MACHINE LEARNING AND LINK ANALYSIS FOR WEB CONTENT MINING

Moreno Carullo and Elisabetta Binaghi
Department of Computer Science and Communication, University of Insubria, 21100 Varese, Italy

Keywords: Web content mining, Hyperlinks, Machine learning, Radial basis function networks.

Abstract: In this work we define a hybrid Web Content Mining strategy aimed to recognize within Web pages the main entity, intended as the short text that refers directly to the main topic of a given page. The salient aspect of the strategy is the use of a novel supervised Machine Learning model able to represent in an unified framework the integrated use of visual pages layout features, textual features and hyperlink description. The proposed approach has been evaluated with promising results.

1 INTRODUCTION

The web content is huge and dynamic, mainly constituted by unstructured, partially-structured text data or by low-volumed structured data having different non-standard scheme. Web documents are characterized by both their heterogeneous contents represented with different media and format, and the organization of these contents within graphical page layouts more and more consistent with usability guidelines. Server logs information, link structure and description constitute additional relevant information. All these aspects make the extraction of useful knowledge a challenging research problem.

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 and the goal of the learning algorithm is to predict the output values of never seen input values. In unsupervised learning training examples are constituted only by input patterns; the learning algorithm is able to generalize from input patterns to discover similarities among data (Michalski et al., 1983; Mitchell, 1997).

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).

Proceeding from these considerations, we intend to define and experimentally investigate a hybrid WCM strategy aimed to recognize within Web pages the *main entity*, intended as the short text that refers directly to the main topic of a given page. We consider this application of great importance for the building and the maintenance of intelligent web agents oriented to advanced web-based user services. The salient aspect of the strategy is the use of a novel supervised ML model able to represent in an unified framework the integrated use of visual layout features, textual features and hyperlink descriptions.

Carullo M. and Binaghi E..
MACHINE LEARNING AND LINK ANALYSIS FOR WEB CONTENT MINING.
DOI: 10.5220/0003065401560161
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2010), pages 156-161
ISBN: 978-989-8425-28-7

2 WEB MAIN ENTITY RECOGNITION

Every and each page in the web has a definite topic that can be explicitly declared in the page itself with a short text 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.

Each page p can be modeled as a collection of *text entities* B: where the *main entity* $e \in B$ is the one best describing the topic or title of the page. In addition 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.

In the *main entity recognition problem* (MERP) the goal is to consider a page p and the set of Incoming Link Anchor Texts (ILA) $A = \{a_1, \ldots, a_n\}$ and to discover the text block $\hat{e}$ recognized as the most plausible *main entity* in the $H \subset B$ set of candidate *main entity* blocks.

Suppose we have the possibility to collect a supervised truth for each web page, that is a set D of tuples (p, e, A) where p is a web page in a given domain, e is the *main entity* and A is the set of ILA for p. The supervised truth can be collected by providing domain experts an environment where they can select the *main entity* $e \in B$ for each p.

MERP can be seen as a supervised classification task where the object to classify is $(b, A) \in \mathbb{B} \times \mathbb{A}$ and the candidate labels are "main entity" and "non main entity". The main issue lies in the fact that A is a set of elements of variable cardinality while b is a single element. In order to allow a ML model to discover the relationship between (b, A) and the label, a proper low-level representation of features is supposed to take place.

3 THE PROPOSED APPROACH

The approach we propose 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 suitable ML model for MERP.

We model each web page p as a semi-structured source of information where $B = \{b_1, \ldots, b_m\}$, $B \subset \mathbb{B}$ is the set of text blocks in $p \in \mathbb{P}$, where $\mathbb{B}$ is the domain of all textblocks and $\mathbb{P}$ is the domain of pages. With the term "text block" we refer to a string of text that appears in p as a the inner text of a DOM [1] node. The set B is obtained through the $extract\_blocks : \mathbb{P} \rightarrow \mathbb{B}$ function, further explained in Sec. 3.4.

3.1 Feature Extraction

In the *main entity recognition problem* the object to classify is a text block $b \in B$ given its page p and the A set for p. We therefore partition the features in two subsets: textblock 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 $g : \mathbb{A} \times \mathbb{B} \rightarrow \mathbb{R}$.

Textblock 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:

- f_{intitle} : expresses the percentage of matching between the terms of the textblock 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 $b =$ "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_{fsize} : expresses the font size of the textblock b normalized w.r.t. a maximum value defined a priori. The idea behind this feature is that the *main entity* is often presented with a big font-size to be easily identified by the user.
- f_{fbold} : expresses whether or not the considered textblock b 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 the *main entity* is often presented with a bold font.

Anchortext features take into account anchors and textblocks 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 textblock b and the anchortext a. The employed similarity measure is the *Dice coefficient* (Frakes and Baeza-Yates, 1992), computed as:

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

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.

[1] http://www.w3.org/TR/DOM-Level-2-HTML/

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 $\Theta \subset \mathbb{R}^n$, $n = |F|$ textblock 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 A Multi-source Machine Learning Algorithm

In this section we describe the ML model used to solve the MERP. 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.

The MERP defines the textblock b as the object to be classified, with the additional context of its page p and the anchortext set A. As introduced in the previous section, we define two 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 w.r.t 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. In (Zhang and Zhou, 2006) for instance a RBFN has been adapted to Multi-Instance Learning problems by using the Hausdorff distance, 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 $h : \Theta \times \Gamma \to \Omega$ that maps the problem space to the categories space as a result of the learning phase on the training set $TrS = \{((\theta_1,\gamma_1),y_1),\ldots,((\theta_N,\gamma_N),y_N)\}$.

The network is structured as follows:

1. a hybrid first level of:
   (a) M_1 units $\phi_i : \Theta \to \mathbb{R}$ 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:

$$o_c(\gamma,\theta) = \sum_{i=1}^{M_1} \phi_i(\gamma)\cdot w_{i,c} + \sum_{j=1}^{M_2} \psi_j(\theta)\cdot w_{(j+M_1),c} \quad (1)$$

$$f(\gamma,\theta) = \text{argmax}_{c=1,\ldots,|\Omega|}\ o_c(\gamma,\theta) \quad (2)$$

The training scheme is two-phased as in the original RBFN (Moody and Darken, 1989): one is unsupervised and selects μ_{θ_i}, $i = 1,\ldots,M_1$ and μ_{γ_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 μ_{θ_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 $\Phi_{n,i} = \phi_i(\theta_n),\quad i = 1,\ldots,M_1$ and $\Phi_{n,(M_1+j)} = \psi_j(\gamma_n),\quad j = 1,\ldots,M_2$ 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.

3.3 Training and Generalization

Let us now consider the training phase of the algorithm and the step-by-step phases for the recognition of the *main entity*.

Given the supervised dataset D, the textblock features F, the anchortext features G and the MS-RBFN parameters M_1,M_2 and π, the first step is to train the algorithm with data extracted from D.

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 (*main entity*) if $b = e$ and to ω_2 (not *main entity*) otherwise.
      iv. Add the $((\hat{\theta},\hat{\gamma}),y)$ element to TrS.
4. Train MS-RBFN with TrS.

The model can then be used to solve the MERP using the generalization phase. Be (p,A) an input pair, and H the output set of recognized *main entities*:

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}.

3.4 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.

3.4.1 Page Rendering

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. 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 rendering engine program is based on XULRunner [2] 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.

3.4.2 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

[2] https://developer.mozilla.org/en/XULRunner

Table 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"

Table 2: 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

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". The latter block stems from the merge of the sub-nodes with the "em", "span" and "u" names.

4 EXPERIMENTS

In this section we report the experimental evaluation of the effectiveness of the proposed approach as an automated main entity recognition tool. In particular the experiments address the two main questions:

1. to quantify whether or not the combined use of anchortext and textblock features is able to improve the recognition process.
2. to isolate the contribution of non-conventional visual features.

Since to the best of our knowledge no suitable datasets are available for this problem, we have built and published an initial 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 [3]. 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 1 we report some data from the (p,e,A) tuples.

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 extracton process, see section 3.1.

4.1 Evaluation Metrics

Standard ML 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 the main entity 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) targeting the recognition of correct *main entities*. 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 Results

The results obtained by the overall set of experiments are reported in table 2. In the best configuration using

[3] http://www.dicom.uninsubria.it/ moreno.carullo/thesis/la/

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

Moreover 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.

5 CONCLUSIONS AND FUTURE WORKS

In this paper we have shown that ML techniques can be used in conjunction with LA to achieve automatic recognition of the *main entity* from pages with a given topic and well-known web-usability-driven structure. Experimental results show encouraging results on the proposed dataset and highlight the advantage of combining the two sources of information, text blocks with their visual formatting styles and incoming anchor texts. Future works include the experimentation on other website domains and the extension of the current set general purpose features with additional domain specific features.

REFERENCES

Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. (2003). Extracting content structure for web pages based on visual representation. In *Web Technologies and Applications: 5th Asia-Pacific Web Conference, APWeb 2003, Xian, China, April 23-25, 2003. Proceedings*, page 596.

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

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.

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

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

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

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.

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

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

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

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

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.

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.

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

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