LEARNING TEXT FROM THE WEB: AN APPLICATION OF CLASSIFICATION METHODS

Advanced Data Analysis Project[1]

Stella M. Salvatierra
Department of Statistics
Carnegie Mellon University

Advisory Committee
Stephen Fienberg [2]
Brian Junker[2]
Tom Mitchell [3]

ABSTRACT

The WWW contains a huge amount of information but understands little. Many WWW searching mechanisms produce poor results. One approach that appears to have much more efficient text retrieval is the system RAINBOW, created by the CMU CS Text-Learning group. RAINBOW's approach for statistical text classification is based on the naive Bayes classifier and its results are impressive. This report demonstrates that the "standard" statistical tools of discriminant analysis lead to results that are even better than those obtained from the naive Bayes approach and also emphasizes that adding "non informative" words increases the noise of the model producing poor results.

KEY WORDS: Discriminant analysis, Logistic regression, Bayes classifier, Machine learning, WWW searches.

1 Introduction

The World Wide Web contains millions of pages and it has more information that we can handle. Given a query, we would like to find the relevant documents. Usually searches based on "key words" produce a huge number of documents most of them being irrelevant for the particular query. This feature motivated a research project at the Department of Computer Science of CMU. The team is leading by Professor Tom Mitchell and the long term goal of the project is to automatically create and maintain a computer-understandable knowledge base whose content mirrors that of the World Wide Web.

An approach might be (see Craven et al, 1998) to develop a trainable system that can be taught to extract various type of information by automatically browsing the web. A first attempt to achieve this goal is a system with the following two inputs:

[1]This research was supported by National Science Foundation Research Grant No. REC-9720374
[2]Department of Statistics, Carnegie Mellon University
[3]Department of Computer Science, Carnegie Mellon University

- A specification of the classes and relations of interest (called an "ontology"). The classes are given by: "student", "faculty", "research project", etc and the relations are, for example, "student of", "advisor of", etc.
- Training samples that describe instances of the classes and the relations.

Given such an ontology and a set of training samples, the system attempts to learn a general procedure for extracting new instances of these classes and relations from the web.
This is an information retrieval problem using training examples. To understand the problem better, let us look at Figure 1, which shows a typical web page.

Tom Mitchell's Home Page

Welcome! Hope you find something of interest in here.

Professor of Computer Science and Robotics,
Director, Center for Automated Learning and Discovery
Carnegie Mellon University

☎ 412-268-2611, ✉ *Tom.Mitchell@cmu.edu*

Research: Machine Learning, Computer Science

How can we make computers improve automatically from experience? This question drives my research. It includes

- How can software agents learn to help their users?
  - See our text learning and software agents research
- How can robots learn by experimenting in their environment?
  - See our learning robots lab
- How can computers learn from (i.e., data mine) large historical databases?
  - One example is described in "Using the Future to Sort Out the Present: Rankprop and Multitask Learning for Medical Risk Analysis," (postscript) R. Caruana, S. Baluja, and T. Mitchell, *Neural Information Processing 7*, December 1995.

Textbook: Machine Learning

- *Machine Learning,* Tom Mitchell, McGraw Hill, available March, 1997.

Courses

- Artificial Intelligence, 15-780 and 16-731 , Spring 1998.

Figure 1: Example of a typical home page

Any web page may contain text, graphics, sounds and also links to other web pages. From looking at the page showed above we can conclude, for example, that it corresponds to a Faculty member of the Computer Science Department of Carnegie Mellon University. This particular feature is represented by the sentence: "Professor of Computer Science and Robotics". That is, the words come in a sequence that contain a lot of relevant information. Also, the words that appear in the links have information that might be useful for some queries. But we can understand the page because it was written by humans for other humans. The WWW does not understand anything by itself. Observe that the data are highly noisy and the main difficulty is that the information that we need to satisfy the query is not uniquely determined. Statistically speaking, the covariates or "discriminators" are not uniquely defined. For example, what are the relevant covariates we would need in order to identify a list of all the research projects of CMU?
There is an extra problem and it is due to the fact that the pages were written using the HTML code. A small piece of the HTML code corresponding to the page showed above is the following.

```
<html>
<head>
<title>Tom Mitchell's Home Page</title>
</head>

<body bgcolor="#eff7ff" alink="#5e5a80">
<h1>Tom Mitchell's Home Page</h1>
<img src= "http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/tom3.gif"><p>
Welcome!  Hope you find something of interest in here.<p>
Professor of <A HREF="http://www.cs.cmu.edu/Web/FrontDoor.html">
Computer Science</a> and
<A HREF="http://www.ri.cmu.edu/"> Robotics</a>,<br>
Director, <A HREF="http://www.cs.cmu.edu/~cald"> Center for Automated Learning and Discovery </a> <BR>
Carnegie Mellon University<p>
<img src= "./telephone.xbm"> 412-268-2611, <img src= "./mail.xbm"><i><a href="mailto:tom.mitchell@cmu.edu"> Tom.Mitchell@cmu.edu</a></i>
<p>
<img src= "http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/colorsep.gif">

<H3><a name="research_interests">Research: Machine Learning, Computer Science</a></H3>
<i> How can we make computers improve automatically from experience?
</i> This question drives my research.  It includes

<ul>
<li> How can software agents learn to help their users? <ul>
<li> See our <A
href="http://www.cs.cmu.edu/afs/cs/project/theo-4/text-learning/www/index.html">
text learning and software agents </a> research </ul>
<br>
```

Figure 2: An example of a HTML code

In other words, we need an extra effort to transform the HTML code in the covariates of interest.

The CMU CS text-learning research group has already created a system which, for a restricted web database, is able to answer questions like: "give me all the professors who taught course xx " (the questions are written in symbolic form). The system learns from the training samples and is not only able to give the name of the professors but also it additional information that may be of interest such that "students of those professors". The method that they use is based on "naive Bayes" classifiers involving word counts. (See Craven et al, 1998). The model underlying is incorrect (e.g. it assumes independence of words counts within WWW pages) but the results remain impressive.

The organization of the remaining sections is as follows. Section 2 describe the populations, the units and also the assumptions due to the not unique definition of the ontology classes. Section 3 presents the data sets. Section 4 describes the models that will be used in the next section. Section 5 shows the results obtained from 3 different classifiers. Section 6 summarizes the results remarking that the the traditional statistical tools lead to results that are even better that those of the system already created (RAINBOW) and also emphasizes that adding "non informative" words increases the noise the model producing poor results. Section 7 is related to the future research plan.

2 Model assumptions

Our goal in this project is to use "standard" statistical tools of discriminant analysis and logistic regression to make inference on all the members associated with four Computer Science Departments who have a home page or are cited on a web page of the corresponding department. We plan to compare the results of the use of more traditional approaches with the results from the naive Bayes classifier referred to above. This database includes faculties, staff, students and all the courses and research projects contained or cited in the main web page of the corresponding department. In other words the ontology classes to be considered are faculties, staff, students, courses and research projects.

The instances of each ontology class are not uniquely defined. For example, an instance of a Person may be described by a single page (the person's home page), or by a reference to the person in a string of text within an arbitrary web page , or by a collection of interconnected web pages that jointly describe the person. This means that each instance of an ontology class is represented by one or more contiguous segments of hypertext in the web. By "contiguous segment of hypertext" we mean either a single web page, or a contiguous string of text within a web page or a collection of several web pages interconnected by hyperlinks.

At this stage of the problem, we make the following assumptions:

- Each class instance is represented by a single web page. If an instance happens to be described for multiple pages, the current system is trained to classify only the primary home page as a description of the class instance and ignore the neighboring affiliated pages. If the system encounters two non-contiguous web pages that represent instances of the same class, it creates two distinct instances of this class in its knowledge base.
- The units are related each other. The relation are defined in advance and they are of the form: “student of”, “research of”, “taught by”, etc. They will have only two arguments, so that the can be denoted by R(A,B), e.g., “A is a student of B”.
- Even when the web pages were written in different times we will not care about this at this stage of the problem.
- There might be people from very different cultures and different native languages. The style of writing may be very different among the different cultures. We will not take into account this point.

3 The Data

The data consist of a set of pages and hyperlinks drawn from the WWW files of some Computer Science Departments including those of University of Texas at Austin, Cornell University, University of Washington, University of Wisconsin and Carnegie Mellon University. For each University there are 2 files: one corresponding to the pages and other corresponding to the hyperlinks interconnecting the pages.

Here we assume that,

- Each web page represent an instance of an ontology class.
- Each instance of a relation is assumed to be represented on the web in one of two ways:
  - R(A,B) may be represented by a contiguous segment of hypertext that connects the segment representing A to the segment representing B. For example a hyperlink from somewhere in some faculty to somewhere in course web page.
  - R(A,B) may be represented by a contiguous segment of text representing A that contains the segment that represent B. For example, the an instance of the relation “taught by” might be represented because the home page of a faculty contains the phrase “Course XX” in a particular context.

This means that the connectivity files contain examples of the relations defined in the ontology. The files containing the web pages includes examples of both the classes and the relations defined in the ontology.

- We will ignore the graphics which might appear in the web pages. We will take into account only the words and the hyperlinks contained in each web page.
- We will consider that the web pages are homogeneous respect to the time in which they were written, the style of writing of each person, the different cultures and different native languages

In the file containing the pages, each of them is labeled as one of the following cases: faculty, student, staff, research project, course or other. There are 4,127 pages. Each file looks like the example below.

```
faculty:
http://www.cs.cornell.edu/Info/Department/Annual95/Faculty/VanLoan.html
http://www.cs.cornell.edu/Info/People/skeshav/
http://www.cs.cornell.edu/Info/People/rdz/rdz.html
http://www.cs.cornell.edu/Info/Department/Annual94/Faculty/Salton.html
http://www.cs.cornell.edu/Info/People/tve/tve.html
http://www.cs.cornell.edu/Info/People/halpern/

course:
http://www.cs.cornell.edu/Info/Courses/Current/CS212/CS212.html
http://www.cs.cornell.edu/Info/Courses/Current/CS611/home.html
http://www.cs.cornell.edu/Info/Courses/Spring-96/CS211/CS211.html
http://www.cs.cornell.edu/Info/Courses/Current/CS314/
http://www.cs.cornell.edu/Info/Courses/Current/CS631/home.html
http://www.cs.cornell.edu/Info/People/tah/cs611.html
```

The file corresponding to the hyperlinks contains a record for every link in the file of pages. There are 10,945 hyperlinks interconnecting the pages. Each of these files looks like the example below.

```
(#:g22435 *novalue*
     (underlined.text "Om Damani")
     (column 20)
     (line 55)
     (url "http://www.cs.utexas.edu/users/damani")
        (link.range texas-om-p.-damani)
        (link.domain texas-cs304p---teaching-assistants---spring-1996)
     (generalizations (u.connection)))
```

```
(#:g22443 *novalue*
     (underlined.text "1994-5 Annual Report")
     (column 6)
     (line 22)
     (url "http://www.cs.utexas.edu/users/UTCS/report/1995/")
        (link.range texas-contents)
        (link.domain texas-utcs-publications)
     (generalizations (u.connection)))

(#:g22433 *novalue*
     (underlined.text "Shailesh Kumar")
     (column 20)
     (line 58)
     (url "http://www.cs.utexas.edu/users/skumar")
        (link.range texas-shailesh-kumar)
        (link.domain texas-cs304p---teaching-assistants---spring-1996)
     (generalizations (u.connection)))
```

In the last file each link goes from *link.domain* to *link.domain*.

4 The Models

4.1 Notation

Assume that

- There is a vocabulary base of p words denoted by $\{w_1, ..., w_p\}$.

- Each unit is represented by a $(p+1)$-dimensional vector of the form, $(Y, \mathbf{X})$ where,
  - y_j can take any values on $\{1, ..., K\}$ with probabilities $\pi_1, ..., \pi_K$, respectively.
  - y_j indicates from which population $\mathbf{x}_j$ comes from.
  - $\mathbf{X}$ is a p-dimensional vector with components, we say, $(X_1, ..., X_p)$ where each x_i is the number of times that the word w_i appears in the page.

- The training set is denoted by $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n)$, where $\mathbf{X}_i = (X_{i1}, ..., X_{ip})$ for each $i = 1, .., p$.

4.2 The problem of discrimination between K populations

Assume that we ask the system for all the students of a particular University, we say University XX. The trained system has to browse the web and see whether each page belong to a student of XX. In other words, the trained system has to be able of classifying each page as either "student of XX" or "not a student of XX". After that, the system will give us the required list of students. That is, the problem of discrimination between K populations plays an important role here.
If $(y, \mathbf{x})$ denote a new individual and $(y_1, \mathbf{x}_1), ..., (y_n, \mathbf{x}_n)$ is a training sample, we predict $y = i$ if

$$i = argmax\{P(y = j/\mathbf{x}), j = 0, ..., K\} \qquad (1)$$

For the initial steps of the problem we assume that the components of $\mathbf{x}$ are given for $\{x_1, ..., x_p\}$ where each x_i is the number of times that the word w_i appears in the page.
We plan to apply three different methods of classification: polytomous logistic model, linear discriminant model for normal populations, and naive Bayes classifier.

The probabilities given by (1) are different for each of the methods cited above.

- The polytomous logistic model assumes that

$$P(y = i/\mathbf{x}) = \frac{exp(\beta_{0i} + \beta_i^{'}\mathbf{x})}{1 + \sum_{j=1}^{k} exp(\beta_{0j} + \beta_j^{'}\mathbf{x})} \quad i = 1, ..., K$$

$$P(y = 0/\mathbf{x}) = 1 - \sum_{j=1}^{K} P(y = j/\mathbf{x})$$

  The β 's are estimated from the training samples by maximizing the conditional likelihood function given by,

$$f(y_1, ..., y_n/\mathbf{x}_1, ..., \mathbf{x}_n) = \prod_{j=1}^{n} \frac{\prod_{l=1}^{K} exp[(\beta_{0l} + \beta_l^{'}\mathbf{x}_j)I_{[y_j=l]}]}{1 + \sum_{i=1}^{k} exp(\beta_{0i} + \beta_i^{'}\mathbf{x}_j)}$$

- Case of normally distributed populations. Here, it can be shown that $P(y = j/\mathbf{x})$ has the same form as in the logistic model but the β 's are related with the mean and the covariance matrix of the populations. The relation is the following (see Efron, 1975):

$$\beta_{0i} = log\frac{\pi_i}{\pi_0} - \frac{1}{2}(\mu_i^{'}\Sigma^{-1}\mu_i - \mu_0^{'}\Sigma^{-1}\mu_0)$$

$$\beta_i^{'} = (\mu_i - \mu_0)^{'}\Sigma^{-1}$$

  where $\mathbf{x}/y = j$ has multivariate normal distribution with parameters (μ_j, Σ) The μ's and the matrix Σ are estimated by maximum likelihood using the training sample (see Anderson, 1984).

- Naive Bayes classifier. Here we assume that the components of $\mathbf{x}$, let's say $(x_1, ..., x_p)$ are independent each other. Thus we have,

$$P(y = j/\mathbf{x}) = \frac{P(\mathbf{x}/y = j)\pi_j}{\sum_{i=0}^{K} P(\mathbf{x}/y = i)\pi_i}$$

which in this case results to be equal to

$$P(y = j/\mathbf{x}) = \frac{\pi_j \Pi_{l=1}^{p} P(x_l/y = j)}{\sum_{i=0}^{K} \pi_i \Pi_{l=1}^{p} P(x_l/y = i)}$$

For each j ($j = 1, ..., p$), $\mathbf{x}/y = j$ has multinomial distribution. So the maximum likelihood estimator for each $P(x_l/y = j)$ is given by the number of times that the word w_l appears in the documents of class j over the total number of words observed in class j. Thus the observed zeros for the frequency of some words in some of the classes will produce the scores for the loosing classes to gravitate toward zero and the score for the winning class to gravitate toward 1. For this reason, sometimes a smoothing version of the involved probabilities helps to improve the results. Some proposes for that

4.3 The RAINBOW system

RAINBOW is a system developed by the CMU CS text-learning research group and created to classify web pages. The approach is an adaptation of the Naive Bayes Classifier with some modification based on Kullback-Leibler Divergence. So, it works assuming that all the possible words of the web pages of the universe are in a "Bag Of Words" and each page is made by randomly drawing a given number of words. The last part of the name RAINBOW are exactly the initials of "Bag Of Words" (the first part, RAIN, is just a nice word that sounds good).

It splits the training sample in 2 pieces: one of them is used to build the classifier (70 % by default) and the rest is used for the cross-validation. More precisely, it classifies a document $(y, \mathbf{x})$ as belonging to class m if

$$m = argmax \left\{ \frac{log \pi_j}{n} + \sum_{i=1}^{p} P(x_i/\mathbf{x}) log \left(\frac{P(x_i/y = j)}{P(x_i/\mathbf{x})} \right), j = 0, .., K \right\}$$

where n is the total number of words in $\mathbf{x}$, i.e. $n = \sum_{i=1}^{p} x_i$.

This method makes exactly the same classification as Naive Bayes but produces classifications that are less extreme. Observe that the parameters to be estimated correspond to the probabilities $P(X_i = x_i/y = j)$. Since there are many observational zeros, the estimator has to be robust in that sense. To estimate these probabilities, RAINBOW uses the Witten-Bell smoothing estimate (see Witten-Bell, 1991) given by,

$$\hat{P}(X_i = x_i/y = j) = \frac{\frac{p}{p-T_j} I_{[N(w_i,j)=0]} + N(w_i,j) I_{[N(w_i,j)\neq 0]}}{T_j + \sum_{l=0}^{K} N(w_l,j)}$$

where,

- $N(w_i,j) = \sum_{k=1}^{n} x_{ki} I_{[y_i=j]}$ is the total number of times that word w_i occurs in the training data of class j

- T_j is the total number of unique words in class j

Observe that this is nothing else than a smoothing version of the MLE.

Note that thousands of different words can be found in the web pages. By default, RAINBOW uses 2000 words as a vocabulary base (i.e., $p = 2000$). These words are the most informative words as measured by information gain (see Mitchell, 1995). This is a method commonly used by the data miners to do variable selection and it seems that works well among several alternatives (see Yang and Pederson, 1997). For more details see also Craven et al (1998).

5 Results

We used 1038 web pages, distributed as follows,

	Students	Projects	Faculties	Courses	Totals
Texas	147 59%	20 8%	46 18%	38 15%	251 100%
Cornell	127 56%	20 9%	34 15%	44 20%	225 100%
Washington	126 49%	21 8%	31 12%	77 30%	255 100%
Wisconsin	155 50%	25 8%	42 14%	85 28%	307 100%
Totals	555	86	153	244	1038

Table 1: size of the training samples

As we explained above, RAINBOW is based on the Naive Bayes classifier. Even when we know that the words don't occur independently, the system does a very good job classifying the 4 most populous classes (students, faculties, courses and research projects). RAINBOW can be run using any set of words as a vocabulary base but the work done to date by the research group that created RAINBOW, has used 2000 words as a vocabulary base (see Section 4.3).

There are 2 relevant features to evaluate a classifier: "accuracy" and "coverage" of each class. For a particular class, "accuracy" is the percentage of individuals of each class that were well classified. "Coverage" is the percentage of documents classified in each class that truly came from that class.

In all this work the procedure is based on drawing 70% of the training sample randomly and using the rest for the cross validation. For each class the selection is proportional to its sample size.

Table 2 shows a confusion matrix obtained from RAINBOW and also the accuracy and coverage for each class.

Classified as

		Students	Projects	Faculties	Courses	Totals
True	Students	150	1	9	9	169
	Projects	7	9	9	2	27
	Faculties	14	1	29	1	45
	Courses	1	0	2	67	70
	Totals	172	11	49	79	311

	Students	Projects	Faculties	Courses
Accuracy	89%	33%	64%	96%
Coverage	87%	81%	59%	85%

Table 2: Confusion matrix, accuracy and coverage from RAINBOW using 2000 words

It seems that "students" and "courses" are classified reasonable well but the rate of proper classification for "projects" is indeed low. We can also think about an overall measure of accuracy given as the the ratio between the total number of web pages that were well classified and the total number of individuals. According to that measure, the average overall accuracy of RAINBOW was around 82% (255 out of 311). Observe that the data for the cross-validation are sampled at the same rate as the training samples. This produces a measure of accuracy that tends to be high, despite the fact that RAINBOW classifies research projects poorly. But, since most of the real cases come from the very well classified classes, one might adopt this as a measure of overall accuracy.

Of course, if we take a different random sample, the results will differ. To have an estimate of the variability of the accuracy and coverage, we repeated the procedure 30 times. The results are shown in Figures 3a) and 3b).

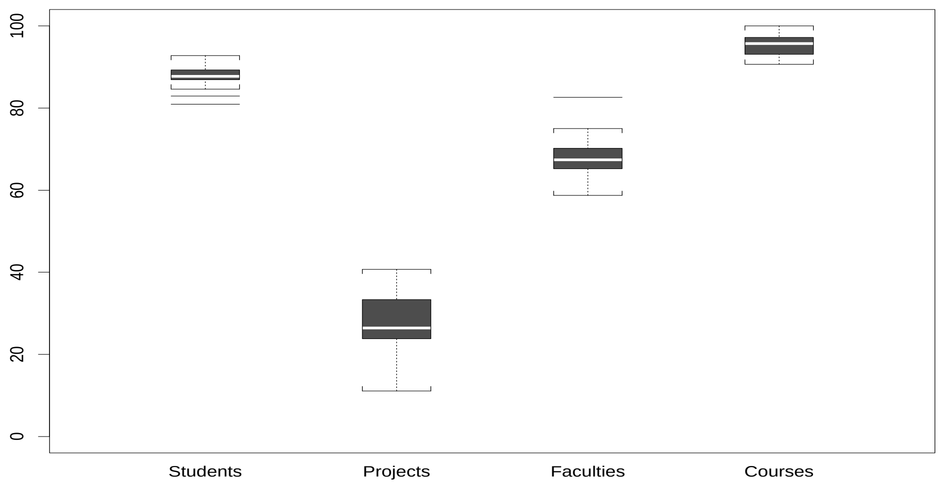


Figure 3a). Accuracy obtained from RAINBOW using 2000 words as vocabulary base

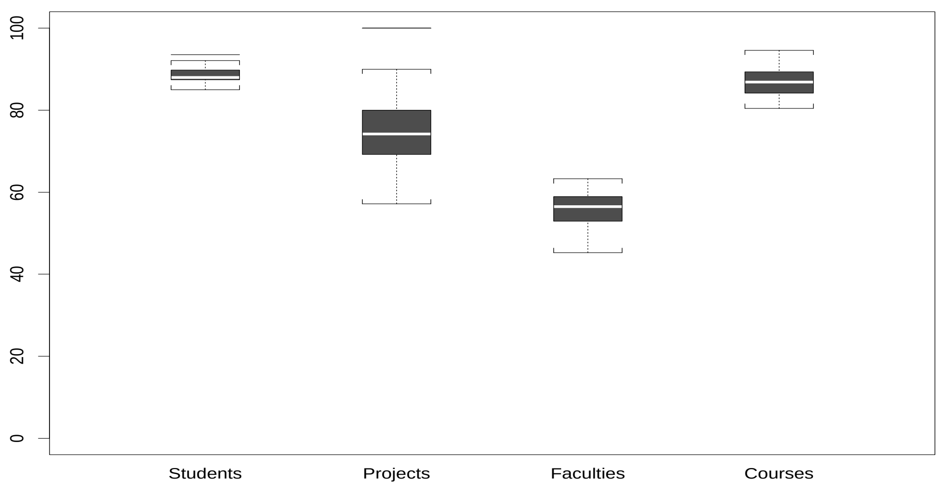


Figure 3b). Coverage obtained from RAINBOW using 2000 words as vocabulary base

Observe that the classifier was very effective for classifying student and course web pages. It was fairly good for classifying faculties but poor to classify research projects.

We could not fit the other methods of discrimination mentioned in Section 4 because the statistical software available are not able to handle 2000 covariates.

It is important to note that the number of parameters to be estimated is huge. The methods needs to estimate the probability of occurrence of each word given each class. For a vocabulary basic size of 2000, it implies that 8000 parameters are estimated.The first question that arises is whether is possible to reduce the dimensionality of the problem, i.e., can we find a

set of good words in the sense that they are powerful to discriminate among the classes?. The next steps are devoted to solving this problem.
Looking at the data, we would see that they consist mainly of a bunch of zeros, i.e., most of the words do not occur in a typical page. Appendix 1 contains a small piece of the database and also some univariate frequency tables. The first attempt to find a smaller set of words was to use the 101 most frequent words (we choose 101 instead of 100 because the last two have the same frequency). These words are shown in Table below. They are not sorted by their frequency.

"edu"	"office"	"rainbowthreedigit"
"rainbowtwodigit"	"rainbowdigits"	"rainbowonedigit"
"rainbowtime"	"are"	"in"
"other"	"can"	"also"
"be"	"by"	"students"
"course"	"home"	"page"
"algorithms"	"this"	"the"
"for"	"and"	"we"
"will"	"an"	"of"
"data"	"as"	"design"
"with"	"from"	"some"
"information"	"links"	"more"
"homework"	"you"	"mail"
"cs"	"rainbowphonenum"	"is"
"that"	"to"	"all"
"one"	"not"	"class"
"lecture"	"may"	"at"
"have"	"on"	"software"
"program"	"last"	"rainbowfourdigit"
"university"	"computer"	"science"
"department"	"fall"	"programming"
"group"	"language"	"new"
"your"	"here"	"or"
"if"	"email"	"time"
"postscript"	"student"	"lab"
"project"	"am"	"using"
"it"	"professor"	"system"
"which"	"work"	"performance"
"phone"	"he"	"about"
"engineering"	"systems"	"based"
"computing"	"web"	"has"
"research"	"my"	"me"
"parallel"	"distributed"	"rainbowfivedigit"
"conference"	"interests"	

Table 3: 101 most frequent words

Eight of the words shown in Table 3 begin with "rainbow" (remember that this is the name of the system). Words like "rainbowonedigit", "rainbowtwodigits", etc, represent any string of 1 digit, any string of two digits, etc. Words like "rainbowtime" or "rainbowphonenum" are identifying any string related to time or phone number, an so on. In the U.S. the courses are usually identified by numbers, so words representing string of numbers might be helpful in the discrimination.

Table 4 contains a confusion matrix obtained from RAINBOW using this 101 most frequent words and also the accuracy and coverage of each ontology class.

		Classified as				
		Students	Projects	Faculties	Courses	Totals
True	Students	143	6	8	8	165
	Projects	5	12	2	1	20
	Faculties	16	0	27	1	44
	Courses	12	3	1	56	72
	Totals	176	21	38	66	301

	Students	Projects	Faculties	Courses
Accuracy	87%	60%	61%	78%
Coverage	77%	57%	71%	85%

Table 4: Confusion matrix, coverage and accuracy from RAINBOW using the 101 most frequent words.

It seems that this set of words improved the accuracy of "project" but produced a lower coverage. The results of 30 runs of RAINBOW for this subset of words is shown in Figures 4a) and 4b).

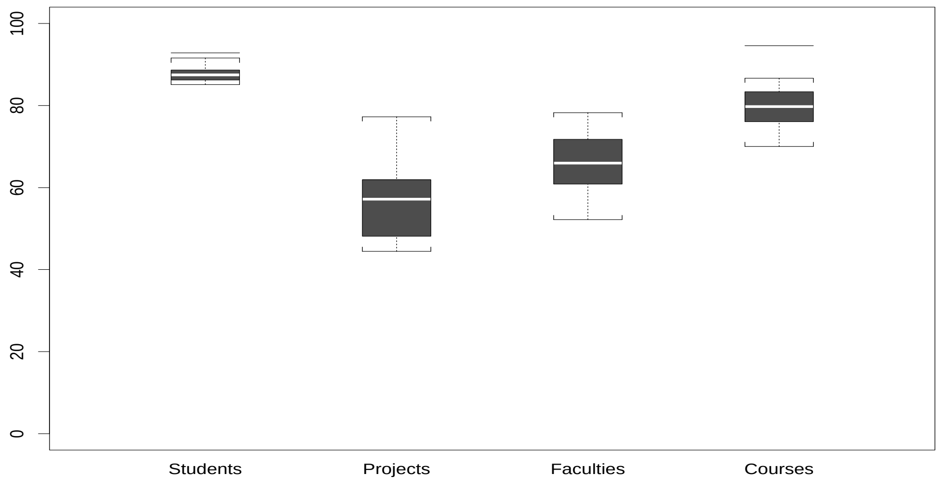


Figure 4a). Accuracy obtained from RAINBOW using 101 words as vocabulary base

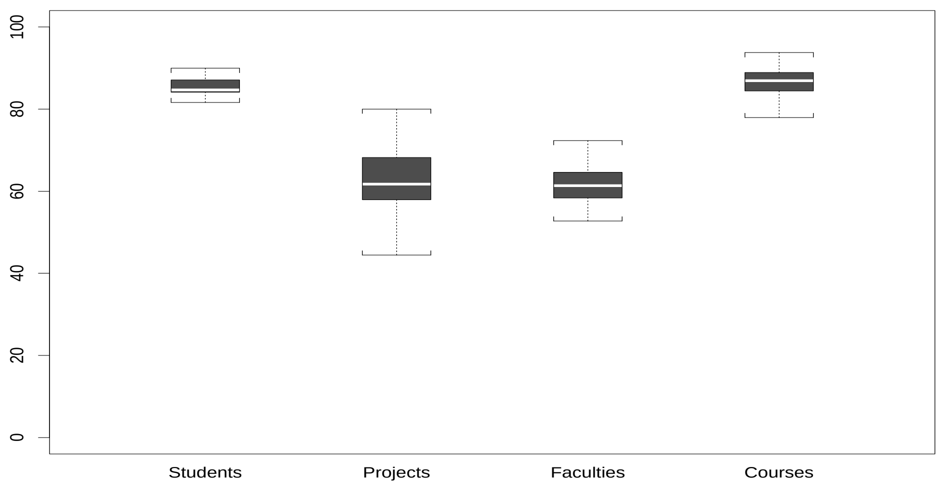


Figure 4b). Coverage obtained from RAINBOW using 101 words as vocabulary base

Comparing Figures 3a) and 4a), we observe that the smaller set of words improves considerable the accuracy of "projects" but is a little worse in "faculties" and "courses". With respect to the coverage, Figures 3b) and 3b) show that, even when we gain a little bit in "faculties", overall we losse by using a smaller set of words.

We also tried a similar analysis using the linear discriminant model. The normal model is incorrect but so it is the independence assumption of the Naive Bayes classifier. We used function "lda" of *Splus* to fit the linear discriminant model. Table 5 contains a confusion matrix for this situation and the corresponding accuracy and coverage for each class.

Classified as

		Students	Projects	Faculties	Courses	Totals
True	Students	153	0	10	4	167
	Projects	5	17	3	1	26
	Faculties	12	3	31	0	46
	Courses	20	1	3	49	73
	Totals	190	21	47	54	312

	Students	Projects	Faculties	Courses
Accuracy	92%	65%	67%	67%
Coverage	81%	81%	66%	91%

Table 5: Confusion matrix, coverage and accuracy from the LD model using the 101 most frequent words.

It seems that the linear discriminant model does not work too badly and might be an alternative to the approach used in RAINBOW. The results after 30 runs of this classifier are given in Figures 5a) and 5b). Comparing RAINBOW vs. the linear discriminant model (LD model) both based on 101 words, observe that none was better in accuracy but, overall, the coverage for the LD model was better. We also note that, overall, the coverage for the LD model based on 101 words was better that that of RAINBOW based on 2000 words (Figures 3b) and 5b)).

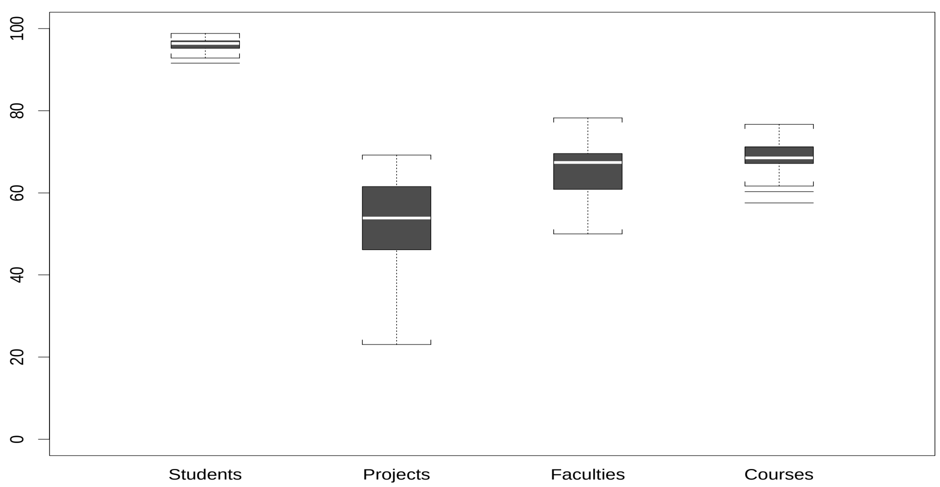


Figure 5a). Accuracy obtained from the LD model using 101 words as vocabulary base

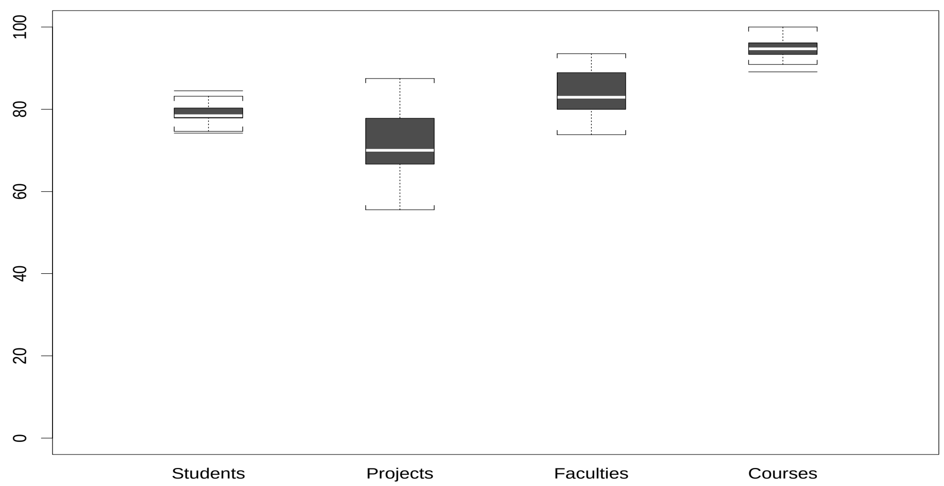


Figure 5b). Coverage obtained from the LD model using 101 words as vocabulary base

Even though the results based on 101 words seem to be fairly good, we do observe that we lose with respect to coverage. If we look at Table 3 again, we see that there are words such "as", "the", "of", etc. It is hard to believe that such words would help to discriminate the groups. Instead of adding information, there are increasing the noise. (For a good explanation about why these words are of little or no value, see Mosteller and Wallace, 1984). Therefore we decided to eliminate the set of noninformative words given in Table 6.

"edu"	"are"	"in"	"other"	"can"	"also"	"be"	"by"	"this"
"the"	"for"	"and"	"we"	"will"	"an"	"of"	"as"	"with"
"from"	"some"	"more"	"you"	"mail"	"cs"	"is"	"that"	"to"
"all"	"not"	"may"	"at"	"have"	"on"	"your"	"or"	"if"
"email"	"am"	"it"	"which"	"he"	"about"	"web"	"has"	

Table 6: Noninformative words

Thus we have now reached the point where we are keeping only 57 out of 2000 words. This means a huge reduction of the dimensionality. Tables 7 and 8 show confusion matrices obtained from RAINBOW and the linear discriminant model using these 57 words. The results after 30 runs of each method are shown in Figures 6a), 6b), 7a) and 7b).

Classified as

		Students	Projects	Faculties	Courses	Totals
True	Students	144	4	10	5	163
	Projects	2	16	7	0	25
	Faculties	9	0	32	1	42
	Courses	8	8	3	54	73
	Totals	163	28	52	60	303

	Students	Projects	Faculties	Courses
Accuracy	88%	64%	76%	74%
Coverage	88%	57%	62%	90%

Table 7: Confusion matrix, coverage and accuracy from RAINBOW using 57 words.

Classified as

		Students	Projects	Faculties	Courses	Totals
True	Students	159	1	6	1	167
	Projects	10	15	0	1	26
	Faculties	10	2	34	0	46
	Courses	20	1	0	52	73
	Totals	199	19	40	54	312

	Students	Projects	Faculties	Courses
Accuracy	95%	58%	74%	71%
Coverage	80%	79%	85%	99%

Table 8: Confusion matrix, coverage and accuracy from the LD model using the 57 words.

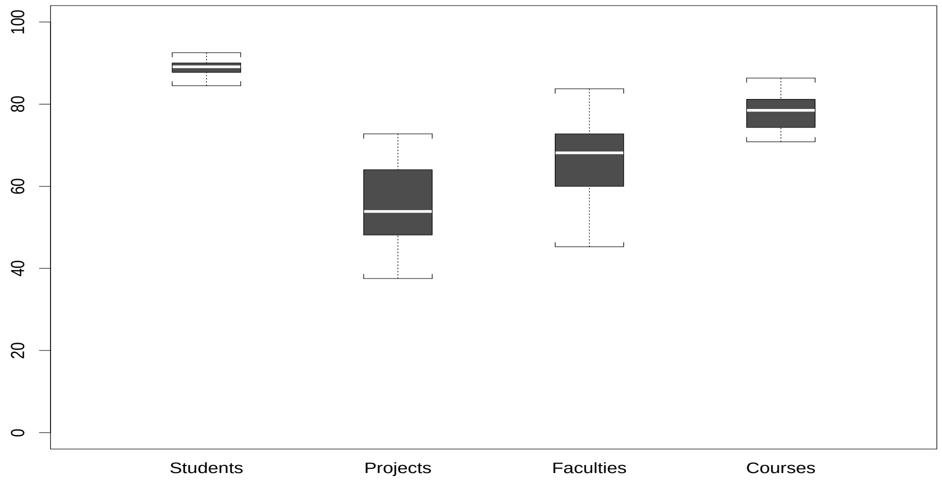


Figure 6a). Accuracy obtained from RAINBOW using 57 words as vocabulary base

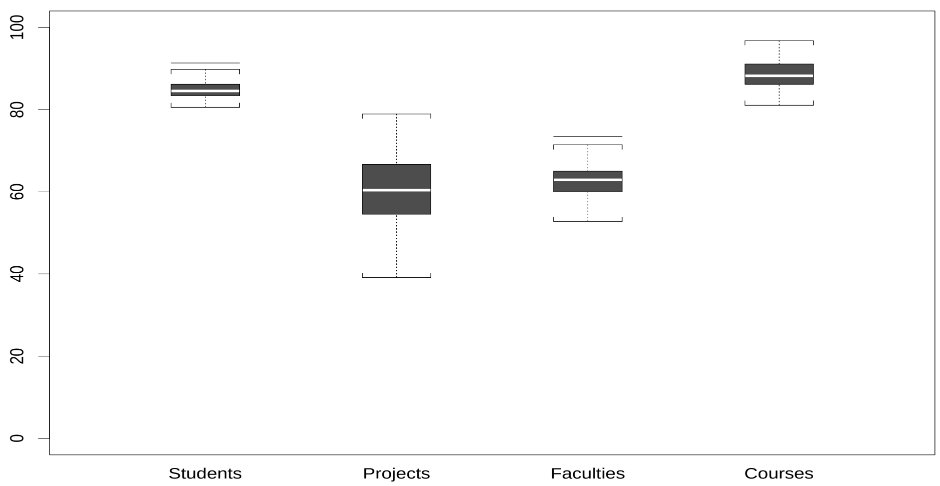


Figure 6b). Coverage obtained from RAINBOW using 57 words as vocabulary base

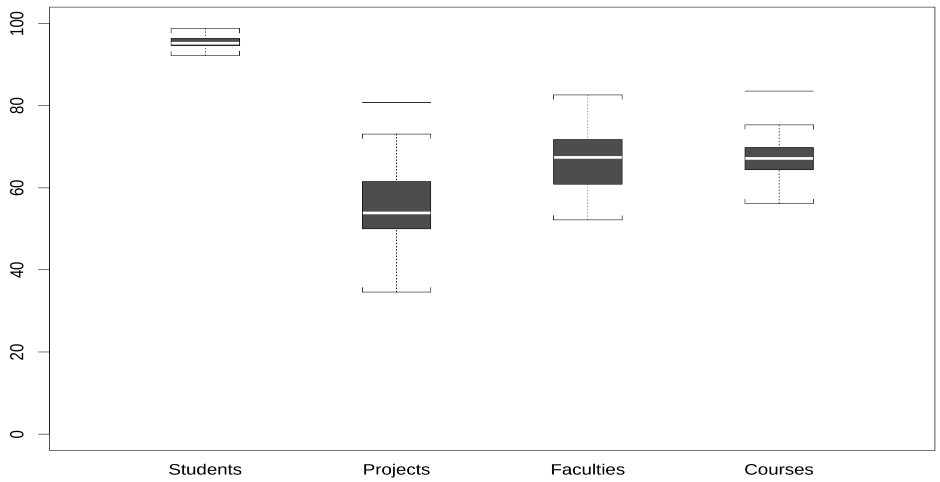


Figure 7a).Accuracy obtained from the LD model using 57 words as vocabulary base

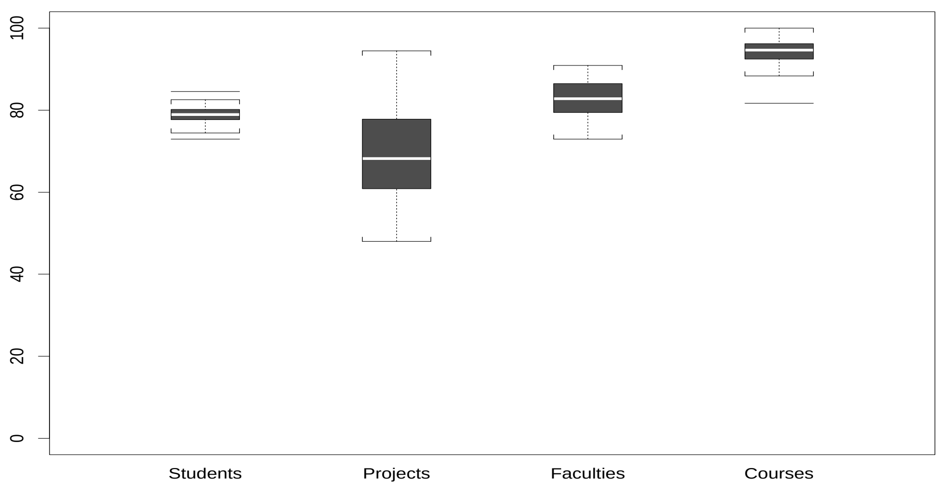


Figure 7b). Coverage obtained from the LD model using 57 words as vocabulary base

By comparing Figures 3a) and 6a), obtained from RAINBOW using 2000 and 57 words as a vocabulary base respectively, we see that, even when we loss accuracy in the classification of “course”, we improve considerably for “faculties” and “projects”. The results of this comparison emphasize that the subset of 57 words works better than the 2000 initial words. With respect to the coverage, the behavior is more or less the same (see Figures 3b) and 6b)). In other words, there is no reason to think that 2000 words produce better results that the final 57 words when RAINBOW is used.

By comparing Figures 6b) and 7b) we see that, overall, the LD model produces a better coverage than RAINBOW (both using 57 words). With respect to accuracy, neither is a clear winner.

Can we reduce the dimensionality even more? The answer appears to be no. Any subset of these 57 words turned out to poorly discriminate among the classes.

Finally, we consider logistic regression as another classifier. Most statistical software for logistic regression cannot handle such a large number of unknown parameters (even when we use 57 words as a vocabulary base). In order to avoid computational problems, we therefore tried the following nested model:

```
students vs. {research projects, faculties, courses}
research projects vs  {faculties, courses}
faculties vs. courses
```

In other words, we tried a nested model containing three dichotomic logistic regressions. Since "students" were always well classified, we decided to start by comparing them with the other three categories. The sparseness of the data and 57 covariates are too much for *Splus* to handle here.

But we were able to implement a mixture of logistic and linear discriminant models. We try a logistic model for "students vs. { research projects, faculties, courses } " and then a linear discriminant function for the remaining three groups. To find the best model (with no interactions), we used the function "step" of Splus applied to a "glm" object (in this case the logistic model). The significant coefficients for the logistic regression correspond to the words listed in Table 9. The subset of most powerful words for the remaining three groups is given in Table 10 (these were obtained by a stepwise procedure). The confusion matrix is given in Table 11.

"office"	"rainbowonedigit"	"students"	"course"
"page"	"information"	"links"	"office"
"student"	"one"	"class"	"lecture"
"last"	"rainbowfourdigit"	"computer"	"science"
"department"	"fall"	"group"	"language"
"time"	"student"	"project"	"professor"
"phone"	"systems"	"based"	"research"
"my"	"me"	"distributed"	"rainbowfivedigit"
"interests"			

Table 9: Most powerful words to discriminate between "students" and the remaining classes

```
"rainbowthreedigit" "rainbowonedigit"   "rainbowtime"
"course"            "home"              "homework"
"class"             "fall"              "group"
"time"              "project"           "system"
"performance"       "systems"           "based"
"research"          "me"
```

Table 10: Most powerful words to discriminate between “research projects, faculties and courses”.

Classified as

True		Students	Projects	Faculties	Courses	Totals
	Students	152	5	9	0	166
	Projects	6	9	8	3	26
	Faculties	6	2	32	6	46
	Courses	2	2	3	66	73
	Totals	166	18	52	75	311

	Students	Projects	Faculties	Courses
Accuracy	92%	33%	69%	90%
Coverage	91%	52%	61%	88%

Table 11: Confusion matrix, coverage and accuracy from the mixed model.

This particular mixed model does not work as well as the LD model. Also note that the LD model is much simpler.

6 Summary

The accuracy of the linear discriminant model is at least as good as that obtained from RAINBOW but its coverage is much better.

We also emphasize that the real dimensionality, p of the models we ended up fitting is 3% of the original dimensionality explored in RAINBOW (57 words vs. 2000 words). The linear discriminant model shows an accuracy of the same order than that of RAINBOW but is much better in coverage.

In order to see what is the general appropriateness of using the linear discriminant model, we made the classifier using all the 4 universities (1038 data) and use 3157 extra data to do the cross validation. This is important because in real applications, the classifiers will be estimated by using a very small part of the WWW. Table 12 shows the results for this cross validation. Accuracy (except accuracy for "students") and coverage lye in the range of the boxplots given by Figure 5a) and 5b). There is a considerable loss of accuracy for "students". The loss is due to the fact that the new data are more spread than that of the 4 universities.

Classified as

		Students	Projects	Faculties	Courses	Totals
True	Students	1026	27	24	6	1083
	Projects	169	202	33	14	418
	Faculties	343	19	591	18	971
	Courses	222	21	23	419	685
	Totals	1760	269	671	457	3157

	Students	Projects	Faculties	Courses
Accuracy	95%	48%	61%	61%
Coverage	58%	75%	88%	92%

Table 12: Cross-validation using 3157 extra data (LD model)

If we do the same with RAINBOW, the results are the following.

Classified as

		Students	Projects	Faculties	Courses	Totals
True	Students	966	19	88	10	1083
	Projects	121	186	100	11	418
	Faculties	394	21	546	10	971
	Courses	123	31	66	465	685
	Totals	1604	257	800	496	3157

	Students	Projects	Faculties	Courses
Accuracy	89%	44%	56%	68%
Coverage	60%	72%	68%	94%

Table 13: Cross-validation using 3157 extra data (RAINBOW).

In order to check how sensitive the LD model (based on 57 words) is to the size of the training sample, we randomly draw subsets with different sizes from the big dataset containing more than 4000 web pages. The accuracy and coverage are shown in Appendix 2. It seems that neither the accuracy nor the coverage change too much as the sample size changes. The results based on a sample size of 1000 do not differ too much from those based on a size of

3700. This is another very good behavior of the LD model.

Summarizing, the linear discriminant model works even a little better than the naive Bayes classifier and can be taken as a good alternative to classify web pages.

7 Future Work

The work presented here did not include an analysis of the hyperlinks of the web pages. Those are an important source of information that we still need to take into account.

The two inmediate steps in the continuation of this work will be:

- Model the data on hyperlinks appearing in the web pages without including the words. That is, analyse only the hyperlink structure. This approach would thus utilize directed graphs and one possibility is to try the model for social networks proposed by Fienberg, Meyer and Wasserman (1985) or a variant of it.

- Incorporate both text and hyperlinks in the same model. There are some attempts of this model in Blum and Mitchell (1998) using a conditional independence argument.

Once we have a good model including the words and the hyperlinks that occur in the pages, we will try to include the category "Other" in the ontology classes. This would be a very good approximation to the real situation in which most of the web pages to be classified come from that new category.

References

Anderson, T.W. (1984). *An Introduction to Multivariate Statistical Analysis*, 2nd ed. New York: Wiley.

Blum, A. and Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-Training. *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT'98).*

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S. (1998). Learning to Extract Symbolic Knowledge from the World Wide Web. *Technical Report 98-122*. School of Computer Science, Carnegie Mellon University.

Efron, B. (1975). The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis. *Journal of the American Statistical Association*, Vol.70, Number 352, p.892-898.

Fienberg, S., Meyer, M. and Wasserman, S. (1985). Statistical Analysis of Multiple Sociometric Relations. *Journal of the American Statistical Association*, Vol. 80, p.51-67.

Mosteller, F. and Wallace, D.(1984). *Applied Bayesian and Classical Inference: The Case of the Federalist Papers.* Springer-Verlag.

Mitchell, T. (1997). *Machine Learning.* McGraw-Hill, Inc.

Witten, I. and Bell, T. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory 37(4).*

Yang, Y. and Pederson, J. (1997). Feature selection in statistical learning of text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, p.412-420. Morgan Kaufmann.

APPENDIX 1

Example of a data set

Any small piece of the dataset looks like the following (each row is a web page and each column is the frequency of a given word).

	V4	V5	V6	V7	V8	V9	V17	V18	V19	V20	V21	V30	V32	V36	V37
739	0	1	1	0	2	1	0	0	0	0	0	0	1	1	0
740	2	2	0	0	5	6	0	0	0	0	0	0	0	1	0
741	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
742	0	3	0	0	0	0	1	3	0	0	0	0	1	0	0
743	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
744	0	1	0	0	1	2	0	3	0	0	0	0	0	2	0
745	0	0	3	0	7	1	0	3	0	0	0	0	0	2	0
746	2	4	2	4	6	3	1	2	0	1	0	0	0	0	1
747	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
748	0	0	8	6	3	0	0	4	1	1	0	0	0	3	0
749	2	4	3	1	3	3	1	14	2	2	0	0	5	4	1

Some univariate tables

"OFFICE"

value	0	1	2	3	4	6	7	8	9	13	14	36
frequency	686	163	101	36	26	12	8	1	1	1	1	2

"IN"

value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
frequency	272	148	119	104	76	51	59	32	22	26	19	17	15	19	6	1	9	5

value	18	19	20	21	22	23	25	26	27	28	31	33	36	37	42	43	77
frequency	2	5	7	2	5	2	1	2	2	1	2	1	1	1	2	1	1

"STUDENTS"

value	0	1	2	3	4	5	8
frequency	885	106	25	15	4	2	1

"COURSE"

value	0	1	2	3	4	5	6	7	8	9	12	13	14	16	19
frequency	785	111	52	36	17	10	4	7	5	4	2	1	2	1	1

"AND"

value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
frequency	224	117	104	69	77	65	47	46	31	25	21	27	18	22	13	9	10

value	17	18	19	20	21	22	23	24	25	26	27	28	29	30	32	33	34	35
frequency	14	14	7	4	5	6	6	4	4	4	5	2	1	4	2	4	6	4

value	37	38	39	42	44	47	49	53	54	56	57	70	72	124
frequency	2	2	1	2	1	1	1	1	1	1	1	1	1	1

"PROFESSOR"

value	0	1	2	3	4	5	6
frequency	859	146	18	7	6	1	1

"SYSTEM"

value	0	1	2	3	4	5	6	7	8	9	10	11	14	15	16
frequency	784	118	55	34	14	13	4	3	4	1	1	2	3	1	1

APPENDIX 2
Splus codes

```
f.70.fun <- function(dato.mat,x)
{

# INPUTS:
#        "dato.mat" is a matrix containing the entire data set
#        "x" is a number between 0 and 100
#        "clases" is a vector containing the class label
#                 of each individual

# This function draws randomly x% of de training data,
# use this new sample to estimate a linear discriminant model
# and make the cross-validation with the remaining data.

stu <- dato.mat[clases==1,]
pro <- dato.mat[clases==2,]
fac <- dato.mat[clases==3,]
cou <- dato.mat[clases==4,]

m1 <- dim(stu)
m2 <- dim(pro)
m3 <- dim(fac)
m4 <- dim(cou)

n1 <- round(x*m1[1]/100,0)
n2 <- round(x*m2[1]/100,0)
n3 <- round(x*m3[1]/100,0)
n4 <- round(x*m4[1]/100,0)


s.st <- sample(c(1:m1[1]),n1)
s.pr <- sample(c(1:m2[1]),n2)
s.fa <- sample(c(1:m3[1]),n3)
s.co <- sample(c(1:m4[1]),n4)

stu.ld <- stu[s.st,]
pro.ld <- pro[s.pr,]
fac.ld <- fac[s.fa,]
cou.ld <- cou[s.co,]

stu.n <- stu[-s.st,]
pro.n <- pro[-s.pr,]
```

```
fac.n <- fac[-s.fa,]
cou.n <- cou[-s.co,]


dat.ld <- rbind(stu.ld,pro.ld,fac.ld,cou.ld)
dat.ne <- rbind(stu.n,pro.n,fac.n,cou.n)
clase <- c(rep(1,n1),rep(2,n2),rep(3,n3),rep(4,n4))
class.cl <- c(rep(1,m1[1]-n1),rep(2,m2[1]-n2),rep(3,m3[1]-n3),
            rep(4,m4[1]-n4))

ld <- lda(dat.ld,clase)
pred <- predict.lda(ld,dat.ne)$class
crosstabs(~class.cl+pred)
}



inf.rel.fun <- function(conf)
{
# INPUT: "conf" is a confusion matrix

# This function give the accuracy and the coverage for
# each class of given confusion matrix

a <- (conf[1,1]+conf[2,2]+conf[3,3]+conf[4,4])*100/sum(conf)
pres <- round(a,2)

b <- c(conf[1,1]/sum(conf[1,]),conf[2,2]/sum(conf[2,]),
      conf[3,3] sum(conf[3,]),conf[4,4]/sum(conf[4,]))*100
well.class <- round(b,2)

c <- c(conf[1,1]/sum(conf[,1]),conf[2,2]/sum(conf[,2]),
       conf[3,3]/sum(conf[,3]),conf[4,4]/sum(conf[,4]))*100
well.ret <- round(c,2)

m <- rbind(well.class,well.ret)
m
}



inf.tot <- function(k,datos.mat,perc)
{
# INPUTS: "k" is the number of rund of the LD model
```

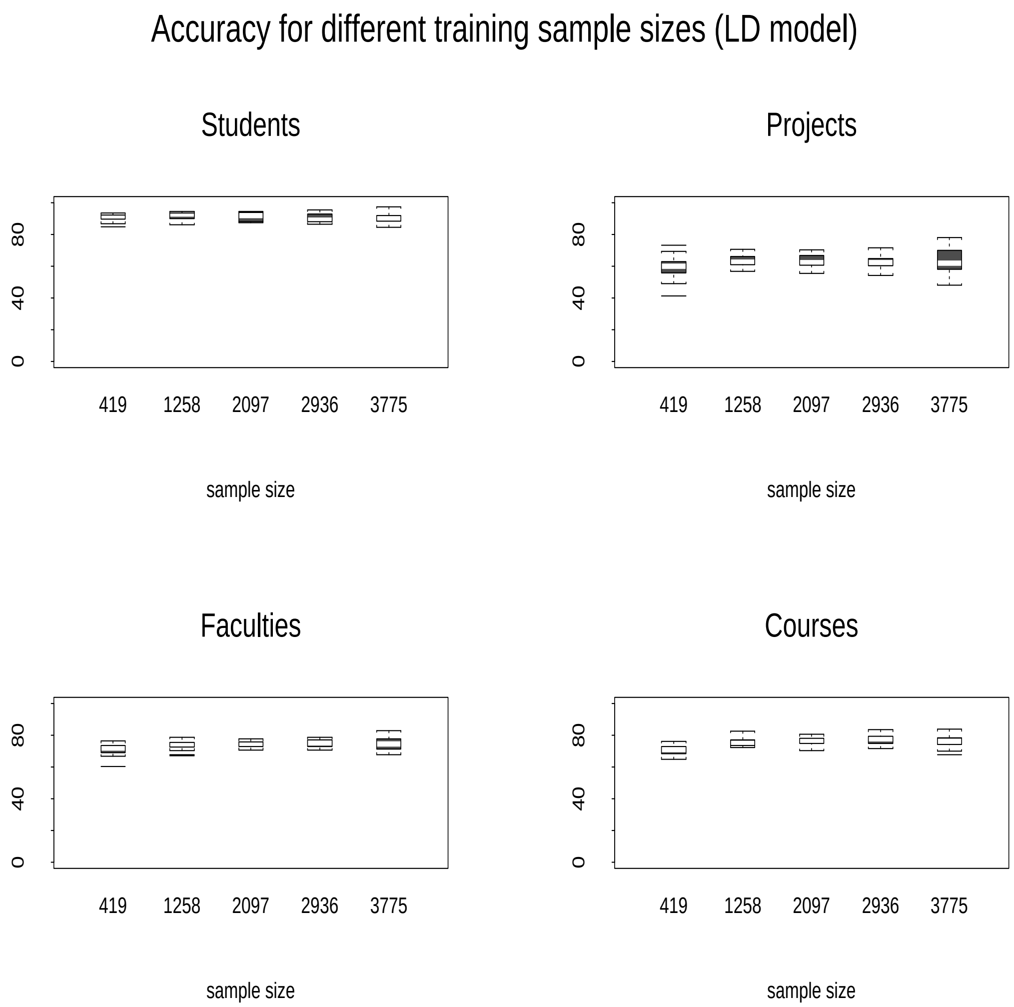
```
#         "datos.mat" is the training data matrix
#         "perc" is a number between 0 amd 100

# This function runs "k" times an LD model. Each run uses "perc"
# percent of the data to make the classifier and the rest for
# the cross-validation. The output is a matrix containing the
# accuracy and the coverage of each class for each run.

well <- matrix(0,nrow=k,ncol=4)
ret <- matrix(0,nrow=k,ncol=4)
for(i in 1:k)
{
   conf <- f.70.fun(datos.mat,perc)
   m1 <- inf.rel.fun(conf)
   well[i,] <- m1[1,]
   ret[i,] <- m1[2,]
   tot <- cbind(well,ret)
   tot
}
}



f.log.fun <-  function(dato.mat,x,clases)
{

# INPUTS:
#        "dato.mat" is a matrix containing the entire data set
#        "x" is a number between 0 and 100
#        "clases" is a vector containing the class label
#                 of each individual

# This function draws randomly x% of de training data,
# use this new sample to estimate a mixed model
# and make the cross-validation with the remaining data.

stu <- dato.mat[clases==1,]
pro <- dato.mat[clases==2,]
fac <- dato.mat[clases==3,]
cou <- dato.mat[clases==4,]

m1 <- dim(stu)
m2 <- dim(pro)
m3 <- dim(fac)
```

```
m4 <- dim(cou)

n1 <- round(x*m1[1]/100,0)
n2 <- round(x*m2[1]/100,0)
n3 <- round(x*m3[1]/100,0)
n4 <- round(x*m4[1]/100,0)


s.st <- sample(c(1:m1[1]),n1)
s.pr <- sample(c(1:m2[1]),n2)
s.fa <- sample(c(1:m3[1]),n3)
s.co <- sample(c(1:m4[1]),n4)

stu.ld <- stu[s.st,]
pro.ld <- pro[s.pr,]
fac.ld <- fac[s.fa,]
cou.ld <- cou[s.co,]

stu.n <- stu[-s.st,]
pro.n <- pro[-s.pr,]
fac.n <- fac[-s.fa,]
cou.n <- cou[-s.co,]

dat.ld <- rbind(stu.ld,pro.ld,fac.ld,cou.ld)
dat.ne <- rbind(stu.n,pro.n,fac.n,cou.n)
rem <- n2+n3+n4
clas.st <- c(rep(1,n1),rep(0,rem))
m <- cbind(clas.st, 1-clas.st)
mod <- glm(m~., family=binomial,data=dat.ld)
dat.3 <- rbind(pro.ld,fac.ld,cou.ld)
clase.3 c(rep(2,n2),rep(3,n3),rep(4,n4))
ld.3 <- lda(dat.3,clase.3)
cp <- 0.5

pred.dat <- predict.glm(mod,dat.ne,type="response")

a1 <- m1[1]-n1
a2 <- m1[1]-n1+m2[1]-n2
a3 <- m1[1]-n1+m2[1]-n2+m3[1]-n3
a4 <- m1[1]-n1+m2[1]-n2+m3[1]-n3+m4[1]-n4

stu.pr <- pred.dat[1:a1]
pro.pr <- pred.dat[a1+1:a2]
fac.pr <- pred.dat[a2+1:a3]
```

```
cou.pr <- pred.dat[a3+1:a4]

a <- stu.pr[stu.pr>cp]
b <- pro.pr[pro.pr>cp]
c <- fac.pr[fac.pr>cp]
d <- cou.pr[cou.pr>cp]

a1 <- length(a)
b1 <- length(b)
c1 <- length(c)
d1 <- length(d)

class.true <- c(m1[1]-n1-a1,m2[1]-n2-b1,m3[1]-n3-c1,m4[1]-n4-d1)

r1 <- c(a1,b1,c1,d1)

a <- stu.pr[stu.pr<cp]
b <- pro.pr[pro.pr<cp]
c <- fac.pr[fac.pr<cp]
d <- cou.pr[cou.pr<cp]

dat <- rbind(a,b,c,d)

pred.3 <- predict.lda(ld.3,dat)$class

a <- crosstabs(~class.true+pred.3)

r2 <- a[,1]
r3 <- a[,2]
r4 <- a[,3]

r <- cbind(r1,r2,r3,r4)
r
}
```

APPENDIX 3

Sensitivity of the LD model to different sizes of the training samples



Coverage for different training sample sizes (LD model)