Which classification algorithm works best with stylistic features of Portuguese in order to classify web texts according to users' needs?

Rachel Aires[1,2], Aline Manfrin[1], Sandra Aluísio[1], Diana Santos[2]

[1]NILC/ICMC-USP, [2]Linguateca, SINTEF
Instituto de Ciências Matemáticas e de Computação - ICMC-USP, SINTEF ICT
Av. Trabalhador São-carlense, 400. Centro. 13566-970. São Carlos – SP – Brazil, Pb 124 Blindern, 0314 Oslo, Norway
raires@icmc.usp.br, aline@nilc.icmc.usp.br, sandra@icmc.usp.br, diana.santos@sintef.no

Abstract. In order to improve Web Information Retrieval, we have, in a previous work (Aires et al., 2004), investigated the use of stylistic features of Web texts in Portuguese to classify web pages according to users' needs, using in most of the experiments the classification algorithm J48 (the Weka implementation of C4.5). From that study, we concluded that it was possible to identify some of the categories reliably, but we should investigate whether it was possible to get even better classification schemes using other algorithms. Language is a different domain, and the fact that C4.5 has been used successfully in other applications (even others dealing with written language) does not imply that it is also the best solution for our problem. In this paper, we document the replication of the experiments presented in Aires et al (2004), using all relevant Weka algorithms, also providing more information on the linguistic features used and on the issues concerning algorithm choice.

Keywords: Web Information Retrieval, stylistic features, users' needs, Portuguese language

1. Introduction

The actual size of the Web and its variety of texts allow us to find almost any type of information. The size of the web in Portuguese was estimated in 5,090,230,228 words in early November 2002 (Aires & Santos, 2002). Current search engines do a good job in matching the documents' topic with the user's search topic. Although texts can be about the topic that the user is looking for, they may not fulfil his/her needs. The reason for this is that the user might be looking for a text about the same subject of the recovered texts, but belonging to a different genre, text type, register type, style or quality.

According to Karlgren (2000), style is the difference between two ways of saying the same thing. Systematic stylistic variation can be used to characterize the genre of documents. Genre depends upon context and can be defined as a group of documents that are stylistically consistent and intuitive to accomplished readers of the communication channel in question.

Biber (1988) has studied English texts variation using several variables, and found that texts vary along five dimensions. Registers would then differ systematically along each of these dimensions, relating to functional considerations such as interactiveness, involvement, purpose, and production circumstances, all of which have marked correlates in linguistic structure.

Other work that has explored relatively stable characteristics of texts to be used on text categorization consists of the studies presented in Karlgren (2000), two of which are particularly interesting for our work. The first one (Karlgren, 2000: Chapter 7) was carried out with features similar to Biber's, but concentrating on those easy to compute with a POS tagger. Using texts from the Brown Corpus, three experiments were performed, with two, four or fifteen categories, respectively, correctly classifying 478, 366 and 258 texts out of 500. The second study (Karlgren, 2000: Chapter 16) explored how an interactive system could be designed to incorporate stylistic information in its interface, categorizing retrieval results by genre, and displaying the results using this categorization. In this experiment eleven categories were employed and an user-centred evaluation was performed. The users were asked to execute two tasks each, using the prototype of the interface that uses stylistic features and the web search engine Altavista. Karlgren concluded that most users used the interface as intended and many searched for documents in the genres the results could be expected to show up in.

We believe that simple stylistic items like word-based statistics, text-based statistics and statistics on specific items, used by Biber, Karlgren and others, can be used as well to automatically classify texts according to basic users´ needs, decreasing the user effort to find the information he is looking for.

The goal of our study was to find regularities in a corpus composed by web pages in Portuguese, which could be used in rules to classify texts according to users´ needs. This work is part of a larger project that consists on the development of a linguistically motivated approach for information retrieval

for Portuguese, named *Linguarudo*[1]. *Linguarudo* explores features of the language (Portuguese) during the interpretation of queries, matching and ranking. The results of the work presented here will be used on the dialog interface with the users. Our approach, by default, tries to detect automatically the user's need from his enquiries in natural language, based on pre-defined typical ways of posing questions, but also allows the user to choose the type of need his query is related to.

In the following sections we present the setup of our study and the results of the experiments carried out. The paper ends up with a discussion on the issues concerning algorithms choices.

2. Experimental Setup

2.1 Seven users´ needs

The classification in seven categories of users´ needs was the outcome of a qualitative analysis of two TodoBr[2] logs (a major Brazilian search engine). We selected these seven items as the most common users' needs by analysing the logs of November 1999 and July 2002. This classification is based on what the user wants:

1) A definition of something or to learn how or why something happens. For example, what are the northern lights? For this need, the best results would be presented by dictionaries and encyclopaedias, or even textbooks, technical articles and reports and texts of the informative genre.

2) To learn how to do something or how something is usually done. For example, find a recipe of his favourite cake, learn how to make gift boxes, or how to install Linux on his computer. Typical results are texts in the instructional genre, such as manuals, textbooks, readers, recipes and even some technical articles or reports.

3) A comprehensive presentation or survey about a given topic, such as a panorama of 20th century American literature. In this case, the best results should be texts of the instructional, informative and scientific genres, e.g. textbooks, reportages and long articles.

4) To read news about a specific subject. For example, what is the current news about the situation in Israel, what were the results of the soccer game on the day before or find about a terrible crime that has just happened in the neighbourhood. The best answers in this case would be texts of the informative genre, e.g. news in newspapers and magazines.

5) To find information about someone or some company or organization. For example, the user wants to know more about his blind date or to find the contact information of someone he met in a conference. Typical answers here are personal, corporation and institutional web pages.

6) To find a specific web page that he wants to visit, but does not remember its URL. For this type of need the results could be from any type of text or genre. The only way to identify this need would be the interface asking the user what type of page he is looking for.

7) To find URLs where he can have access to a given online service. For example, he wants to buy new clothes or to download a new version of software. The best answer to this kind of request is commercial text types (companies or individuals offering products or services).

These seven types are not claimed, however, to cover all kinds of user needs. Users may do all kinds of unpredictable searches, and we are not presuming to be able to recover their intentions by looking only at the logs[3].

2.2 The Corpus of Web texts

According to Gorsuch (1983: 332, apud Biber 1988: 65), the data in a factor analysis should include five times as many texts as linguistic features to be analysed. Although we are carrying out a different kind of analysis, we followed this recommendation.

In our experiment we created a corpus with 511 texts extracted from the Web, 73 for each type of need[4] plus additional 73 texts that would not answer any of the six types used (we call it "others"), in order to have a balanced corpus. Picking up the same number of texts for each type we ended up with

---

1 http://www.nilc.icmc.usp.br/nilc/projects/linguarudo.html

2 www.todobr.com.br

3 See Aires & Aluísio (2002) for a preliminary investigation on making intentions explicit.

4 Except for type 6, which, as explained above, can correspond to any kind of text.

considerable differences in the size of the parts of the corpus concerning the number of words, as can be seen in Table 1. We did not consider this difference in the size in words a problem for our study as the training instances are the texts, not their words.

The selection of the texts was carried out by five different persons who were instructed to maximize the variety of genres and subjects that could be relevant for the types of needs 1 to 5 and 7. We have used websites which were already known to contain the sort of things we look for. All the text in the page was used (the web pages were automatically converted into plain text, resulting in losing any text that was part of a picture), and links were not followed. As the variants of Portuguese differ on the lexical, morphological and syntactic levels we decided to use only one variant – the Brazilian Portuguese – in order to prevent interference in the classifier training. The resulting corpus has 640,630 words.

1	2	3	4	5	7	others
76,841	51,959	19,6450	39,533	67,601	39,951	168,295

Table 1: Corpus size per type of user need

It should be noted that while Biber's 481 texts amounted to a corpus with approximately 960,000 words, due to the fact that Web pages/texts are often smaller than texts in other media we only achieved 640,630 words. Another alternative to create the corpus would be to randomly select from a Brazilian Web collection like WBR-99 (Calado, 1999). We avoided this alternative because we would have to classify those pages according to the user's needs we were interested in.

2.3 Stylistic Features

The 46 features[5] used in our study were based on the ones in Biber (1988) and Karlgren (2000). We did not rely on POS taggers, parsers or analysis in other levels, in order not to have to revise manually their output, otherwise errors could interfere with our results. We used mainly closed lists[6]. The 46 features are shown on Figure 1.

Word-based statistics
Type/token ratio (**3**), capital type token ratio (**4**), digit content (**5**), average word length in characters (**6**), long words (>6 chars) count (**7**)
Text-based statistics
Character count (**1**), average sentence length in characters (**2**), sentence count (**8**), average sentence length in words (**9**), text length in words (**10**)
Other statistics
the subjective markers "acho", "acredito que", "parece que" and "tenho impressão que" ("I think so", "I believe that", "it seems that", "have the impression that") (**11**)
the present forms of verb to be "é/são" ("is/are") (**12**);
the word "que" (can be: noun, pronoun, adverb, preposition, conjunction, interjection, emphatic particle) (**13**)
the word "se" ("if/whether" and reflexive pronoun) (**14**);
the discourse markers "agora", "da mesma forma", "de qualquer forma", "de qualquer maneira" and "desse modo" ("now", "on the same way", "anyway", "somehow" and "this way") (**15**)
the words "aonde", "como", "onde", "por que", "qual", "quando", "que" and "quem" on the beggining of questions (wh-questions) (**16**)
"e", "ou" and "mas" as sentence-initial conjunctions ("and", "or", "but") (**17**);
amplifiers (**18**). Amplifiers scale upwards (Quirk et al, 1992), denoting or an upper extreme of a scale or a high degree, high point on the scale. Some examples are: "absolutamente" (absolutely), "extremamente" (extremely), "completamente" (completely) and "longe" (far).
conjuncts (**19**). Most conjuncts are adverbs and prepositional phrases (Quirk et al, 1992). Some examples are: "além disso" (moreover), "conseqüentemente" (accordingly), "assim" (thus) and "entretanto" (however).
downtoners (**20**). Downtoners have a lowering effect on the force of the verb and many of them scale gradable verbs, they can have a slight lowering effect, scale downwards considerably or serve to express an approximation to the force of the verb (while indicating its non-application) (Quirk et

[5] Numbers after the description of the category indicate the feature number used in the classifier.

[6] The lists were elaborated based on both the examples presented in Portuguese grammars, taking out, in some cases, words that could be ambiguos and the examples known by us.

al, 1992). Some examples are: "com exceção" (with the exception), "levemente" (slightly), "parcialmente" (partially) and "praticamente" (practically).
emphatics (**21**). Emphatics (emphasizers) have a general heightening effect (Quirk et al, 1992). Some examples are: "definitivamente" (definitely), "é óbvio que" (it is obvious that), "francamente" (frankly) and "literalmente" (literally).
suasive verbs (**22**). Some examples are the verbs: *aderir* (to adhere), *distinguir* (to distinguish), *crer* (to believe) and *dar* (to give).
private verbs (**23**). Some examples are the verbs: *partir* (to leave), *ter* (to have), *averiguar* (to check) and *guardar* (to keep).
public verbs (**24**). Some examples are the verbs: *abolir* (to abolish), *promulgar* (to promulgate), *mencionar* (to mention) and *declarar* (to declare).
number of definite articles (**25**); number of indefinite articles (**26**)
first (**27**), second (**28**) and third person pronouns (**29**)
number of demonstrative pronouns (**30**)
indefinite pronouns and pronominal expressions (**31**)
number of prepositions (**32**)
place adverbials (**33**); time adverbials (**34**)
number of adverbs (**35**)
number of interjections (**36**)
contractions (**37**)
causative (**38**), final (**39**), proportional (**40**), temporal (**41**), concessive (**42**), conditional (**43**), "conformative" (**44**), comparative (**45**) and consecutive conjunctions (**46**)

Figure 1 - The 46 features selected

The 46 features used to train the classifiers were calculated over the texts using a Perl script.

2.4 The Classification Algorithm

In our first study (Aires et al., 2004) we used mainly the J48 algorithm available on the Weka collection of machine learning algorithms (Witten & Frank, 2000). J48 is the Weka implementation of the decision tree learner C4.5. C4.5 was chosen for several reasons: it is a well-known classification algorithm, it has already been used in similar studies (Karlgren, 2000) and it can originate easily understandable rules. There are also some other reasons why C4.5 can be a good solution:

- We do not always know the distribution of our features, and some methods presuppose a specific kind of distribution, normal, for example, for all features. However, much linguistic data are not normally distributed (see e.g. Katz, 1996, Bahl et al, 1989, and Yarowsky, 1993). For example, some variables may have a polar or binary distribution. In those cases, it is better to use non-parametric measures (features with no a priori approximation of their value space) than measures based on complex distributional assumptions. Otherwise one would have to hope that the discrepancy would not affect the results, or to recode the variables to follow the required distribution or to try understanding the distribution of the features;
- Many knowledge-based systems combine information from different sources by weighting the sources, most usually doing a linear combination of measurements. But there is no reason always to assume that variables always engage in a relationship that is suitable for linear combination. C4.5 does not assume they do;
- C4.5 is designed to classify into predefined discrete categories (classes) that the training data belong to (as in our case);
- Since C4.5 only processes features one by one, it does not matter that there are interactions between the features we are using, C4.5 only allows interactions in the form of multi-part conditions. That may result in missing some positive effects of the interaction of features, but does not risk a false positive result, originated from a multivariate test based on false assumptions;
- On the case of Languages studies, particularly, Karlgren (2000: Chapter 10) states that "It is also wasteful of linguistic knowledge in the sense that linguists know or should know about interrelations between linguistic variables and not need to throw that information out in order to rediscover some of it in probabilistic formulae. Most importantly, results of this form have low or no explanatory power. Better ways of combining evidence, through production rules, decision trees, general pattern matching techniques, algebraic techniques, and combinations thereof are necessary to be able to make use of and understand linguistic data."

Despite of all advantages of C4.5, we must remember that there are several others classification algorithms available for use with similar characteristics, some also developed for non-parametric features. Those algorithms are, in addition, freely available to experiment with. Moreover, language can be seen from different angles; different tasks considering written language can display a different behaviour of the features considered in each case. Furthermore, there is not such vast comparison of classification algorithms for the domain of linguistic features as for other domains, and given that the costs are not high, there is no reason to restrict ourselves to the algorithms found better for other domains. This is why in this paper we decided to replicate the experiments of Aires et al (2004) using all the Weka algorithms which could deal with non-nominal features, with non-numerical classes, with the number of classes we needed (maximum 7) and which did not present errors related to the standard deviation of our features for any of our classes. Forty-four algorithms were used: Naive Bayes, Naive Bayes Multinomial, Naive Bayes Updateable, Multilayer Perceptron, SMO, Simple Logistic, IB1, IBK, KStar, LWL , AdaBoostM1, Attributive Selected Classifier, Bagging, Classification via regression, CV parameter selection, Decorate, Filtered classifier, Logit Boost, Multiclass classifier, Multi Scheme, Ordinal class classifier, Raced incremental logit boost, Random committee, Stacking, Stacking C, Vote, FLR, HyperPipes, VFI, Decision Stump, J48, LMT, Random Forest, Random Tree, REP Tree, User classifier, ZeroR, Conjunctive Rule, OneR, Decision Table, Part, NNGe, Ridor and JRIP. To test the generated classifiers we did a 10-fold cross-validation test.

3. Results

We have trained classifiers using 2, 3 (2 categories plus "others"), 4, 5 (4 categories plus "others"), 6 and 7 categories (6 categories plus "others") (Table 2).

2 categories	4 categories	6 categories
1) the union of needs 1, 2, 3, 4 and 5 2) need 7	1) the union of needs 1, 2, 3 2) need 4 3) need 5 4) need 7	Need 1 Need 2 Need 3 Need 4 Need 5 Need 7

Table 2: Categories used

Table 3 presents the percentage of correct classifications for the classifiers trained using the different algorithms, considering 7 categories. All the results were calculated using 10 folds cross-validation, except when mentioned not to. Table 4 presents the percentage of corrects for all the sets of categories for the algorithms J48 (1), Logistic Model Tree (LMT) (2 – the second best one) and Sequential Minimal Optimisation (SMO) (3 – the one with the best percentage of corrects for most of the classification schemes (see Table 4). LMT (Landwehr et al, 2003) is a classification algorithm for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves. SMO implements Platt's (1998) sequential minimal optimisation algorithm for training a support vector classifier using scaled polynomial kernels. Transforms output of SVM into probabilities by applying a standard sigmoid function that is not fitted to the data. This implementation does not perform speed-up for linear feature space and sparse input data. It globally replaces all missing values, transforms nominal attributes into binary ones, and normalizes all numeric attributes.

Algorithm used	Percentage of correct
Bayes	
Naive Bayes	53.62%
Naive Bayes Multinomial	45.83%
Naive Bayes Updateable	53.62%
Functions	
Multilayer Perceptron	53.44%
SMO	58.31%
Simple Logistic	57.33%
Lazy	
IB1	42.54%

IBK	42.54%
KStar	45.32%
LWL	39.98%
Meta	
AdaBoostM1	26.2%
Attributive Selected Classifier	13.7%
Bagging	54.9%
Classification via regression	54.02%
CV parameter selection	13.7%
Decorate	56.29%
Filtered classifier	13.7%
Logit Boost	55.22%
Multiclass classifier	13.7%
Multi Scheme	13.7%
Ordinal class classifier	13.72%
Raced incremental logit boost	13.7%
Random committee	50.16%
Stacking	13.7%
Stacking C	13.7%
Vote	13.7%
Misc	
FLR	30.81%
HyperPipes	30,81%
VFI	47.8%
Trees	
Decision Stump	
J48	45.32%
LMT	57.5342 %
Random Forest	54.28%
Random Tree	36.62%
REP Tree	47.55%
User classifier	13.7%
Rules	
ZeroR	13.7%
Conjunctive Rule	25.73%
OneR	29.31%
Decision Table	44.36%
Part	44.95%
NNGe	45.47%
Ridor	45.55%
JRIP	44.84%

Table 3: Percent of corrects with the seven categories for the forty-four algorithms[7]

Figure 3 shows precision and recall results divided by needs for the algorithms J48 (1), LMT (2) and SMO (3).

Number of categories	Percentage of correct1	Percentage of correct2	Percentage of correct3
2 categories	90.93%	93.8356 %	93.379 %
3 categories	76.97%	82.9746 %	81.9961 %
4 categories	65.06%	71.9178 %	73.7443 %
5 categories	56.56%	64.9706 %	67.9061 %
6 categories	52. 01%	62.7854 %	63.6986 %
7 categories	45.32%	57.5342 %	58.31%

Table 4: Percent of corrects for the algorithms J48, LMT and SMO

[7] All the algorithms were trained with their Weka original settings.

The classification with 2 categories decides whether a page gives any kind of information about a topic or gives access to an online service. The corresponding resulting tree, which uses 10 features, is shown in Figure 2.

```
feature25 <= 2.578269
|  feature34 <= 0.453858
|  |  feature33 <= 0.053419
|  |  |  feature22 <= 0.041494
|  |  |  |  feature6 <= 4.481243: Need7 (16.0)
|  |  |  |  feature6 > 4.481243: Need12345 (2.0)
|  |  |  feature22 > 0.041494: Need12345 (3.0/1.0)
|  |  feature33 > 0.053419: Need7 (33.0)
|  feature34 > 0.453858: Need12345 (3.0)
feature25 > 2.578269
|  feature9 <= 11.322034
|  |  feature14 <= 0.451467
|  |  |  feature28 <= 0.287356
|  |  |  |  feature31 <= 0.613027
|  |  |  |  |  feature43 <= 0: Need12345 (8.0)
|  |  |  |  |  feature43 > 0: Need7 (11.0/1.0)
|  |  |  |  feature31 > 0.613027: Need12345 (24.0)
|  |  |  feature28 > 0.287356
|  |  |  |  feature14 <= 0.344828: Need7 (14.0/3.0)
|  |  |  |  feature14 > 0.344828: Need12345 (2.0)
|  |  feature14 > 0.451467: Need12345 (25.0)
|  feature9 > 11.322034: Need12345 (297.0/2.0)
```

Figure 2: J48 tree to classify in 2 categories

	Precision1	Recall1	Precision2	Recall2	Precision3	Recall3
2 categories						
Need12345	0.94	0.951	0.949	0.978	0.952	0.97
Need7	0.739	0.699	0.871	0.74	0.833	0.753
3 categories						
Need12345	0.866	0.901	0.856	0.959	0.85	0.959
Need7	0.636	0.671	0.736	0.726	0.688	0.753
Others	0.426	0.315	0.7	0.288	0.737	0.192
4 categories						
Need123	0.737	0.781	0.768	0.863	0.761	0.872
Need4	0.556	0.479	0.683	0.562	0.719	0.562
Need5	0.431	0.384	0.491	0.356	0.561	0.438
Need7	0.692	0.74	0.747	0.808	0.808	0.808
5 categories						
Need123	0.663	0.63	0.705	0.817	0.707	0.872
Need4	0.57	0.671	0.603	0.562	0.75	0.534
Need5	0.278	0.274	0.491	0.384	0.5	0.411
Need7	0.553	0.644	0.714	0.753	0.704	0.781
Others	0.35	0.288	0.527	0.397	0.625	0.411
6 categories						
Need1	0.395	0.428	0.61	0.493	0.547	0.479
Need2	0.446	0.452	0.708	0.699	0.658	0.658
Need3	0.478	0.438	0.541	0.63	0.58	0.644
Need4	0.632	0.589	0.63	0.63	0.704	0.685
Need5	0.358	0.329	0.533	0.548	0.538	0.575
Need7	0.671	0.699	0.757	0.767	0.803	0.781
7 categories						
Need1	0.409	0.521	0.522	0.493	0.542	0.534
Need2	0.411	0.411	0.563	0.493	0.609	0.534
Need3	0.507	0.493	0.568	0.685	0.53	0.603
Need4	0.577	0.562	0.648	0.63	0.676	0.63

Need5	0.361	0.301	0.507	0.521	0.539	0.562
Need7	0.606	0.589	0.707	0.795	0.655	0.781
Others	0.296	0.288	0.484	0.411	0.525	0.438

Figure 3: Accuracy by class for the 6 classifications for the algorithms J48, LMT and SMO

The classification with 4 categories differentiates among information about something, someone or some company/institution/organization, news, and online services. Finally, the classification with 6 categories is the full one we have presented in Section 2.1, excluding type 6 that can be of any type of text or genre.

The class "others" contains text types like blogs, jokes, poetry, etc. Although it makes the classification task harder, it cannot be ignored, as is often done in works dealing with classifiers for closed domains or those not dealing with real world applications. Since we are going to use this work in *Linguarudo*, we will be dealing with many different texts that are not from the seven users' needs types considered in its dialogue interface. Then, examples from those different types should be used during classifier training to be able to reliably identify the seven types vs. the others not catered for by *Linguarudo*.

Using a cross-validation strategy we obtain worse but more reliable figures. For example, for seven categories, using 90% of the corpus for training and 10% for testing we got 49.42% of correct results against 45.32% using cross-validation and the algorithm J48. The same happened with the algorithms LMT and SMO, as we got 57.69 and 59.62% (respectively) of correct results against 57.53 and 58.31% (respectively) using cross-validation.

4. Discussion and Further Work

The work reported in Aires et al. (2004) can be considered preliminary, but it is the first, as far as we know, that tried to automatically categorise, in terms of user needs, the texts in Portuguese on the Web (actually, for the best our knowledge, it is the first one for any language, no matter this having been pointed out as relevant in (Broder, 2002)). Our hypothesis behind that paper was that it is going to be easier for an user to choose among types of needs than between genres or text types; this has to be confirmed later using a user-centred evaluation.

In the work presented on this paper we investigated the hypothesis that other algorithms could perform better than J48. We confirmed the initial results presented in Aires et al. (2004), where some algorithms performed better for the classification in 7 categories. For this paper we investigated thirty-two more algorithms for the classification in 7 categories and evaluated LMT and SMO (the two best ones) for the other four classification schemes (in 2, 3, 4, 5 and 6 categories). Fourteen of the thirty-two new algorithms achieved, regarding percentage of corrects, the same as (one of them) or better than (13 of them) J48.

SMO achieved better results for the classification in 4, 5, 6 and 7 categories, while LMT achieved better results for the classification in 2 and 3 categories. However, the difference on the percentage of corrects for the classification in 2 and 3 categories on both algorithms is very small. We believe that this does not justify choosing one algorithm for two schemes of classification and other one for the rest.

The best 2 algorithms for our task, SMO and LMT, don't have an output as easy to understand as J48. The functions used on their output can be easily used in our application, but they can not be as easily interpreted as simple rules, especially because of the weights used on them.

Nevertheless, it was shown that it is possible to discriminate reliably at least among some of the categories, and this should have a positive impact on the usability of a Web system. Just to separate between pages that give information and those that offer services (a task with a success rate of 90.95%, 93.83%, 93.38%, respectively in J48, LMT and SMO) seems intuitively useful.

The fact that there are other more precise algorithms than the J48 for our task (classification of web texts according to users´ needs) is going to be considered in our future work. The next experiments will be done using SMO (the one with the best results in most cases) and J48. J48 will be still used to allow future comparison with our work. As further work we have the following agenda:

- to enrich and reclassify the corpus in order to accept the fact that texts can belong to more than one need[8], and to increase it in size so that it may be employed later on also by other researchers in IR of Portuguese, following the general philosophy of Linguateca (www.linguateca.pt)

[8] See Santos (1998) for a general claim that linguistic classification should allow vagueness in membership, or vague categories.

- to devote considerable work to find more specific discriminating features. The ones we have used are too generic and neither have they been developed for the Web nor for the Portuguese language;
- to perform a detailed study of the discrimination features. As can be seen in Figure 2, for J48, only 10 of the 46 features have been employed to distinguish between two categories. For the 7 categories classification 40 features were used. These 2 cases exemplify the importance of analysing the resulting rules and eliminating those features that have not been used.
- to compare the results using simple features with a new study using also features depending on PoS taggers or parsers, and also using lemmas instead of simple forms;
- to investigate whether good results can be obtained by always classifying one class against all others, i.e. turning the classification into a set of binary ones; and
- finally, to study the use of a more flexible classification in terms of axes such as formal/informal, short/elaborated, contextualized or not, involved/detached, etc. allowing customized choices.

Acknowledgements

We thank Akwan Information Technologies (www.akwan.com.br) for the TodoBr logs; Luiz Carlos Genoves Jr. and Marcos Felipe Tonelli de Carvalho for the script to calculate the features and Crislaine Aparecida Francisco, Vanessa Silva Marquiafável and Lucélia Helena de Oliveira for building the corpus. This work was supported by Fundação para a Ciência e Tecnologia through the grant POSI/PLP/43931/2001 and co-financed by POSI.

References

Aires, R.; Manfrin, A.; Aluísio, S. Santos, Diana (2004) What is my Style? Using Stylistic Features of Portuguese Web Texts to classify Web pages according to Users' Needs. To appear in Proceedings of LREC 2004, Lisbon, Portugal.

Aires, R.V.X. & Aluísio, S.M. (2003). Como incrementar a qualidade das máquinas de busca: da análise de logs à interação em Português. Revista Ciência da Informação, 32(1), 5--16.

Aires, R. & Santos, Diana (2002). Measuring the Web in Portuguese. In Proceedings of the Euroweb 2002 Conference (pp. 198--199). Oxford, UK.

Bahl, L., P. Brown, P. de Souza & R. Mercer. (1989) A Tree-based Statistical Language Model or Natural Language Speech Recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 37, No. 7, July, pp. 1001-8.

Biber, D. (1988) Variation across speech and writing. Cambridge University Press. Cambridge, UK.

Broder, A. (2002)A Taxonomy of Web Search. SIGIR Forum 36 (2), Fall 2002, pp.3-10.

Calado, P. (1999) The WBR-99 Collection: Description of the WBR-99 Web collection data-structures and file formats. LATIN - Laboratório para o Tratamento de Informação, Dep. de Computação, Universidade Federal de Minas Gerais, Brazil.

Karlgren, J. (2000) Stylistic Experiments for Information Retrieval. PhD Dissertation. Stockholm University, Department of linguistics.

Katz, Slava M. (1996) Distribution of content words and phrases in text and language modelling, *Natural Language Engineering* **2**, pp.15-59.

Landwehr, N, Hall, M, Frank, E. (2003) Logistic Model Trees. ECML 2003: 241-252

Platt, J. (1999) Fast training of support vector machines using sequential minimal optimization. In Advances in kernel methods: support vector learning. B. Schölkopf, C. Burges, and A. Smola, eds. MIT Press, pp. 185-208.

Quirk, R.; Greenbaum, S.; Leech, G. & Svartvik, J. A (1992) A grammar of contemporary English. Longman Group Ltd. Harlow, UK.

Santos, Diana. (1998) The relevance of vagueness for translation: Examples from English to Portuguese, *TradTerm **5.1***, Revista do centro interdepartamental de tradução e terminologia, FFLCH - Universidade de São Paulo, 1998, pp.71-98

Witten, I.H. & Frank, E. (2000) Data Mining: Practical machine learning tools with Java implementations. San Francisco: Morgan Kaufmann.

Yarowsky, David. (1993) One sense per collocation, Proceedings of Human Language Technology, ARPA. San Francisco, Calif.; Morgan Kaufmann, pp. 266-71.