Ranking and Classifying Legal Documents using Conceptual Information

Kees van Noortwijk, Johanna Visser and Richard V. De Mulder
Centre for Computers and Law
Erasmus University Rotterdam
vannoortwijk@law.eur.nl

Abstract

A substantial part of all written legal information is published in electronic form these days. Existing retrieval systems, however, are increasingly found to be inadequate. Conceptual ranking and retrieval, in this case based on Bayesian statistics, can be a powerful alternative. Working prototypes of two applications are described. The first one provides the user with the possibility to define, test and save retrieval concepts. Such concepts can be used to rank documents retrieved from a database. The second application reads the saved concepts and calculates the probability that a new document is relevant to the concept.

Keywords

Information retrieval, conceptual retrieval, classification, Bayesian classifier, legal databases, WIPO, Codas, domain names

1. Introduction

The most common method for retrieving a document, in use since more than 40 years and to be found in almost every legal databank, is still based on ‘Boolean searching’. The user has to specify one or more keywords, after which the retrieval system produces a list of documents containing those words. This search method, although widely known and accepted, has considerable limitations. The most important of these, as has already been indicated in Van Noortwijk & De Mulder 1997, is that the searching is completely based on the *form* of the documents (to be more precise, on the *words* that these contain). This means that lawyers who want to look up certain legal cases have to make a crucial conversion. They know what the cases should be *about*. But for the search operation, they have to speculate *which words* should be present in such cases. The set of words must be assembled with care, avoiding common words but also too specific terms. Finally, the output that most current retrieval systems generate often has limitations. In many cases, the list of documents that is the result of a search request is not ordered according to the (expected) relevance of the documents. Even if an attempt is made to sort the list, it is often not clear what criterion is applied for that. It could be that a relatively unimportant keyword that is part of the search request is responsible for the high ranking of a certain document, just because that single keyword appears in it unlike in the other documents.

Improvements in this field, for instance in the form of a more powerful retrieval mechanism, can be of great benefit to lawyers, dependent as they have become of the variety of electronic sources. A very interesting idea in this respect is the construction of ‘conceptual’ retrieval systems, capable of locating information that conforms to a specific retrieval concept. This could be a legal concept, like ‘tort’, ‘trade secret’ or ‘fundamental breach’, but also a more ad-hoc retrieval concept like

‘all documents from set X that deal with article Y from regulation Z’ or ‘arbitral cases in which bad faith plays a role’.

Legal experts tend to define concepts in a normative way. But for concepts that are used in document retrieval tasks, this can present a problem. More dynamic concept definitions are often necessary for that, to reflect certain changes or developments in society. It is possible that in court proceedings and especially in alternative dispute resolution proceedings new legal concepts are defined and existing legal concepts are altered. When using basic search methods, like standard Boolean searching, it can be very difficult to refer to such a legal concept by means of a limited set of keywords. Therefore Boolean searching will often fall short when ranking the found documents according to the concept a user is looking for. In general, it is obvious that the formulation of a concept has to be based on knowledge about the field (in this case, the law)[1].

In an earlier article in JILT[2] we already introduced some basic techniques for document comparison, such as the calculation of a similarity score based on word use. Since then, several publications on this subject have emerged. This article, however, reports on two new applications that take the principle a step further. Not only do they make it possible to define concepts interactively, but they can also store concepts and re-use these, for future database search operations but also to classify single (new) documents.

2. Conceptual retrieval in practice

Several possibilities to construct conceptual retrieval systems for legal documents have been proposed in the last decades[3]. A characteristic of many of these approaches and/or systems is that they intend to model established legal concepts and use these for retrieval purposes. An example of this is document retrieval based on structured *knowledge* about a (legal) domain, for instance in the form of a so-called *ontology*[4]. Several researchers have reported on this recently.[5] Constructing an ontology of a non-trivial domain is a complex task, however, and when it is finished, it constitutes a very rigid structure that can not be changed or adapted easily, at least not by end users.

In this paper, we have taken a different approach. Instead of working with predefined retrieval concepts, the program provides users with the possibility to define their own. This could be strict legal concepts, like ‘theft’ or ‘breach of contract’, but also more general ones like ‘documents that refer to terrorism’. The user can test retrieval concepts and apply these to retrieve information from a case law database. A well-defined concept makes it possible to perform a search operation with a much higher recall and precision than would otherwise be possible. Furthermore, concepts that have been defined can be reused many times. They can also be refined to improve their quality, using the results of search operations.

An important question is of course if this conceptual approach is specific to the legal field or if it can be applied in other disciplines as well. In fact application in other fields is certainly possible. But in the field of law it is a necessity, because

- lawyers depend heavily on text material (regulations, case law);
- most of this material is nowadays stored in huge (and always growing) but unstructured electronic document collections;
- finding or not finding a certain piece of information can mean the difference between winning or loosing the case;

- retrieving all (or even a modest percentage) of the relevant documents using just Boolean searching proves to be very difficult, if not impossible.

Conceptual retrieval – in any form – is often presented as a superior methodology for retrieving documents. Practical implementations differ greatly, however, even if we limit ourselves to the field of law. One of the reasons for this is probably that the term 'concept' can refer to different entities[6] and that concepts can be defined in a number of ways[7]. For practical applications, the way that is chosen should depend on the intended purpose of the concept. In this case, the main purpose is to retrieve legal documents from a dataset. Therefore, the definition of the 'retrieval concept' should be connected to that. It should specify a set of documents from the database, including as many relevant documents as possible and excluding irrelevant ones.

The way we have chosen to implement this type of concept is the following: the user has to identify *example documents*. These are documents from the dataset that the user considers to be relevant to the concept that is being defined. We will call such documents *exemplars*. Furthermore, the user can indicate documents that, even though they might resemble the exemplars, are not relevant to the concept. We will call these *counter-exemplars*. The searching facility of the retrieval system will then search for other documents that are similar to the exemplars (and dissimilar to the counter-exemplars). To fulfill that task, certain attributes of the (counter) exemplars and the other documents have to be compared. When the attributes show a sufficient match, the document is considered to be similar and therefore possibly relevant to the retrieval concept. By measuring the characteristics or the values of the attributes a *matching score* can be calculated. Such a score makes it possible to not only accept or reject documents, but also to *rank* them according to their matching score and, with that, to their expected relevance to the concept.

3. Calculating a matching score for documents

In theory, all kinds of attributes could be used when comparing documents. Salton (1971), for instance, already described a method to form clusters in a database by means of word use statistics. His method, however, depends on manually selected index words, which prevents the creation of a system for fully automatic classification of documents. In the last 10 to 15 years, research on what is often called 'probabilistic searching' – as the *probability* that a document is relevant is calculated from certain document attibutes – has intensified with a focus on several statistical techniques. One of these, usually referred to as the 'Naive Bayes Model' or 'Naive Bayes Classifier' to a certain extend plays a role in the application described here as well. The document attributes that this method is applied to here are (all) the word types (different words, or vocabulary) from the set of documents.

In principle, every word present in one or more documents can be used as an attribute to compare these documents and to calculate a matching or similarity score. If a word is present in document A and also in document B (a 'hit'), this should increase the similarity score for these two documents. If a word is present in one of the two, but not in the other (a 'miss'), this represents a dissimilarity and the score should be decreased accordingly. A special case is represented by words that are present in other documents in the dataset, but not in documents A and B. As this is in fact a characteristic that documents A and B have *in common* (they *both* miss the word), such a word should also *increase* their similarity. A further point to consider is that not all words should have the same impact on similarity: very common words should increase similarity only a little bit if they are found in both document A and B,

whereas very rare words should have a much higher 'weight'. Van Noortwijk & De Mulder 1997 contains a detailed description of this type of similarity calculations[8].

Similarity measures are especially useful when comparing document *pairs*. Applied to, for instance, a case law database such a measure could answer the question: "What other document from the database matches this document X most closely?". To build a conceptual retrieval system, however, that is not enough. We want to establish a full ranking of the documents, not just calculate similarity between two of them. Also, we want this ranking to be based on the combined characteristics of a set op exemplars and counter exemplars. For that, we need a different technique.

4. Ranking documents using Bayesian statistics

The application to define concepts and to rank and classify documents with these, as described here, is based on the use of Bayesian statistics[9]. The basic idea is that information changes the odds that a certain outcome will occur. For instance, the information that a student has not prepared himself for a test (information x) will change the probability that he will get a high mark (fact f). The odds after the information are nevertheless a function of the original (or 'a priori') odds. According to Bayes

$$\frac{p(f \mid x)}{p(not\ f \mid x)} = \frac{p(x \mid f)}{p(x \mid not\ f)} \times \frac{p(f)}{p(not\ f)} \qquad (1)$$

or

$$odds(f \mid x) = odds(x \mid f) \times odds(f) \qquad (2)$$

in which $p(f)$ means: the probability of f and $p(x|f)$ means: the probability of x given f.

The values at the right of the equal sign are often known, which makes it possible to calculate the value on the left. Therefore, if we have information x, we can estimate to what extent the probability of f increases or decreases, if the probability of x given f as well as that of x given not-f are known. In our example, $p(x \mid f)$ would mean: the probability that the student has not prepared himself for the test, given that he has got a high mark. This probability is usually low (for most students). Therefore, the new odds $(f \mid x)$ will be considerably lower than the a priori odds(f). The model can be extended when more information is available: if not only x but also y is given, we can include this (and therefore calculate the odds of f given x *and* y) by multiplying the factor to the right of the equal sign with $odds(y \mid f)$.

This theory has been applied to document retrieval in a number of ways. One of these is particularly relevant here: the so-called 'Naive Bayesian Classifier'[10]. Here, the same technique described above is used to classify (or categorize) documents in a database. The probability that a certain document belongs to a certain class is calculated from the probabilities of each of its attributes, given that it belongs to the class. The classification is called 'naive' because this probability calculation is in fact only allowed if the attributes (which take the place of the information items x and y from the example above) are *independent* of each other. When we use words as attributes, that is of course a requirement that is impossible to fulfill. Nevertheless, even though the prerequisites for the calculation are in fact not met, the classification using this method is often found to be correct.

We want to rank documents according to the probability that they are relevant to a certain retrieval concept (in terms of the above: documents that are in the class 'relevant to the specified concept'). A computer program by the name of CODAS, which stands for Conceptual Document Analysis System, developed at Erasmus University is capable of doing that. What is *specific* in this program (apart from the fact that it is a working application, not just a model) is not so much the Bayesian statistics involved (as we already stated, numerous researchers have defined and used these) but the method by which the retrieval concept (or class of desired documents) is defined and the fine-tuning to the type of concepts and datasets that are used in the legal field.

To start with the first, the definition of the retrieval concept, this is done by using example documents from the actual dataset. When the program is initialized the user has to specify several *exemplars* and *counter exemplars* first: documents from the dataset that are known to be relevant and documents that are known to be irrelevant. Especially the last category is specific to this implementation. The set of examples is in fact the specification of the retrieval concept: the user wants to retrieve as many documents as possible that are similar to the *exemplars*, but dissimilar to the *counter exemplars*. After this initial step, the program calculates odds for every document in the dataset. These odds represent the probability that the document is relevant to the specified retrieval concept. After this a list of all documents, sorted according to each document's odds, is shown on screen.

The calculation of the document odds is crucial in this process. Like in the example outlined above, it is based on information. The a priori odds that a document is relevant are very low, possibly as low as 1 / N (where N represents the total number of documents) if there is only one document we are looking for. Information to supplement this is obtained from the word use in the documents. This word use is compared, which generates an extended series of probabilities $p(x|f)$ – the probability that a certain word is present given the fact that the document is relevant – from the exemplars and a series of $p(x|not\ f)$ – the probability that a word is present given the fact that the document is irrelevant – from the counter exemplars. With these probabilities, the odds for every document can be calculated. As we compare the odds of all documents, the a priori odds are in fact not relevant here.

Of course the calculation and comparison of document odds, although relatively uncomplicated in theory, yields unexpected problems. For instance, because of the high number of attributes and documents, probabilities can become very low. This means that calculations have to be performed with high precision (many decimals), as otherwise too much information will be lost. Furthermore, the integration of odds calculated from exemplars and from counter exemplars complicates the algorithm. Details on the solutions for this will possibly be the subject of a future report.

Another important consideration is of course the practical usability of this method. Specifying a sufficient number of exemplars and counter exemplars might seem a lot of work for 'just' document retrieval, a task that many users have a basic understanding of. However, tests show that the time to define a retrieval concept using the method described below decreases quickly with a little training. In our experience, after a few hours of trial and error most users are capable of specifying a basic set of exemplars and counter exemplars in less than ten minutes. After that, the set – and therefore the concept it represents – can be extended and refined, but results are already visible in the form of a first ranking of documents. Furthermore, sets can be saved and re-used (in original form or with alterations) for future retrieval operations. On the average, the technique should not take more time than a carefully performed 'regular' search operation using Boolean operators.

5. An example: ordering a series of cases

To demonstrate the practical use of the method, a sample databases with case law on the subject of Internet domain names was compiled. A homogeneous set of cases on this subject was taken from the online database of WIPO (World Intellectual Property Organization). The WIPO Arbitration and Mediation Center[11] offers Alternative Dispute Resolution (ADR) options including arbitration and mediation services for the resolution of international commercial disputes between private parties. In this case, only arbitral decisions concerning Internet domain name disputes have been used. Since December 1999, the Center has administered over 5000 proceedings. WIPO provides online publications of cases and decisions[12].

The set of decisions that was used here consists of about 400 arbitral decisions that were picked randomly from arbitral publications. From this group twenty cases were again randomly selected and set apart for later use (see the chapter on classifying documents). In the arbitral decisions the registration and use of an internet domain name is disputed. WIPO arbiters have to apply the *Uniform Dispute Resolution Policy* (UDRP-rules)[13], also known as ICANN Policy. For this particular research project we have concentrated on one of the three grounds necessary for awarding a complaint and transferring a domain name to the complainant:

> *There is evidence of registration and use in bad faith.*

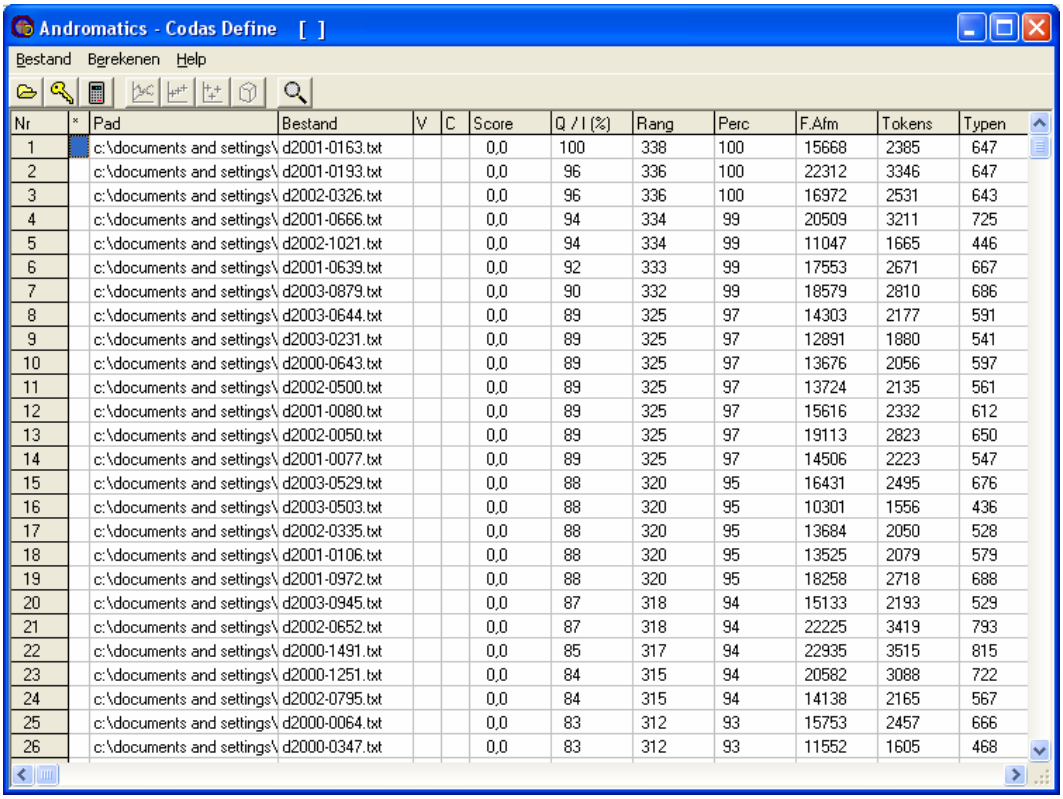
Although this ground contains the legal term ‘bad faith’, according to the ICANN Policy the concept of ‘bad faith’ is used in a more limited sense than is the case in other legal domains. In general, more than 80% of all cases are awarded by WIPO. If the ground mentioned above is not present or unproven, the complaint should be denied. We have used the ranking program, which is called Codas Define, to locate cases in which the ground is not satisfied or is missing, leading to the denial of the claim of complainant.

5.1 Using Codas Define

The program Codas Define (and also the Classify module, described in the next chapter) has been developed as part of the CODAS project. It is an easy to use Windows application that shows most results in a list on the screen but is also capable of representing the results graphically[14].

Working with the program to sort a set of cases takes place as follows. First, the location of the documents (or database) has to be set. In this case, separate documents in MS-Word format were used. These documents had been downloaded from the WIPO Internet-site. The program needs the original document but also a copy in the .TXT format (i.e. without layout).

After these settings have been entered (they can also be saved for later use) the user issues the ‘Calculate’ command. A list of documents appears, for the moment ordered according to the so-called ‘Initial score’.



Nr	*	Pad	Bestand	V	C	Score	Q / I (%)	Rang	Perc	F.Afm	Tokens	Typen
1		c:\documents and settings\	d2001-0163.txt			0,0	100	338	100	15668	2385	647
2		c:\documents and settings\	d2001-0193.txt			0,0	96	336	100	22312	3346	647
3		c:\documents and settings\	d2002-0326.txt			0,0	96	336	100	16972	2531	643
4		c:\documents and settings\	d2001-0666.txt			0,0	94	334	99	20509	3211	725
5		c:\documents and settings\	d2002-1021.txt			0,0	94	334	99	11047	1665	446
6		c:\documents and settings\	d2001-0639.txt			0,0	92	333	99	17553	2671	667
7		c:\documents and settings\	d2003-0879.txt			0,0	90	332	99	18579	2810	686
8		c:\documents and settings\	d2003-0644.txt			0,0	89	325	97	14303	2177	591
9		c:\documents and settings\	d2003-0231.txt			0,0	89	325	97	12891	1880	541
10		c:\documents and settings\	d2000-0643.txt			0,0	89	325	97	13676	2056	597
11		c:\documents and settings\	d2002-0500.txt			0,0	89	325	97	13724	2135	561
12		c:\documents and settings\	d2001-0080.txt			0,0	89	325	97	15616	2332	612
13		c:\documents and settings\	d2002-0050.txt			0,0	89	325	97	19113	2823	650
14		c:\documents and settings\	d2001-0077.txt			0,0	89	325	97	14506	2223	547
15		c:\documents and settings\	d2003-0529.txt			0,0	88	320	95	16431	2495	676
16		c:\documents and settings\	d2003-0503.txt			0,0	88	320	95	10301	1556	436
17		c:\documents and settings\	d2002-0335.txt			0,0	88	320	95	13684	2050	528
18		c:\documents and settings\	d2001-0106.txt			0,0	88	320	95	13525	2079	579
19		c:\documents and settings\	d2001-0972.txt			0,0	88	320	95	18258	2718	688
20		c:\documents and settings\	d2003-0945.txt			0,0	87	318	94	15133	2193	529
21		c:\documents and settings\	d2002-0652.txt			0,0	87	318	94	22225	3419	793
22		c:\documents and settings\	d2000-1491.txt			0,0	85	317	94	22935	3515	815
23		c:\documents and settings\	d2000-1251.txt			0,0	84	315	94	20582	3088	722
24		c:\documents and settings\	d2002-0795.txt			0,0	84	315	94	14138	2165	567
25		c:\documents and settings\	d2000-0064.txt			0,0	83	312	93	15753	2457	666
26		c:\documents and settings\	d2000-0347.txt			0,0	83	312	93	11552	1605	468

Figure 1 - Top initial scores

The initial score in column ‘Q / I (%)’ indicates the average similarity (based on word use) of a document with all other documents. It is intended as an aid for finding cases that can be used as exemplars and counter exemplars[15]. In this case, document d2001-0163.txt has the highest (and maximum) initial score of 100. A great deal of its contents can probably also be found in many of the other documents. In other words, the top of the list contains the very ‘common’ documents. We can inspect some of these documents by double-clicking on the document name, to see if the ‘bad faith’ ground (see above) is present and accepted as proven in that specific case. It probably will be in most of these ‘common’ cases. Every case that is found to contain the ground is identified by placing a ‘+’ in column ‘V’. It is important to identify as varied a selection of these positive cases as possible, to cover the different forms in which these are found. The selection can be modified later, if necessary. Interestingly, two counter exemplars were also found in this part of the list. These were marked with a ‘–‘ in column ‘V’.

Nr	*	Pad	Bestand	V	C	Score	Q / I (%)
1	*	c:\documents and settings\	d2001-0163.txt	+		0,0	100
2	*	c:\documents and settings\	d2001-0193.txt			0,0	97
3	*	c:\documents and settings\	d2002-0326.txt	+		0,0	97
4	*	c:\documents and settings\	d2001-0666.txt	-		0,0	94
5	*	c:\documents and settings\	d2002-1021.txt	+		0,0	94
6	*	c:\documents and settings\	d2001-0639.txt			0,0	92
7	*	c:\documents and settings\	d2003-0879.txt	+		0,0	91
8		c:\documents and settings\	d2003-0644.txt			0,0	90
9		c:\documents and settings\	d2002-0500.txt			0,0	90
10		c:\documents and settings\	d2002-0050.txt			0,0	90
11	*	c:\documents and settings\	d2001-0080.txt			0,0	90
12	*	c:\documents and settings\	d2001-0077.txt	-		0,0	90
13	*	c:\documents and settings\	d2001-0972.txt	+		0,0	90
14		c:\documents and settings\	d2000-0643.txt			0,0	90
15	*	c:\documents and settings\	d2003-0231.txt			0,0	90
16		c:\documents and settings\	d2001-0106.txt			0,0	89
17	*	c:\documents and settings\	d2003-0503.txt	+		0,0	89
18	*	c:\documents and settings\	d2003-0529.txt			0,0	89
19		c:\documents and settings\	d2002-0335.txt			0,0	89

Figure 2 - (Counter) exemplars at the top of the list

At this point, two counter exemplars that did *not* contain the chosen ground (bad faith) had already been found. It is a fact that only a minority of the decisions does not contain the ground (as 80% of all cases is decided in favor of the complainant, which implies that in those cases the ground is present). 'Uncommon' or 'atypical' documents are to be expected at the *bottom* of the list, because it is sorted to the initial score. Therefore, it was expected that there would be more documents that lacked the 'bad faith' ground at that location. Indeed, when the lower 10 documents from the list were inspected, three more counter exemplars were found.

326	*	c:\documents and settings\	d2001-0527.txt			0,0	24
327	*	c:\documents and settings\	d2001-0184.txt			0,0	24
328	*	c:\documents and settings\	d2000-0996.txt			0,0	24
329	*	c:\documents and settings\	d2000-1674.txt			0,0	24
330	*	c:\documents and settings\	d2000-0169.txt			0,0	22
331	*	c:\documents and settings\	d2000-0491.txt	-		0,0	20
332	*	c:\documents and settings\	d2000-1786.txt	-		0,0	19
333	*	c:\documents and settings\	d2000-1220.txt			0,0	18
334	*	c:\documents and settings\	d2002-0617.txt			0,0	17
335	*	c:\documents and settings\	d2002-0485.txt			0,0	15
336	*	c:\documents and settings\	d2000-0616.txt			0,0	7
337	*	c:\documents and settings\	d2000-0906.txt	-		0,0	2
338	*	c:\documents and settings\	d2000-0621.txt			0,0	1

Figure 3 - Counter exemplars at the bottom of the list

Although the identification of this concept, which could be called 'Presence of bad faith', is still relatively weak – only six exemplars and five counter exemplars have been indicated – the program can already use it. When the Calculate button is pressed a new document score, based on Bayesian odds, is calculated. Next, the list is sorted to this score. The top and the bottom of the list now looked like this.

Nr	*	Pad	Bestand	V	C	Score	Q / I (%)
1	*	c:\documents and settings\	d2001-0163.txt	+		405,6	100
2	*	c:\documents and settings\	d2003-0879.txt	+		403,2	91
3	*	c:\documents and settings\	d2001-0972.txt	+		376,5	90
4	*	c:\documents and settings\	d2002-1021.txt	+		374,5	94
5	*	c:\documents and settings\	d2002-0326.txt	+		358,9	97
6	*	c:\documents and settings\	d2003-0503.txt	+		314,4	89
7		c:\documents and settings\	d2003-1035.txt			248,3	61
8		c:\documents and settings\	d2003-0945.txt			244,6	88

331		c:\documents and settings\	d2001-0722.txt			94,5	65
332		c:\documents and settings\	d2000-0853.txt			48,4	59
333		c:\documents and settings\	d2001-0074.txt			47,1	51
334	*	c:\documents and settings\	d2001-0077.txt	-		14,5	90
335	*	c:\documents and settings\	d2000-0906.txt	-		-67,3	2
336	*	c:\documents and settings\	d2000-0491.txt	-		-91,2	20
337	*	c:\documents and settings\	d2001-0666.txt	-		-91,3	94
338	*	c:\documents and settings\	d2000-1786.txt	-		-309,8	19

Figure 4 - Top and bottom after recalculation

The exemplars and counter exemplars are now grouped together at the top and the bottom of the list, respectively. That is what could be expected, as the new document scores are closely related to the contents of these documents. But the score the other documents now have is even more interesting. The documents on locations 7 to 12, for instance, have almost as high a score as the exemplars and consequently we would expect them to be positive decisions as well. Upon inspection of the documents, this indeed proved to be the case with all of them. At the bottom of the list, say from location 326 to 333, we would expect to find cases lacking the bad faith-ground. Indeed this proved to be true with cases 326 (d2002-1110), 328 (d2002-0404), 329 (d2000-1470) and 333 (d2001-0074). If we mark these cases (case 7 to 12 with a '+' and 326, 328, 329 and 333 with a '–'), the concept gains strength considerably. It now contains twelve exemplars and nine counter exemplars. This part of the process in fact incorporates a form of *relevance feedback*, a set of techniques of which some were already introduced in the 1960's. The purpose of these is to improve retrieval effectiveness, for instance by doing automatic query reformulation. Salton & Buckley (1990) describe and compare several of these techniques. Our approach is in fact what they indicate as a 'probabilistic feedback method', although a little different from the methods of this type that are described by them (for example, our method works with closed sets of documents, while their methods are intended to be used in open sets).

It is possible to continue inspecting documents and adding (counter) examples for a few more rounds. For the present example, we have chosen not to do so but to save the concept in its present state. As will be demonstrated, even a limited retrieval concept like this can already be used in the second and final step of the process, which consists of the classification of new documents based on the information from the present set of documents.

5.2 The automatic classification of new documents

Defining a retrieval concept, as specified in the previous chapter, is straightforward most of the time. Depending on the concept and the type of documents, the user seldom has to inspect more than a few dozens of documents to find suitable exemplars and counter exemplars. After these have been specified, the information drawn from them (which is used to calculate the document scores, based on Bayesian odds) is usually quite effective for sorting all the documents. The required documents can then be located easily.

However, the method described thus far only works for a 'closed' set of documents. The usability of a retrieval concept would be greatly enhanced if it would be possible to use it for other (sets of) documents as well. This is especially true for the legal domain, where for instance case law databases expand on an almost daily basis. It is necessary that users can 'classify' new documents with the existing concept, or even with multiple predefined concepts.

The CODAS Classify application does exactly this. It uses the concept information from the Define program. This program stores, after every recalculation, a list of all exemplars and counter exemplars (we call this the concept file), as well as a file that contains the essentials of the word use in all documents (we call this the dictionary). The Classify program reads these two files. Any new document the user then specifies is compared to the information from these two files.

Of course the Classify program can never identify new documents that do and do not conform to the concept with 100% accuracy. It is possible that the document contains new information, resulting in a word use that is different from all the documents in the existing set. The program can, however, determine quite precisely the location the new document would have if it where part of the existing database (on which the retrieval concept is based). It does this by giving a *percentile* value. A percentile of 100% means: this new document would be at the top of the list. Such a document would almost certainly conform to (be relevant for) the retrieval concept. Another document could have a percentile of 10%, which would mean that nine out of ten documents in the existing set would have a higher value – in other words, the probability that this is a document that does *not* conform to the concept is high.

To test the merits of this Classify program we used the following method. During the random selection of cases from the WIPO database, 20 cases had been set apart. These cases were not part of the original selection that was used to define the retrieval concept. Five of these 20 cases (25%) lacked the ‘Bad Faith’ ground, therefore the a-priori probability to select a case containing the ground was 75% (15/20). This was consistent with the rest of the dataset. Each of these new cases was now classified using the program. The output for one of the cases is shown in figure 5.

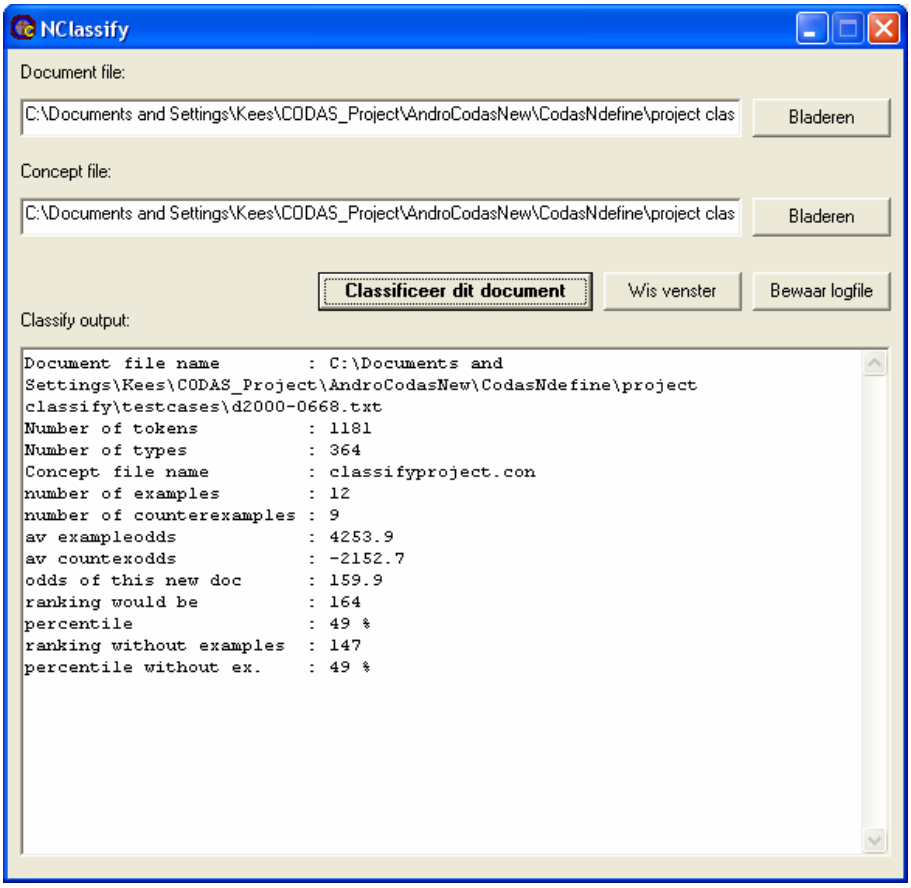


Figure 5 - The Classify program

The Classify program has a somewhat different user interface. The two edit boxes at the top are for the specification of the (new) filename and of the name of the concept file. When the button ‘Classificeer dit document’ (Classify this document) is clicked, the new document is read and the results are shown in the output window. Especially the percentile values shown in the last lines of this window are of importance here. The first value is the estimated percentile if all documents in the original database are taken into account, the second value is the percentile that would apply if the exemplar and counter exemplar documents are not counted (because of their influence on the original sorting order). The example document shown here has identical percentile values of 49%, which places it in the middle of the list. That means that this is probably not a decision lacking the ‘bad faith’ ground, as we would expect percentile values under 30% (because the a priori probability is over 70%) for such documents. The results for all 20 new documents were as follows (only the first percentile value is shown, the values usually differed only slightly).

Number	Case	Percentile	Bad faith	Remarks
1	d2000-1079	2%	Y	
2	d2000-1104	4%	N	
3	d2000-0728	10%	N	No proof of use
4	d2000-1756	13%	Y	
5	d2001-0700	14%	Y	
6	d2001-0233	15%	N	Only proof in Court
7	d2002-0196	18%	Y	
8	d2002-0481	19%	Y	
9	d2001-1055	29%	Y	
10	d2001-0067	39%	Y	
11	d2000-0850	41%	Y	
12	d2002-0732	44%	Y	
13	d2000-0668	49%	Y	
14	d2000-1393	50%	N	Language confusing
15	d2000-0253	51%	Y	
16	d2003-1045	55%	Y	
17	d2003-0697	58%	Y	
18	d2003-0850	67%	Y	
19	d2003-0773	76%	Y	
20	d2001-1028	85%	N	Denied for other reason

Table 1 - Classification of new documents (sorted according to percentile value)

As can be concluded from the table, the results are not perfect. The cases that are in fact lacking the 'Bad Faith' ground have percentile values of 4%, 10%, 15%, 50% and 85%, respectively. This means that with this limited concept (only 21 exemplars and counter exemplars), three out of the five documents without the ground are identified correctly (by means of their low percentile values) and 3 out of 6 documents with percentile values of 15% maximum are indeed cases in which the 'bad faith' ground is not present. This means that only 2 out of 20 documents are classified wrongly and 18 correctly, which is a clear improvement over the a-priori probability

Case d2000-1104 is a very clear one. None of the elements of a violation of the ICANN Policy are established in this case.
In case d2000-0728 the domain name has been registrated in bad faith but there is no proof of *use* in bad faith. For the complaint to succeed both elements, registration as well as use in 'bad faith', have to be proven according to the ICANN Policy. Therefore presence of 'bad faith' as stated by the ICANN Policy is denied and this case is classified as such.
In case d2001-0233 the Panel considers that: 'Maybe the Respondent did register the name in bad faith. But that fact cannot be decided on the papers – only a Court would be able properly on evidence to so decide.' Therefore the concept of bad faith does not comply with the ICANN Policy.

Two cases are predicted wrongly. In case d2000-1393 the language used by the arbiters might be confusing to the program. The decision contains a somewhat unclear double denial. In case d2001-1028 the complaint is denied for other reasons, but with the same result as if bad faith would have been absent.

6. Using this techniques with existing databases

An important question is if these techniques for defining and applying retrieval concepts could already be used with existing legal databases, like for instance Lexis/Nexis or Westlaw. A complicating factor for this is of course that these

databases are commercial products of considerable value, which makes it unlikely that the publishers who own them would supply documents from these databases separate from the existing user interface (containing the existing retrieval functions). Given that situation, our options to facilitate this are somewhat limited.

- One possibility would be to adapt the conceptual retrieval software in such a way that it uses the commercial database in its current form, but hides its user interface. The actual retrieval functions are performed by means of predefined scripts, contained in the conceptual retrieval software. This method is difficult to implement, probably quite slow and error prone, for instance when changes are made to the interface of the commercial database.

- A far better way would be the incorporation of concept definition and classification functions in the existing retrieval software by the publisher. The software would then be able to access the data directly, providing the user with a choice to select the best tool for a given retrieval task.

- An 'in between' option would be that publishers of commercial databases provide their products with 'hooks' or an 'application programming interface' (API) to enable third parties to use the contents directly, without the original user interface. End users could then install different retrieval software – like the applications described here – on their own computer, giving them the possibility to customize their search tools as needed. Many commercial databases already contain such options, although the use of these can be costly.[16]

7. Conclusion

Document retrieval is still a relatively underdeveloped activity in the field of law, while lawyers become increasingly dependant on the information in electronic databases. Statistical techniques like the ones shown here prove to be useful for legal document classification and could lead to improved precision and recall rates when used in the retrieval process. The use of a separate classification program, capable of classifying new documents by using information from existing texts, yields interesting new possibilities. The programs described in this report, although not perfect, can already be used to improve productivity and effectiveness for people with specific information needs. Another possibility would be to adapt them in such a way that they can be used as a front end for commercial databases.

Literature

Jon Bing, 'Designing text retrieval systems for conceptual searching', in: Proceedings of the first international conference on Artificial intelligence and law, Boston, Massachusetts, United States 1987, p. 43 – 51.

Sylvie Despres & Sylvie Szulman, 'Construction of a Legal Ontology from a European Community Legislative Text', in: Thomas F. Gordon (Ed.), Legal Knowledge and Information Systems, proceedings of the 17th Annual Jurix conference, Amsterdam: IOS Press 2004, p. 79-88.

Dick, J.P., 'Representation of Legal Text for Conceptual Retrieval', in: Proceedings of the third international conference on Artificial Intelligence and law, Oxford, England 1991, p. 244 – 253.

Pompeu Casanovas et al., 'Iuriservice II: Ontology Development and Architectural Design', in: Anne v.d. L Gardner et al. (eds.), Proceedings of the 10th International Conference on Artificial Intelligence and Law, New York: ACM 2005, p. 188-194.

Charles Elkan, Naïve Bayesian Learning, San Diego: University of California 1997.

Eyheramendy, S., Lewis, D. D., & Madigan, D. (2003). 'On the naive bayes model for text categorization', in: Bishop, Ch.M & Frey, B.J. (eds.), Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, Fl: Society for AI & Statistics 2003.

Hafner, C. D., An Information Retrieval System Based on a Computer Model of Legal Knowledge, Ph.D. Thesis, The University of Michigan, UMI Research Press: Ann Arbor, MI 1981.

Hafner, C. D. & Berman, D.H., 'The Role of Context in Case-Based Legal Reasoning: Teleological, Temporal, and Procedural', in: AI & Law, Volume 10, September 2002, Kluwer Academic Publishers 2002, p.19-64.

Philip Leith & Amanda Hoey, The Computerised Lawyer, London: Springer 1998.

Lewis, D.D., 'Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval', in: Proceedings of the 10th European Conference on Machine Learning, London: Springer Verlag 1998, p. 4 – 15.

D.V. Lindley, Making Decisions, 2nd edition, London: John Wiley and Sons 1971.

Mitchel, T., Machine Learning, McGraw Hill 1997.

R.V. De Mulder, M.J. van den Hoven & C. Wildemast, 'The Concept of Concept in "Conceptual Legal Information Retrieval"', in: 8th BILETA Conference Pre-proceedings, Warwick: CTI Law Technology Centre 1993, p. 79-92.

R.V. De Mulder & C.J.M. Combrink-Kuiters, 'Is a computer capable of interpreting case law?', in: The Journal of Information, Law and Technology (JILT), Warwick: CTI Law Technology Centre 1996.

Kees van Noortwijk & Richard V. De Mulder, 'The Similarity of Text Documents', in: The Journal of Information, Law and Technology (JILT), Warwick: CTI Law Technology Centre 1997.

G. Salton (ed.), The SMART Retrieval System, Experiments in Automatic Document Processing, Englewood Cliffs N.J.: Prentice Hall 1971.

G. Salton & Ch. Buckley, 'Improving Retrieval Performance by Relevance Feedback', in: Journal of the American Society for Information Science, June 1990, p. 288-297.

C.A.M. Wildemast & R.V. De Mulder, 'Some Design Considerations for a Conceptual Legal Information Retrieval System', in: C.A.F.M. Grütters, J.A.P.J. Breuker, H.J. van den Herik, A.H.J. Schmidt & C.N.J. de Vey Mestdagh (eds.), Legal Knowledge Based Systems: Information Technology and Law, Jurix 1992, Lelystad: Koninklijke Vermande 1992, p. 81-92.

[1] See Wildemast & De Mulder 1992 for an overview of attempts to build such retrieval systems.
[2] Van Noortwijk & De Mulder 1997.
[3] See for example Hafner 1981, Bing 1987 and Dick 1991.
[4] The term 'ontology', according to Wikipedia (http://en.wikipedia.org), is used to indicate an exhaustive conceptual schema of a certain domain, containing all entities and their relationships together with applicable rules. It can be seen as a knowledge model of the domain, to be used for reasoning but also data retrieval purposes.
[5] See for example Casanovas *et al.* 2005, p. 190 and Despres & Szulman 2004, p. 80.
[6] Hafner & Berman (2002, p. 21) for instance mention “open-textured concepts (i.e., legal concepts that do not have clear definitions to determine their applicability, but which depend on experience and common sense, such as the concept of recklessness)”.
[7] See for examples De Mulder *et al.* 1993.
[8] Van Noortwijk & De Mulder 1997, p. 3-9.
[9] A good introduction to Bayesian statistics is given in Lindley 1971.
[10] See for instance Elkan 1997, Mitchel 1997, Lewis 1998 and Eyheramendy *et al.* 2003.
[11] http://arbiter.wipo.int/center/wipo-adr.html
[12] The decisions used in this example can be found at http://arbiter.wipo.int/domains/decisions/index-gtld.html
[13] Internet Corporation For Assigned Names and Numbers (ICANN, 1999). Uniform Domain Name Dispute Resolution Policy. http://www.icann.org/udrp/udrp-policy-24oct99.htm.
[14] The program version shown here has a user interface in Dutch, but international versions are also in preparation.
[15] In fact, the program contains another tool for this purpose. The user can specify certain keywords and sort the documents according to the presence of these. This method is not used here.
[16] Lexis Nexis, for instance, offers a 'Web Services Kit' that seems to have this functionality; see http://www.lexisnexis.com/webserviceskit/.