Is linguistic information relevant for the classification of legal texts?

Teresa Gonçalves
Departamento de Informática
Universidade de Évora
7000-671 Évora, Portugal
tcg@di.uevora.pt

Paulo Quaresma
Departamento de Informática
Universidade de Évora
7000-671 Évora, Portugal
pq@di.uevora.pt

ABSTRACT

Text classification is an important task in the legal domain. In fact, most of the legal information is stored as text in a quite unstructured format and it is important to be able to automatically classify these texts into a set of concepts.

Support vector machines (SVM) have shown to be good classifiers for text bases [Joachims, 2002]. In this paper, SVM are applied to the classification of legal texts – the Portuguese Attorney General's Office Decisions – and the relevance of linguistic information in this domain, namely lemmatisation and part-of-speech tags, is evaluated.

The obtained results showed that linguistic information can be successfully used to improve the classification results and, simultaneously, to decrease the number of features needed by the learning algorithms.

1. INTRODUCTION

The learning problem can be described as finding a general rule that explains data given a sample of limited size. In supervised learning, we have a sample of input-output pairs (the *training sample*) and the task is to find a deterministic function that maps any input to an output such that the disagreement with future input-output observations is minimised. If the output space has no structure except whether two elements are equal or not, we have a *classification* task. Each element of the output space is called a *class*. The supervised classification task of natural language texts is known as *text classification*.

Text classification is also an important task in the legal domain. In fact, most of the legal information is stored as text in a quite unstructured format and it is important to be able to automatically classify these texts into a set of concepts.

Research interest in this field has been growing in the last years. Several learning algorithms were applied such as decision trees [Tong & Appelbaum, 1994], linear discriminant analysis and logistic regression [Schütze *et al.* , 1995], naïve Bayes algorithm [Mladenić & Grobelnik, 1999] and Support Vector Machines (SVM)[Joachims, 2002].

In the legal domain, much work has been done in data and text classification tasks. For instance, [Wilkins & Pillaipakkamnatt, 1997] used decision trees to extract rules to estimate the number of days until the final disposition of cases; [Zeleznikow & Stranieri, 1995] developed rule based and neural networks legal systems; [Borges *et al.* , 2003] used neural networks to model legal classifiers; [Thompson, 2001] proposes a framework for the automatic categorisation of case law; [Schweighofer & Merkl, 1999, Schweighofer *et al.* , 2001] describes the use of self-organising maps (SOM), to obtain clusters of legal documents in an information retrieval environment and explores the problem of text classification in the context of the European law; [Liu *et al.* , 2003] describes classification and clustering approaches to case-based criminal summaries and [Brüninghaus & Ashley, 2003, Bruninghaus & Ashley, 1997] describe also related work using linear classifiers for documents. However, in these research work the relevance of linguistic information in legal classification tasks is not studied in detail.

In our work, the application of support vector machines to the problem of legal text classification is described and an evaluation of the relevance of linguistic information is performed.

In previous work, we evaluated the SVM performance compared with other Machine Learning algorithms [Gonçalves & Quaresma, 2003] and in [Silva *et al.* , 2004], linguistic information was applied to the preprocessing phase of text mining tasks. In this one, we apply a linear SVM to a legal text base, the Portuguese Attorney General's Office dataset – PAGOD [Quaresma & Rodrigues, 2003], performing a thorough study on several preprocessing techniques such as feature reduction, feature subset selection and term weighting.

The relevance of using some linguistic information, such as lemmatisation and part-of-speech tags (POS), to reduce the number of features is studied in detail and showed that it is possible to strongly reduce the number of features and the complexity of the legal text classification problem without loosing accuracy.

We also considered another experiment trying to evaluate the impact of the imbalance nature of this dataset: we made a balancing experiment by over-sampling and concluded that it generates better performance, especially for those categories with worse results when using the original number of positive and negative examples.

In Section 2, a brief description of the Support Vector Machines theory is presented, while in Section 3 the PAGOD dataset is characterised. Section 4 describes our experimental setup and Section 5 our experiments. Conclusions and future work are pointed out in Section 6.

2. SUPPORT VECTOR MACHINES

Support Vector Machines, a learning algorithm introduced by Vapnik and coworkers [Cortes & Vapnik, 1995], was motivated by theoretical results from the statistical learning theory. It joins a kernel technique with the structural risk minimisation framework.

Kernel techniques comprise two parts: a module that performs a mapping into a suitable feature space and a learning algorithm designed to discover linear patterns in that space. The *kernel function*, that implicitly performs the mapping, depends on the specific data type and domain knowledge of the particular data source. The *learning algorithm* is general purpose and robust. It's also efficient, since the amount of computational resources required is polynomial with the size and number of data items, even when the dimension of the embedding space grows exponentially [Shawe-Taylor & Cristianini, 2004]. Four key aspects of the approach can be highlighted as follows:

- Data items are embedded into a vector space called the feature space.
- Linear relations are discovered among the images of the data items in the feature space.
- The algorithm is implemented in a way that the coordinates of the embedded points are not needed; only their pairwise inner products.
- The pairwise inner products can be computed efficiently directly from the original data using the kernel function.

These stages are illustrated in Figure 2.

The *structural risk minimisation* (SRM) framework creates a model with a minimised VC dimension. This developed theory [Vapnik, 1998] shows that when the VC dimension

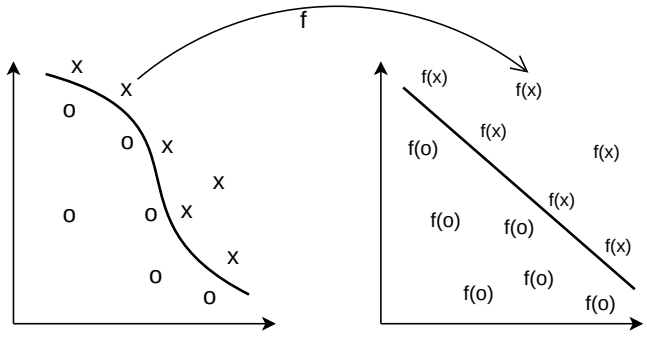

Figure 1: Kernel function: The nonlinear pattern of the data is transformed into a linear feature space.

of a model is low, the expected probability of error is low as well, which means good performance on unseen data (good generalisation).

SVM can also be derived in the framework of the regularisation theory instead of the SRM one. The idea of regularisation, introduced by Tickonov and Arsenin [Tikhonov & Arsenin, 1977] for solving inverse problems, is a technique to restrict the (commonly) large original space of solutions into compact subsets.

3. DATASET DESCRIPTION

The working dataset (PAGOD – Portuguese Attorney General's Office Decisions), has 8151 legal documents and represents the decisions of the Portuguese Attorney General's Office since 1940. It is written in the European Portuguese language, and delivers 96 MBytes of characters. All documents were manually classified by juridical experts into a set of categories belonging to a taxonomy of legal concepts with around 6000 terms.

Each PAGOD document is classified into multiple categories so, we have a multi-label classification task. Normally, this task is solved by splitting it into a set of binary classification tasks and considering each one independently.

A preliminary evaluation showed that, from all potential categories only about 3000 terms were used and from all 8151 documents, only 6388 contained at least one word on all experiments. For these documents, we found 77723 distinct words, and averages of 1592 words and 362 distinct words per document.

Table 1 presents the top ten categories (the most used ones) and the number of documents that belongs to each one.

4. EXPERIMENTAL SETUP

category	# docs
pensão por serviços excepcionais / excepcional services pension	906
deficiente das forças armadas / army injured	678
prisioneiro de guerra / war prisoner	401
estado da Índia / India state	395
militar / military	388
louvor / praise	366
funcionário público / public officer	365
aposentação / retirement	342
competência / authority	336
exemplar conduta moral e cívica / exemplary moral and civic behavior	289

Table 1: PAGOD's top ten categories: label and number of documents.

This section presents the choices made in our study: how did we represent a document, the kind of procedure we used to reduce/construct features, the process for obtaining the part-of-speech tags and how we measured learners' performance.

The linear SVM was run using the WEKA [Witten & Frank, 1999] software package from the Waikato University from New Zealand, with default parameters performing a 10-fold cross-validation procedure.

To represent each document we chose the bag-of-words approach, a *vector space model* (VSM) representation: each document is represented by the words it contains, with their order and punctuation being ignored. From the bag-of-words we removed all words that contained digits.

To measure learner's performance we analysed precision, recall and the F_1 measures [Salton & McGill, 1983] of the positive class. These measures are obtained from contingency table of the classification (prediction *vs.* manual classification). For each performance measure we calculated the micro- and macro-averaging values of the top ten categories.

Precision is the number of correctly classified documents divided by the number of documents classified into the class.

Recall is given by the number of correctly classified documents divided by the number of documents belonging to the class.

F_1 is the weighted harmonic mean of precision and recall and belongs to a class of functions used in information retrieval, the F_β-*measure*. F_β can be written as follows

$$F_\beta(h) = \frac{(1+\beta^2)prec(h)rec(h)}{\beta^2 prec(h) + rec(h)}$$

Macro-averaging corresponds to the standard way of computing an average: the performance is computed separately for each category and the average is the arithmetic mean over the ten categories.

Micro-averaging does not average the resulting performance measure, but instead averages the contingency tables of the various categories. For each cell of the table, the arithmetic mean is computed and the performance is computed from this averaged contingency table.

All significance tests were done regarding a 95% confidence level.

5. EXPERIMENTS

This section presents all SVM experiments made. First we used some typical information retrieval preprocessing techniques. Then, using the best setup, we used part-of-speech tags as a feature selection procedure. For each of these two classes of experiments we also considered balancing the dataset by over-sampling.

5.1 IR preprocessing experiments

We considered three classes of preprocessing experiments: feature reduction/construction, feature subset selection and term weighting. For each of class, we considered several different values. This subsection describes them.

5.1.1 Feature Reduction/Construction

On trying to reduce/construct features we used some linguistic information: we applied a Portuguese stop-list (the set of non-relevant words such as articles, pronouns, adverbs and prepositions), and POLARIS (a lexical database) to generate the lemma for each Portuguese word. We made three different experiments:

- rdt_1: consider all words of the original documents (except, as already mentioned, the ones that contained digits)
- rdt_2: consider all words except the ones that belong to the stop-list
- rdt_3: consider all words (but the ones that belong to the stop-list) transformed onto its lemma

5.1.2 Feature Subset Selection

For the feature subset selection we used a filtering approach, keeping the features that receive higher scores according to different functions:

- scr_1: *term frequency*. The score is the number of times the feature appears in the dataset; only the words occurring more frequently are retained;

- scr_2: *mutual information*. It evaluates the worth of an attribute by measuring the mutual information with respect to the class. Mutual Information, $I(C;A)$, is an Information Theory measure [Cover & Thomas, 1991] that ranks the information received to decrease the uncertainty. The uncertainty is quantified through the Entropy, $H(X)$.

- scr_3: *gain ratio* – $GR(A,C)$. The worth is the gain ratio with respect to the class. Mutual Information is biased through attributes with many possible values. Gain ratio tries to oppose this fact by normalising mutual information by the feature's entropy.

Mutual information and *gain ratio* are defined in terms of the probability function $p(x)$ where C is the class and A is the feature. $H(C|A)$ is the class entropy when we know the feature's value. These quantities are defined by the following expressions:

$$H(X) = -\sum_x p(x)\log_2 p(x)$$

$$\begin{aligned} I(C;A) &= H(C) - H(C|A) \\ &= -\sum_c p(c)\log_2 p(c) + \\ &\quad +\sum_a p(a)\sum_c p(c|a)\log_2 p(c|a) \end{aligned}$$

$$GR(A,C) = \frac{I(C;A)}{H(A)}$$

For each filtering function, we tried different threshold values. This threshold, for the term frequency scoring function, is the number of times the feature appears in all documents. We performed experiences for thr_1, thr_{50}, thr_{100}, thr_{200}, thr_{400}, thr_{800}, thr_{1200} and thr_{1600}, where thr_n means that all words appearing less than n are eliminated.

For the mutual information and gain ratio scoring functions, thr_n means that we select the same number of features that would be selected by the term frequency scoring function with the thr_n threshold value.

Table 2 shows the number of features obtained for each combination of feature reduction/construction and feature subset selection experiments. The last two rows show, per document, the average number of all (avg_{all}) and distinct ($avg_{distinct}$) features.

Term Weighting

Term weighting techniques usually consist of three components: the document component, the collection component and the normalisation component. In the final feature vector x, the value x_i for word w_i is computed by multiplying the three components.

	rdt_1	rdt_2	rdt_3
thr_1	68886	68688	42423
thr_{50}	9479	9305	5983
thr_{100}	6439	6275	4413
thr_{200}	4238	4085	3147
thr_{400}	2578	2440	2115
thr_{800}	1515	1390	1332
thr_{1200}	1076	962	956
thr_{1600}	831	724	743
avg_{all}	1339	802	768
$avg_{distinct}$	306	277	215

Table 2: Number of features for each threshold value and feature construction/reduction combination.

Document component captures statistics about a particular term in a particular document. Its basic measure is the *term frequency* – $TF(w_i, d_j)$. It is defined as the number of times word w_i occurs in document d_j.

The collection component assigns lower weights to terms that occur in almost every document of a collection. Its basic statistic is the *document frequency* – $DF(w_i)$, *i.e.* the number of documents in which w_i occurs at least once.

The normalisation component adjusts weights so that small and large documents can be compared on the same scale.

We made experiments for the following combination of components:

- wgt_1: *binary* representation. Each word occurring in the document has weight 1; all others have weight 0. The resulting vector is normalised to unit length.

- wgt_2: *raw term frequencies.* – $TF(w_i, d_j)$: is the number of times word w_i occurs in document w_j.

- wgt_3: *normalised term frequencies.* It uses $TF(w_i, d_j)$ normalised to unit length.

- wgt_4: *TFIDF representation.* Is $TF(w_i, d_j)$ multiplied by $log(N/DF(w_i))$ where N is the total number of documents and $DF(w_i)$ is the number of documents in which w_i occurs. The quantity is normalised to unit length.

These IR preprocessing experiments can be represented graphically in a n-dimensional space. First we have a three dimension space with one axis for feature reduction/construction, feature subset selection and term weighting. In each axis there are three or more possible values that represents different experiments. The feature subset selection axis is then "sub-divided" in another two: the scoring function and the threshold value. Figure 5.1.2 shows one of the possible experiments.

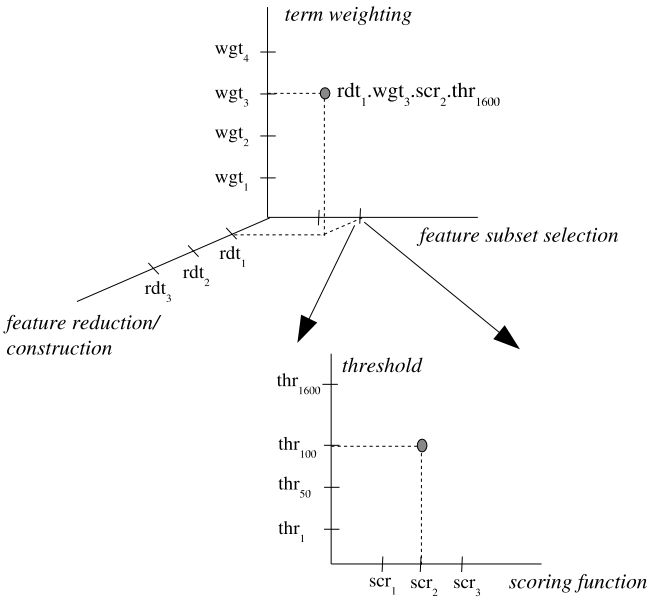

Figure 2: Graphical representation of the IR experiments.

5.1.3 Results

Here, we only show the F_1 micro- and macro-averaging mean values for all feature subset selection threshold values (thr_1, thr_{50}, ..., thr_{1600}). Tables 3 and 4 show the results.

		wgt_1	wgt_2	wgt_3	wgt_4
scr_1	rdt_1	0.743	0.666	0.733	0.723
	rdt_2	0.749	0.665	0.751	0.748
	rdt_3	0.746	0.642	**0.754**	0.749
scr_2	rdt_1	0.750	0.697	0.740	0.731
	rdt_2	**0.756**	0.695	0.752	0.745
	rdt_3	0.749	0.673	0.750	0.740
scr_3	rdt_1	0.740	0.672	0.731	0.714
	rdt_2	0.740	0.671	0.735	0.719
	rdt_3	0.734	0.642	0.731	0.712

Table 3: Micro-averaging mean F_1 value for each scoring function, feature reduction and term weighting combination.

The rdt_1, scr_3, wgt_2 and wgt_4 experiments presented the worst values. Other combinations of experiments showed very similar, better results.

We used the $rdt_2.scr_2.wgt_3.thr_{400}$ setting for the remaining experiments: rdt_2 is easier and faster to obtain, src_3 and wgt_3 produce more stable results and thr_{400} was always on the set of the best values and presents a good trade-off between the performance and the time consumed to generate the model.

5.2 Part-of-speech tag experiments

		wgt_1	wgt_2	wgt_3	wgt_4
scr_1	rdt_1	0.645	0.479	0.628	0.612
	rdt_2	0.651	0.480	0.650	0.638
	rdt_3	0.648	0.457	**0.654**	0.639
scr_2	rdt_1	0.645	0.513	0.621	0.608
	rdt_2	**0.647**	0.511	0.633	0.618
	rdt_3	0.634	0.475	0.624	0.603
scr_3	rdt_1	0.599	0.478	0.585	0.567
	rdt_2	0.593	0.479	0.584	0.565
	rdt_3	0.586	0.439	0.577	0.554

Table 4: Macro-averaging mean F_1 value for each scoring function, feature reduction and term weighting combination.

To obtain each word's POS tag we used a parser for the Portuguese language. This parser – PALAVRAS[1] [Bick, 2000] was developed in the context of the VISL (Visual Interactive Syntax Learning) project in the Institute of Language and Communication of the University of Southern Denmark.

The POS tagger incorporated in PALAVRAS is reported to have more than 95% accuracy for texts written in Portuguese. Possible tags are:

noun (*n*)
proper noun (*prop*)
adjective (*adj*)
verb (*v*)
article (*det*)
pronoun (*pron*)
adverb (*adv*)
numeral (*num*)
preposition (*prp*)
interjection (*in*)
conjunction (*conj*)

From all tags, we just considered *n*, *prop*, *adj* and *v*.

Note that Portuguese is a rich morphological language: while nouns and adjectives have 4 forms (two *genres* – male and female and two *numbers* – singular and plural), a regular verb has 66 different forms (two *numbers*, three *persons* – 1st, 2nd and 3rd and five *modes* – indicative, conjunctive, conditional, imperative and infinitive, each with different number of *tenses* ranging from 1 to 5).

The parser's output is the syntactic analysis of each phrase and the POS tag associated with each word. For example, the morphological tagging of the phrase "O Manuel ofereceu um livro ao seu pai./Manuel gave a book to his father." is:

```
o [o] <artd> <dem> DET M S
Manuel [Manuel] PROP M S
ofereceu [oferecer] V PS 3S IND VFIN
um [um] <quant> <arti> DET M S
livro [livro] N M S
a [a] <prp>
o [o] <artd> <dem> DET M S
```

[1] http://www.visl.sdu.dk/

```
seu [seu] <pron-det> <poss> M S
pai [pai] N M S
```

To have a base value of comparison, we also present the values for the best setting of the IR experiments (here named *base*). Besides using rdt_2 and thr_{400}, we also examined the generated models using thr_1 and rdt_3 setup from the previous subsection.

Table 5 shows the number of features obtained for each POS tag experiment for original words (rdt_2) and their lemmas (rdt_3). Table 6 shows the averages per document (of all and distinct features) for each threshold value.

	thr_1		thr_{400}	
	rdt_2	rdt_3	rdt_2	rdt_3
nn	24597	20388	1168	1026
vrb	27689	8899	601	542
$nn + vrb$	49838	27031	1752	1533
$nn + adj$	33431	25720	1535	1349
$nn + prop$	35273	30123	1329	1165
$nn + adj + prop$	43229	34877	1679	1473
$nn + vrb + adj$	58052	31981	2122	1855
$nn + vrb + prop$	59742	36287	1917	1669
base	68688	42423	2440	2115

Table 5: Total number of features for each POS experiment.

	all		distinct	
	rdt_2	rdt_3	rdt_2	rdt_3
nn	437	424	126	110
vrb	212	184	120	76
$nn + vrb$	638	598	237	179
$nn + adj$	559	540	175	148
$nn + prop$	547	514	149	130
$nn + adj + prop$	668	630	196	166
$nn + vrb + adj$	759	714	285	216
$nn + vrb + prop$	747	688	260	198
base	1592	912	362	255

Table 6: Average of words per document (all and distinct) for each POS experiment.

5.2.1 Results

For each experiment, we, once again, analysed *precision*, *recall* and F_1 measures and calculated the micro- and macro-averaging of the top ten categories. Tables 7 and 8 shows F_1 micro- and macro-averaging values for each experiment.

Considering macro-averaging F_1 values, the worst significant experiments were vrb (with words or lemmas, for both threshold values) and nn (lemmas with thr_1). The micro-averaging F_1 worst significant values were obtained for the same experiments and also for $nn + vrb$ and $nn + adj$ (with lemmas with thr_1) and nn (words with thr_1).

The best values were obtained $nn + prop$, $nn + adj + prop$, $nn + adj + vrb$, $nn + prop + vrb$ and the *base* experiments. If we take into account the number of features we can reduce

	rdt_2		rdt_3	
	thr_1	thr_{400}	thr_1	thr_{400}
nn	0.787	0.814	0.781	0.812
vrb	0.770	0.786	0.762	0.783
$nn + vrb$	0.799	0.809	0.787	0.813
$nn + adj$	0.793	0.818	0.790	0.817
$nn + prop$	0.794	0.821	0.791	0.817
$nn + adj + prop$	0.801	0.817	0.797	0.822
$nn + vrb + adj$	0.801	0.809	0.797	0.818
$nn + vrb + prop$	0.803	0.809	0.798	0.818
base	–	–	0.800	0.811

Table 7: F_1 micro-averaging values for each POS, feature construction and threshold value combination.

	rdt_2		rdt_3	
	thr_1	thr_{400}	thr_1	thr_{400}
nn	0.727	0.728	0.719	0.721
vrb	0.649	0.642	0.639	0.645
$nn + vrb$	0.737	0.742	0.732	0.747
$nn + adj$	0.735	0.745	0.733	0.739
$nn + prop$	0.735	0.746	0.734	0.738
$nn + adj + prop$	0.743	0.749	0.742	0.754
$nn + vrb + adj$	0.738	0.747	0.743	0.756
$nn + vrb + prop$	0.740	0.745	0.743	0.754
base	–	–	0.743	0.753

Table 8: F_1 macro-averaging values for each POS, feature construction and threshold value combination.

from 2115 (*base* with rdt_3 and thr_{400}) to 1165 ($nn + prp$ with rdt_3 and thr_{400}) features without loosing accuracy.

5.3 Balancing dataset

This experiment was made to evaluate the impact of the imbalance nature of the datasets, since, as referred for example in [Japkowicz, 2000], this can be a source of bad results. In fact, in the PAGOD dataset there are much more negative than positive examples. For example, the ratio for the most used category is about seven to one while for tenth most used is about 22 to one (see Table 1).

We balanced the PAGOD dataset by over-sampling the positive examples of each learner (category) in order to have an equal number of positive and negative ones. We made experiments for each winning setting of the previous subsections – IR ($rdt_2.scr_2.wgt_3.thr_{400}$) and POS tag ($nn + adj + prop.rdt_3.scr_2.wgt_3.thr_{400}$) experiments.

5.3.1 Results

Table 9 presents the original values for the IR experiment while Table 10 presents the corresponding over-sampling results.

As can be seen, we achieve much better results by over-

sampling the datasets, especially on those categories with very bad values for the original setting (like the *funcionário público* and *aposentação* categories).

	precision	*recall*	F_1
aposentação	0.655	0.607	0.630
competência	0.408	0.322	0.360
deficiente... armadas	0.984	0.972	0.978
estado da Índia	0.992	0.982	0.987
exemplar... cıivica	0.940	0.869	0.903
funcionário público	0.477	0.203	0.285
louvor	0.813	0.806	0.809
militar	0.520	0.470	0.494
pensão... excepcionais	0.974	0.962	0.968
prisioneiro... guerra	0.993	0.993	0.993

Table 9: IR's original results for the top ten categories.

	precision	*recall*	F_1
aposentação	0.850	1.000	0.919
competência	0.793	0.883	0.836
deficiente... armadas	0.997	0.997	0.997
estado da Índia	0.998	1.000	0.999
exemplar... cıivica	0.993	0.996	0.994
funcionário público	0.915	0.894	0.904
louvor	0.976	0.955	0.965
militar	0.965	0.955	0.959
pensão... excepcionais	0.993	0.997	0.995
prisioneiro... guerra	0.998	0.999	0.998

Table 10: IR's over-sampling results for the top ten categories.

This conclusion can also be thrown from the POS experiment. Tables 11 and 12 show, respectively, the original and over-sampling results.

	precision	*recall*	F_1
aposentação	0.657	0.589	0.621
competência	0.422	0.209	0.279
deficiente... armadas	0.990	0.978	0.984
estado da Índia	0.992	0.985	0.989
exemplar... cıivica	0.957	0.851	0.901
funcionário público	0.420	0.258	0.320
louvor	0.850	0.885	0.867
militar	0.615	0.512	0.559
pensão... excepcionais	0.976	0.975	0.975
prisioneiro... guerra	0.998	0.995	0.996

Table 11: POS's original results for the top ten categories.

6. CONCLUSIONS AND FUTURE WORK

In this work the application of support vector machines to the classification of Portuguese legal documents was described and evaluated. Several information retrieval techniques were used to reduce and select the document features. Moreover, the use of part-of-speech information was also studied and the impact of balancing the dataset (positive and negative examples).

	precision	*recall*	F_1
aposentação	0.861	1.000	0.925
competência	0.799	0.891	0.842
deficiente... armadas	0.999	0.999	0.999
estado da Índia	0.999	1.000	0.999
exemplar... cıivica	0.993	0.999	0.996
funcionário público	0.921	0.912	0.916
louvor	0.976	0.965	0.970
militar	0.971	0.959	0.965
pensão... excepcionais	0.997	0.998	0.997
prisioneiro... guerra	0.999	1.000	0.999

Table 12: POS's over-sampling results for the top ten categories.

It was possible to identify a good combination of these factors having a F_1 micro-averaging for the top ten categories of 0.822: POS $= nn + adj + prop.$ $rdt_3.scr_2.wgt_3.thr_{400}$. This means that it is a good approach to use only words tagged as nouns, adjectives or proper nouns, lemmatised, ordered with mutual information and weighted with normalised term frequency.

Using the referred combination, it was possible to reduce the number of features from a total 68886 distinct words to 1473 and to increase the F_1 micro-averaging for the top 10 categories from 0.740 to 0.822.

In conclusion, it is possible to state that linguistic information, such as, lemmatisation and part-of speech tags improve SVM classifiers and strongly reduce the computational complexity of the task.

As future work, and in order confirm these results, we intend to make the same experiments with legal datasets written in other languages and with non-legal datasets. It will be important to evaluate if these results are binded to the Portuguese language and/or the legal domain.

On the other hand, and aiming to develop better classifiers, we intend to address the document representation problem by trying more powerful representations than the bag-of-words, allowing us to use word order and syntactical and/or semantical information in the representation of documents. To achieve this goal we plan to use other kind of kernel such as the string kernel (see, for example, [Shawe-Taylor & Cristianini, 2004]).

7. REFERENCES

Bick, E. 2000. *The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press.

Borges, F., Borges, R., & Bourcier, D. 2003. Artificial Neural Networks and Legal Categorization. *Pages 11–20*

of: 16th International Conference on Legal Knowledge Based Systems. IOS Press.

Bruninghaus, S., & Ashley, K. 1997. Finding factors: learning to classify case opinions under abstract fact categories. *Pages 123–131 of: 6th International Conference on Artificial Intelligence and Law*. ACM.

Brüninghaus, Stefanie, & Ashley, Kevin D. 2003. Predicting Outcomes of Case-Based Legal Arguments. *Pages 233–242 of: ICAIL*.

Cortes, & Vapnik. 1995. Support-vector networks. *Machine Learning*, **20**(3).

Cover, Thomas M., & Thomas, Joy A. 1991. *Elements of Information Theory*. Wiley Series in Telecomunication. New York: John Wiley and Sons, Inc.

Gonçalves, T., & Quaresma, P. 2003. A preliminary approach to the multilabel classification problem of Portuguese juridical documents. *Pages 435–444 of:* Moura-Pires, F., & Abreu, S. (eds), *11th Portuguese Conference on Artificial Intelligence, EPIA 2003*. LNAI 2902. Évora, Portugal: Springer-Verlag.

Japkowicz, N. 2000. The Class Imbalance Problem: Significance and Strategies. *Pages 111–117 of: Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, vol. 1.

Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines*. Kluwer academic Publishers.

Liu, Chao-Lin, Chang, Cheng-Tsung, & Ho, Jim-How. 2003. Classification and Clustering for Case-Based Criminal Summary Judgement. *Pages 252–261 of: ICAIL*.

Mladenić, D., & Grobelnik, M. 1999. Feature selection for unbalanced class distribution and naïve Bayes. *Pages 258–267 of: Proceedings of ICML-99, 16th International Conference on Machine Learning*.

Quaresma, P., & Rodrigues, I. 2003. PGR: Portuguese Attorney General's Office Decisions on the Web. *Pages 51–61 of:* Bartenstein, Geske, Hannebauer, & Yoshie (eds), *Web-Knowledge Management and Decision Support*. LNCS/LNAI 2543. Springer-Verlag.

Salton, G., & McGill, M. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.

Schütze, H., Hull, D., & Pedersen, J. 1995. A comparison of classifiers and document representations for the routing problem. *Pages 229–237 of: Proceedings of SIGIR-95, 18th International Conference on Research and Developement in Information Retrieval*.

Schweighofer, E., & Merkl, D. 1999. A Learning Technique for Legal Document Analysis. *Pages 156–163 of: 7th International Conference on Artificial Intelligence and Law*. ACM.

Schweighofer, Erich, Rauber, Andreas, & Dittenbach, Michael. 2001. Automatic text representation, classification and labeling in European law. *Pages 78–87 of: ICAIL*.

Shawe-Taylor, J., & Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Silva, C.F., Vieira, R., Osorio, F.S., & Quaresma, P. 2004 (August). Mining Linguistically Interpreted Texts. *In: 5th International Workshop on Linguistically Interpreted Corpora*.

Thompson, Paul. 2001. Automatic categorization of case law. *Pages 70–77 of: ICAIL*.

Tikhonov, V.M., & Arsenin, V.Y. 1977. *Solution of Ill-Posed Problems*.

Tong, R., & Appelbaum, L.A. 1994. Machine learning for knowledge-based document routing. *In:* Harman (ed), *Proceedings of 2nd Text Retrieval Conference*.

Vapnik, V. 1998. *Statistical learning theory*. NY: Wiley.

Wilkins, D., & Pillaipakkamnatt, K. 1997. The effectiveness of machine laerning techniques for predicting time to case disposition. *Pages 39–46 of: 6th International Conference on Artificial Intelligence and Law*. ACM.

Witten, I., & Frank, E. 1999. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann.

Zeleznikow, J., & Stranieri, A. 1995. The Split-up system: Integrating neural networks and rule based reasoning in the legal domain. *Pages 195–194 of: 5th International Conference on Artificial Intelligence and Law*. ACM.