TEXT CATEGORIZATION OF COMMERCIAL WEB PAGES

E. Binaghi, M. Carullo, I. Gallo and M. Madaio
Universitá degli Studi dell'Insubria
Via Mazzini 5, 21100 Varese, Italy
email: elisabetta.binaghi@uninsubria.it

ABSTRACT
In this paper we describe a new on-line document categorization strategy that can be integrated within Web applications. A salient aspect is the use of neural learning in both representation and classification tasks. Within text documents conceived as images, the regions of interest (RoI) containing information meaningful for categorization are identified with the support of a supervised neural network. Text within RoI is represented according to a simple solution that consider the first K words in the text and code them properly. A Kohonen Self-Organizing Map (SOM) is applied to cluster documents that are subsequently labelled by applying a simple majority voting mechanism. Solutions adopted were evaluated by conducting experiments within the context of on-line price comparison services. Results obtained demontrate that the overall classification strategy is able to categorize documents satisfectorily taking into account the high variability of Web pages.

KEY WORDS
Text categorization, Kohonen Self-Organizing Map, neural network, multilayer perceptron

1 Introduction

The World Wide Web is a great resource for all type of information and offers a high potential of efficient on-line services in several application domains. The on-line availability of ever larger numbers of commercial information, for example, creates the premise for profitable price comparison services allowing individual to see lists of prices for specific products.
However, the rapid growth of heterogeneous information usually coded in a relatively free text format poses a challenge to information management solutions that become more and more expensive and frustrating. The problem can be addressed by conceptually organizing the huge amount of data, forming content-based categories within which search and/or mining tools can be efficiently applied.
Automated Text Categorization (ATC) is a long-term research topic dealing with the task of building software capable of classifying text (or hypertext) documents under predefined categories. ATC techniques are the premise for improving web search engines in finding relevant documents and Web mining application [3].
There are several Machine Learning (ML) algorithms that have been successfully applied to text categorization. They include Neural Networks, Naïf Bayes, Support Vector Machine and k-Nearest Neighbors. Each of these methods has their advantages and limitations; the choice of the categorization algorithm depends upon many factors such as scale and dimensionality [6].
Considerable interest has been devoted to Self-Organized Maps [4] that approximate an unlimited number of input data items by a finite set of models. This property makes the SOM useful for organizing large collections of data in general, including document collections.

The representation of documents is a central issue in all of the approaches in ATC having a strong impact on the generalization accuracy of a learning system. The representation should be suitable for the classification task and for the specific learning algorithm adopted. For most of classification techniques, the documents, which typically are strings of characters, have to be transformed in quantitative, attribute-values patterns. This requirement agrees with the traditional document encoding method in Information Retrieval, called Salton's vector space model [5], which is based on the computation of the frequency of occurrence of each word in a document and its collection into a vector. This method has been widely and successfully used in several categorization tasks based on different learning strategies. However, it is impracticable to encode the documents in a large collection using the vector space model as such. Other techniques were proposed in literature alternative to the original vector space model or complementary. Heuristic problem driven preprocessing and/or feature selection strategies are included in an attempt to reduce the size of the vector space [3, 2].

The objective of our work is to design and implement a new on-line document categorization strategy named **Tc-system**. It makes use of neural learning in both representation and classification tasks. Web pages are conceived as images: within the overall image, the region of interest containing information meaningful for categorization is identified with the support of a supervised neural network. Text within RoI is represented according to a simple solution that consider the first k words in the text and code them properly. A second task based on Kohonen Self-Organizing Map is applied to cluster documents; clusters

are subsequently labelled by applying a simple majority voting mechanism.
Experiments were conceived and conducted within the context of price comparison services (also known as shopping comparison or price engine) allowing individuals to see lists of prices for specific products.

2 Preprocessing: automated RoI extraction

Web usability guidelines in the design of Web sites allow users don't waste time reading all items they see in a web page, and let them to localize immediately interesting information. Consistently with our cognitive and perceptual attitudes, textual information in a Web page are organized within a graphical layout. The Web-page is then perceived initially in a pictorial form and a region of interest (RoI) is immediately localized; then our experience and attitude let us to understand its meaning and interpret detailed textual information. We transport these concepts within our document modelling and representation strategy. Initially Web documents are viewed as images. Within the overall image, regions corresponding to textual sections composing the specific page layout, are identified. Subsequently a Neural Network is used to recognize the RoI, where the relevant information are located. Figure 1 illustrates the phases in which the RoI extraction process is articulated.

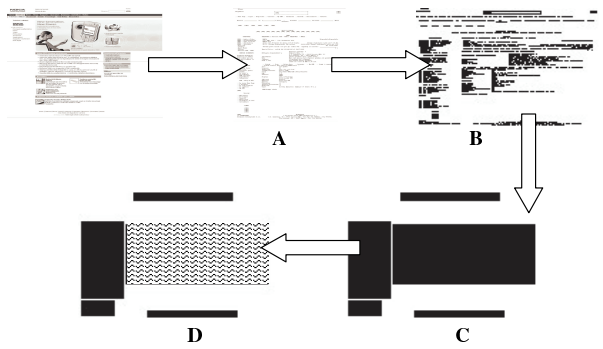

Figure 1. Main region determination: (A) the page is transformed in text; (B) text is converted in a binary image; (C) regions are delimited; (D) The RoI is detected and extracted.

2.1 Setting a page like an image

The preprocessing starts using a text-browser. It takes a web page and discards pictures, tags, fonts etc...(Fig. 1-A) The web-page is converted in an image format, where every character is a black pixel (Fig. 1-B). For our purposes, an image can just be thought as a binary function $f(x,y)$ of size $M \times N$

2.2 Searching borders

At this point we need to define a set of regions $\widehat{R} = \{R_1, R_2, ...R_I\}$ delimiting I areas in the image.

A window W of size $J \times K$ slides upon the image pixel after pixel obtaining a new-image g. Formally:

$$g(x,y) = \begin{cases} 1 & \text{if} \sum_{s=0}^{J} \sum_{t=0}^{K} f(x+s, y+t) > T \\ 0 & \text{else} \end{cases}$$

where T is a threshold heuristically assessed.

The sliding process is performed by using two windows W_v and W_h of size 12×1 and 2×12 respectively obtaining two new images $g_v(x,y)$ and $g_h(x,y)$ where respectively vertical borders and horizontal borders are emphasized. Window dimensions were heuristically assessed during the experiments.

We merge the vertical and the horizontal borders to find an unique borders-image $b(x,y)$. The merging is made by an OR operation: $b(x,y) = g_v(x,y)$ OR $g_h(x,y)$

2.3 Creating rectangular regions

The extracted regions are processed for approximating their shape to a rectangle. We do this with an algorithm we called "snake90" (see 1).

Algorithm 1 *Snake90* algorithm

repeat
  Select the first black point B_s starting from the top-left corner in the image;
  Identify the boundary of the region using the following strategy;
  $Direction \leftarrow Right$;
  repeat
    if $Direction = Right$ **then**
      try to go Up or *Right* or *Down*
    else if $Direction = Left$ **then**
      try to go *Down* or $Left$ or Up
    else if $Direction = Down$ **then**
      try to go *Right* or *Down* or $Left$
    else if $Direction = Up$ **then**
      try to go $Left$ or Up or *Right*
    end if
  until current point $\neq B_s$
  Extract the most left and highest point visited (x_1, y_1);
  Extract the most right and lowest point visited (x_2, y_2);
  Identify the rectangular region $R = ((x_1, y_1), (x_2, y_2))$ if it is not too small;
  Remove region R from the image;
until there are regions in the page

2.4 Region identification by neural network

The last step of the preprocessing stage is to extract the RoI.

Web users accomplish this task implicitely: it could be difficult to handcraft the set of rules undertaking the process. The problem can be addressed by automatically induce these rules from a set of examples. Several studies were conducted demostrating the great potential of neural networks in dealing with recognition tasks supported by visual inspection of images [1]. To this purpose we use a MultiLayer Perceptron neural model (MLP) trained by the standard Backropagation learning algorithm.

During training a supervised set of examples is provided to the network.
Formally a supervised training example is constituted by a pair of elements $(\overline{a}, \overline{b})$ where:
$\overline{a} = (R_1, ...R_I)$ (see eq.**??**) and
$\overline{b} = (o_1, ...o_I)$ with $o_j = 1$ if R_j is the RoI, $o_j = 0$ otherwise
.
The trained network receives in input a set of coordinates identifying regions extracted by the "'snake90 algorithm"'. In output, the region associated with the output neuron having the maximum activation value is interpreted as the RoI.

3 Text representation: First-K strategy

Preliminary to the text encoding phase, our strategy includes a module for the automatic construction of the dictionary. It collects all the words found in all the documents of the dataset, removing uncommon words and sorting all the meaningful ones in a proper file.
In this phase a string tokenizer acts on sequences of alphanumerics strings: it builds up a string of symbols until it finds a symbol belonging to a numbers (0-9) or letters (azA-Z). Consequently the resulting dictionary does not contain mixed words of numbers and letters. This approach is consistent with the nature of commercial documents: in which products are mentioned with identifiers crafted in many different ways (e.g BMW 320i, BMW 320-i, BMW 320 i).
Assuming that the RoI preliminarily identified during the preprocessing task contains text meaningful for classification purposes, we adopt a simple direct solution for encoding this text. The first K tokens found in the RoI are automatically selected and corresponding addresses in the dictionary are considered as feature values. A pattern is then constructed sorting the extracted feature values. The best value for K can be heuristically assessed for a specific context and types of documents. Patterns are provided in input to the machine learning algorithm for classification purpose.

4 Text classification

We have developed a classification strategy which is essentially based on the use of Self Organizing Map, one of the most distinguished unsupervised artificial neural network models which has shown great potentialities in the organization of on-line document archives [2]. It provides a form of cluster analysis by producing a mapping of high-dimensional input data onto a usually two dimensional output space while preserving the topological relationships between the input data items as much as possible. Each output unit i is associated with a model vector m_i representing input data. During the learning process, the model vectors change gradually so that the output map forms an ordered non -linear regression of the model vectors into the data space. During each iteration t , an input pattern, generated in our context by the first-k strategy, is presented to the system. The node c that best represents the input is then searched using the Euclidean distance between the input pattern and that unit. The winner unit is then adapted in such a way as to decrease the difference between that unit's weight vector m_c and the input pattern x. Adaptation takes place during each training iteration and is realized as the gradual reduction of the difference between the respective components of input and weight vector. The amount of adaptation is guided by means of a learning-rate that gradually decreases in the course of training. A predefined, time-varying neighbor of the winner units is also involved in the adaptation. The spatial range of units around the winner that are subject to adaptation may be described by means of a time-decreasing neighborhood function h_{ci} taking into account the distance between unit i currently under consideration and winner unit c. Formally:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \qquad (1)$$

During repeated applications of the adaptation rule with different input patterns, model vectors associated with neighboring map units become gradually similar. We designed the SOM system heuristically setting learning rate and neighboring function parameters. Clusters found by the SOM are associated with predefined categories by applying a simple majority voting mechanism which makes use of a minimal supervised set of examples.

5 Experiments

The experimental activity aimed to evaluate quantitatively the accuracy of the overall strategy in categorizing commercial web documents. Document collection is taken from the price comparison service Shoppydoo (http://uk.shoppydoo.com and http://www.shoppydoo.it) . The total number of documents collected is partitioned in eight categories: cell phone, video-camera, printer, palmtop, mp3 reader, camera, tv lcd and notebook. Experiments addressed the following main questions:

- how did the neural stage for RoI extraction contribute to the overall performance?
- how did the performance of the proposed tc-system depend upon the main parameters involved in the text representation strategy?

5.1 Performances of RoI extraction stage

In this section we describe the experiment aimed to quantify performances of the automated neural extraction of the RoI in the pre-processing stage. The MLP was trained with a supervised training set of 250 samples and accuracy was evaluated using a test set of supervised 94 samples. Table 1 reports the results obtained having set the max number of regions identifiable to 10. Accuracy values are acceptable taking into account the high variability of web page layout considered.

Table 1. Contingency table and evaluation measures for neural RoI. *region i* in row and corresponding *ri* in column identify the regions of interest selected by classifier and by supervisor respectively.

	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10	Row
region 1	28	2	1	0	0	0	0	0	0	0	31
region 2	0	12	0	0	0	0	0	0	0	0	12
region 3	1	1	10	0	0	0	0	0	0	0	12
region 4	2	1	1	20	0	0	0	0	0	0	24
region 5	0	0	0	1	5	0	0	0	0	0	6
region 6	0	0	0	1	1	1	0	0	0	0	3
region 7	0	0	0	0	0	0	1	0	0	0	1
region 8	0	0	0	0	0	0	0	4	0	0	4
region 9	0	0	0	0	0	0	0	0	1	0	1
region 10	0	0	0	0	0	0	0	0	0	0	0
Col	31	16	12	22	6	1	1	4	1	0	94
recall	0,9	0,75	0,83	0,91	0,83	1	1	1	1	1	
precision	0,9	1	0,83	0,83	0,83	0,33	1	1	1	1	
f-measure	0,9	0,86	0,83	0,87	0,83	0,5	1	1	1	1	

overall acc.	mic.av.recall	mic.av.precision
0,8723	0,9228	0,8736

5.2 Sensitivity analysis

We attempted to evaluate the effects of systematically varying the key parameter K in the text representation task performed by the first-k strategy in order to find an optimal setting for the application domain considered. Nine different configurations were considered for the first-k algorithm, distinguished by an increasing number of tokens extracted which assumed values from 10 to 50 with step 5. For these experiments we partitioned the documents available in training and test sets including a number of 5005 and 2502 data respectively.

Results obtained are shown in table 2 where overall accuracy values (OA) obtained by the tc-system using the different configuration of the first-k strategy are reported.

Table 2. Overall accuracy values (OA) obtained by the tc-system using the different configuration of the first-k strategy

K of firstK	10	15	20	25	30	35	40	45	50
overall acc.	94,56	94,2	93,28	92,72	92,52	91,6	91,2	90,48	90,52
corrects	2366	2357	2334	2320	2315	2292	2282	2264	2265
totals	2502	2502	2502	2502	2502	2502	2502	2502	2502

Performances are strongly affected by the variation of the parameter K; best results were obtained setting the parameter k to 10. Values less than 10 provided arbitrary results. Additional experiments demonstrate that performances are not influenced by varying other parameters in classification task such as SOM output map dimension and number of training epoch.
Setting the best value for K parameter, a detailed performance analysis was conducted creating a global contingency table and then deriving the micro-averaged recall and precision; to provide a single-numbered performance index the f-measure is also computed [7].

Table 3. contingency table and derived evaluation measures obtained by tc-system

	cell-p.	video-c	print.	palmto	mp3	camera	tv lcd	noteb	row
cell-phone	253	0	1	3	5	10	0	1	273
video-c.	10	193	0	0	1	3	0	1	208
printer	2	0	195	0	0	1	2	0	200
palmtop	3	1	1	102	0	1	8	1	117
mp3.r	1	0	2	1	198	2	1	16	221
camera	0	0	0	3	13	558	1	3	578
tv lcd	0	9	0	0	4	4	408	2	427
notebook	0	0	2	7	0	1	8	459	477
Col	270	203	201	116	221	580	428	483	2502
recall	0,937	0,951	0,97	0,879	0,896	0,962	0,953	0,95	
precision	0,927	0,928	0,975	0,872	0,896	0,965	0,956	0,96	
f-measure	0,932	0,939	0,973	0,876	0,896	0,964	0,954	0,96	

overall acc.	mic.av.recall	mic.av.precision
0,9456	0,9473	0,9350

Results in Table 3 demonstrate that the overall classification strategy is able to categorize Web documents.

6 Conclusions

The objective of this work was to design and experimentally evaluate a neural network based strategy for automatic text categorization of commercial web documents. As seen in our experimental context, the allied use of the novel text representation strategy and neural computing in preprocessing and classification stages benefits to web document categorization. The solution proposed can be considered a sound basis for further processing activities in Web search and mining applications.

References

[1] I. Gallo E. Binaghi and M. Pepe. A cognitive pyramid for contextual classification of remote sensing images. *Geoscience and Remote Sensing, IEEE Transactions on*, 41:2906– 2922, 2003.

[2] Timo Honkela, Samuel Kaski, Krista Lagus, and Teuvo Kohonen. Websom - self-organizing maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June*

4-6, pages 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1998.

[3] Peter Jackson and Isabelle Moulinier. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. John Benjamins Publishing Co., Amsterdam, first edition, 2002.

[4] Teuvo Kohonen. *Self-Organizing Maps*. Springer, December 2000.

[5] Gerard Salton and Michael J. Mcgill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

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

[7] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90, 1999.