Legal Documents Categorization by Compression

Antonio Mastropaolo
Dipartimento di Scienze Economiche e Politiche
Università di Aosta
Strada Cappuccini, 2
11100 - Aosta, Italy

Francesco Pallante
Dipartimento di Giurisprudenza
Università di Torino
Lungo Dora Siena, 100
10153 - Torino, Italy

Daniele P. Radicioni*
Dipartimento di Informatica
Università di Torino
Corso Svizzera 185,
10149 - Torino, Italy
radicion@di.unito.it

ABSTRACT

In this paper we investigate how to categorize text excerpts from Italian normative texts. Although text categorization is a problem of broader interest, we single out a specific issue. Namely, we are concerned with categorizing the set of subjects in which Italian Regions are allowed to produce norms: this is the so-called *residual legislative power* problem. It basically consists in making explicit a set of subjects that was originally defined only in a residual and negative fashion. The categorization of legal text fragments is acknowledged to be a difficult problem, featured by abstract concepts along with a variety of locutions used to denote them, by convoluted sentence structure, and by several other facets. In addition, in the present case subjects are often partially overlapped, and a training set of sufficient size (for the problem under consideration) does not exist: all these aspects make our task challenging. In this setting, classical feature-based approaches provide poor quality results, so we explored algorithms based on compression techniques. We tested three such techniques: we illustrate their main features and report the results of an experimentation where our implementation of such algorithms is compared with the output of standard machine learning algorithms. Far from having found a silver bullet, we show that compression-based techniques provide the best results for the problem at hand, and argue that these approaches can be effectively coupled with more informative and semantically grounded ones.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.5 [**Pattern Recognition**]: Clustering—*Similarity measures*; I.7 [**Document and Text Processing**]: Index generation

Keywords

Automatic Text Categorization, Compression Techniques

*Corresponding author.

ICAIL '13, June 10 - 14 2013, Rome, Italy

1. INTRODUCTION

The text categorization (TC) task is to classify a given data instance into a predefined set of categories: in particular, given a set of categories (subjects) and a collection of text documents, text categorization is the process of finding the correct subject for each document. TC techniques are applied in a plethora of diverse contexts, ranging from spam filtering, to Web pages categorization, automatic generation of metadata, detection of text genre, author detection, plagiarism detection and so forth. Text categorization has been of the utmost importance in the last decade, due to the growth of the volume of digital documents: documents (and elements therein) indexing and retrieval have become hot topics in machine learning, and in the larger AI community. This problem is particularly relevant in the legal field, where more and more sophisticated access and elaboration of digital information is today required by both law professionals and scholars.

Legal text retrieval and categorization are often based on *external* knowledge sources such as thesauri and classification schemes, thereby requiring accurate hand-crafted indexing of the documents and maintenance of the indexed documents. As a result, only a fraction of legal documents required by users is currently available for information retrieval purposes [23]. Conversely, in the realm of digital documents, user information needs are becoming more and more sophisticated and demanding, often determining requests for small document partitions or connections amongst them, instead of full documents. Unfortunately, identifying inter- and intra-documents links is frequently left beyond the scope of the work of human annotators, with the effect that only a portion of actual users queries can be fulfilled. This fact implies that from a 'practical' perspective, legal professionals who mostly use electronic documents cannot access the appropriate (parts of) documents. In addition, systematic investigations in the legal field are badly affected by the lack of automatic tools to classify legal documents and their finer grained sub-elements.

This work aims at bridging the gap: we compare different classification techniques to categorize heterogeneous size text fragments, be they whole documents or small excerpts, by starting from a reduced training set. Two elements of interest are mixed in our work: *i)* we are concerned with text categorization to examine in an automatic and systematic way the problem of *residual legislative power* (described below, in Section 2.1); *ii)* we show how algorithms based on compression techniques compare with standard approaches, also providing results in line with and above those reported

in literature on similar classification problems. Although not new, to the best of our knowledge this kind of approach has never been used before to classify legal documents.

The paper is organized as follows: we first illustrate the problem under consideration (Section 2), we then describe in full detail the proposed approach (Section 3), report and discuss the results of an experimentation (Section 4) and survey related works (Section 5). Conclusions will close the paper.

2. PROBLEM DESCRIPTION

2.1 Residual Legislative Power

The problem of *residual legislative power* (RLP henceforth) arises from the definition of regional legislative powers as described by the the Italian Constitution, amended in 2001 [25, 7]. The Article 117 of the Constitution provides, in relation to the State and Regions with ordinary statute, three different types of legislative power:

A. the exclusive jurisdiction of the State, in the matters listed in paragraph 2;

B. the concurrent jurisdiction between the State (concerned with fundamental principles) and Regions (concerned with detailed issues) in the matters identified in paragraph 3;

C. the residual powers of the Regions, including (in accord with paragraph 4) all areas other than those mentioned in A and B.

We note that RLP is a widespread problem: let us consider, in fact, that in slightly different terms, the problem is present in every law system where some kind of twofold center-periphery structure exists, e.g., USA, Canada, Australia, Germany, Belgium, Spain, etc.. We can draw a distinction between centralized and decentralized systems by considering whether the State or the devolved administrations exercise the RLP. In centralized systems the State exercises legislative power to a large extent. Conversely, in decentralized systems such as federal states, residual power is devolved to the periphery. In the Italian system the residual clause worked until 2001 in favor of the central State, and since 2001 it has worked in favor of the Regions.

From the perspective of legal hermeneutics, it is relevant to determine the sphere of competences of Regions, by compiling a list of the matters that are actually included in their residual power. This question has a practical impact, and is at the base of the broader theme of democratic citizenship practice, as witnessed by the EU Fundamental Rights and Citizenship Funding Programme,[1] which aims at promoting "information and civic education initiatives on the active participation of Union citizens in the democratic life of the Union and, in particular, participation in European Parliament and municipal elections".

Unfortunately, identifying the matters falling within the scope of the RLP[2] is difficult. In addition to matters whose exercise is unquestionably of regional competence, there are other ones whose attribution has been –and, definitely, *is*– still under debate. We therefore decided to analyze the object of the judgments of the Italian Constitutional Court that ruled on the grey zone claimed by both State and Regions. In more detail, we focussed on the analysis of the judgments of the Constitutional Court from 2002 to 2012, related to residual competence issues. The underlying idea is that analyzing this body of decisions (overall amounting to a hundred elements) and the laws which they refer to allows us to identify the essential characteristics of the residual legislative powers.

Once the set of matters falling within the regional competence limits is identified, it will be possible to define the matters in relation to the national and regional legislation, so as to provide a tool for the classification of the entire regional legislation. This will permit to investigate the directions taken by the regional legislator in those twelve years.[3] Also, in a more general perspective, this research will contribute to making the legislation, which is to date confused and in fact inaccessible, knowable and transparent.

2.2 Problem Formalization

Legal texts categorization is usually acknowledged to be a difficult problem, due to several reasons, such as the presence of abstract concepts, and the wide variety of expressions that can be used to convey the same abstract concepts [23]. Furthermore some distinguishing elements characterize the present case: we have to cope with a small set of training examples, featured by partially overlapped text excerpts. All these aspects make it difficult to directly employ most standard classification approaches and encoding schemes, such as the standard feature-based representation. While in the long term we intend to exploit knowledge based methods (such as thesauri [5]), for the present we are concerned with clearly defining classes and experimenting with available techniques. In particular, our work relies on a group of compression-based classification algorithms we found promising to approach the categorization problem, and that could be then used coupled with more semantically motivated classification approaches.

We implemented a system to automatically extract the object provision(s) from a decision, to query institutional sites[4] to retrieve the normative sources, and to extract the text excerpts that actually constitute the object provisions. We are presently concerned with recognizing the topic of the *object* of judgments decisions, and defer its full illustration to a future work. For the sake of self-containedness, we briefly recall the judgments formalization to introduce our work. For a detailed description of the encoding of Italian Constitutional Court judgments, please refer to [10].

A judgement contains at least one *decision*. The decision

[1] `http://ec.europa.eu/justice/grants/programmes/fundamental-citizenship/index_en.htm`

[2] A provisional list of the subjects has been provided by Law scholars also based on judgments of the Italian Constitutional Court; it includes the following subjects: Agriculture; Assistance and social services; Crafts; Education and training; Fishing; Health organization; Hunting; Incentives to businesses; Industry; Local public services; Local public transport; Mineral and thermal waters; Mountain communities; Police and local government; Promotion of activities and cultural heritage; Promotion of heritage sites; Public housing; Public works of local interest; Quarries and peat bogs; Regional legal and administrative organization; Regional public employment; Right to education; Tourism; Trade.

[3] E.g., one would be interested in answering questions as "in which matters fall most regional laws? is this distribution proper to a specific Region, or is it common? how can we globally compare the legislative activities of Regions?"

[4] `http://www.normattiva.it`

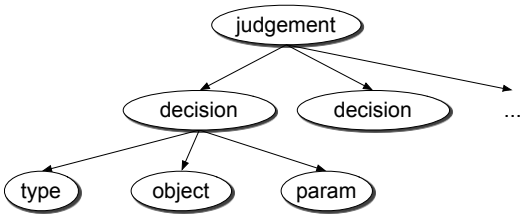

Figure 1: The main elements of the judgement.

is a complex object, having a type, an *object*, a *parameter*. A graphical account of the main elements of the judgement is provided in Figure 1. The *object* of a decision is the provision about which the Court is asked to state whether it is not compliant to the Italian Constitution. The object is composed by one or more object provisions. The *parameter* is the normative source upon which the pronouncement is based; in turn, a parameter is composed by one or more parameter provisions. Both the object provision and the parameter provision are a source, and sources are defined based on a *source type* (e.g., Law, Decree, the Italian Constitution, etc.), an optional *number*, an optional *year* (e.g., the Constitution or the Civil Code have no number and year associated), and an *article*, containing further information about paragraphs and finer grained partitions.

3. OBJECT PROVISIONS CATEGORIZATION

Most supervised learning approaches to TC extract features from text documents, and feature vectors corresponding to documents are then used to learn how to classify new documents. In order to reduce the dimensionality of such vectors, feature selection algorithms are commonly used to identify the most meaningful features, based on standard methods such as TF-IDF, mutual information, information gain, and other statistics collected from data [30]. All these approaches represent documents as bags of words, in that word order and contextual information are disregarded. Yet, usually to extract features from documents, some sort of further preprocessing (like stemming or lemmatization of words that passed the stop-words filtering steps) needs to be done. Also attempts at integrating semantic level descriptions and terminologies into the feature vector model have been carried out, in order to partially overcome such limitations [16, 8]. However, such approaches suffer from a known bottleneck in the acquisition of the needed information (e.g., ontological knowledge), and still do not provide competitive results in terms of accuracy, and in the trade-off between results and employed efforts.

We compare three approaches based on compression, whose theoretical tenets are rooted in information theory. This setting permits to formulate an intuitive and theoretically sound notion of *similarity* between documents, which is easy to implement and requires virtually no preprocessing of the input data. Compression-based classification techniques provide several attractive properties listed in a seminal work by Frank and colleagues [13]: the focus on the document as a whole, instead of filtering some features in the preprocessing stage; the uniform treatment of morphological variants of words; the possibility to cope with phrasal effects spanning over word boundaries; and the reduction of arbitrary decisions that are usually needed to implement any learning scheme. More generally, since compression techniques are mainly character-based, they allow to automatically capture non-word features, such as punctuation and word-stems, and features spanning more than one word.

3.1 Background in Kolmogorov complexity

We now introduce the notion of Kolmogorov complexity, following the notation provided by [20], then we survey some distance measures, and finally introduce the algorithms actually used in our experimentation.

Compression based techniques can be used in text categorization to train classifiers on labeled documents; the rationale behind this approach is that learning can be thought of as the problem of identifying (thus being able to generalize) regular traits in data. In turn, identifying some sort of regularity allows describing data with fewer resources, so that for a given set of hypotheses $\mathcal{H}$ and data set $D = \{C_1, C_2, \ldots, C_n, \}$, to learn regularities underlying the classes C in D we look for the hypothesis in $\mathcal{H}$ that compresses D most. Then, given a new document to be classified through $\mathcal{H}$, it will be assigned to the class C_i that permits to obtain the highest compression rate [14].

Be a text (in our present setting) coded as a string x over the binary alphabet. The set of such strings is denoted as $\{0,1\}^*$. The integer $K(x) = |x|$ is the length of the shortest binary program emitting x, also known as the Kolmogorov complexity of x. The *conditional* Kolmogorov complexity $K(x|y)$ of x relative to y is the length of a shortest program to compute x if y is provided as an auxiliary input to the computation. The notation $K(xy)$ denotes the length of a shortest binary program that outputs x concatenated to y.

In these terms, the distance between two strings x and y can be defined [20] as

$$d_k(x,y) = \frac{K(x|y) + K(y|x)}{K(xy)}. \tag{1}$$

Although $K(\cdot)$ is incomputable, there exist algorithms, called compressors, devised to approximate it. A compressor takes a file and rewrites it attempting to encode it as the shortest possible file. Given a data compression algorithm, we define $C(x)$ as the size of the compressed size of x and $C(x|y)$ as the compression achieved by first training the compression on y, and then compressing x.

The theoretical distance d_k in Eq. (1) can thus be approximated by the distance d_c based on a compression algorithm c:

$$d_c(x,y) = \frac{C(x|y) + C(y|x)}{C(xy)} \tag{2}$$

where $C(x|y)$ is the size of x, compressed by using the compression model built for y.

Another way to measure the distance between strings relies on the notion of *information distance* [9]. Information distance $E(x,y)$ is defined in terms of the shortest binary programs that with input x computes y, and that with input y computes x:

$$E(x,y) = \max\{K(x|y), K(y|x)\}$$

Its normalized version, the *normalized information distance* is defined as

$$\mathrm{NID}(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}} \tag{3}$$

By approximating the *NID* using a compressor C we obtain the normalized compression distance NCD: a compressor C approximates the information distance $E(x, y)$ by the compression distance $E_C(x, y)$ defined as

$$E_C(x, y) = C(xy) - \min\{C(x), C(y)\}.$$

The normalized version of $E_C(x, y)$ is called the *normalized compression distance* [9]:

$$\mathrm{NCD}(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \quad (4)$$

3.2 Classification procedures

We tested three different compression-based categorization strategies known in literature. The two former procedures are variants of the *minimum description length* approach: the Approximate Minimum Description Length [19, 18], and the Best-Compression Neighbor [3]. The latter procedure builds on the Normalized Compression Distance [9].

As a compression program we chose the open-source Gzip utility, that proved to be effective in text classification (e.g., in spam filtering [12]). Gzip implements a dictionary-based compressor, and is virtually ubiquitous in UNIX systems: it relies on the Lempel-Ziv (LZ77) compression algorithm [31]. It looks for duplicated strings in the input data: the second occurrence of a string is replaced by a pointer to the previous string, in the form of a pair (distance, length). Distances are limited to $32K$ bytes, and lengths are limited to 258 bytes. When a string does not occur anywhere in the previous $32K$ bytes, it is emitted as a sequence of literal bytes.[5] That is, according to the principles stated above, the chief idea of the Gzip algorithm is to encode more recurring sequences with few bytes and to use further bytes for seldom seen sequences.

We briefly report the description of the algorithms using the notation provided by [21]. All of these procedures are based on the following intuition. Analyzing two compressed documents both individually and concatenated, we can compute a measure of how similar they are: the greater the observed compression rate, the more similar the documents. That is, if two documents are very similar then the size of the compressed file containing both documents concatenated together will only slightly increase with respect to the compressed size of a single document. Vice versa, this does not hold when documents are significantly different.

*Approximate minimum description length (*AMDL*)*

Given a set of training documents taken from n categories, $C_1, C_2, \ldots, C_n$, all documents in the category C_i are filed in a single archive A_i. The compression program is then run on each A_i, yielding as output a compressed file $\mathcal{A}_i$ of length $|\mathcal{A}_i|$. Given a test file T, AMDL appends T to each A_i, producing A_iT. It then runs the compression program on each A_iT to produce a compressed file $\mathcal{A}_i\mathcal{T}$. Finally, it assigns T to the class C_i that minimizes the compressed size difference $v_i = |\mathcal{A}_i\mathcal{T}| - |\mathcal{A}_i|$.

*Best-compression neighbor (*BCN*)*

The BCN procedure is similar to AMDL, but instead of concatenating all the training documents in a class into a single input file, each training document D is kept in a separate file. The test document T is concatenated to each D, forming DT, and the difference between the size of the compressed versions of DT and D is computed as $v_{DT} = |\mathcal{DT}| - |\mathcal{D}|$. Then T is assigned to the class containing the document D that minimizes v_{DT}. This procedure is actually a kNN approach using v_{DT} as the distance measure.

*Normalized Compression Distance (*NCD*)*

The NCD is an approximation of the incomputable Normalized Information Distance. The test document T is concatenated to each D, forming DT; at each step the original documents T and D are compressed (to form $\mathcal{T}$ and $\mathcal{D}$, respectively), and their concatenation DT is compressed, as well (to form $\mathcal{DT}$). For each pair $\langle D, T\rangle$ the NCD metric is computed as:

$$\mathrm{NCD}_{(D,T)} = \frac{\mathcal{DT} - \min(\mathcal{D}, \mathcal{T})}{\max(\mathcal{D}, \mathcal{T})}.$$

Then T is assigned to the class containing the document D that minimizes $\mathrm{NCD}_{(D,T)}$. Similar to the BCN procedure, we implemented a *k-nearest-neighbor* algorithm with k set to 10: that is, the k nearest documents are selected, and the class is assigned based on a majority vote.

4. EXPERIMENTATION

The whole set of topics used for categorization is composed by 24 classes (see Section 2.1). One challenging property of the dataset is that, as it is inherent in the fact that classes are defined only in a *residual* manner, class definition is somehow elusive, and classes are not clearly separated. Let us consider, for example, that norms referred to the class 'industry' could be easily confused with 'incentives to businesses'. Similarly, the category 'Assistance and social services' has links with 'Regional legal and administrative organization', in that actions in the former field involve the creation of ad-hoc departments (e.g., in the case of drug prevention) that pertain the latter field. Our dataset is composed of a hundred object provisions resulting from the systematic analysis of judgements on the residual legislative power. This set of documents includes *all* the objects mentioned by these judgements. Due to the reduced number of such documents, we pruned classes for which less than 5 documents were present, and used the remaining 70 documents as our dataset. Such documents are arranged in 7 classes: Agriculture, Assistance and social services, Trade, Public housing, Education and training, Regional legal and administrative organization, and Tourism. Overall, the 70 files amount to 628,177 bytes. The collected provisions are highly variable in length, ranging from an article paragraph composed of few words (e.g., the smallest document size is in the dataset is 231 bytes) to an entire law, whose size is 45,707 bytes. Also the level of detail of the concepts and terms in such texts is widely varying. The average file size is 8,973 bytes.

It is known in literature that unbalanced training data produces bias effects on the acquired classifiers, and there exist several techniques to overcome such limitation [21]. It is possible to concatenate all documents that belong to a given class in a single file, then truncating the file when it reaches a fixed size. Also, it is possible to balance training data by sampling chunks from the files in each class, until a given threshold file-size is reached. Since our dataset was too small to undertake any automatic categorization approach, we simply tried to enlarge it, and adopted the

[5] `http://www.gzip.org/algorithm.txt`

Table 1: The size of files (in bytes) in the dataset. The last column reports about the support document *sd* that we added to each class. For space reasons, some class names have been shortened.

class	# files	files size	*sd* size
Agriculture	7	49,565	203,354
Assistance	10	32,614	273,488
Trade	8	137,922	361,221
Public housing	7	67,798	332,137
Education	12	138,693	268,243
Regional organization	16	124,122	292,317
Tourism	10	77,463	504,439
Total	70	628,177	2,235,199

following strategy. For each class we added a support document *sd* –which of course has been used only for training purposes–, containing provisions taken from regional legislation downloaded from the Internet, and having the same subject as that class. The final dataset we used is detailed in Table 1.

Baseline classifiers. In order to provide a baseline against which to compare the results of the three outlined procedures, we tested a batch of standard classifiers. In this case we had to build a feature vector representation: we implemented a standard approach, consisting of stop words filtering, lemmatization and extraction of *TF-IDF* features. In particular, for a collection of N documents with m features (with n_t documents containing term t), each weight $w(d,t)$ for a given term t in document d is computed through the familiar formula

$$w(d,t) = \frac{\mathrm{tf}(d,t) \cdot \log(N/n_t)}{\sqrt{\sum_{j=1}^{m} \mathrm{tf}(d,t_j)^2 \cdot (\log(N/n_{t_j}))^2}}.$$

Three classifiers were trained based on such data, and tested on a 10-fold cross validation basis. Results were averaged through 50 executions of the experiment. Specifically, we used the J48, NaiveBayes, and SMO algorithms. They are all popular (and general-purpose) implementations taken from the Weka workbench [15]: J48 is a Java implementation of the decision tree learning algorithm C4.5; NaiveBayes implements a simple naïve Bayesian classifier; and SMO implements Platt's sequential minimal optimization algorithm for training support vector classifiers [26].

Results and Discussion

The accuracy of the tested algorithms is reported in Table 2: the best results are obtained by the BCN and NCD procedures (75.71% and 64.29% accuracy, respectively), which is nearly approached by the NaiveBayes algorithm (61.14% accuracy).[6] A first remark is that the problem confirms to be a challenging one, since no algorithm provided satisfactory results. Classical approaches (J48, NaiveBayes and SMO) seem to suffer the reduced size of the dataset more than compression based ones. Interestingly, if we consider 2 nearest neighbors (the two most voted classes) rather than only the first one, the success rate raises to 88.71% for the BCN approach, and 80.65% for the NCD approach, respectively. By considering the three most voted classes, we obtain 97.14% correct results with BCN and 87.14% with NCD.

[6]The results obtained with no support document: AMDL 54.29% accuracy; BCN 74.29% accuracy, and NCD 61.43% accuracy.

Table 2: The accuracy of the 6 compared approaches.

J48	NaiveBayes	SMO
36.00%	61.14%	50.86%
AMDL	**BCN**	**NCD**
55.71%	75.71%	64.29%

Table 3: The accuracy of the two best classification schemes for each class.

class	BCN	NCD
Agriculture	85.71%	42.86%
Assistance	50.00%	90.00%
Trade	50.00%	37.50%
Public housing	71.43%	42.86%
Education and training	83.33%	50.00%
Regional organization	87.50%	81.25%
Tourism	90.00%	80.00%
Weighted average	75.71%	64.29%

The detailed results of the BCN procedure, which attained the highest accuracy, are reported in Appendix A. A closer examination of the errors reveals some interesting cases. Some documents should have been annotated with more than one single label. For example, a document labeled as "Agriculture", but containing norms about agritourism has been misclassified as "Tourism". Elsewhere we notice that classes are not clearly separated: let us consider, e.g., that the classes "Assistance and social services" and "Public housing" are at least partially overlapped, and in some cases they would be confusing for human beings, too. However, to fully assess our results, it would be useful to record the inter-annotator agreement, especially for ambiguous cases. Moreover, the fact that classes are not well separated and that an inspection of class contents reveals subclass relationships, suggests that multiclass and hierarchical classification schemes should be considered to categorize these documents.

A deeper inspection of the two best classification schemes –BCN and NCD– is provided in Table 3: it seems to suggest a correlation between the accuracy rate and the size of the dataset, so we decided to test the considered approaches in a further experiment. We tested our implementation of the mentioned algorithms on a widely studied task, that is the authorship attribution [19, 3, 24]. In particular, we tried to replicate the experimentation described in [3]: a dataset composed of 97 files, amounting to 34,588,616 bytes storage (more than 15 times the object provisions dataset) was downloaded from the same site used by the authors.[7] The detail of the files available per author and the overall files size are provided in Table 4.

The results are reported in Table 5. The first important fact is that the accuracy of BCN grows as the files size increases, in spite of a larger number of classes (11 authors were present), thus scaling better than competitors to a more realistic setting. As regards as NCD, it increases the accuracy obtained in the object provisions dataset. This procedure confirms to be robust to larger datasets, and that

[7]`http://www.liberliber.it`

Table 4: The size of classes (in bytes) in the dataset.

class	# files	files size
D'Annunzio	10	3,035,621
Dante	10	1,563,576
Deledda	10	2,733,274
Fogazzaro	9	4,708,338
Guicciardini	9	5,537,608
Machiavelli	7	2,151,209
Manzoni	5	1,860,768
Pirandello	9	2,139,583
Salgari	10	3,905,974
Svevo	8	4,267,395
Verga	10	2,685,270

Table 5: The accuracy of the compared approaches tested on the authorship attribution problem.

J48	NaiveBayes	SMO
33.09%	62.55%	37.45%
AMDL	**BCN**	**NCD**
13.40%	80.41%	70.10%

it can be fruitfully employed to handle cases were larger data is available. The last considered procedure, AMDL, degrades to a very poor performance, slightly superior to random guess. In this case, we suspect to have missed some critical implementation details. The same should be said about the Weka implementation of J48, NaiveBayes and SMO, whose performance was much lower than expected.

Considering the first two most voted classes, the success rate of BCN raises to 85.57%, and that of NCD reaches 74.23%. The correct solution is found, at the best of the three highest scored classes, in 92.78% of cases by BCN, and in 83.50% of cases by NCD. Similar to the object provisions categorization, the 'shortlisting' approach provides encouraging results. Such figures represent the upper bound to the accuracy of a further classifier considering only the short list composed of two or three classes. An extended architecture can be drawn, based on a two-fold strategy: at the first step a small subset of classes can be selected (we presently considered 2 and 3 nearest neighbors); at a later stage, semantically-grounded techniques can be exploited to disambiguate among these few classes. This attempt would allow bounding the increased computational costs due to the adoption of semantic technologies, such as those based on ontological knowledge.

5. RELATED WORK

The problem of automatically categorizing legal texts has a long tradition in the AI & Law community, and many approaches have been proposed in literature. Since the early attempts, one main strategy has been that of generating legal thesauri and then trying to categorize documents based on some sort of proximity between terms in the thesaurus and in the documents; another ubiquitous design choice has been that of representing documents as feature vectors.

One pioneering work dealing with legal documents categorization proposed the system KONTERM. KONTERM was designed as a tool for automatically indexing documents: after the creation of the thesaurus, where along with terms proper, some surrounding context was recorded based on the assumption that most terms meaning is generally conveyed by terms context [28]. Most often categorization has been considered as part of the investigation on argumentation techniques in case law, where the problem of identifying the similarity between cases is a principal one. This approach has been adopted, e.g., in a hybrid CBR-IR system [27]. The relevance of cases stored in the case base was determined by maximizing the shared *dimensions* between the case under consideration and those in the KB. Dimensions were analogous to features, and they were used for indexing and for comparison purposes. A CBR approach was also adopted by the system SPIRE [11], designed to identify legal passages containing relevant information about features present in court opinions.

Also at the intersection of CBR and IR is the work [4], where the problem of classifying case opinions was tackled in the frame of an intelligent tutorial environment (CATO) aimed at teaching argumentation to law students. Unfortunately, the task proved to be a very hard one, and the authors declared that "Since the generalization power of purely inductive algorithms [...] does not measure up to the complexity of the concepts [...], the learning algorithms' performance is not satisfactory yet." In a subsequent work the same authors enriched obtained enhanced categorization accuracy by providing the learning algorithms with a legal thesaurus and text parsing information [5]. In 2001 the model was extended by adding further elements, such as accounting for negation and roles, based on the observation that the presence of proper names may prevent classifiers from correctly categorizing texts (vice versa, the information on roles supports correct classification) [6]. Subsequently, the authors proposed three different sorts of representation: bag of words, with the mentioned roles, and also with "propositional patterns" including roles information. In the latter case, additional information is retained about sequences of words that fall within predefined syntactic relationships (subject-verb, verb-object, verb-prepositional phrase, and verb-adjective) [2].

The work by [29] investigates how to automate the indexing of the West Legal Directory, an online legal retrieval system [1]. In particular, the paper provides a comparison of classification results obtained with a C4.5, with kNN using TF-IDF as distance measure and with Ripper, a rule induction algorithm. The feature selection adopted is interesting to our present concerns, in that features were selected from a set of manually assigned keywords taken from the West Legal Directory: in particular, for each category a set of 300 features with higher TFIDF score was retained from the whole feature set (initially composed of 900 keywords) and paired with 300 further features associated to the category.

Also the work proposed in [22] tackles the problem of automatically categorizing arguments in legal texts; particular emphasis is given to assessing different feature sets, including lexical, syntactic, semantic and discourse properties of the analyzed texts. The resulting feature vectors are then used to train a Multinomial naïve Bayes classifier and a Maximum Entropy classifier. The feature set includes unigrams, bigrams, trigrams, adverbs (used to detect argumentative information), verbs, modal auxiliary verb, word couples (all combinations of two words in the sentence); text statistics (including sentence length, average word length, number of

punctuation marks); punctuation; key words (a set of keywords used as predictors of argumentation); parse features (in particular tree depth and number of subclauses). It is noteworthy that the best results were obtained by combining "word couples selected by their POS-tag, verbs and statistics on sentence length, average word length and number of punctuation marks (accuracy of 73.75%)" [23, Sec. 4.3]: with the exception of the word couples, such information seems quite similar to that grasped through the compression-based methods used in the present work. However, although less informative about their decisions, the algorithms we tested seems have one strong advantage over these works: they are parameter free [17], and do not require deciding which combination of features to use (features are in the order of thousands, and thus finding the best combination entails another optimization problem).

6. CONCLUSIONS AND FUTURE WORK

We illustrated some algorithms for the categorization of legal texts, namely the object provisions of Italian Constitutional Court. The problem proved to be a fascinating and challenging one, due to the mentioned peculiar traits of legal language. Although the techniques we explored are not new, to the best of our knowledge they had never been used before to categorize legal texts and compared to standard approaches. In a preliminary experimentation compression based algorithms provided encouraging results; in the meantime we argued that knowledge richer additions (such as using thesauri and other sorts of information) can be paired to present algorithms to improve results in realistic settings. Also, based on the characteristics of the implemented kNN procedures, we elaborated on how to extend our current approach to employ such richer representation. This will be our future work.

One interesting result is that compression schemes proved to be effective in dealing with our dataset (and in the Italian authors dataset, too, where they were known to work fine). Provided that the reduced size of datasets menaces to undermine learning based approaches, the results obtained on the object provisions dataset are appreciable and encouraging enough. Still, these algorithms are parameter-free, and therefore they overcome the drawbacks coming from incorrect settings (which may result in classification failures) and in general from arbitrary choices; and they do not require any sort of preprocessing and/or feature selection.

A concluding remark about the overall impact of the present work. We have illustrated algorithms that we employed for the analysis of regional legislation. It is our opinion that legal texts categorization can be fruitful in providing a response to the demand for more transparency and knowability, as opposed to a law-making that is to some extent confusing, fragmented and inaccessible.

7. ACKNOWLEDGMENTS

We are grateful to Roberto Esposito for the discussion on Kolmogorov complexity issues, and for his precious advices on earlier versions of the paper.

8. REFERENCES

[1] West'sLawFinder: A Legal Resources Guide. West Publishing Company.

[2] K. Ashley and S. Brüninghaus. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law*, 17(2):125–165, 2009.

[3] D. Benedetto, E. Caglioti, and V. Loreto. Language trees and zipping. *Physical Review Letters*, 88(4):48702, 2002.

[4] S. Brüninghaus and K. Ashley. Finding factors: learning to classify case opinions under abstract fact categories. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law*, pages 123–131. ACM, 1997.

[5] S. Brüninghaus and K. Ashley. Toward adding knowledge to learning algorithms for indexing legal cases. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law*, pages 9–17. ACM, 1999.

[6] S. Brüninghaus and K. Ashley. Improving the representation of legal case texts with information extraction methods. In *Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 42–51. ACM, 2001.

[7] S. Calzolaio. State And Regional Legislation In Italy In The Decade After The Constitutional Reform. *Italian Journal of Public Law*, 2:399–454, 2012.

[8] P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2):261–272, 2007.

[9] R. Cilibrasi and P. Vitányi. Clustering by compression. *Information Theory, IEEE Transactions on*, 51(4):1523–1545, 2005.

[10] G. Damele, M. Dogliani, A. Mastropaolo, F. Pallante, and D. P. Radicioni. On Legal Argumentation Techniques: Towards a Systematic Approach. In M. A. Biasiotti and S. Faro, editors, *From Information to Knowledge – Online Access to Legal Information: Methodologies, Trends and Perspectives*, Frontiers in Artificial Intelligence and Applications, pages 119–127, Amsterdam, Netherlands, 2011. IOS Press.

[11] J. Daniels and E. Rissland. Finding legally relevant passages in case opinions. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law*, pages 39–46. ACM, 1997.

[12] S. Delany and D. Bridge. Feature based and feature free textual CBR: a comparison in spam filtering. In D. Bell, P. Milligan, and P. Sage, editors, *Proceedings of the 17th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'06)*, pages 244–253, 2006.

[13] E. Frank, C. Chui, and I. Witten. Text categorization using compression models. In *Proceedings of DCC-00, IEEE Data Compression Conference*, pages 200–209, 2000.

[14] P. Grünwald. A tutorial introduction to the minimum description length principle. *arXiv preprint math/0406077*, 2004.

[15] G. Holmes, A. Donkin, and I. Witten. Weka: A machine learning workbench. In *Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems*, pages 357–361. IEEE, 1994.

[16] A. Hotho, A. Maedche, and S. Staab. Ontology-based text document clustering. *Künstliche Intelligenz (KI)*, 16(4):48–54, 2002.

[17] E. Keogh, S. Lonardi, and C. Ratanamahatana. Towards parameter-free data mining. In *Conference on Knowledge Discovery in Data: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 22, pages 206–215, 2004.

[18] D. Khmelev and W. Teahan. A repetition based measure for verification of text collections and for text categorization. In *Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 26 th annual international ACM SIGIR conference on Research and development in informaion retrieval*, volume 28, pages 104–110. ACM, 2003.

[19] O. Kukushkina, A. Polikarpov, and D. Khmelev. Using Literal and Grammatical Statistics for Authorship Attribution. *Problems of Information Transmission*, 37(2):172–184, 2001.

[20] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi. The similarity metric. *Information Theory, IEEE Transactions on*, 50(12):3250–3264, 2004.

[21] Y. Marton, N. Wu, and L. Hellerstein. On compression-based text classification. In D. E. Losada and J. M. Fernández-Luna, editors, *ECIR*, volume 3408 of *Lecture Notes in Computer Science*, pages 300–314. Springer, 2005.

[22] M. Moens, E. Boiy, R. Palau, and C. Reed. Automatic Detection of Arguments in Legal Texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pages 225–230. ACM, 2007.

[23] M.-F. Moens. Innovative techniques for legal text retrieval. *Artificial Intelligence and Law*, 9(1):29–57, 2001.

[24] W. Oliveira Jr, E. Justino, and L. Oliveira. Authorship attribution of documents using data compression as a classifier. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, 2012.

[25] F. Pallante. L'oggetto della potestà legislativa regionale residuale: l'esperienza piemontese a confronto con quella lombarda. *Federalismi*, 3, 2010.

[26] J. Platt. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 42–65. MIT Press, Cambridge, MA, 1998.

[27] E. Rissland and J. Daniels. Using CBR to drive IR. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 400–407. Citeseer, 1995.

[28] E. Schweighofer, W. Winiwarter, et al. Intelligent information retrieval: Konterm-automatic representation of context related terms within a knowledge base for a legal expert system. In *Proceedings of the 25th Anniversary Conference of the Istituto per la documentazione giuridica of the CNR: Towards a Global Expert System in Law,(Padua, Italy)*. Citeseer, 1994.

[29] P. Thompson. Automatic Categorization of Case Law. In *Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 70–77. ACM, 2001.

[30] 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*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[31] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *Information Theory, IEEE Transactions on*, 23(3):337–343, 1977.

APPENDIX

A. RESULTS OF THE BCN PROCEDURE

Table 6: Detailed results of the *BCN* classification procedure.

File #	Correct class	*BCN* output	File #	Correct class	*BCN* output
1	Agriculture	Agriculture	36	Education	Education
2	Agriculture	Agriculture	37	Education	Education
3	Agriculture	Agriculture	38	Education	Education
4	Agriculture	Tourism	39	Education	Education
5	Agriculture	Agriculture	40	Education	Education
6	Agriculture	Agriculture	41	Education	Education
7	Agriculture	Agriculture	42	Education	Tourism
8	Assistance	Public housing	43	Education	Education
9	Assistance	Assistance	44	Education	Education
10	Assistance	Public housing	45	Regional organization	Regional organization
11	Assistance	Assistance	46	Regional organization	Regional organization
12	Assistance	Assistance	47	Regional organization	Regional organization
13	Assistance	Public housing	48	Regional organization	Regional organization
14	Assistance	Assistance	49	Regional organization	Regional organization
15	Assistance	Public housing	50	Regional organization	Regional organization
16	Assistance	Tourism	51	Regional organization	Tourism
17	Assistance	Assistance	52	Regional organization	Regional organization
18	Trade	Regional organization	53	Regional organization	Regional organization
19	Trade	Trade	54	Regional organization	Regional organization
20	Trade	Education	55	Regional organization	Regional organization
21	Trade	Trade	56	Regional organization	Regional organization
22	Trade	Tourism	57	Regional organization	Public housing
23	Trade	Trade	58	Regional organization	Regional organization
24	Trade	Trade	59	Regional organization	Regional organization
25	Trade	Tourism	60	Regional organization	Regional organization
26	Public housing	Public housing	61	Tourism	Tourism
27	Public housing	Public housing	62	Tourism	Tourism
28	Public housing	Tourism	63	Tourism	Agriculture
29	Public housing	Public housing	64	Tourism	Tourism
30	Public housing	Public housing	65	Tourism	Tourism
31	Public housing	Public housing	66	Tourism	Tourism
32	Public housing	Tourism	67	Tourism	Tourism
33	Education	Regional organization	68	Tourism	Tourism
34	Education	Education	69	Tourism	Tourism
35	Education	Education	70	Tourism	Tourism