

Naive Bayes Web Page Classification with HTML Mark-Up Enrichment

Víctor Fresno

Dpto. de Lenguajes y Sistemas Informáticos
Universidad Rey Juan Carlos (URJC)
ESCET
victor.fresno@urjc.es

Soto Montalvo

Dpto. de Lenguajes y Sistemas Informáticos
Universidad Rey Juan Carlos (URJC)
ESCET
soto.montalvo@urjc.es

Raquel Martínez

Dpto. de Lenguajes y Sistemas Informáticos
Facultad de Informática
U. Nacional de Educación a Distancia (UNED)
raquel@lsi.uned.es

Arantza Casillas

Dpto. de Electricidad y Electrónica
Facultad de Ciencias
Universidad del País Vasco (UPV-EHU)
arantza.casillas@ehu.es

Abstract

In text and web page classification, Bayesian prior probabilities are usually based on term frequencies, term counts within a page and among all the pages. However, new approaches in web page representation use HTML mark-up information to find the term relevance in a web page. This paper presents a Naive Bayes web page classification system for these approaches. In the representation phase, the feature relevance is obtained from a combination of criteria extracted from the HTML tags. In the supervised learning phase, central limit theorem is assumed to find the relevance of a feature given a class. Finally, a Gaussian density function gets the probability that a word appears in a class, whose parameters are obtained by a maximum likelihood estimator method. We also address a comparison between two different models for the prior probabilities, the event and the gaussian model, applied to representations based on frequencies and relevance.

1. Introduction

The fast expansion of the Web has turned Internet into a huge source of information; therefore, achieving a better organization of its contents will make easier and more efficient the access to such information. In this context, Web Mining finds a fertile field.

Web mining consists of Web usage mining, Web structure mining, and Web content mining. Web usage mining refers to the discovery of user access patterns from Web usage logs, Web structure mining tries to discover useful

knowledge from the structure of hyperlinks, and Web content mining aims to extract useful information or knowledge from web page contents. This paper focuses on Web Content Mining.

When we want to find information in the Web, we usually access to it by search engines, which return a ranked list of web pages in response to our request. This way to access information works well when we want to retrieve homepages, websites related to corporations, institutions or specific events, and finding quality portals [4]. However, when we want to relate information from several sources, this way has some lacks: the ranked lists are not conceptually ordered and information in different sources is not related. Other way to find information is using web directories organized by categories, such as Open Directory Project (<http://www.dmoz.org>). In this case, the maintenance of these directories can be too arduous if it is not assisted by a machine process. Automatic web page classification can be very useful for tasks such as relating pages retrieved by search engines, or creation and maintenance of web directories.

Most of the applied web page classification techniques are inherited from automatic text classification: a supervised learning task, defined as assigning pre-defined category labels to new documents, based on the likelihood suggested by a training set of labeled documents. Therefore, an increasing number of learning approaches have been applied to classify web pages [16]. In this paper, we focus on Naive Bayes approach, which is one of the most effective approaches for text document classification. In addition, it is a straightforward method that has exhibited good results in previous studies [[15],[1]].

In this paper, we focus on web page text content, so hy-

perlinks and multimedia data are not considered. In this way, the representation and later classification of web pages is similar to the representation and classification of any text, except for the mark-up information. Instead of using the plain unstructured text, we process semi-structured data to help our representation task. Several researches have applied this mark-up information in different ways. In [16] a study of hypertext classification methods is carried out, where five hypertext regularities are considered to explore hypothesis about the structure of the hypertext. In [9] a hierarchy of tag classes are presented, and normalized frequencies in each one of them are compounded in an analytical combination.

This paper presents a Naive Bayes classification that consider HTML mark-up information to enrich the term relevance in the representation phase. For that reason, the well-known Luhn's curve, according to which the terms with low-to-medium frequency are the most informative ones, could be reconsidered before its application in bayesian prior probabilities for web page classification. In these kind of representations, the most informative terms have the biggest term relevance. In the supervised learning, we assume that Internet contains a "large number" of pages relating to any theme and so, central limit theorem can be assumed to find the relevance of a term given a class. Therefore, in the classification phase, a gaussian function gets the prior probability that a term belongs to a class, whose parameters are obtained by a maximum likelihood estimator method. In this work, we also address a comparison between two different models for the prior probabilities, the *event* and the *gaussian model*, applied to representations based on term frequencies and relevances.

The remainder of the paper is organized as follows: Section 2 summarizes the main approaches in web document representation and shows the different weight functions we study. Section 3 describes the Naive Bayes classification algorithm and the prior probabilities that we evaluate in each model. Section 4 presents the benchmark we use for evaluation. Section 5 describes the experiments, the two types of feature reduction considered, the evaluation method, and the results. Finally, Section 6 includes the main conclusions.

2. Web Document Representation

Some of the elements which can be distinguished in a web document are: plain text, text enriched with HTML tags, metacontent, hyperlink structure, text associated with hyperlinks or a set of statistical parameters (media types, size, number of images or links, etc.)[16]. Thus, the representation of a web page can be defined according to these (and other) elements and can be taken separately or combined. The studies in web page representation has focused

mainly on two approaches: context and content.

In representation by context, the underlying information of the links is explored, and multimedia components are treated [5], [6], [14]. In representation by content, information is extracted from text and neither document structure nor topology is studied. In this group we can find representations based on concept identification, exploring techniques in Neural Networks or semantic and linguistic analysis [16].

This paper is focused on web page representation by text content. The terms or features which represent web pages will be extracted from the text of them and some information from HTML tags will be taken into account. We evaluate four term weighting functions of the features based solely on pure text on the web page, and two more which, in addition to pure text, combine HTML tags for emphasis and the "title" tag with term position.

2.1. Evaluated Representations

We represent web documents using the vector space model. In this model, each document is represented by means of a vector, where each component is the weight of a feature in the document, which is tokenized using simple rules, such as whitespace delimiters in English, and tokens stemmed to canonical form (eg. 'reading' to 'read'). Each canonical token represents an axis in the Euclidean space. This representation ignores the sequence in which words occur and is based on the statistical about single independent words. This Independence Principle between the words that co-appear in a text, or appear as multiword terms, is a certain error but reduce the complexity of our problem without loss of efficiency. In this way, the different representations are obtained using different functions to assign the value of each component in the vector representation.

2.2. Representations based only on Term Counts

First, we create four representations of web documents using well known functions. All four representations use only the plain text of the HTML documents. These functions are:

- **Term Frequency (TF)**. Each term or feature is assumed to have importance proportional to the number of times it occurs in the document. The weight of a feature t in a document d is given by: $W(d, t) = TF(d, t)$, where $TF(d, t)$ is the term frequency of the feature t in the document d .
- **Binary Inverse Document Frequency (BinIDF)**. This looks at feature occurrence across a collection

of documents. The importance of each feature is assumed to be inversely proportional to the number of documents that contain the feature. The IDF factor of a feature t is given by: $IDF(t) = \log \frac{N}{df(t)}$, where N is the number of documents in the collection and $df(t)$ is the number of documents that contain the feature t . Then, the weight of a feature in BinIDF representation is: $W(d, t) = B(d, t) \times IDF(t)$, where $B(d, t) = \{0, 1\}$ is a binary function that represents if d contains t .

- **TF-IDF**: It is the combination of TF and IDF to weight terms. The combination weight of a feature t in a document d is given by: $W(d, t) = TF(d, t) \times IDF(t)$.
- **WIDF**. This is an extension of IDF to incorporate the feature frequency over the collection of documents. The WIDF weight is given by: $WIDF(d, t) = TF(d, t) \sum_{i \in D} TF(i, t)$.

The expressions of all these function can be found in [11].

2.3. Representations based on a Criteria Combination

In addition to those four representations we use two which combine several criteria. Both consider as main features of a web document a selection of the set of words it contains. The combined criteria are: text content with the word frequency in the text, the words appearance in the title of the web page, positions throughout the text, and whether or not the word appears in emphasized tags. For the last criterion a set of HTML elements are selected because they capture the author's intention and highlight some parts from others. These two representation are the Analytic Combination of Criteria (ACC) [3] and the Fuzzy Combination of Criteria (FCC) [10]. The difference between them lies in how they evaluate and combine the criteria. The first uses a linear combination, whereas the second combines them by using a fuzzy system.

We start with a brief description of ACC. Once the criteria are defined, the functions to catch them and their combination have to be described. The final relevance assignation function is constructed as a linear combination of the criteria where we use the following functions:

- **The frequency function of a word on a web page**: $f_f(i) = \frac{n_f}{N_{tot}}$ Where $n_f(i)$ is the number of occurrences of a word i on the page and N_{tot} is the total number of words on the web page. This definition allows the function to be normalized using $\sum_1^k f_f(i) = 1$, where k is the number of different words in the document.

- **The frequency function of a word in the title**: $f_t(i) = \frac{n_t}{N_{tit}}$ Here $n_t(i)$ is the number of occurrences of a word i in the title, and N_{tit} is the total number of words in the title. As previously, $\sum_1^k f_t(i) = 1$, where k is the number of different words in the title.
- **The emphasized function of a word in the title**: $f_e(i) = \frac{n_e}{N_{emph}}$ Here $n_e(i)$ is the number of times that a word is emphasized and N_{emph} the total number of words that are emphasized in the whole document. As in former cases, $\sum_1^k f_e(i) = 1$.
- **The position function**. To compute the position criteria the web page is split into four equal parts. Considering $n_{tot}(i) = n_{1,4}(i) + n_{2,3}(i)$, and if $n_{1,4}(i)$ and $n_{2,3}(i)$ are the number of times that the term i appears in the introduction or in the conclusion - first and fourth parts -, and in the document body - second and third parts - respectively, the function is: $f_p(i) = \frac{2n_{1,4}(i) + n_t(i)}{2n_{tot}(i) + N_{tot}(i)}$ Where $n_{1,4}(i)$ and $n_{2,3}(i)$ are the number of times that the term i appears in the introduction or in the conclusion -first and fourth parts- and in the document body -second and third parts- respectively. This expression is a particularization of a more general expression [3], where we impose a weight ratio of 3/4 and 1/4 for preferential and standard quarters respectively.

The combination of these criteria for each feature i in ACC representation is given by the following relevance function: $r_i = C_1 f_f(i) + C_2 f_t(i) + C_3 f_e(i) + C_4 f_p(i)$. The coefficients were estimated as a result of previous research [2] on the influence of each criterion in the representation. As result, the best coefficients of the linear combination were estimated as: C_1 (Frequency) 0.30, C_2 (Title) 0.15, C_3 (Emphasis) 0.25, C_4 , (Position) 0.30. These are the coefficients we use this paper.

The FCC representation uses the same criteria to evaluate feature relevance on a web page. It is a fuzzy system for the assignation of weight functions and their combination. Thus, the linguistic variables of the fuzzy system are:

- **text-frequency** with fuzzy sets “Low”, “Medium and High”;
- **title-frequency** with fuzzy sets “Low” and “High”;
- **emphasis** with fuzzy sets “Low”, “Medium” and “High”; and
- **global-position**, with fuzzy sets “Preferential”, and “Standard”

To make the fuzzy rules independent of document size, the inputs of *text-frequency*, *title-frequency* and *emphasis*

are normalized to the greatest of the frequency in each criterion. So, in the $f_f(i)$, $f_t(i)$ and $f_e(i)$ functions defined in ACC, the denominator is the maximum number instead of the total number.

In the position criterion, the variable *global-position* is calculated from an auxiliary fuzzy system because a term can be different appearances. This auxiliary system has as input the **line-position** variable, with fuzzy sets “Introduction”, “Body” and “Conclusion”, and as output the *global-position*, with fuzzy sets “Standard” and “Preferential”. The output value is calculated from the different positions where a feature appears. As output of the fuzzy module we have only one variable for each feature on a web page: *relevance*, with fuzzy sets “NonRelevant”, “LowRelevant”, “Medium-Relevant”, “HighRelevant”, and “VeryRelevant”. The inference engine is based on a Center of Mass Algorithm (COM) that weights the output of each rule against the truth degree of each antecedent.

The foundation of the rules is based on the following points: (1) A web page may have no emphasized words; (2) A word appearing in the title may not always be relevant (for instance, the title can be generated automatically by a HTML editor); (3) In general, position is a criterion that gives more weight on longer pages than on shorter ones; (4) A word with a high frequency on a page could mean that the word is an all-purpose word and so not highly discriminate. The set of rules and a detailed explanation of this representation can be found in [10].

It is important to emphasize that TF, ACC and FCC representations are independent of the collection information; in other words, they only need information from the web page itself to represent it. However, the Binary-IDF, TF-IDF and WIDF representations need the whole collection information to represent each web page.

3 Naive Bayes Classification

Bayes theorem provide a method to calculate the probability of a hypothesis based on its prior probability, the probability of observing various data given the hypothesis, and the observed data itself [8]. Then, considering a set of web documents D belonging a set of known classes C , the most probable classification of a new web page instance is obtained combining the predictions of all hypothesis (the prior probabilities of each one of them) weighted by their posterior probabilities. The naive part in this approach is the assumption of word independence: the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category. This assumption makes the computation more efficient than considering word combinations as predictors [15] and so, $P(c_j|d) = \frac{P(c_j) \times P(d|c_j)}{P(d)} = \frac{P(c_j) \times \prod_i P(t_i|c_j)}{P(d)}$

where $P(c_j|d)$ is the probability of a web page $d \in D$ belongs to $c_j \in C$ and $P(t_i|c_j)$ is the probability of a term $t_i \in V$ belongs a class c_j . The vocabulary V is created from the union of all terms that appear in D . Thus, if does not exit a prior knowledge about the prior probabilities of all the classes and pages, they can be assigned as equals and then the most probable class, given a new web page, will be $c_j = \operatorname{argmax}_{c_j} \prod_i P(t_i|c_j)$. Rather than maximizing this expression, its logarithmic transformation is chosen, due to the logarithm of f is a non-decreasing function of f . Therefore, maximizing the logarithm of f also implies the maximization the function f . Finally, $c_j = \operatorname{argmax}_{c_j} \sum_i \ln P(t_i|c_j)$ In this point, we must search expressions to estimate the prior probabilities $P(t_i|c_j)$ that optimize the classification task.

3.1 Classification Algorithms

We evaluated different probabilities functions that were grouped in two models, *event* and *gaussian*, due to they have the same parameters inside the group and different outside the model. In both cases, their optimal values should be learned in two different way.

3.2. Event Models

Into this group, we consider two well known functions in automatic text classification, *m-estimate* and *multinomial*, whose estimations are only based on the values of term counts within a page and among the collection.

- **m-estimate.** In this case, the probability is estimated calculating the frequency of each term t_i over all the documents in a class c_j , supplemented with Laplace smoothing to avoid zero probabilities, $P(t_i|c_j) = \frac{1 + \sum_{d \in C_j} N(t,d)}{|V| + \sum_{t \in V} \sum_{d \in C_j} N(t,d)}$ being $N(t,d)$ the term frequency of t in d .

- **Multinomial NB model.** In this case, the model capture the term frequency information in documents. We restrict our analysis to the most popular multinomial NB classifier [7]. It estimates the posterior probability of test document d as a member of category c_j using the following formula: $P(d|c_j) = \prod_{i \in d} \left(\frac{1 + \sum_{d \in C_j} N(t,d)}{|V| + \sum_{t \in V} \sum_{d \in C_j} N(t,d)} \right)^{N(t,d)}$

In event models, learning is restricted to count the term frequency values within a page and among all the collection. This counts are stored in inverted files. However, these functions do not capture all the information calculated from the ACC and FCC representations about term relevance in

a document. Because the event models only consider frequencies, we lose qualitative information about if a feature has appeared not much times but in the title, or if some times was emphasized and in a preferential position. With event models this qualitative information captured from ACC and FCC is lost in the learning process. This is one reason to search continuous variable functions, as gaussians, for the learning and classification tasks.

3.3. Gaussian Models

We present three different probability functions that not consider term frequencies but term relevance. These relevance can be obtained from the combinations of criteria, such as described in ACC and FCC representations. This consideration is important because allow us to make an assumption based on *central limit theorem*¹. The relevance of a given term t_i in web pages that belong a given class c_j will follow a Normal distribution. So, the class descriptor will be a matrix $3 \times N$ that represents the class, where the first row are the words t_i that belong to that class c_j , and the second and third ones hold the μ_i and σ_i parameters that define the Normal distribution for each t_i . In this way, the gaussian functions we try are:

- **Gaussian model.** The probability function of a term t_i given a class c_j is estimated as a Normal function:

$$P(t_i|c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(r_i - \mu_{ij})^2}{\sigma_{ij}^2}}$$

- **Weighted Gaussian model.** In this case, the probability is weighted with the relevance of t_i , $P(t_i|c_j) =$

$$\frac{r_i}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(r_i - \mu_{ij})^2}{\sigma_{ij}^2}}$$

- **LogNormal model.** We consider a LogNormal function to estimate the prior probability. $P(t_i|c_j) =$

$$\frac{1}{r_i \sigma_{ij} \sqrt{2\pi}} e^{-\frac{(\ln r_i - \mu_{ij})^2}{\sigma_{ij}^2}}$$

In the learning phase, we assume that classes are independent of each other, as well as words in a same class. Then, for each word in each page of the sample set, μ and σ are obtained by a maximum likelihood estimator method. Then, $\mu_{ij} = \frac{1}{N_{ij}} \sum_k r_{ik}$, $\sum_{ij}^2 = \frac{1}{N_{ij}} \sigma_k (r_{ik} - \mu_{ij})^2$ where N_{ij} is the number of web documents belonging c_j and k is an index that goes round those documents, being r_{ik} the relevance of t_i in the web page k . Once the parameters are estimated, the probability functions can be apply.

¹The mean of a random sample from any distribution with finite variance σ^2 and mean μ is approximately distributed as a normal random variable with mean μ and variance $\frac{\sigma^2}{n}$

4 Web Page Collection

We use a subset of the Benchmark Dataset as web page collection to evaluate the performance of the classification system. The Benchmark [13] is a dataset of 11,000 web documents pre-classified into 11 equally-sized categories, each containing 1,000 web documents. It was generated by Mark Sinka and David Corne, from Reading University in the U.K., with the main aim of proposing a general dataset for web document clustering and similar experiments. The selected collection consists of two separate categories: Astronomy (G) and Motor Sport (J), where the 70% of the web pages were used for training, and the 30% for testing.

We use the vector space model, so that each web document is represented by using a vector, where each component is the weight of a feature. In this work, a feature is a character stream between two space characters. We fix the maximum length of the feature to be 30 characters. In order to calculate the values of the vector components for each document we follow these steps:

1. We eliminate all the punctuation marks except some special marks that are used in URLs, e-mail addresses, and multiword terms.
2. The words of a stoplist used in Information Retrieval are eliminated.
3. We considered only the features that fulfilled two conditions: (1) occur more than one time in the document and (2) appears in more than one document.
4. We obtain the stem of each feature by using Porter's stemming algorithm.
5. We count the number of times each feature appears on each web page, and the number of web pages where each feature appears.
6. To calculate the ACC and FCC representations, we memorize the position of each feature throughout the web page, and whether or not the feature appears in emphasized tags.

In the collection, the 70% of the web pages were selected to extract the class descriptors in the learning phase, and the 30% were used to evaluate the system performance. The page selection to learn had 1,231 pages and 909,323 words, that were reduced to 45,128 features after those steps. The average number of document features was 282, the maximum frequency of one word was 924, and the minimum was 2.

5 Experiments

We tried the six representations in the collection by means of a Naive Bayes classification process.

5.1 Features Reduction

One of the main problems in representation and later classification is the high number of features that have to be taken into account when documents are dealt with. In these experiments we tried two types of feature reduction:

1. Only the features that appear more than $FFmin$ times in more than $DFmin$ web pages, and less than $FFmax$ times in less than $DFmax$ web pages are selected. This is a variant of [12]. This type of reduction is applied to the six weight functions. We called this type of reduction “Mm”.
2. The proper weight functions are used as reduction method. So, the n most relevant features (those of higher function values) on each web page are selected. In this case, this type of reduction has been able to be applied to every weight function. We have tried seven values for N , from 1 to 70. We called this type of reduction “ $f(n)$ ”, where f represents the weight function.

We tried several values for the different variables ($FFmin$, $DFmin$, $FFmax$, $DFmax$, N) in order to obtain different magnitude reductions. We fix the maximum number of features to be one magnitude order less than the initial number of features.

5.2 Evaluation Measures

We test the performance of the classification algorithms with the different representations and reductions. We carry out an external evaluation to determine the quality of the classification results by means of the accuracy, estimated in terms of the *contingence table* as $A = \frac{TP+TF}{TP+TN+FP+FN}$. Here, FP are false positives, the number of test pages incorrectly classified; TN are true negatives; and FP and FN are false positives and false negatives counts respectively.

5.3 Results

The results of the experiments are summarized in Figure 1 which shows the accuracy values after Naive Bayes classification with bag of features of different size. The proposed *event* (m-estimate and multinomial) and *gaussian* models (Gaussian, WeightedGauss and LogNormal) are applied with different weight functions (BinIDF, TF, TF-IDF, WIDF, ACC and FCC) and reductions (MinMax and proper reduction) to obtain optimal performances in Naive Bayes classifications. Only the best performances in each case (combinations of weight function - reduction - classification function) are shown in Figure 1.

The best performances in our collection are reached by FCC in conjunction with Gaussian and Weighted Gaussian

probabilities, and proper function reduction. Only in these cases the accuracy rate overcomes the 0.9. In addition, in these conditions, TF-IDF performance is better than using any representation in a event model in most features dimensions. As it was to be expected, gaussian models performs better than event ones when mark-up information is considered and proper reductions are applied. However, the LogNormal function do not obtain so good results.

In general, event models obtain worse performances than the gaussians, because the accuracy absolute values are lower. Thus, proper reduction is more efficient than “Mm” one in most of the cases. The TF representation in conjunction with m-estimate and proper reduction obtain the best results in event models. In general, the worst accuracy rates are obtained by event models and “Mm” reductions.

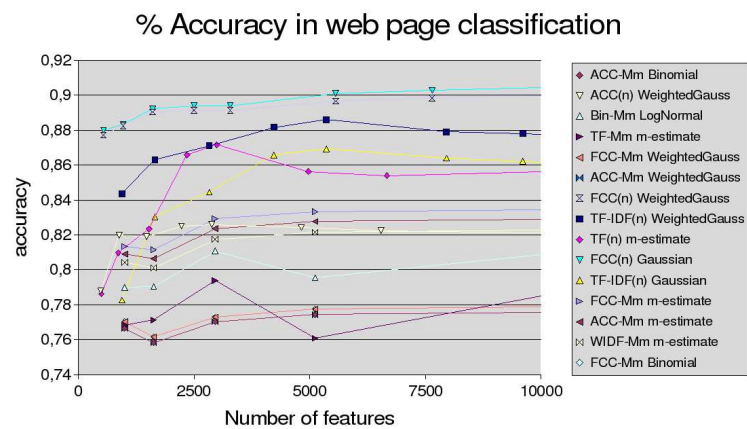


Figure 1. Summary of the best performances in each classification system (weight function - reduction-class-function).

6 Conclusions

Nowadays, several approaches have been introduced to represent and classify web pages. Some of them use HTML mark-up information to enrich the term relevance in a document. However, the main approaches to web page classification have been inherited from text classification and so, only consider feature counts to carry out this task. In this paper, a Naive Bayes classification system is presented for the new enriched web page representations.

Six weight functions have been tried: five well known functions that use only the plain text of the HTML documents and are based on frequencies, and two weight functions which combine several criteria including information from HTML mark-up. In the later kind of representation, the most informative terms have the biggest term relevance and so, the “Luhn’s law” for term frequencies could not

be followed in the classification phase design. Then, we also address a comparison between two different models for bayesian prior probabilities, the *event* and the *gaussian model*, applied to representations based on frequencies and relevance, respectively.

The experiments were carried out with a benchmark of web documents created for general use in web document research. We selected one subset of that collection, consisting of two separate categories. The experiments show that, in general, gaussian models obtain better accuracy rates than event models when enriched representation are considered. On the other hand, gaussian models works well in conjunction with representation based on term counts and in some cases, these performances are better than the events ones. In future works the experiments will be extended to closer classes and to more than two categories.

References

- [1] S. Chakrabarti, S. Roy, and M. Soundalgekar. Fast and Accurate Text Classification via Multiple Linear Discriminant Projections. *The VLDB Journal*, 12(2):170–185, 2003.
- [2] V. Fresno and A. Ribeiro. Features Selection and Dimensionality Reduction in Web Page Representation. In *Proceedings of the International ICSC Congress on Computational Intelligence: Methods and Applications*, Bangor, Wales, 2001.
- [3] V. Fresno and A. Ribeiro. An analytical approach to concept extraction in html environments. *Journal of Intelligent Information Systems - JIIS, Kluwer Academic Publishers*, pages 215–235, 2004.
- [4] J. Gonzalo. Hay vida después de google? Technical report, In the Software and Computing System seminars. Escuela Superior de Ciencias Experimentales y Tecnología. Universidad Rey Juan Carlos. (<http://sensei.lsi.uned.es/julio/>), 2004.
- [5] R. N. Guglielmo, E. J. Natural language retrieval of images based on descriptive captions. *ACM Transactions on Information Systems*, 14(39):237–267, 1996.
- [6] S. M. D. M. D. Harmadas, V. Images retrieval by hypertext links. In *Proceeding of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 269–303, 1997.
- [7] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, 1998.
- [8] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [9] A. Molinari, G. Pasi, and R. A. M. Pereira. An indexing model of html documents. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 834–840. ACM Press, 2003.
- [10] A. Ribeiro, V. Fresno, M. García-Alegre, and D. Guinea. A fuzzy system for the web page representation. In *Intelligent Exploration of the Web. Springer-Verlag Group*, pages 19–38, 2002.
- [11] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1988.
- [12] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
- [13] C. D. W. Sinka, M. P. A large benchmark dataset for web document clustering. *Soft Computing Systems: Design, Management and Applications, Frontiers in Artificial Intelligence and Applications*, 87:881–890, 2002.
- [14] R. K. Srihari. Automatic indexing and content-based retrieval of captioned images. *Computer*, 28(9):49–56, 1995.
- [15] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.
- [16] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *J. Intell. Inf. Syst.*, 18(2-3):219–241, 2002.