Retrieving Information in Online Dispute Resolution Platforms: A Hybrid Method

Davide Carneiro
Universidade do Minho
Campus of Gualtar
Braga, Portugal
+351 253 604 437
dcarneiro@di.uminho.pt

Paulo Novais
Universidade do Minho
Campus of Gualtar
Braga, Portugal
+351 253 604 437
pjon@di.uminho.pt

Francisco Andrade
Universidade do Minho
Campus of Gualtar
Braga, Portugal
+351 253 601 817
fandrade@direito.uminho.pt

José Neves
Universidade do Minho
Campus of Gualtar
Braga, Portugal
+351 253 604 466
jneves@di.uminho.pt

ABSTRACT

Information Retrieval is a theme that is so multifaceted as it is its significance to any knowledge-based sphere of influence. This is true in The Law, especially when we judge under the angle of the so-called On-line Dispute Resolution. Indeed, there is the need to analyze and develop efficient information retrieval methods that may improve the course of actions that depend on such techniques. It was under this line of thought that we look at two different methods for information retrieval, and then strengthen its advantages into a third one. The results of this effort are now being applied in UMCourt, an Online Dispute Resolution platform that helps disputant parties and software agents to interact and make their decisions.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval – *relevance feedback, retrieval models, search process, selection process.*

General Terms

Algorithms, Performance, Experimentation.

Keywords

UMCourt, Classification, Association, Case-based Reasoning.

1. INTRODUCTION

Given the growing amount of information we must deal with on a daily basis, efficient information retrieval tools are nowadays essential. In effect, such tools may be generalized to any knowledge-based universe, once the matter of fact is the same. In this context, the legal field is not an exception. Given the considerable amount of information that includes legal norms, legal texts or past cases, information retrieval tools are useful for both law practitioners and disputant parties. If we consider the more specific and recent advent of the so-called Online Dispute Resolution [1] platforms, efficient information retrieval is also essential. In fact, such platforms generally need to provide access, preferably in an autonomous mode, to all sorts of information in order to increase its availability for disputant parties and implement efficient dispute resolution algorithms [6].

ICAIL'11, June 6–10, 2011, Pittsburgh, PA, USA.

In this information we can include the BATNA and the WATNA - Best and Worst Alternative to a Negotiated Agreement, respectively. These concepts denote the best and worst outcomes in a litigation scenario [5]. Moreover, this information should also include the space between the BATNA and the WATNA, which may represent a measure of the risk being taken in accepting or not accepting a proposal. This space is evidently related to the Zone of Possible Agreement proposed by Raiffa [8]. Another important information for a party involved in a dispute is the MLATNA– Most Likely Alternative to a Negotiated Agreement [7], which indicates the region in which an outcome is more likely. Following the same line of thought, it would also be interesting for a party to consider the most and less likely cases. Only being aware of all this information can a party take rational and weighted decisions.

This paper is about implementing contextualized information retrieval methods in such a framework. Specifically, we are looking for efficient ways to retrieve past cases that can be central in a given dispute resolution process. In particular, these results are being applied in the UMCourt platform, addressing three different Portuguese legal domains: Labor Law [2], Consumer Law [3] and Heritages and Divorce's share [4].

2. TWO INFORMATION RETRIEVAL METHODS

Under this setting, in which past cases are retrieved to build knowledge, several issues must be considered. One of them is related with the amount of cases retrieved. In fact, it is expectable to have some kind of control about the amount of cases retrieved, namely: (1) there is no sense in retrieving a huge amount of cases if most of them have a low probability of being considered, and (2) there should be a minimum amount of cases that can provide the needed information for the parties to take rational and informed decisions. Another issue is related with the nature of the ODR platform: these methods must be autonomous, i.e., they cannot depend on human experts at running-time. The two methods presented bellow have this requirements into consideration

2.1 SIMILARITY FUNCTION

Unlike database searches that target a specific value in a record, retrieval of cases from the case-base must be equipped with heuristics that perform partial matches, since in general there is no existing case that exactly matches the new case [9]. In this approach, we are using a nearest neighbor algorithm that is able to compute a value of similarity between two cases by comparing some key characteristics. Cases are then selected according to their value of similarity with the new case: if they are above a

given threshold, they are selected [10].

$$sim = \frac{\sum_{i=1}^{n} W_i * fsim_i(Arg_i^N, Arg_i^R)}{\sum_{i=1}^{n} W_i} \quad (1)$$

In equation 1, the closest neighbor algorithm is shown. In this equation we have:

- n – number of elements to consider to compute the similarity;
- Wi – weight of element i in the overall similarity;
- Fsim – similarity function for element i;
- Arg – arguments for the similarity function representing the values of the element i for the new case and the retrieved case, respectively N and R.

Basically, the similarity function looks at each of the components that characterize a case and assigns it a value of similarity. Each of these values has a given significance for the computation of the overall similarity (e.g. the legal norms used by the parties may be much more important than the dates of occurrence of the two cases). In this algorithm, these weights are, at this moment, determined by a law expert, based on the substance that, according to his/her experience, each of the components of the similarity measure has.

It is now time to detail the information of the case that is considered to be relevant for the computation of the similarity value, i.e., the components. According to our scope of application, we consider three types of information: the objectives stated by each party in the beginning of the dispute, the norms addressed by each party and by the eventual witnesses, and the date of the dispute. Both norms addressed and the objectives are lists of elements, thus the similarity function consists in comparing two lists (equation 2). The similarity is higher when the two lists have a higher percentage of common members. As for the date, the similarity function verifies if the two dates are within a given time range, having a higher similarity when the two dates are closer.

$$fsim_{list} = \frac{|L_N \cap L_R|}{n}, n = \begin{cases} |L_N|, & |L_N| \geq |L_R| \\ |L_R|, & |L_N| < |L_R| \end{cases} \quad (2)$$

Once all the values of the several similarity functions are summed in accordance to their weights, a value of similarity is obtained that describe to which extent a past known case is similar to the new one. By applying this algorithm to each known case, it is possible to select the most similar cases.

The main disadvantage of this approach is that the algorithm requires some computational power and may take some time to perform, depending on the size of the case-base. Another disadvantage is that, for each new case or problem under consideration, all the values of similarity must be computed again as each new case is different from the previous ones. On the opposite, the main advantage is that once all the similarity values are computed for a given new case, it is easy to determine which cases to select: this is done by changing the similarity threshold. This is especially useful for controlling the number of cases that are retrieved. Consider, for example, a case-base that is fairly big. It is likely that under this scenery, a relatively high amount of cases will be retrieved. That is not desirable as it may, for instance, confuse a disputant party or a practitioner that might have requested some cases to compare with his/her own. However, using this approach, retrieving a moderate number of cases is as easy as changing the similarity threshold. That is, if too many cases were retrieved, one could increase the similarity threshold, ensuring that fewer cases will be retrieved, with a higher value of similarity. On the other hand, if too few cases are selected, the disputant party could not be able to get the whole picture. In that sense, it might be useful to decrease the similarity threshold in order to retrieve more cases. Once the similarity values are computed, this process becomes straightforward and results can easily be adapted as needed.

2.2 ASSOCIATION RULES

The aim of this method is to identify relationships between the values of given variables that make up a case. This is a fairly common task in data mining, having a wide range of applications. The main objective is to find hidden patterns that may help to explain or determine some system behavior. The most traditional example is the use of association rules in a supermarket environment to determine the behavior of the customers. Generally, registers of product purchases are analyzed to determine which products are bought, with the aim of better placing the products in the store. From this analysis, rules can be defined that describe such conducts. For example: “seventy percent of the people who buy beer also buy appetizers”. Alternatively, if we think in the legal domain, we can consider rules such as “sixty percent of cases in which norms A and B are used by one party, that party wins” Such rules are stated in the form:

$$if\ X\ then\ Y \quad \text{or} \quad X \Rightarrow Y$$

in which *X* is the antecedent of the rule (“use norms A and B” in the first example) and *Y* is the consequent (“party wins”). In order to support the generation of the rules and select the ones that are essential, statistical aspects can be considered such as the support factor, confidence factor or the expected confidence factor. Generally, only rules that have a confidence factor above a given threshold are considered.

However, there is still the need to determine which of the rules make sense given the domain of application as some rules do not contain any useful information. For example, we could be interested in a rule stating that cases in which norm1 and norm2 are used and the objective of the party is to solve the dispute at all cost, that party wins. The work of selecting the relevant rules is done by a legal practitioner. Although it might be quite an extensive work, it must only be done once for each database of cases. Once this is done, these rules can be used to create categories or classes of cases. Then, cases are assigned to categories according to the rules they comply with.

In order to follow this approach, the information contained in the database about the cases is represented differently, according to the vector space model. This is a fairly simple algebraic model for representing text documents in which instead of using textual fields, a case is represented as a vector. Specifically, in this work, a case is seen as a vector *V* of binary entries, in which each entry *i* < *N* corresponds to a fixed descriptor from the descriptor vector *D* of size *N*. Thus, the value of each binary entry denotes the presence or absence of that descriptor on the case. Descriptors denote important components of a case (e.g. legal norms, objectives of the party, winner of the case). Thus, one can look at a vector which represents a case and, considering the descriptors vector *D*, determine which information is or is not present on the case (Figure 1).

Basically, this representation of a case allows us to see each party’s addressed norms, which are their objectives and which is

the outcome. It is thus a very concise manner of representing all this information, demanding very few resources to handle and to store. Following the same line of thought, a database with *m* cases in which each case is described by *N* descriptors can be represented as an *m*-by-*N* matrix in which each line is a vector representing a case (Figure 2).

Given this data representation, it is possible, as stated before, to apply association procedures to determine relationships between the data. The objective is to create groups of documents, or cases, in which the same rules return a truth-value true. Then, the retrieval process becomes relatively simple, i.e., whenever cases have to be retrieved for a given problem, the system first determines which rules return a truth-value true for the new case. This will allow determining to which category the case belongs to. Then, all the cases of that group can be retrieved, as they are potentially similar and appealing to find a solution to the new problem. Basically, this approach consists in classifying cases using association procedures. The purpose is thus to group the cases in such a way that retrieval will be faster.

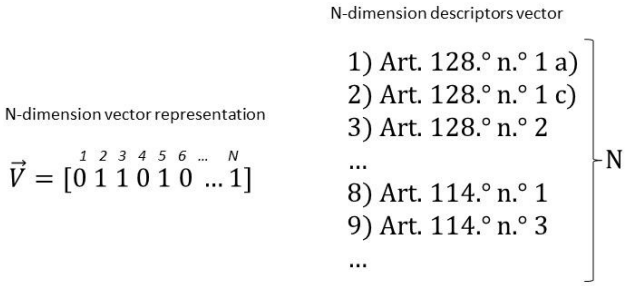

Figure 1. N-dimension vectorial representation of a case.

The main advantage of this method is indeed its effectiveness. Once all the cases of the database are classified, it becomes very easy to retrieve the cases from a given group. However, on the down-side, there is no control on the number of cases that are retrieved. That is, this method cannot actively control the amount

$$DB = \begin{matrix} 1 & 2 & 3 & \ldots & N \end{matrix} \begin{bmatrix} 1 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ \ldots \\ N \end{matrix}$$

Figure 2. Representation of a database of cases as a matrix.

of cases that are retrieved to be presented to the users. In fact, if the new case is classified as belonging to a group that contains hundreds of cases, all those cases will be retrieved, rendering the results nearly useless for a human user.

3. A HYBRID METHOD

Let us now present a method for case retrieval which is intended to combine the advantages of the two previously given methods in order to make it a dynamic, efficient and autonomous one. Recalling, the first method presented had as main advantage the ability to control with precision the amount of cases retrieved. On the other hand, the second method presented had as main advantage a fast retrieval by means of case classification. These are the two advantages that we merge in this hybrid one. Specifically, this method has a preparation phase and two running phases, that is to say pre-select and evaluation (Figure 4). In the pre-select phase, association rules are used to pre-select cases in an efficient way. In the evaluation phase, the system assesses the amount of cases retrieved and may refine the pre-selection by means of similarity functions.

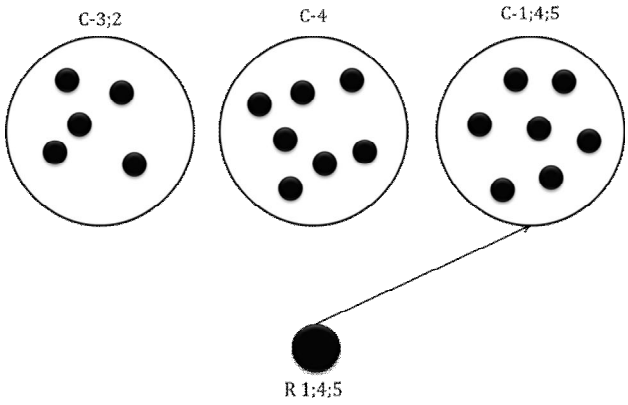

Figure 3. To pre-select cases, the system looks at the rules that are true in the new case and selects cases in which the same rules are also true.

In the preparation phase, association rules are discovered in the database. Then, a human expert determines which of these rules are to be considered and which ones are to be discarded. Once this task is finished, the system looks at all the cases in the database and classifies them according to the rules that they authenticate. This will organize the cases into groups or categories. As a consequence, each case in a category is more similar to other cases in the same category than it is to outer ones. Away from here, the information compiled is ready to be used by the system to retrieve cases.

This process starts with the pre-select phase. This phase makes use of the previously mentioned method that relies on association rules to select cases. In that sense, it starts by determining which rules return a truth-value true with the new case. Having done that, the system determines to which category or group the new case belongs. This phase provides as output all the cases that belong to that same category. Figure 3 considers a scenario in which in the new case the rules 1, 4 and 5 return a truth-value true. In this example, all the cases from category C-1;4;5 would be pre-selected. Up to this point, this process is quite efficient.

In the second phase, the evaluation one, the system analyzes the results of the pre-selection and determines if further actions are needed. If the amount of the retrieved cases is inside the desirable range (this depends on the request or target user), the process ends and all the cases pre-selected are retrieved. On the other hand, the amount of cases pre-selected may not be the desirable one. Now, two scenarios are possible.

In the first one, there are few cases selected. In such a scenario, the system will try to relax the pre-selection rules. Still considering the previous example of Figure 3, let us now assume that category C-1;4;5 does not exist. Under this assumption, no case would have been pre-selected as no group matched the same rules of the new case. In that sense, the system would relax the pre-selection rules by looking at the rules individually. Thus,

under this assumption, the cases from category C-4 would be pre-selected, given that in all of them rule 4 returns the truth-value true, like in the new case. This would ensure the needed amount of cases, although their similarity would also be lower.

In the second scenario, too many cases are selected. In this event, a fine-tuning of the pre-selection must be performed in order to select fewer cases. However, this fine-tuning must be performed under the requirement that the less similar cases are discarded in favor of the most similar ones. In this scenario, this method makes use of a similarity function to decide on which cases to discard and which ones to consider. This will associate each pre-selected case with a similarity value. Away from here, the system only has to change the similarity threshold in order to change the amount of cases retrieved and their similarity value. In that sense, one of two different approaches may be selected.

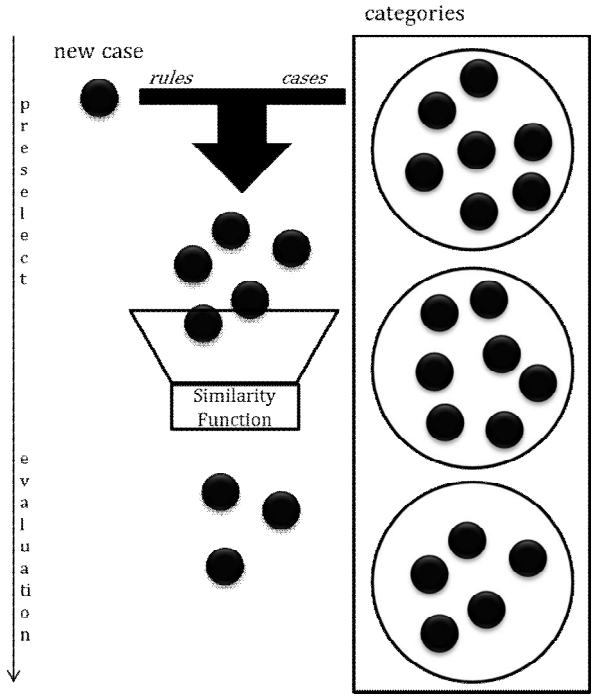

Figure 4. This hybrid method uses classification rules to make a pre-selection of cases and a similarity function to decide, among the cases inside a category, which ones are more relevant.

On the one hand, the system can make use of the similarity function presented before. This method is straightforward and consists on the application of the algorithm described. Nevertheless, it involves retrieving the cases from the database in their original form in order for the algorithm to be applied. However, as the indexes of cases are already known from the pre-selection phase, the process is relatively fast. In this algorithm, output values range from 0 to 1, with 0 meaning that there is no similarity at all and 1 denoting an exact match.

A faster approach, however, would be to use the vectorial representation of the cases, which is available from the pre-selection phase, to compute the similarity. This can be done by means of the cosine similarity. In fact, the similarity between two vectors can be determined by finding the cosine of the angle between them. Given two vectors of attributes A and B, with N entries each, the cosine similarity, θ, is determined as shown in equation 3. Given the definition of data as vectors of binary entries described previously, the cosine similarity of two cases will range from 0 to 1, i.e., the angle between two vectors cannot be greater than 90°.

This second method of computing similarity is quite simpler and faster as it uses the vectors of binary entries. However, contrary to the previous similarity function, it does now allow to assign weights to the several components of the case. This may or may not be a disadvantage, depending on the scope of the application.

$$sim = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{N} A_i * B_i}{\sqrt[2]{\sum_{i=1}^{N} (A_i)^2} * \sqrt[2]{\sum_{i=1}^{N} (B_i)^2}} \qquad (3)$$

4. RESULTS

The method presented in this paper allows for the retrieval of past cases in a wide range of applications. Specifically, the information retrieved can be presented to the parties, in order for them to take more informed decisions. Moreover, it can be presented to a mediator or arbitrator, which will analyze the past cases in order to take better decisions, supported by previous occurrences. Alternatively, the information can be forwarded to a software agent in order to perform some additional operations. An example of this is depicted in Figure 5.

In this example, a software agent used the information provided by the system to build a graphical representation with some added value, to show to a disputant party. This consists in retrieving all the relevant cases similar to the one of the disputant party as well as their similarity values. The software agent is able to compute a value of utility for each case. Merging this information with the similarity values, it is possible to draw a graphical representation of the information retrieved, as it is shown in Figure 6, in which each circle represents a case. Moreover, the software agent can also group the cases in clusters, according to their significance, allowing a faster understanding of the situation from the part of the user. For each cluster, the mean values of similarity and utility are also shown. Looking at this representation, the user is also able to see the BATNA and the WATNA, as well as the MLATNA, represented by the green section of the graphic. Therefore, this information retrieval method results in an efficient ant multifaceted one, enabling the implementation of very different functionalities that can considerably improve the experience of people involved in dispute resolution processes.

5. CONCLUSION AND FUTURE WORK

In this paper we have presented a hybrid method for retrieving information in the context of an Online Dispute Resolution platform. The main functionality implemented is to retrieve past cases that are significant for the solution of a given problem, typically a dispute resolution one. In that sense, this work is integrated into the UMCourt ODR platform, enabling a wide range of applications to be implemented.

Future work will be aimed at increasing the adaptability of this method. Basically, we will increase the actions that can be performed to better retrieve cases. Specifically, we are now adding a hierarchical component to this method. This is supported by the Portuguese concept of legal norm, which is organized as follows: a Law is organized into chapters and articles, which in turn can have one or more numbers and each number can have

Figure 5. A detail of the graphical representation of the information retrieved.

one or more items. In that sense, when one party addresses a norm, we can look at the level of the article, to know the high level subject being addressed, or we can look at the level of the number of the item and get to know the specific issues. As an example, article 128.° of the Law n.º7/2009 (Portuguese law) is inserted in chapter "Rights, duties and warranties of the parties", and it deals with the obligations of the worker. Thus, *obligations of the worker* is the general concept addressed. However, each item inside this article addresses specific issues. As an example, no. 1 a) states that the employee should be respectful to his/her superiors and to the remaining co-workers, contributing to the maintenance of a cooperative and healthy working environment.

Indeed, our approach consists in creating the association rules on different hierarchical levels. Thus, rules created considering the numbers and items of each number will be more specific and precise, but less frequent. On the other hand, rules created considering only the article of each norm addressed will be more general and thus more frequent. Therefore, at run-time, the system will be able to drill-down or drill-up on the categories in order to better adjust the pre-selection of the cases.

6. ACKNOWLEDGMENTS

The work described in this paper is included in TIARAC - *Telematics and Artificial Intelligence in Alternative Conflict Resolution Project* (PTDC/JUR/71354/2006), which is a research project supported by FCT (Science & Technology Foundation), Portugal.

7. REFERENCES

[1] Katsch, E. and Rifkin, J. 2001. Online dispute resolution – resolving conflicts in cyberspace. Jossey-Bass Wiley Company, San Francisco.

[2] Carneiro, D., Novais, P., Andrade, F., Zeleznikow, J. and Neves, J. 2010. Using Case-based Reasoning to Support Alternative Dispute Resolution. In *Distributed Computing and Artificial Intelligence*, Carvalho, A., Rod-González, S., Paz, J. and Corchado, J. M., Eds. Proceedings of the 7th International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2010), Valencia, Spain), Springer - Series Advances in Intelligent and Soft Computing, vol. 79, ISBN: 978-3-642-14882-8, 123-130.

[3] Costa, N., Carneiro, D., Novais, P., Barbieri, D., Andrade F. 2010. An Agent-Based Approach to Consumer´s Law Dispute Resolution. In *Proceedings of the 12th International Conference on Enterprise Information Systems*, (ICEIS 2010, Madeira, Portugal, 8th -12th June 2010), ISBN: 978-989-8425-05-8, 103-110.

[4] Café, A., Carneiro, D., Andrade, F., Novais P. 2010. Online Dispute Resolution for Property Share – Divorce and Hereditary. Proceedings of the INFORUM - 2º Simpósio de Informática, Barbosa, L. and Correia, M. Eds, Braga, Portugal, ISBN 978-989-96863-0-4, 779-790. *(in portuguese)*

[5] Notini, J. 2005. Effective Alternatives Analysis In Mediation: "BATNA/WATNA" Analysis Demystified. Available in http://www.mediate.com/articles/notini1.cfm, accessed January, 2011.

[6] Fisher, R., Ury, W. 1981. Getting To Yes: Negotiating Agreement Without Giving In. Boston: Houghton Mifflin.

[7] Steenbergen, W. 2005. Rationalizing Dispute Resolution: From best alternative to the most likely one. In *Proceedings of the 3rd ODR workshop*, Brussels.

[8] Raiffa, H. 1982. The art and science of negotiation: how to resolve conflicts and get the best out of bargaining. Cambridge, The Belknap Press of Harvard University Press.

[9] Watson, I., Marir, F. 1994. Case-based reasoning: A Review. In *Knowledge Engineering Review*, vol. 9, 327–354.

[10] Ashley, K. 1991. Modeling Legal Arguments: Reasoning with Cases and Hypotheticals, MIT Press, Cambridge, MA.