PROLEGIS: Intelligent Search in Legislation Databases

Hugo Miguel de Jesus Lopes, Instituto Superior Técnico

***Abstract*—Portuguese legislation, similarly to other countries, is not published in an organized way, being it by topics or concepts. Instead, it is organized by a numbering system which follows the publication order. For a common citizen or even researchers, searching for information about a subject or a specific problem is an hard and complex task.**

The categorization of legal texts, besides requiring specialized labour, is a task which would need a great amount of time due to the quantity of published documents. Because of that, the Text Categorization (TC) task often relies on Machine Learning (ML) algorithms.

The focus of this work will be on the supervised domain, nevertheless, an unsupervised clustering analysis is also explored. Multiple supervised classification algorithms are experimented, using a set of pre-classified documents, in order to comparatively evaluate their classification performances. Support Vector Machines, K-Nearest Neighbours, Multinomial Naive Bayes and Decision-Trees were used individually and, in order to seek to enhance the results, in conjunction with various techniques for pre-processing features. Latent Semantic Indexing, feature selection with different metrics and stemming were analysed.

***Index Terms*—Portuguese Legislation, Legal Texts, Text Categorization, Supervised Classification**

I. Introduction

Researchers and citizens are posed with a difficult problem when searching for topics of interest in legislation. In which documents lies the pertinent information? The analysis and narrowing of relevant information in large databases is not a trivial task, and can be quite time consuming. Moreover, political and legislative documents could be of interest for researchers since in some way they reflect the social, economical and political situation of a country in a certain time frame. It is possible to observe trends of the political wing, reflections of economical influences and responses for new problematics of the world.

Nowadays most of the legislation is published freely by the governments in publicly available on-line databases, offering easily the data for researchers to work with and citizens to explore. The problem lies in the absence of systematic organization of the legal information, creating obstacles to its knowledge and access to both citizens and legal experts.

There is already some effort in creating a set of categories which represents the main topics of political agendas. The *Policy Agendas Project*[1] aims to create a coding scheme with 19 major topics and 225 specific subtopics. The main purpose is to use a generalized and organized code system to categorize, over time, the legislative content. A side project, *Comparative Agendas Project*[2], extends the initial project to other participating countries, being Portugal one of those, aiming for comparability among nations. A research unit group from CIES (Centro de Investigação e Estudos de Sociologia) belonging to the ISCTE - University Institute of Lisbon started this task and provided a set of 1810 of already classified documents using the Policy Agendas Project code scheme. The goal of our work is to use the provided set of classified documents and demonstrate that there is a possibility of mitigating the necessity of extensive human labelling labour. By automating this process, it could allow researchers to dedicate their investigation time into other complex questions and aid citizens in the search of information in the legal turmoil. Moreover, it will be shown that legal text classification is practicable with relative high accuracies.

II. Text Classification

Formally, given a set of documents $D = \{d_1, .., d_n\}$ and a predefined set of categories $C = \{c_1, .., c_m\}$, TC consists in the task of approximating an unknown category assignment function $F(d, c) : D \times C \rightarrow \{0, 1\}$. The classifier is the approximating function $\hat{F}$, where the goal is to build one as close as the F function [1].

With unsupervised techniques it is not possible to guess which categories will emerge, and because of that, these techniques are, most of the time, only used to discover possible groups of classes. Since the main purpose is to categorize in a topic based fashion, the focus of this work will be on the supervised domain, while also an unsupervised evaluation, using a clustering algorithm, will be performed to evaluate if the documents organize in a fashion similar to the given categories.

A. Feature Identification and Document Representation

Most TC tasks usually only consider individual words as features, ignoring punctuation and often numbers as well. In the legal domain punctuation does not seem relevant and numbers are often related with accounting which are highly variable and carry few class related information.

The most popular and straightforward approach among TC for feature representation is the Bag-of-Words (BOW) model. Its effectiveness has been widely demonstrated in many studies [2]. This method consists in directly representing the document as a set of found words, where each word maps one feature, as an n-dimensional vector $d_m = \{w_1, ..., w_n\}$. Due to the simplistic nature of this representation there is

[1]Policy Agendas Project: http://www.policyagendas.org/

[2]Comparative Agendas Project: http://www.comparativeagendas.info/

no consideration about the order of the words, meaning the structure of the document is lost.

It is common to filter out words which are too frequent in the language since they tend to be statistically irrelevant. For instance "the", "a" or "but" can be found almost in every document and therefore they carry no value in distinguishing the documents from each other. These words are called *stop words*. To tackle the problem of potential irrelevant features, the weighing schemes become an important factor.

B. Feature Weighting

One of the most popular feature weighting schemes in TC is the Term-Frequency/Inverse Document Frequency (TFIDF). It is composed by the Term-Frequency (TF), which is simply the number of word occurrences, and the Inverse Document frequency (IDF), which is a measure of whether the term is common or rare across all documents [3]. Formally, the weight, w_{ij}, of the term t_i in the document d_j is:

$$w_{ij} = tf_{ij} \times \log(idf_i) = tf_{ij} \log \frac{|D|}{N_i} \tag{1}$$

where tf_{ij} is the term i frequency in document j, $|D|$ is the total number of documents in the corpus and N_i is number of documents which contain the term i.

TFIDF aims to favour terms which allow to better distinguish certain individual documents from the corpus, while penalizing the ones which are too frequent in the whole collection, like the previously referred *stop words*.

C. Scaling

Feature scaling or normalization is the process of standardizing the range of the features. The vector based representation of a document collection can be viewed as a set of vectors in a vector space, in which there is one axis for each term. If two documents discuss the same topic it is probable that both contain the same terms, however if one of the documents is bigger than the other, then the relative frequencies might be identical in the two documents, but the absolute term frequencies of the bigger document will be larger.

Euclidean normalization converts the document vectors into unit length vectors. For each document d_j its scaled version d'_j is computed as:

$$d'_j = \frac{d_j}{||d_j||^2} = \frac{d_j}{\sqrt{\sum_{i=1}^{|T|} d_{ij}^2}} \tag{2}$$

where $|T|$ and the total number of terms in the document, d_{ij} and the term i in the document j.

D. Dimensionality Reduction

Due to the complexity of human languages, as the corpus grows, it is often expected to have a bigger dimension of the feature space. This happens specially when working in domain specific areas, where there is quite specific vocabulary. High dimensionality can increase drastically the computation time of some algorithms or even make them unusable. Moreover, many of the features are irrelevant (like the *stop words*), carrying few discriminating information, affecting negatively the accuracy of our classifier. Reducing dimensionality is a way of reducing *over-fitting* of our classifier, which happens when a classifier is tuned to the characteristics of the training data rather than the real characteristics of the categories. Furthermore, most of the times, dimensionality reduction can be applied without affecting significantly the performance of the classification [4].

There are two main approaches for producing a reduced set of features: *Feature Selection* and *Feature Transformation*.

1) Feature Selection: can increase the classification performance by removing features which might mislead the classifier into erroneous classifications. If for instance the word "violencia" (violence) only shows in documents were we are talking about "direitos" (rights) the classifier might not classify a document with the word "violencia" in crime related laws. This is over-fitting introduced by the dataset, which can be potentially avoided by a careful selection of features.

Several feature selection methods have been studied in [4], [5]. Here we present some of those.

- Gini Index (GI)

The Gini Index is a way of measuring the discrimination level of a feature. Being C the set of categories, the Gini Index $G(t)$ for a given word t is defined by:

$$G(t) = \sum_{i=1}^{|C|} P(c_i|t)^2 \tag{3}$$

where $P(c_i|t)$ is the probability of the class c_i knowing that the term t belongs to it.

- Information Gain (IG)

IG measures the quantity of information obtained for category prediction by knowing the presence or absence of a term in a document. Being C the set of categories, the information gain $I(t)$ for a given word t is:

$$I(t) = -\sum_{i=1}^{|C|} P(c_i) \log_2 P(c_i) + P(t) \sum_{i=1}^{|C|} P(c_i|t) \log_2 P(c_i|t) + P(\bar{t}) \sum_{i=1}^{|C|} P(c_i|\bar{t}) \log_2 P(c_i|\bar{t}) \tag{4}$$

where $P(t)$ is the probability of the documents having the word t, $P(\bar{t})$ is the probability of the documents not having the word t, $P(c_i)$ is the probability of class c_i, $|C|$ is the number of categories.

- Mutual Information (MI)

Measures the variables mutual dependence, which in this case will be the mutual dependence between features and classes. The $MI(t, c_i)$ between the word t and class c_i is given by:

$$MI(t, c_i) = \log \frac{P(t|c_i)}{P(t)} \approx \log \frac{N_{it} \times |D|}{(N_{it} + c_{i\bar{t}})(N_{it} + \bar{N}_{it})} \tag{5}$$

where N_{it} is the number of documents where t and c_i co-occur, $N_{i\bar{t}}$ is the number of documents where c_i occurs without

t, $\bar{N}_{it}$ is the number of documents where t occurs without c_i, $|D|$ is the number of documents in the corpus.

There are two main popular final scores, by averaging or by choosing the maximum value:

$$MI_{avg}(t) = \sum_{i=1}^{|C|} P(c_i) MI(t, c_i) \tag{6}$$

$$MI_{max}(t) = \max_{i=1}^{|C|} MI(t, c_i) \tag{7}$$

- χ^2 Statistic

The χ^2 statistic is used in statistics to test the independence of two events and the goodness of a fit. In TC is other way of measuring the dependence between a word and a class. It can be defined in terms of probabilities by:

$$\chi^2(t, c_i) = \frac{|D| \times (N_{it}\bar{N}_{i\bar{t}} - N_{i\bar{t}}\bar{N}_{it})}{(N_{it} + N_{i\bar{t}})(\bar{N}_{it} + \bar{N}_{i\bar{t}})(N_{it} + \bar{N}_{it})(N_{i\bar{t}} + \bar{N}_{i\bar{t}})} \tag{8}$$

where N_{it} is the number of times t and c_i co-occur, $N_{i\bar{t}}$ is the number of times c_i occurs without t, $\bar{N}_{it}$ is the number of times t occurs without c_i, $\bar{N}_{i\bar{t}}$ is the number of times neither c_i nor t occurs and $|D|$ is the number of documents in the corpus.

The weighted average can be combined to express the score respective to each term:

$$\chi^2_{avg}(t) = \sum_{i=1}^{|C|} P(c_i) \chi^2(t, c_i) \tag{9}$$

2) Feature Transformation: can be seen as a way of combining features instead of removing them, resulting in a new set of reduced dimensionality with different feature values.

- Stemming

Stemming is a technique which is often applied in the pre-processing phase. It consists in transforming inflected or derived words into their stem, base or root form. This results in singular, plural and different tenses to be consolidated into a single stem, reducing considerably the dimension of the data. For instance, the words "stemming", "stemmed" and "stems" would all map to the word base "stem". There are multiple approaches, such as trained statistic models, or by looking at pre-defined tables, but the most common approach is the *suffix stripping* method [6], [7].

- Latent Semantic Indexing (LSI)

When dealing with word features without further analysis, each word is, as previously stated, assumed to be independent from each other. LSI tries to capture relations in the terms which are implied (latent semantics) [8]. The occurrence of some patterns of words gives a strong clue to the occurrence of others. Moreover, it addresses the problem of the use of synonymous, near-synonymous and polysemous in texts, which consists in multiple features (multiple dimensions) which could correspond only to a single feature in terms of meaning. Since the independence assumption is not true, this technique finds dependence among the features of the corpus. It then maps this newly found information into new set of vectors.

In the processing phase, the documents will represented in a document term matrix X, where the lines correspond to the documents in the corpus, the columns to words, and each cell to the weight of the word in the corresponding document. LSI consists in applying Singular-Value Decomposition (SVD) to the document term matrix X. SVD results in the decomposition of X into three matrices, where their product reconstructs the original document term matrix:

$$X = T_0 S_0 D_0^T \longrightarrow \hat{X} = TSD^T \approx X \tag{10}$$

where T_0 and D_0 are orthonormal columns and S_0 is a diagonal matrix of singular values. T_0 and D_0 are called left and right singular vectors respectively and S_0 is the diagonal matrix of singular values. The singular values of S_0 are ordered in decreasing magnitude.

Since the singular values of S_0 are ordered, dimensionality reduction is performed by keeping the first *k* largest values and setting the smaller ones to zero, resulting in the a matrix $\hat{X}$ which is approximately similar to X but of rank *k*.

III. Algorithms

A. Unsupervised Algorithm

In unsupervised methods the data does not need to be labelled, allowing to be applied to any text data. The goal of unsupervised clustering is to cluster the documents without external intervention and additional knowledge besides its features. The resulting clusters should be in such way that the documents within a cluster are more similar than the documents belonging to different clusters.

1) K-Means Clustering: is widely applied in many different areas of text mining and information retrieval [9].

The k-means algorithm aims to split the data into *k* clusters where each observation belongs to the cluster with the nearest mean, therefore its name. Each cluster mean is the mean of the documents assigned to that cluster [10]. Usually *n* documents are randomly selected to form the initial mean centroids, meaning that the formed clusters vary with the initial selected documents.

One of the drawbacks of the K-means, since it is not guaranteed that the algorithm finds a global minimum, is that it becomes significantly sensitive to the randomly selected centres. To reduce this effect usually multiple trials are computed. Also, the chosen number of clusters can affect drastically the final distribution of the documents in the clusters.

B. Supervised Algorithms

Supervised algorithms require a set of labelled documents which are used to train and allow to provide a mapping function to the predefined class labels, based on the document features. Here are presented some of the most popular classifiers in the TC research.

1) Multinomial Naive Bayes (MNB): assumes a mixture model for the generation of the documents where each class is a component of the mixture. Each mixture component provides the probability of sampling a particular term of its respective class. The term *naive* refers to the assumption where all features are independent of each other, given their respective class.

In the MNB the word frequency (or weight) information is taken into account. It follows the BOW representation, where the word position in the text is assumed to not affect its probability. With a given a set of training documents it is possible to estimate the probability $P(c_k|d_i)$ of a certain document d_i belonging to class c_i

While the independence assumption of this model is not true in most of the real cases, it often performs fairly well and, due to its simplicity, gained a lot of popularity in early TC research [11]–[13].

2) Support Vector Machines (SVM): are a kind of *linear classifier* which attempts to separate the document's different classes [14], [15]. The SVM consists in a convex quadratic optimization problem which can be solved using Lagrangian methods [16].

Known to be tailored for high dimensional problems, SVM only has to take into account a small part of the (training) data to construct the optimal separating hyperplane. It was suggested that SVM was suitable to work with text classification since in corpora datasets it often has to deal with high-dimensional data. Document text data is often quite sparse, since documents in big collections usually have few entries in the document term matrix which are not zero, allowing to separate linearly the classes.

SVM are one of the most popular classifiers in TC achieving really high performances compared to other classifiers, as shown in [17], [18].

3) Decision Trees (DT): is a hierarchical decomposition of the training data in binary trees. The splitting of the data is performed at the nodes by a rule and, the same rule is applied to each child node recursively [19].

The goal is to decide which words of the training documents, makes the best splitting based on their classes. The best splitter (word feature) is the one that decreases the diversity, i.e. increases homogeneity of the training samples by the greatest amount [20]. The problem corresponds to finding the best words t in the whole vocabulary T to associate to the splits in the decision tree.

The biggest problem with decision trees is that they tend to over-fit to the training sets. However in some studies, DT shows competitive results compared to MNB [21], [22].

4) K-Nearest Neighbours (KNN): is one of the simplest classification algorithms that has shown to perform well comparative with other popular classifiers [22].

The algorithm is based on the similarity shared between the documents and their labels. In the training phase, the algorithm simply stores the multidimensional feature vectors with their respective labels, without having actual "learning" or internal parameters estimation. For the test phase it considers, according to a distance metric, the *k* nearest training examples which are closest. The major class among the *k* nearest samples is chosen as the class for the test document [23].

The advantage of the KNN is that the cost of the learning process is zero, remaining all the computation cost to the prediction. Due to this, when the dimensionality of the corpus is large, it can be computationally expensive. Even though KNN is a simple classifier, it can achieve performances comparable to the SVM for some corpus [18].

IV. Evaluation Metrics

In TC, since we are in a single-label perspective, the main objective of the classifier is to fit all the test documents in their correct categories. However, finding only how many were correctly classified does not tell us how the system behaved when it incorrectly guessed a classification. Different metrics usually cover different aspects of the behaviour of our classifier.

In a supervised TC the classification experiment it is necessary to have two sets of documents. One is the train set which is used for training the classifier, and the other is the test set, where its results are used to evaluate the performance of the classier. In unsupervised clustering evaluating the performance is not as simple as comparing labels outputs. Any evaluation metric should not take absolute values of the cluster labels into account but rather if the separations made by the method define the separations of the documents in a fashion similar to the truth known classes or, if the members of some cluster are more similar between each other.

A. Classification Scores for Unsupervised methods

There are two kinds of clustering evaluation: internal and external. Since usually there are no labelled documents in unsupervised methods, most of the evaluation to infer the quality of the algorithms have to rely on characteristics of the data that was clustered. This is called internal evaluation. When there is knowledge about the true labels of the documents, even though it is not used in the process, the quality of the clustering classification can be evaluated with it. The use of previously known information about the data is called external evaluation.

1) Silhouette score: In general, internal clustering evaluation combines two types of measurements. It can be related to the closeness of the cluster elements, compactness, or how distinct two clusters are by measuring their distances relative to each other, separability. The Silhouette score is an evaluation metric which aims to unify this two concepts [24].

For a given cluster X_j the silhouette measure of each document i in it is defined as:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{11}$$

where $a(i)$ is the average distance between the *ith* sample and all the samples in cluster k_j and $b(i)$ is the minimum distance between sample i in cluster k_j and all the samples in the nearest cluster. The final silhouette score is simply the mean of all document silhouettes.

2) V-measure: V-measure score is an entropy based metric which joins two criteria for cluster evaluation [25]. Homogeneity evaluates how pure is a cluster relative to the labels of the documents which belong to it. A cluster method satisfies homogeneity if each of the clusters contains only documents of the same class. The other measure, completeness, evaluates how documents of the same class are aggregated together. A clustering method is said to satisfy completeness if all the documents of the same class belong to the same cluster.

V-measure is the weighted harmonic mean of the completeness and homogeneity:

$$V_\beta = 2\frac{homogeneity \cdot completeness}{homogeneity + completeness} \tag{12}$$

B. Classification Scores for Supervised methods

For evaluating the classifiers the popular F1-score was chosen. It combines two metrics: *recall* and *precision*. Recall is the number of correctly labelled documents divided by the number of elements that actually belong to the class. Precision is the number of correctly labelled documents divided by the number of elements that were labelled as belonging to the class [26].

The F1-score which respect to a class c_i is defined by:

$$F1 = 2\frac{precision \cdot recall}{precision + recall} = \frac{2TP}{2TP + FN + FP} \tag{13}$$

where True Positives (TP) are the documents correctly labelled into a class c_i. False Negatives (FN) the documents which belong to a class c_i but were not labelled as such. False Positives (FP) are the documents incorrectly labelled into c_i which belong to another class.

1) Averaging the scores: How we compute the global scores affects the perception of effectiveness of our system. There are two different averaging approaches, micro-averaging and macro-averaging.

Micro-averaging is computing the score over all individual decisions of every class. It can be expressed formally:

$$precision_m = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \tag{14}$$

$$recall_m = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)} \tag{15}$$

Macro-averaging is computing the score for each class individually, then averaging equally every class:

$$precision_M = \frac{\sum_{i=1}^{|C|} precision_i}{|C|} \tag{16}$$

$$recall_M = \frac{\sum_{i=1}^{|C|} recall_i}{|C|} \tag{17}$$

The F1 averaged scores are then computed with equation 13 where F1 micro-averaged score is denoted by $F1_m$ and the F1 macro-averaged score by $F1_M$.

In micro-averaging every document counts the same in the final average, while in macro-averaging each class counts the same. When classes are very unbalanced, meaning the number of documents in each class is considerably different, the class with the most documents might dominate the classification process. The classes which have fewer training documents will provide less information, and consequently, less generalization power to the classifier. Because of that, results in such categories tend to be lower and, when macro-averaging, this effect will be more pronounced than when micro-averaging.

V. Experimental Setup

A. Datasets

TABLE I
CORPORA STATISTICS

Corpus	Total Words	Unique Words	Mean words per doc
DRE	2 104 896	36 870	1 162.9
DRE-META	40 019	4 851	22.1
DRE-GAP	830 799	19 320	1 042.4

1) Diário da República (DRE) Corpus: The text collection consists in 1810 classified documents and their texts were collected using a *web scrapper*. The classified documents used here only cover the First Series ("*Série I*") of Portuguese legislation, which is the most important one. Using the whole integral text of the documents a dataset was created which will be called simply as *DRE* dataset. The corpus word statistics and categories can be found in tables I and II respectively.

TABLE II
MAJOR CATEGORIES AND DOCUMENT FREQUENCIES FOR THE DRE AND DRE-META CORPUS

Categories	NºDocs
Macroeconomia	105
Direitos e Liberdades civis e das minorias	14
Saúde	41
Agricultura, Pecuária e Pescas	26
Trabalho e emprego	16
Educação	140
Ambiente	12
Energia	10
Transportes	146
Justiça e Direito	112
Políticas Sociais	30
Habitação	87
Sector financeiro, indústria e comércio	86
Defesa	49
Ciência, tecnologia e comunicações	33
Comércio externo	11
Política externa	50
Governo e administração pública	831
Recursos naturais e Património	11

2) DRE-META Corpus: An advantage of collecting the data using a scrapper is that the website, besides the integral text of the document, also allows to collect a small summary and respective meta-data, such as the ministry that emitted it and date, which can be used in classification as well. For experimentation a dataset with the source (ministry that emitted it) plus the summary was created with a label set equal to the *DRE* corpus, which can be found in table II. The word statistics can also be seen in table I.

3) DRE-GAP Corpus: The most populated class in the *DRE* corpus represents approximately 46% of the whole *DRE* corpus. Since it represents a big chunk of the data itself,

a subset with only the "Governo e administação pública" category was created. The minor categories of the *DRE-GAP* will be used as the major categories of the documents.

The goal of the *DRE-GAP* is to try to evaluate the possibility of a TC system being able to categorize the documents into such specific minor categories. The final set of classes and respective document frequencies can be found in table III. The corpus word statistics can be seen in table I.

TABLE III
MAJOR CATEGORIES AND DOCUMENT FREQUENCIES OF THE DRE-GAP CORPUS

Categories	NºDocs
Eficiência governativa	339
Administração Pública	197
Nomeações e exonerações	35
Condecorações, reconhecimento público	119
Compras Públicas	46
Gestão do setor público	21
Administração fiscal	25
Transição e consolidação da democracia	15

B. Preprocessing

The preprocessing phase consists in treating the text collection in a way that it can be properly interpreted by the classifier. Following the BOW model, the process of transforming a sequence of text into individual features can be simply performed by separating words when there is white spaces in between with a regular expression. Moreover, punctuation was discarded, since in most of the cases it only adds noise to feature discrimination, and, for legislation categorization, it is believed that it does not introduce relevant information.

All the words are converted to lower case so they all map into the same feature. Features of length of one are also removed. This features are often resultant from abbreviations, acronyms or paragraph enumeration, which are common in law articles. No numeric characters were considered as features. Stop word removal is applied as well, removing the most frequent words in the Portuguese language.

VI. EXPERIMENTAL RESULTS

In supervised classification a k-fold cross validation methodology was used to evaluate the results of the different supervised experiments. The amount of labelled documents in the *DRE* corpus in some categories is reduced and for a cross folding validation is necessary at least *k* labelled documents in a category. Since the categories are quite unbalanced, the stratified scheme was chosen, where each splits maintain the proportions of the original classes.

The unsupervised clustering is repeated 10 times and the averaged scores are presented.

A. Unsupervised Clustering with fixed number of clusters

A clustering algorithm was applied to the corpora to verify how the categories perform under unsupervised circumstances. In first instance, the main purpose is to evaluate, in a simple form, the possibility of using a completely unsupervised algorithm to categorize the corpus. Moreover it is a form of analysing how documents in the categories are similar to each others

In this experiment a fixed number of clusters, equal to the number of categories respective to each corpus, was used in the process of clustering with K-means. The idea is to observe if the clusters resemble to the original categories.

In table IV the results for the multiple scores discussed in section IV-A are presented. The silhouette score is close to zero, meaning that the average distance between the samples within the clusters are close to the distances of the respective nearest clusters, which is a strong indication that the clusters overlap. The low homogeneity score suggests that the formed clusters contain multiple different categories and the low completeness score that the categories are not concentrated on the same clusters, thus leading to a low V-measure as well.

TABLE IV
CLUSTERING WITH K-MEANS

Corpus	Sillouette	Homogeneity	Completeness	V-Measure
DRE	0.031	0.319	0.283	0.288
DRE-META	0.041	0.294	0.255	0.259
DRE-GAP	0.026	0.432	0.450	0.406

B. Supervised Classification Experiments

1) Baseline and Weighting Impact Evaluation: For each component of the TFIDF weighting scheme a test with different classifiers was performed. Here the Term-Frequency (TF), TFIDF without the log and with, which is denoted by $TFIDF_{log}$ are evaluated. To each different weight scheme the Euclidean norm is applied and evaluated as well.

As can be seen in table V each component of the TFIDF weighting scheme, contributes in many cases for an increase of the classification performance. The F1 score for each category weighted with TFIDF can be seen in Fig. 1 and table V.

Clearly the best classifier is the SVM, as seen in most of the literature comparisons, where it performs the best for both $F1_m$ and $F1_M$ scores, outperforming the other classifiers in both datasets. MNB is the second which provides best results, followed by KNN and lastly DT.

2) Stemming evaluation: In this experiment, feature transformation with stemming using suffix removal was applied to the datasets. The experiments were performed with the normalized TFIDF. To facilitate comparison the baseline results are repeated and summarized in table VI.

In terms of feature reduction, in the DRE corpus, it resulted in a feature space of 16694 unique features, which represents a reduction of almost 55%. In the case of DRE-META, the stemming process resulted in a feature space of 3064 unique features, a reduction of 37% of the dimensionality. For the DRE-GAP there is a reduction of 57% with 8219 unique features. These reductions in the feature space are quite significant with only marginal impact in the performance of the classification.

TABLE V
IMPACT OF FEATURE WEIGHTING IN THE CORPORA

Corpus	Weighting	SVM		MNB		KNN		DT	
		$F1_m$	$F1_M$	$F1_m$	$F1_M$	$F1_m$	$F1_M$	$F1_m$	$F1_M$
DRE	TF	0.800	0.553	0.777	0.543	0.675	0.424	0.595	0.394
	Norm(TF)	0.838	0.649	0.820	0.578	0.759	0.497	0.613	0.430
	TFIDF	0.830	0.597	0.784	0.576	0.685	0.458	0.601	0.399
	Norm(TFIDF)	0.853	**0.660**	**0.823**	**0.609**	0.791	0.541	**0.621**	**0.436**
	$TFIDF_{log}$	0.851	0.606	0.787	0.574	0.647	0.421	0.607	0.407
	Norm($TFIDF_{log}$)	**0.860**	0.647	0.806	0.566	**0.799**	**0.548**	0.613	0.419
DRE-META	TF	0.825	0.611	0.778	0.573	0.697	0.420	0.703	0.517
	Norm(TF)	0.813	0.620	**0.809**	0.584	0.754	0.491	0.662	0.462
	TFIDF	0.836	0.636	0.772	0.577	0.701	0.455	0.694	0.506
	Norm(TFIDF)	0.818	0.634	0.807	**0.605**	0.782	0.534	0.673	0.477
	$TFIDF_{log}$	**0.839**	**0.638**	0.768	0.570	0.696	0.454	**0.707**	**0.522**
	Norm($TFIDF_{log}$)	0.830	0.632	0.808	0.600	**0.788**	**0.534**	0.675	0.483
DRE-GAP	TF	0.858	0.758	0.863	0.775	0.776	0.655	0.746	0.647
	Norm(TF)	0.875	0.791	**0.881**	0.774	0.853	**0.745**	**0.769**	**0.683**
	TFIDF	0.862	0.761	0.861	0.754	0.697	0.539	0.763	0.660
	Norm(TFIDF)	**0.890**	**0.809**	0.880	0.802	0.861	0.739	0.766	0.680
	$TFIDF_{log}$	0.875	0.796	0.871	**0.808**	0.640	0.501	0.752	0.644
	Norm($TFIDF_{log}$)	0.887	0.809	0.878	0.782	**0.862**	0.721	0.760	0.671

TABLE VI
IMPACT OF FEATURE STEMMING IN THE CORPORA

Corpus	LinearSVM		MNB		KNN		DT	
	$F1_m$	$F1_M$	$F1_m$	$F1_M$	$F1_m$	$F1_M$	$F1_m$	$F1_M$
DRE	0.853	0.660	0.823	0.609	0.791	0.541	0.621	0.436
DRE Stemmed	0.821	0.585	0.770	0.540	0.705	0.437	0.600	0.400
DRE-META	0.818	0.634	0.807	0.605	0.782	0.534	0.673	0.477
DRE-META Stemmed	0.814	0.617	0.786	0.558	0.715	0.468	0.700	0.516
DRE-GAP	0.890	0.809	0.880	0.802	0.861	0.739	0.766	0.680
DRE-GAP Stemmed	0.868	0.784	0.856	0.760	0.798	0.672	0.731	0.655

3) Evaluation of Feature Selection Methods: Since selecting features may improve results and avoid over-fitting an experiment with the methods presented in section II-D1 was performed. Concerning the classifiers, feature selection metrics were performed only with SVM due to presentation space and computation times. SVM so far shows to be the best classifier and the goal is to maximize the classification scores. The classification performances for each feature selection metric can be seen in Fig. 2.

Overall none of the methods provided significantly better results than the baseline classifications. However, χ^2 feature selection method showed to be a powerful method, allowing performances similar to the full set of features. With only 20% of the most significant feature set for the DRE, DRE-META and DRE-GAP was necessary to achieve similar classification scores of the original set.

C. Feature Transformation with LSI

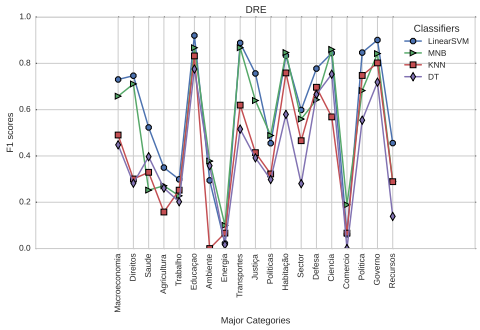
More than just a dimensional reduction method, LSI aims to capture the underlying latent semantics in the texts. To evaluate its effect in the classification, multiple sizes, or in other words, singular vectors, were selected in this experiment. In the literature usually 200 to 500 principal vectors are selected. Since the purpose is to capture the most information with the lowest amount of dimensions, a set of 100 to 2000 dimensions were used, as can be seen in table VII. Since MNB does not support negative features, which the SVD algorithm produces when computing the principal vectors, it was not included in this experiment.

The results can be found in table VII. It is interesting to note that with a very reduced feature set, the classifiers are able to achieve performances close to the original non-transformed features. In overall, F_m score has no significant improvements, while on corpus with fewer features, such as the DRE-META, shown to be marginally worse. It is possible to conclude that for corpus with low dimensional spaces, LSI does not provide great advantages since the classification performance requires bigger dimensions to capture information similar to the original. However, for datasets as the DRE and DRE-GAP, the LSI with only 2000 dimensions, a feature dimensional reduction close to 95%, provides similar results as the original dataset.

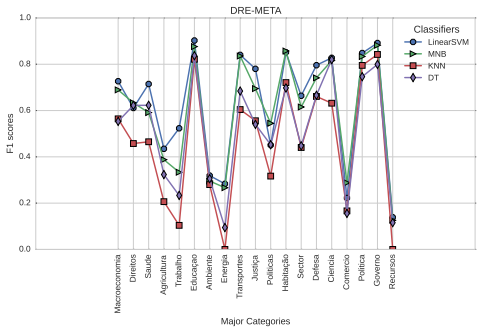
VII. CONCLUSION

As document representation, BOW methodology was applied and each word was treated as a feature. This feature representation showed to be able retain the corpus characteristics well enough to provide reasonable classification scores.

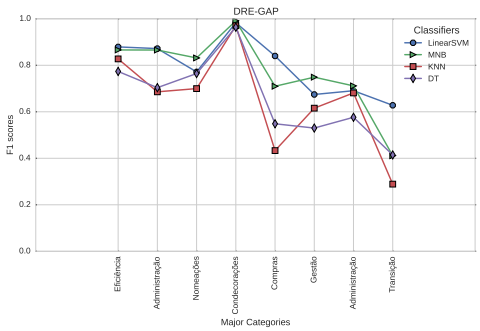
Common techniques of dimensionality reduction were also tested. Stemming algorithm did not show to provide improvements in the classification accuracy of the documents, but it was able to reduce considerably the feature set, while maintaining scores close to the baseline. LSI and feature selection with the χ^2 metric show marginally better results in some cases. However, no significant impact was observed in terms of classification results. Both techniques shown a considerable power as dimensionality reduction tools allowing to achieve similar classification scores with less than 10% of



(a) DRE

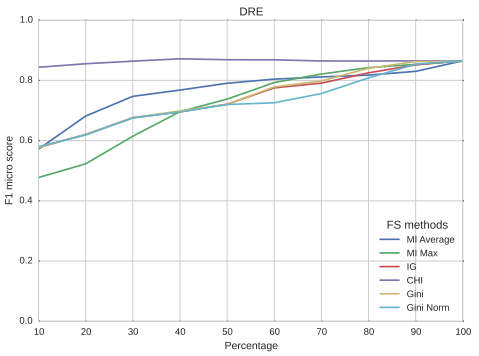


(b) DRE-META

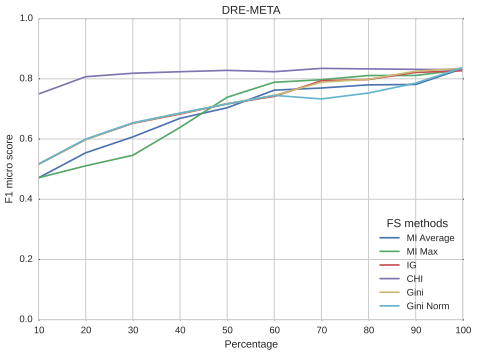


(c) DRE-GAP

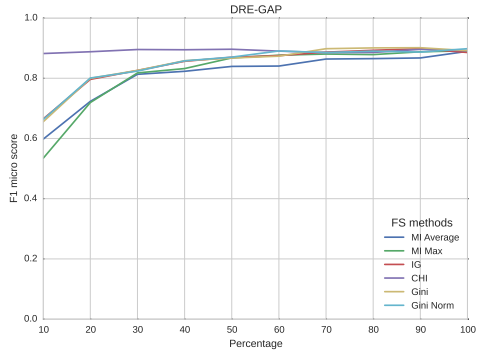
Fig. 1. F1 scores for each category of each corpus and classifier with features weighted with $||TFIDF||$.



(a) DRE



(b) DRE-META



(c) DRE-GAP

Fig. 2. F1 micro scores of each feature selection method of each corpus with SVM.

the original feature set. It should be noted that LSI is also an unsupervised algorithm for dimensionality reduction while χ^2 is a supervised method.

The unsupervised clustering experiments in the corpora did not show clusters which allowed to classify the multiple major categories of the documents. Using this unsupervised algorithm it is not possible to classify the legislation documents with reasonable performances.

Supervised algorithms demonstrated, as expected, more success in correctly classifying the documents. The best classifier is, without doubt, the SVM for the supervised classification. It is interesting to note that SVM is a linear classifier, being clear that the text data is indeed sparse enough and therefore easily separated with only linear algorithms. SVM performance follows closely the tendency observed in the TC literature. MNB, which makes a gross assumption of independence among features, also shows good results being a classifier that is simple and easy to implement.

It was expected lower classification scores in the DRE-META corpus since it has less unique features compared to the DRE corpus. Even so, it was possible to achieve classifications scores in the DRE-META corpus which were close to the DRE corpus. It was demonstrated that it is not a matter of quantity of (unique) features, but instead, how much information they provide to the category, and consequently, to the training of the classifier. With the DRE-GAP corpus it shown that, if provided adequate number of classified document samples for training, the classifiers are able to correctly classify documents which belong to categories within similar topics. This allows the possibility of a finer and detailed classification of the documents.

It has to be noted, that the coded DRE corpus is significantly skewed. One of the biggest difficulties of this work was to

TABLE VII
FEATURE TRANSFORMATION WITH LSI WITH PRE-SELECTED DIMENSIONS FOR MULTIPLE CORPUS AND CLASSIFIERS.

Corpus	Classifiers	avg	Dimensions 100	500	1000	2000
DRE	LinearSVM	F_M	0.612	**0.668**	0.667	0.664
		F_m	0.813	0.845	0.860	**0.867**
	KNN	F_M	0.532	0.571	0.538	0.561
		F_m	0.790	0.819	0.795	0.802
	DT	F_M	0.395	0.360	0.325	0.305
		F_m	0.611	0.564	0.548	0.517
DRE-META	LinearSVM	F_M	0.537	0.616	**0.630**	0.628
		F_m	0.740	0.790	0.816	**0.831**
	KNN	F_M	0.498	0.534	0.529	0.533
		F_m	0.730	0.787	0.780	0.775
	DT	F_M	0.348	0.323	0.283	0.280
		F_m	0.539	0.530	0.499	0.495
DRE-GAP	LinearSVM	F_M	0.812	0.830	0.805	**0.839**
		F_m	0.879	0.882	0.885	**0.896**
	KNN	F_M	0.803	0.719	0.734	0.734
		F_m	0.880	0.860	0.866	0.867
	DT	F_M	0.694	0.677	0.643	0.654
		F_m	0.777	0.765	0.746	0.746

reach reasonable classification scores at the same level of those found in the literature. Often TC is performed with much more documents examples for training the classifier. Still, it was possible to achieve high classification accuracies in some of the well populated classes. The classes for which there are fewer pre-classified examples, the results do not achieve desirable classification scores.

Overall, the classification of legislation has proved to be possible with satisfying results. Some of the categories did not reached the same levels of success, however, it became clear that it is fundamental to have a good sample of pre-classified examples to produce acceptable classification results.

REFERENCES

[1] R. Feldman and J. Sanger, *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2006.

[2] A. Moschitti and R. Basili, "Complex linguistic features for text classification: A comprehensive study," in *Advances in Information Retrieval*. Springer, 2004, pp. 181–196.

[3] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," in *INFORMATION PROCESSING AND MANAGEMENT*, 1988, pp. 513–523.

[4] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Mach. Learn. Res.*, vol. 3, pp. 1289–1305, 2003.

[5] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997, pp. 412–420.

[6] M. F. Porter, "Readings in information retrieval," K. Sparck Jones and P. Willett, Eds. Morgan Kaufmann Publishers Inc., 1997, ch. An Algorithm for Suffix Stripping, pp. 313–316.

[7] V. M. Orengo and C. Huyck, "A stemming algorithm for the portuguese language," in *String Processing and Information Retrieval, 2001. SPIRE 2001. Proceedings.Eighth International Symposium on*, 2001, pp. 186–193.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[9] J. A. Hartigan, *Clustering Algorithms*, 99th ed. John Wiley & Sons, Inc., 1975.

[10] E. W. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," *Biometrics*, vol. 21, pp. 768–769, 1965.

[11] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Mach. Learn.*, pp. 103–130, 1997.

[12] N. Friedman, D. Geiger, M. Goldszmidt, G. Provan, P. Langley, and P. Smyth, "Bayesian network classifiers," in *Machine Learning*, 1997, pp. 131–163.

[13] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*. AAAI Press, 1998, pp. 41–48.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[15] T. Joachims, "Text categorization with suport vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*. Springer-Verlag, 1998, pp. 137–142.

[16] C. C. Aggarwal, *Data Mining*. Springer International Publishing, 2015.

[17] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *Proceedings of the Seventh International Conference on Information and Knowledge Management*. ACM, 1998, pp. 148–155.

[18] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. ACM, 1999, pp. 42–49.

[19] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[20] K. Aas and L. Eikvil, "Text categorisation: A survey," *Raport NR*, vol. 941, 1999.

[21] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization," in *In Third Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 81–93.

[22] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. ACM, 2006, pp. 161–168.

[23] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," *Michigan State Universiy*, vol. 2, 2006.

[24] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.

[25] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure." in *EMNLP-CoNLL*, vol. 7, 2007, pp. 410–420.

[26] D. D. Lewis, "Evaluating text categorization," in *In Proceedings of Speech and Natural Language Workshop*. Morgan Kaufmann, 1991, pp. 312–318.