solutions. In order to consider both the WEBNEWS-1 and COMMOFF-1 dataset we report the minimal training analysis for the *Image of Interest* experiment.

The D_{TeS} partitions used in section 4.4.1 are kept as they are, while the D_{TrS} partitions are properly resized to evaluate the impact of the reduction of the training set in the discrimination of the IOI field class. The feature set used for both datasets is the F_{best} described in the previous section. The main variable in each minimal training experiment is the amount of data ν used to actually train the system, expressed as a fraction of the original D_{TrS}. Given ν and D_{TrS} the steps are:

1. build TrS_ν by extracting r random elements from D_{TrS} where $r = \nu|D_{TrS}|$.
2. train the system on TrS_ν and evaluate on D_{TeS}.
3. repeat from 1) five times and record the average P,R and F_1.

Table 4.8 – *Experimental results on the e-commerce dataset. The MS-RBFN model was trained with* $M_1 = M_2 = 50$ *and* $\pi = 5$.

Feature set (G)	Feature set (F)	P	R	F_1
-	intitle, fsize, fbold	0.78	0.66	0.72
dice	intitle	0.87	0.81	0.83
dice	intitle, fsize, fbold	0.88	0.84	0.86

The experiments are performed considering $\nu \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}$ and the results of the minimal training experiments are reported in figures 4.3 and 4.4.

Considering the COMMOFF-1 dataset, with training set TrS_ν with $|TrS_\nu| = .6 * |TrS_{news}| \simeq 400$ the F_1 quality measure starts to reach a robust figure of 0.80, but with smaller datasets the performance degrades almost linearly. The WEBNEWS-1 dataset behaves in a stable manner also with small training sets, where the number of learned IOI is about 35.

The two domains have datasets of different dimensions, and consequently an absolute comparison between the figures is not appropriate. However by comparing results obtained with similar training set dimension one can note that the COMMOFF-1 dataset needs more elements to obtain satisfactory performance. This is related to the nature of e-commerce pages, where a larger set of images similar to the IOI class can appear.

4.5 Web Content Mining with Link Analysis

The results obtained by the overall set of experiments are reported in table 4.8. In the best configuration using the complete set of features we obtained a satisfactory result with a value of $F_1 = 0.86$. The precision value of $P = 0.88$ determines the ability to embrace the proposed solution in real-world scenarios. The addition of anchortext features is able to improve both P and R passing from an overall F_1 of 0.72 to 0.86. This latter observation deserves particular

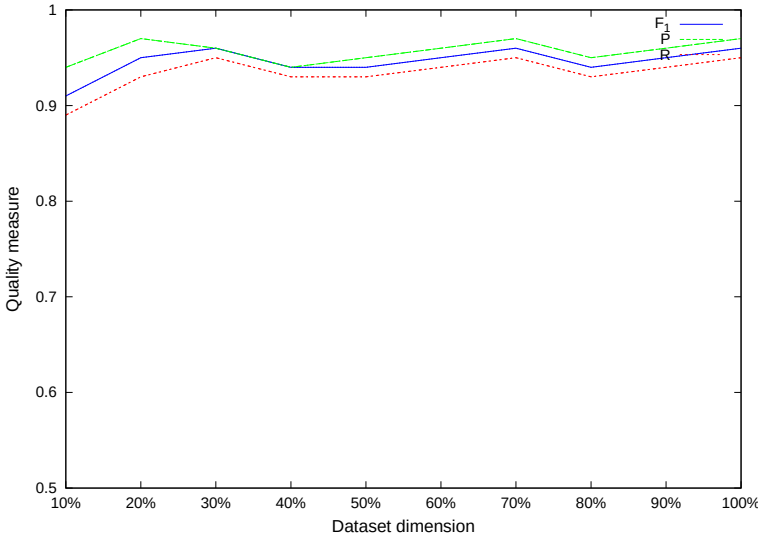


Figure 4.3 – *Experiments with dataset size of growing dimension on the News dataset.*

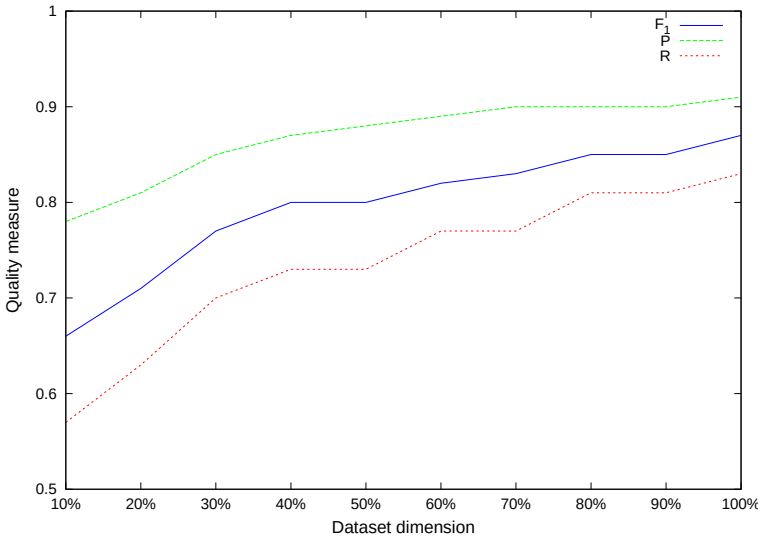


Figure 4.4 – *Experiments with dataset size of growing dimension on the Offers dataset.*

attention since it justifies the definition of the proposed novel ML model.

Experimental results show that the use of the f_{fsize} and f_{fbold} visual features enable a further improvement of the recognition performance, in particular the Recall value that goes from 0.81 to 0.84. This latter experiment can be considered an integration test where the whole set of proposed techniques is combined together in the definition of a complete Web Content Mining technique coped with Web Structure Mining information.

4.6 Case Study: An E-commerce Web Crawler

The overall Web Content Mining approach has been applied to a real-world web crawling process in the e-commerce domain. In particular the problem was to collect automatically from the web and with minimal human aid a big amount of commercial offers from various e-commerce websites unknown a priori.

A self-organizing, distributed and scalable web crawling architecture has been designed and engineered in cooperation with 7Pixel [2] for its price comparison website ShoppyDoo [3]. The software has been scheduled for deployment by the end of Q2 2011.

More than two hundred e-commerce websites in several different languages have been inserted into the system as a first step in a progressive growth of websites number. The real-world operation of the system let us evaluate some additional aspects of the proposed WCM approach, in particular the impact on both speed and quality of the Wrapper Induction process described in section 3.3.

The observed extracted data on random control samples is convincing and confirms the quantitative numerical investigation detailed in the previous sections. Of course (very few) false positive occurs but can be handled by the process with a feedback loop when a manual operator notices them.

The wrapper induction process works as expected, providing a way to speedup the extraction process on the large majority of websites where the data of interest is contained directly in the source of the page. As a reference the average time needed to process a web page (download

[2] `http://www.7pixel.it`

[3] `http://www.shoppydoo.com`

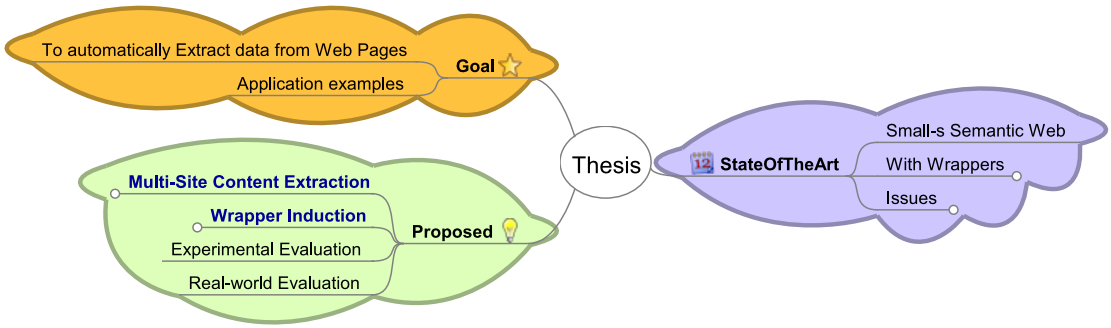
all required contents and attempt to extract data) is lowered from $[5; 30]$ s of the complete page rendering to $[0.1; 1.0]$ s of the source-only wrapper extraction.

The method also works on those websites where the fields of interest are computed dynamically via scripting, with an improvement in the elimination of false positives only. Content pages are the most frequent kinds on e-commerce websites and consequently the ranking strategy used to *induce* the wrapper works as long as the ranking is performed at the end of the first complete crawl. In the operative web crawler system we have introduced this constraint since a premature application of the induction process produced false-positive wrappers (e.g. category listing pages were wrapped).

During the real-world qualitative experimentation we have taken into account an aggressive caching strategy for outgoing HTTP requests by customizing a well-known web caching system [Wessels, 2001] to drop external tracking websites (e.g. Google Analytics), videos, sounds and by forcing graphic files to be cached for a wider time window. Such a caching system has enabled the Machine Learning-only strategy to work considerably faster than before, even though not reaching the order-of-magnitude improvement given by the wrapper induction process.

Chapter 5

Conclusions



The need of structured and semantic-carrying data on the Web is driving the change toward the adoption of Semantic Web standards and protocols. The large-scale deployment of such technologies greatly simplifies the effort required to gather structured data and also permits to perform further reasonings because of the relationships among data that can be expressed.

However the diffusion of such standards is still immature because of the many proposed alternatives and the initial instability of standards. Therefore intelligent techniques able to adapt on the evolving data of the Web are required for high-volumed structured data extraction purposes. In this work we have presented a complete strategy for Web Content Mining aimed at automatizing the collection of data from the web, given a specific page topic.

The key aspects of our approach are: a) the use of inductive techniques to build a *topic* specific instead of a *site* specific model b) the use of visual features from a web browser perspective c) the integration of hyperlink features d) the use of a novel Machine Learning model

e) the use of wrapper induction algorithm able to clean noisy data and improve recognition speed. The combined use of visual features, source features and hyperlink features together with a novel Machine Learning model able to integrate heterogeneous sources has proven to be an effective solution through the extensive experimental analysis on real-world data extraction problems. The proposed approach permits to define an adaptive model able to recognize data within a specific *topic* that can be trained in a reasonable amount of time and with very low human intervention.

The proposed solution has been engineered into a complete web crawling system for automatic e-commerce offers extraction, providing a working proof of the ideas proposed in this thesis. The e-commerce and web news scenario considered during the experimental analysis are quite different making the proposed approach a good candidate for the more general Web Content Mining problem. The development of the visual recognition process allows to mimic the process performed by humans when looking for information on the web at the cost of additional network transfers and longer preprocessing required for adequate page rendering. The proposed two-phased approach includes an integrated Wrapper Induction technique that permits to eliminate the need to compute complex page renderings after a proper warm-up phase on each website. The real-world case study confirms this theory on a large set of unseen websites.

The main drawback of the solution is given by the necessity of building a large dataset in order to build a robust model on unseen websites. However as suggested this issue can be circumvented by building a relatively small set of manual wrappers for a controlled set of websites and by using the extracted data as ground truth.

Another issue of the proposed approach is represented by the warm-up time required to site-specific wrappers that permit to extract data records from a given website with extraction speed limited only by download time of the page source. Although aggressive caching strategies can significantly speed the extraction process as described in the experimental section, whole-page renderings still require an order of magnitude more than source-only extraction.

Future works include the experimentation of the proposed approach on a wider set of Web Content Mining topics, the application of the proposed multi-source Radial Basis Function Network to other Machine Learning problems where heterogeneous features are available.

Colophon

This Ph.D. thesis has been written with the LaTeX typesetting system for its robustness and simplicity in the handling of large, complex books and papers full of mathematical stuff. It has mainly written from my Ubuntu workstation, but also from a Dell notebook and a MacBook Pro.

All the software - except support scripts for running experiments, plotting, etc - were developed embracing an Agile Process [1], in particular by applying Test Driven Development (TDD), Continuous Integration and Pair Programming (for some parts of the software). I think the final quality of the software is greatly due to rigorous application of TDD and the continuous code review process that Pair Programming introduces.

Many different programs have been involved in this thesis, both for typesetting and development. Make, Emacs, Gedit, Bash, Octave, Gnuplot and several others Unix tools have contributed to build the final document by automatizing tedious and repetitive routines. All the "working software" has been developed in the C# language for the .NET Framework using Microsoft Visual Studio 2008 [2]. Day-to-day refactorings were not feasible without the help of the fabulous Resharper [3].

[1] http://www.c2.com/cgi/wiki?AgileProcesses

[2] http://www.microsoft.com/visualstudio/en-us

[3] http://www.jetbrains.com/resharper/

Bibliography

[Adida, 2008] Adida, B. (2008). hgrddl: Bridging microformats and rdfa. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):54–60.

[Andrews et al., 2003] Andrews, S., Tsochantaridis, I., and Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, pages 561–568. MIT Press.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

[Bishop et al., 1975] Bishop, Y., Fienberg, S., and Holland, P. (1975). *Discrete Multivariate Analysis – Theory and Practice*. MIT Press, Cambridge, MA.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*.

[Broomhead and Lowe, 1988] Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355.

[Burges, 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

[Cai et al., 2003] Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. (2003). Extracting content structure for web pages based on visual representation. page 596.

[Carullo et al., 2009] Carullo, M., Binaghi, E., and Gallo, I. (2009). Soft categorization and annotation of images with radial basis function networks. In *VISAPP, International Conference on Computer Vision Theory and Applications*, volume 2, pages 309–314.

[Chakrabarti et al., 1998] Chakrabarti, S., Dom, B., and Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 307–318, New York, NY, USA. ACM.

[Chang et al., 2006] Chang, Kayed, M., Member-Girgis, and Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE Trans. on Knowl. and Data Eng.*, 18(10):1411–1428.

[Congalton, 1991] Congalton, R. (1991). A review of assessing the accuracy of classifications of remotely sensed data. *Remote sensing of environment*, 37(1):35–46.

[Dietterich and Lathrop, 1997] Dietterich, T. G. and Lathrop, R. H. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71.

[Duda et al., 2000] Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience Publication.

[Ecma International, 2009] Ecma International (2009). Ecmascript (Javascript). `http://www.ecma-international.org/publications/standards/Ecma-262.htm`.

[Fiumara, 2007] Fiumara, G. (2007). Automated information extraction from web sources: a survey. In *Proceedings of Between Ontologies and Folksonomies Workshop in 3rd International Conference on Communities and Technology*.

[Frakes and Baeza-Yates, 1992] Frakes, W. B. and Baeza-Yates, R. A., editors (1992). *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall.

[Fürnkranz, 2002] Fürnkranz, J. (2002). Web structure mining - exploiting the graph structure of the world-wide web. *ÖGAI Journal*, 21(2):17–26.

[Gehrke and Turban, 1999] Gehrke, D. and Turban, E. (1999). Determinants of successful website design: Relative importance and recommendations for effectiveness. In *Thirty-second Annual Hawaii International Conference on System Sciences*, page 5042. IEEE Computer Society.

[Halkidi et al., 2003] Halkidi, M., Nguyen, B., Varlamis, I., and Vazirgiannis, M. (2003). Thesus: Organizing web document collections based on link semantics. *The VLDB Journal*, 12:320–332.

[Hammer et al., 1997] Hammer, J., McHugh, J., and Garcia-Molina, H. (1997). Semistructured data: The tsimmis experience. In *First East-European Workshop on Advances in Databases and Information Systems-ADBIS '97,*.

[Hartman et al., 1990] Hartman, E., Keeler, J. D., and Kowalski, J. M. (1990). Layered neural networks with gaussian hidden units as universal approximations. *Neural Comput.*, 2(2):210–215.

[Jain, 1999] Jain, A. K. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323.

[Joachims et al., 2001] Joachims, T., De, T. J., Cristianini, N., and Uk, N. R. A. (2001). Composite kernels for hypertext categorisation. In *In Proceedings of the International Conference on Machine Learning (ICML*, pages 250–257. Morgan Kaufmann Publishers.

[Karayiannis, 1999] Karayiannis, N. B. (1999). Reformulated radial basis neural networks trained by gradient descent. *IEEE Transactions on Neural Networks*, 10(3):657–671.

[Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

[Kosala and Blockeel, 2000] Kosala, R. and Blockeel, H. (2000). Web mining research: a survey. *SIGKDD Explor. Newsl.*, 2(1):1–15.

[Kuhlins and Tredwell, 2003] Kuhlins, S. and Tredwell, R. (2003). Toolkits for generating wrappers. In *NODe '02: Revised Papers from the International Conference NetObjectDays*

on Objects, Components, Architectures, Services, and Applications for a Networked World, pages 184–198, London, UK. Springer-Verlag.

[Kushmerick, 1997] Kushmerick, N. (1997). *Wrapper induction for information extraction*. PhD thesis, University of Washington. Chairperson-Weld, Daniel S.

[Kushmerick, 1999] Kushmerick, N. (1999). Regression testing for wrapper maintenance. In *AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pages 74–79, Menlo Park, CA, USA. American Association for Artificial Intelligence.

[Kushmerick, 2000] Kushmerick, N. (2000). Wrapper verification. *World Wide Web*, 3(2):79–94.

[Laender et al., 2002] Laender, A. H. F., Ribeiro-Neto, B. A., da Silva, A. S., and Teixeira, J. S. (2002). A brief survey of web data extraction tools. *SIGMOD Rec.*, 31(2):84–93.

[Maekawa et al., 2006] Maekawa, T., Hara, T., and Nishio, S. (2006). Image classification for mobile web browsing. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 43–52, New York, NY, USA. ACM.

[Maron and Ratan, 1998] Maron, O. and Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 341–349, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[McKeown et al., 2002] McKeown, K. R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J. L., Nenkova, A., Sable, C., Schiffman, B., and Sigelman, S. (2002). Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the second international conference on Human Language Technology Research*, pages 280–285, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Michalski et al., 1983] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (1983). *Machine Learning, An Artificial Intelligence Approach*. McGraw-Hill.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

[Moody and Darken, 1989] Moody, J. E. and Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294.

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes*, 30(1):1–20.

[Nielsen, 2001] Nielsen, J. (2001). *Designing Web Usability*. New Riders Publishing.

[NIST, a] NIST. Automatic Content Extraction (ACE). `http://www.itl.nist.gov/iaui/894.01/tests/ace/`.

[NIST, b] NIST. Message Understanding Conference (MUC). `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/`.

[Oh et al., 2000] Oh, H.-J., Myaeng, S. H., and Lee, M.-H. (2000). A practical hypertext catergorization method using links and incrementally available class information. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 264–271, New York, NY, USA. ACM.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

[Park and Sandberg, 1993] Park, J. and Sandberg, I. W. (1993). Approximation and radial-basis-function networks. *Neural Comput.*, 5(2):305–316.

[Powell, 1987] Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: a review. pages 143–167.

[Reis et al., 2004] Reis, D. C., Golgher, P. B., Silva, A. S., and Laender, A. F. (2004). Automatic web news extraction using tree edit distance. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 502–511, New York, NY, USA. ACM.

[Rubner et al., 2000] Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121.

[Sarawagi, 2002] Sarawagi, S. (2002). Automation in information extraction and integration. In *The 28th International Conference on Very Large Data Bases (VLDB)*.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

[Spertus, 1997] Spertus, E. (1997). Parasite: mining structural information on the web. *Comput. Netw. ISDN Syst.*, 29(8-13):1205–1215.

[W3C., 1997] W3C. (1997). The Document Object Model. `http://www.w3.org/DOM/`.

[W3C., 1999a] W3C. (1999a). Html markup language. `http://www.w3.org/TR/1999/REC-html401-19991224//`.

[W3C., 1999b] W3C. (1999b). Xml path language. `http://www.w3.org/TR/xpath`.

[W3C., 2000] W3C. (2000). RSS 2.0 specification. `http://validator.w3.org/feed/docs/rss2.html`.

[W3C., 2003] W3C. (2003). Document Object Model (DOM) Level 2 HTML Specification. `http://www.w3.org/TR/DOM-Level-2-HTML/`.

[W3C., 2008] W3C. (2008). RDFa in XHTML: Syntax and Processing. `http://www.w3.org/TR/rdfa-syntax/`.

[W3C., 2009] W3C. (2009). Cascading style sheets. `http://www.w3.org/TR/CSS2/`.

[W3C., 2010] W3C. (2010). XHTML 1.1 - Module-based XHTML - Second Edition. `http://www.w3.org/TR/xhtml11/`.

[Wessels, 2001] Wessels, D. (2001). Squid Internet Object Cache. `http://www.squid-cache.org/`.

[Zhang and Zhou, 2006] Zhang, M.-L. and Zhou, Z.-H. (2006). Adapting rbf neural networks to multi-instance learning. *Neural Process. Lett.*, 23(1):1–26.

[Zhou et al., 2005] Zhou, Z.-H., Jiang, K., and Li, M. (2005). Multi-instance learning based web mining. *Applied Intelligence*, 22:135–147.

[Zhou and Xu, 2007] Zhou, Z.-H. and Xu, J.-M. (2007). On the relation between multi-instance learning and semi-supervised learning. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 1167–1174, New York, NY, USA. ACM.

[Zhu, 2005] Zhu, X. (2005). Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison.