Feature Selection for Web Page Classification

Daniele Riboni
D.S.I., Universita' degli Studi di Milano, Italy
dr548986@silab.dsi.unimi.it

Abstract

Web page classification is significantly different from traditional text classification because of the presence of some additional information, provided by the HTML structure and by the presence of hyperlinks. In this paper we analyze these peculiarities and try to exploit them for representing web pages in order to improve categorization accuracy. We conduct various experiments on a corpus of 8000 documents belonging to 10 Yahoo! categories, using Kernel Perceptron and Naive Bayes classifiers. Our experiments show the usefulness of dimensionality reduction and of a new, structure-oriented weighting technique. We also introduce a new method for representing linked pages using local information that makes hypertext categorization feasible for real-time applications. Finally, we observe that the combination of the usual representation of web pages using local words with a hypertextual one can improve classification performance.

1. Introduction

An HTML document is much more than a simple text file. It is structured and connected with other HTML documents. While a great effort has been made to exploit hyperlinks for classification, the structured nature of web pages is rarely taken into account. We try to find an efficient method for representing web pages considering both these peculiarities. In Section 3 we analyze some techniques for web page representation. In Section 4 we examine the problem of dimensionality reduction. In Section 5 we suggest a weighting technique for exploiting HTML structure. In Section 6 we point out that most methods of hypertextual categorization are unfeasible for real-time classification and introduce a new representation technique for linked documents that does not require to download them. Finally, in Section 7 we compose a hypertextual representation of web pages with the local one and experimentally evaluate it.

2. Experimental setup

For our experiments we used a corpus of 8000 documents belonging to 10 Yahoo! categories. All the categories considered are subcategories of the category `Science`.
We performed our experiments using two different classifiers to verify the robustness of the techniques aside from the algorithm used. We used a probabilistic classifier called *Naive Bayes* [1] and a perceptron-based classifier using kernel functions called *Kernel Perceptron* [2] (or, simply, *Perceptron*). We used a linear kernel in all the experiments except the one of Section 7.
The results of the experiments are averages of 6 random test/training splits of the dataset. The evaluation is performed in terms of *micro-averaged* F_1 (F_1^μ) [3].

3. Text sources for web page representation

Web pages can be represented in various ways. Maybe the simplest way to represent a web page is to extract the text found within the `BODY` element. This representation does not exploit the peculiarities of web pages, i.e. HTML structure and the hypertextual nature of web pages.

3.1. HTML structure

By exploiting HTML structure [4] for web page representation we can choose how a term is representative of the page considering the HTML element it is present in. For example, we can represent a web page using only the words of the title, that is to say the words extracted from the `TITLE` element.
For obtaining good performance in web page representation exploiting HTML structure is important to know where the more representative words can be found. For example, we can think that a word present in the `TITLE` element is generally more representative of the document's content than a word present in the `BODY` element.
We tested five different text sources for web page representation, namely:

- BODY, the content of the `BODY` tag;
- META, the meta-description of the `META` tag;
- TITLE, the page's title;
- MT, the union of META and TITLE content;

Table 1. Classification performance (F_1^μ) for various representations of web pages

CLASSIFIER	BODY	META	TITLE	MT	BMT
NAIVE BAYES	0.4455	0.5374	0.4015	**0.5587**	0.5086
PERCEPTRON	0.4075	0.4727	0.3707	**0.4996**	0.4691

- BMT, the union of BODY, META and TITLE content.

In this experiment we used only the documents of our dataset that had a representation for all the above-stated text sources. This new dataset was made of only about 2500 web pages, because most of web pages had not a meta description.
Experimental results (see Table 1) show that using the meta description and the title for representing web pages results in the best classification performance with both classifiers, while adding the BODY content decreases classification performance. These results are analogous to Pierre's ones [5] and confirm the intuition that metatags meet the requirements of good text features for automated text classification better than other sources of text. Unfortunately, the majority of web pages has not a meta-description. For example, in the set of documents we used for our experiments only one page out of three has such a description. So, we must represent web pages using the text present in the BODY tag, finding a way to exploit, when present, the meta description. In Section 5.2 we expound a weighting technique that can help achieving this end.

3.2. The hypertextual nature of web pages

Another way for representing a web page is to use, instead of the document's content, the content of the linked web pages. A similar technique was used by Chakrabarti et al. [6] for the classification of patents, where the citations between patents were considered as hyperlinks. They tested a naive way of using hypertextual information considering the words in linked patents as they were local. This approach decreased classification accuracy.
Furnkranz [7] tried to represent pages using in-link instead of out-link information. In his experiments the target page was represented using the anchor words and the words near them on all the pages that pointed to it. This approach increased classification accuracy. However, this technique is difficult to implement because generally it is not possible to find a set of pages that point to the target page.
Moreover, both these techniques require in-link or out-link pages to be downloaded, then they are much more time-consuming than usual representation techniques. In Section 6.2 we expound a representation technique for out-link web pages that does not require linked pages to be downloaded from the web.

3.3. Combining local and hyperlink representation

Joachims et al. [8] tried to combine the usual representation of web pages based on local words with a hypertextual one based on the co-citation matrix (called the *co-link matrix*) of the set of web pages. They used an *SVM (Support Vector Machines)* [9] classifier using one kernel for each representation and combining kernels with the technique called *Composite Kernels*. Their experiments were performed on the *WebKB* dataset (*http://www.cs.cmu.edu/~WebKB/*). Their experimental results showed that combining these different representations of web pages improved classification accuracy compared to using the single representations. In particular, the hypertextual representation they used seemed to perform better than the one using local words. However, it is noteworthy that the hyperlink representation based on the co-link matrix is feasible only with a set of web pages rich in inner links as the dataset they used. In Section 7 we expound a method for combining the local representation of web pages with a hypertextual one that is fast and feasible for every set of web pages without requiring to download the linked pages.

4. Dimensionality reduction

Generally the high dimensionality of the term space can make the classifier run slowly and increase *overfitting*, i.e. the phenomenon by which the classifier tends to perform well on reclassifying the examples of the training set and badly on classifying new examples. In this set of experiments we consider two different approaches to the reduction of the dimensionality of the feature space in the context of web page classification.

4.1. Feature selection techniques

Feature selection techniques [10] aim at decreasing the size of the vocabulary without diminishing classification accuracy. We tested three different feature selection techniques, namely

- *information gain*, an information-theoretic function that tries to keep only the terms distributed more differently in the sets of positive and negative examples of the categories;
- *word frequency*, that consists in removing terms that occur less than n times in the training set;
- *doc frequency*, that consists in removing terms that occur in less than n examples of the training set.

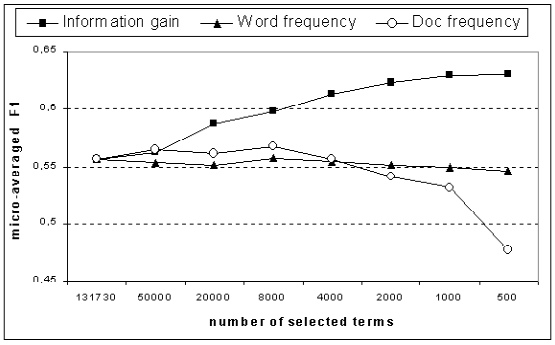

Figure 1. Feature selection: classification performance (F_1^μ) of the *Naive Bayes* classifier

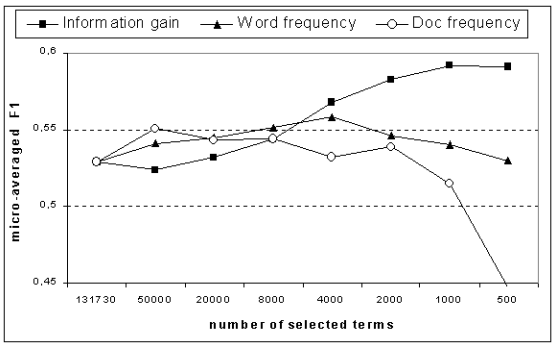

Figure 2. Feature selection: classification performance (F_1^μ) of the *Perceptron* classifier

We perform classification using the same number of terms selected by these three methods.
Experimental results (see Figures 1 and 2) show that *information gain* outperforms the other methods and increases significantly classification accuracy with both classifiers using very few terms (only 1.000 or 500 instead of 131.730).

4.2. *Latent Semantic Indexing (LSI)*

A different approach for the reduction of the dimensionality of the term space is to infer, from the original *term by document* matrix, a new *term by document* matrix in which terms are no more intuitively interpretable but can express the latent semantics of the documents. The technique used is called *Latent Semantic Indexing (LSI)* [11].
We tested this technique with a number of abstract terms k varying from 50 to 300 using only the *Perceptron* classifier because the *Naive Bayes* program we used did not allow using this representation.
Experimental results (see Table 2) show that *LSI* obtains the best performance with 200 terms, and outperforms the feature selection methods tested above.

Table 2. LSI: Classification performance (F_1^μ) of the *Perceptron* classifier

NUMBER OF TERMS (k)	F_1^μ
300	0.5826
250	0.5987
200	**0.6050**
150	0.5966
100	0.5774
50	0.5200

5. Weighting techniques

5.1. *Term frequency (TF)*

The baseline method for computing the weight [12] of a term in a document is to count the number of times the term occurs in the document. This method is usually called *Term Frequency* (TF), and is defined by the function

$$TF(t_i, d_j) = \#(t_i, d_j)$$

where $\#(t_i, d_j)$ denotes the number of times the term t_i occurs in the document d_j.

5.2. *Structure-oriented Weighting Technique (SWT)*

Term Frequency does not exploit the structural information present in HTML document. For exploiting HTML structure we must consider not only the number of occurrences of terms in documents but also the HTML element the terms are present in. The idea is to assign greater importance to terms that belong to the elements that are more suitable for representing web pages (the `META` and `TITLE` elements, see Section 3.1). A similar approach was sometimes used in text categorization for assigning a greater weight to words belonging to the document's title [13] but this weighting technique was never formally defined.
We call this weighting method *Structure-oriented Weighting Technique (SWT)*. It is defined by the function

$$SWT_w(t_i, d_j) = \sum_{e_k} \Big(w(e_k) \cdot TF(t_i, e_k, d_j) \Big)$$

where e_k is an HTML element, $w(e_k)$ denotes the weight we assign to the element e_k and $TF(t_i, e_k, d_j)$ denotes the number of times the term t_i is present in the element e_k of the HTML document d_j.
Term Frequency is a particular case of *SWT* in which the weight 1 is assigned to every element.
In our experiments we defined the w function as:

$$w(e) = \begin{cases} \alpha & \text{if } e = \text{META or } e = \text{TITLE} \\ 1 & \text{elsewhere} \end{cases}$$

Table 3. Classification performance (F_1^μ) of the *Perceptron* classifier with $SWT_w(\alpha)$

WEIGHTING TECHNIQUE	NUMBER OF SELECTED TERMS				
	500	2.000	8.000	50.000	131.730
TF	0.5912	0.5826	0.5441	0.5238	0.529
$SWT(\alpha = 2)$	0.588	0.5812	0.5556	0.5364	0.5285
$SWT(\alpha = 3)$	**0.5993**	0.5902	0.5607	0.5408	**0.5394**
$SWT(\alpha = 4)$	0.5984	**0.5923**	0.5632	0.5414	0.5366
$SWT(\alpha = 6)$	0.5946	0.5895	**0.5748**	**0.5444**	0.531

Table 4. Classification performance (F_1^μ) of the *Naive Bayes* classifier with $SWT_w(\alpha)$

WEIGHTING TECHNIQUE	NUMBER OF SELECTED TERMS				
	500	2.000	8.000	50.000	131.730
TF	0.6298	0.6232	0.5979	0.5623	0.5569
$SWT(\alpha = 2)$	0.6327	0.6318	0.6111	0.5739	0.5591
$SWT(\alpha = 3)$	**0.6465**	0.6397	0.6202	0.5825	0.5653
$SWT(\alpha = 4)$	0.6407	0.6458	0.6217	0.5790	**0.5733**
$SWT(\alpha = 6)$	0.6458	**0.6506**	**0.6359**	**0.5879**	0.5713

We tested *SWT* with $\alpha = 2$, $\alpha = 3$, $\alpha = 4$ and $\alpha = 6$ and compared it to the standard *TF*. Each of the described techniques was evaluated in combination with a feature selection based on the terms' *information gain*.
Experimental results (see Table 3 and Table 4) show that *Structure-oriented Weighting Technique (SWT)* can improve classification accuracy assigning to `META` and `TITLE` elements a greater weight than to the other elements. In particular, *SWT* obtains its best results with $\alpha = 3$ with the *Perceptron* classifier and with $\alpha = 6$ with *Naive Bayes*.

6. Linked pages representation

6.1. Hypertext categorization

In the last few years various techniques have been developed to exploit the hypertextual nature of web pages. These techniques use the hypertextual information in various ways, but they all need to download the linked pages from the web. This operation inevitably slows down the classification, making these techniques unfeasible for those applications in which the classification result must be immediatly available.
We tried to develop a new kind of representation for linked pages that makes use of merely local information, so that the hypertextual nature of web pages can be exploited even in real-time classification.

6.2. Linked pages representation using local information

The idea is to exploit HTML structure [4] for representing linked pages without having to download them. More specifically, we are interested in the content of the `A` element. Here is an example of its use:

```
<A href="./php.html">PHP tutorial</A>
```

In this example, the `A` element is used to link the current page to another page. Users can understand what the linked page talks about by means of the content of the `A` element, that is "`PHP tutorial`".
When a developer links a page to his pages, he tries to explain with few words the content of the linked page using the `A` tag. We can assume that the words used in this description are close to the subject of the linked page, and then we can use this description for representing the linked page without having to download it.

7. Combining local and hyperlink representation

In Section 3.3 we briefly reported how Joachims et al. [8] combined the local representation of web pages with a hypertextual one based on the *co-link matrix*, improving classification accuracy. Their hypertextual representation of web pages is very powerful but unfeasible for sets of pages lacking in inner links. We introduce a new representation of web pages exploiting hyperlinks that does not require the linked pages to be downloaded and that is usable with every set of pages.

7.1. Simple hypertextual representation

A simple way for representing web pages exploiting hyperlinks is to represent documents using the content of the linked pages. The rationale is to represent pages by means of the "type of page" they are linked to. Furthermore, linked pages can be represented using local words by means of the technique expounded in Section 6.2. This representation method is surely less powerful than other hypertextual ones [6, 7, 8, 14], but can be used for real-time categorization and is feasible for every set of web pages.

7.2. Combination of local and hypertextual representations using *Composite Kernels*

We used *Composite Kernels* (see Section 3.3) for combining the usual representation of web pages based on local words with the simple hypertextual one of Section 7.1. The

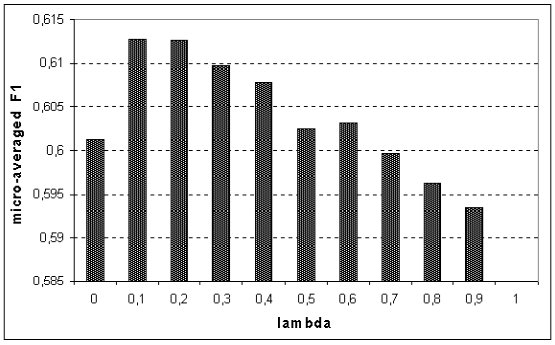

Figure 3. Classification performance (F_1^μ) of the *Perceptron* classifier with *Composite Kernels*

kernel we used was

$$K(x,y) = \lambda K_1(x,y) + (1-\lambda)K_2(x,y)\,,\ \lambda \in [0,1]$$

We used the kernel K_1 for the hypertextual representation and K_2 for the local one.
Experimental results (see Figure 3) show that, in spite of the mediocre performance of the simple hypertextual representation, assigning a value near to 0.2 to the weight λ improves classification accuracy, confirming the usefulness of the combination of standard and hypertextual representations.

8. Conclusions and future work

We observed that the combination of hypertextual and local representations of web pages can improve classification accuracy. The representation of linked pages we introduced can be used for the implementation of much more powerful hypertextual techniques than the one we tested. Furthermore, *Structure-oriented Weighting Technique* can be refined for obtaining better representations.

Acknowledgements

I would like to thank Marko Grobelnik for providing classified web pages, and Alex Conconi for the implementation of *Composite Kernels* for the *Perceptron* classifier.

References

[1] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval", *Proceeding of ECML-98, 10th European Conference on Machine Learning*, Springer Verlag, Heidelberg, DE, 1998, pp. 4-15.

[2] Y. Freund, R. E. Schapire, "Large Margin Classification Using the Perceptron Algorithm", *Computational Learning Theory*, 1998, pp. 277-296.

[3] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization", *Information Retrieval*, Kluwer Academic Publishers, 1999, pp. 69-90.

[4] The World Wide Web Consortium (W3C), "HTML 4.01 Specification", *http://www.w3.org/TR/html4/*

[5] J. M. Pierre, "Practical Issues for Automated Categorization of Web Sites", 2000

[6] S. Chakrabarti, B. Dom, P. Indyk, "Enhanced hypertext categorization using hyperlinks", *Proceedings of SIGMOD-98, ACM International Conference on Management of Data*, ACM Press, New York, US, 1998, pp. 307-318.

[7] J. Furnkranz, "Exploiting structural information for text classification on the WWW", *Proceedings of the 3rd Symposium on Intelligent Data Analysis (IDA-99)*, Springer-Verlag, Amsterdam, Netherlands, 1999.

[8] T. Joachims, N. Cristianini, J. Shawe-Taylor, "Composite kernels for hypertext categorisation", *Proceedings of ICML-01, 18th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, US, 2001, pp. 250–257.

[9] N. Cristianini, J. Shawe-Taylor, *An introduction to Support Vector Machines (and other kernel-based learning methods)*, Cambridge University Press, 2000

[10] Y. Yang, J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization", *Proceedings of ICML-97, 14th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, US, 1997, pp. 412-420.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, "Indexing by Latent Semantic Analisys", *Journal of the American Society of Information Science*, 1990, pp. 391-407.

[12] G. Salton, C. Buckley, "Term-weighting approaches in automatic text retrieval", *Information Processing and Management*, 1988, pp. 513-523.

[13] W. W. Cohen, Y. Singer, "Context-sensitive learning methods for text categorization", *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, ACM Press, New York, US, 1996, pp. 307-315.

[14] H. Oh, S. H. Myaeng, M. Lee, "A practical hypertext categorization method using links and incrementally available class information", *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, ACM Press, New York, US, 2000, pp. 264-271.