Extended conceptual retrieval

Radboud Winkels, Doeko Bosscher, Alexander Boer, Rinke Hoekstra
Department of Computer Science & Law,
University of Amsterdam, The Netherlands

Abstract. In the ESPRIT project CLIME (25.414) we are building a "Legal Information Server" (LIS), an advanced legal information retrieval system. Part of the LIS is a module for what we call "extended conceptual retrieval". In CLIME we elaborate on the notion of CR in several ways. We describe the principles of extended CR, provide data on the quite extensive model we have built so far, and present first evaluation results.

1. Introduction

Access to electronic legal sources – on the internet and on CD-ROMs – is typically handled in the same way as information retrieval from documents in general (e.g. [1]). In traditional text retrieval, documents are indexed by the literal strings that appear in them, and searching is based on string matching techniques, typically enhanced with a query language based on boolean and proximity connectives to combine search strings. When the strings in the search expression match the strings in the document, the document is assumed to be relevant and is returned.

For legal problems, key word matching is limiting, irrespective of the query language, because it requires the user to know the right phrases in the law to be able to use it successfully. Moreover, the quantity and quality of the search result leaves much to be desired. In conceptual retrieval (CR), the idea is that documents are indexed by the concepts denoted by strings rather than the strings themselves. In this case, the retrieval mechanism is intended to return those documents that refer to the concepts in the search expression (cf. [2]). If these concepts are somehow related to each other, typically in type hierarchies, documents can be returned that refer to more abstract concepts than the one a user asks for (e.g. [3]). Although this idea successfully extends retrieval to implied concepts, it is not at all a real problem solving tool; It returns legal information conceptually related to the question, while the user is actually interested in legal information related to the answer.

The alternative approach to legal information systems explicitly aims to assist the user in solving "legal problems". The layman's conceptual view on law in the continental tradition is quite straightforward; It divides legal knowledge into two orthogonal types: rules and cases. The pivotal legal problem in this view is to assess whether a case complies with (or deviates from) the rules. Rules are typically interpreted as deontic sentences and cases as situations or events potentially regulated by rules. This perspective explains i.a. the attention in AI & Law for modal deontic logics, and, more recently, generic and reusable task models and ontologies for legal assessment [4,5]. The latter approach, coined Legal Information Serving by us [6], aims to provide KADS-like blueprints for the development of advanced legal information retrieval systems. In the CLIME project, of which this paper is a result, we develop a Legal Information Server (LIS) using such a blueprint [7].

Radboud Winkels, Doeko Bosscher, Alexander Boer, Rinke Hoekstra, 'Extended conceptual retrieval', in: Joost Breuker, Ronald Leenes and Radboud Winkels (eds.), *Legal Knowledge and Information Systems. Jurix 2000: The Thirteenth Annual Conference.* Amsterdam: IOS Press, 2000, pp. 85-97.

The Legal Information Server contains several software modules to assist the user with the assessment of a (real or hypothetical) situation or event. The assistance that will be offered covers the formulation of a case [8], selection of relevant rules from a large corpus of legal sources, application of those rules to the situation or event at hand, and the resolution of the inevitable apparent conflicts that will arise between rules [9]. The very goal-driven nature of the overall task and restrictive representation of legal knowledge make such a system computationally feasible. The modelling effort – and expertise – required to produce adequate representations for a true reasoning approach to legal assessment, however, are prohibitive for large-scale application in many domains of law.

This paper presents a simpler rule selection module, the extended conceptual retrieval module, that can be used to select rules from partially and superficially modelled legal sources. This module is conceptually closer to the Information Retrieval perspective on developing legal information retrieval systems, but uses a more expressive concept and query language. All methods and techniques developed in CLIME are incorporated in the MILE demonstrator and tested in the paralegal domain of ship classification, a requirement for access to ports, insurance etc. Every ship classification society maintains a set of rules for assessing ships, and assesses compliance with national, international (e.g. IMO), and supranational (e.g. EU) rules. The MILE system will provide better access to this large set of rules, i.a. using the conceptual retrieval techniques described in this paper.

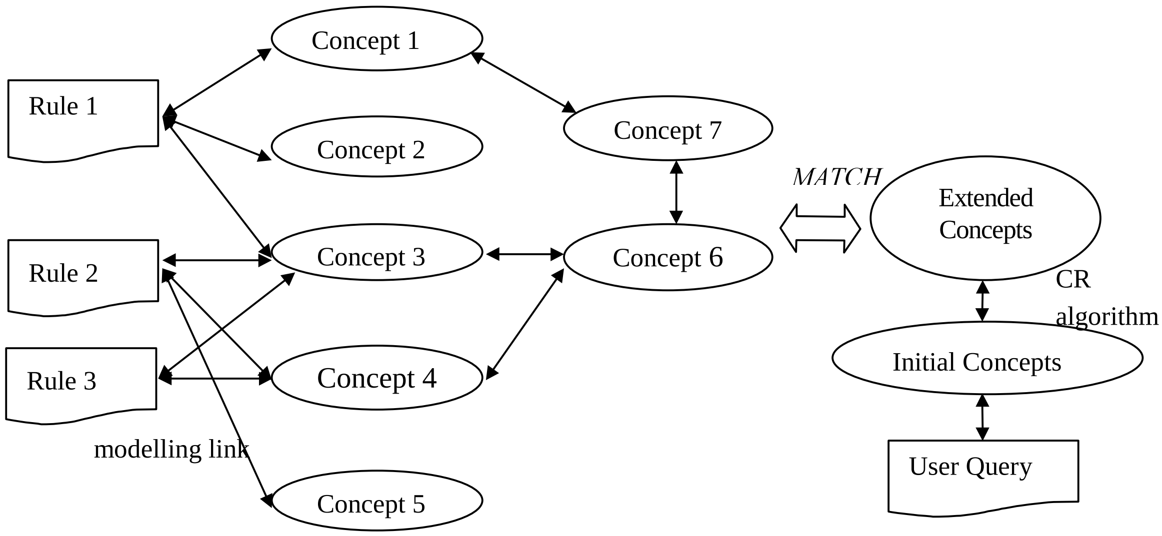


Figure 1: Extended Conceptual Retrieval

2. An Example

We will start with an example: Suppose a user requires all rules that apply to bulk carriers. In text retrieval he might search for the literal string ‘bulk carrier’ and all rules which contain this literal string will be returned. A rule that contains the string ‘bulk-carrier’ will not, neither will a rule that mentions ‘carriers that transport large quantities of goods’ (or whatever a description of a bulk carrier may look like). The first oversight is easily repaired by the user or by building it into the machine (if foreseen), but the second one is already much harder. It is infeasible to come up with all the ways rules may refer to the particular concept one is interested in. The problem is even more complicated than this. Not only the rules that explicitly refer to bulk carriers, in whatever way mentioned, are of importance, also rules that apply to cargo ships, and ships in general are relevant. Bulk carriers are a specific type of cargo ship, so unless the contrary is specifically stated, all rules that apply to cargo ships also apply to bulk carriers. The same applies to rules about ships in general.

Given the nature of the problems tackled by legal information serving, users are also interested in rules that apply to concepts more specific than the one they submit to the system. The user asking about rules that apply to bulk carriers is also helped with rules that are about specific types of bulk carriers – carriers that carry a flammable liquid for instance – that constitute potential exceptions to more general rules (cf. [9]). And the user may not realise this.

Besides the 'isa' type of relation between the search concept and other concepts, the user is also helped with rules that contain related concepts other than more general or more specific ones, i.e. by other than 'isa' relations. An example is that rules about the 'hold' of a (cargo) ship are of importance, because cargo ships 'consist of' or 'contain' one or more holds. This is an extension of how existing approaches to conceptual retrieval work; these only consider type hierarchies or unlabelled (untyped) associative or statistical links (e.g. [3, 10, 11, 12]).

We extend this existing notion of CR in several ways and will discuss this in the next sections. Furthermore, we will provide some data on the domain model we have built so far and present a first evaluation of CR results. We will end with a discussion and some conclusions.

3. Extended Conceptual Retrieval

The process of conceptual retrieval consists of three parts. The first part extends an initial set of concepts from the user's query, using a search expression and the conceptual representation of the legal sources. Output is an extended set of concepts. The second part links these concepts to the original sources. The third part orders and filters the set of links (cf. Figure 1).

The first part of conceptual retrieval is concerned with finding concepts related to those in the user's query, using a search expression and a conceptual representation of the domain. The representation of the content of legislation in the Legal Knowledge Repository (LKR) is explicitly linked to the sources in the Legal Source Repository (LSR) by so called modelling links. There are two types of modelling links: *Definitional links*, and *referential links* [4]. 'Definitional links' keep track of where the information used for modelling a certain element from the knowledge base came from. 'Referential links' are links that keep track of multiple references to the same concept which use a definition made elsewhere. The easiest way to imagine the LKR is to see it as a labelled multi-graph. The graph contains nodes, which are the concepts, such as 'ship', or 'bulk carrier' and edges between the nodes, which are annotated with a relation such as 'part-of'. We currently use the relations presented in Table 1. The concepts in the input of CR form a (small) subset of concepts present in the LKR. The search expression is given as a *finite state automaton* (FSA), where the transitions in the automaton are annotated with a number of relations from the LKR (see Figure 3 for a simple example). The conceptual retrieval algorithm extends the initial set of concepts using the finite automaton. For each concept an *extension* is computed using the transitions of the automaton. The result is an extended set of concepts, a larger subset of all concepts in the LKR. The algorithm is designed so that there may be cycles in the finite automaton (e.g. to express transitive closure of relation such as *has-supertype*) or in the LKR graphs. Using the FSA on the LKR may lead to dead states, i.e. states from which it is impossible either to reach another state or to go beyond a cluster of states. Again, the algorithm is designed to handle this correctly.

Table 1: Domain Relations

Relation	Inverse Relation
HAS-SUBTYPE	HAS-SUPERTYPE
DESCRIBES	DESCRIBED-BY
HAS-PART	PART-OF
CONNECTED-TO	CONNECTED-TO
EXCLUSIVE	EXCLUSIVE
CAUSES	CAUSED-BY
OBSERVABLE	OBSERVABLE-OF
MEASURABLE	MEASURABLE-OF
RELATED-TO	RELATED-TO

A CLIME user still presents his query as a case, just like in normative assessment. Only now the question is not, is this case allowed or not (or what norms are violated given this case), but "What are the relevant norms?" (see Figure 2). This way, the query provides an initial set of concepts that will be extended by the Conceptual Retrieval algorithm according to the specification of the FSA. For most users, formulating FSAs will be too difficult. Therefore, we build them into the system, but what are sensible FSAs?

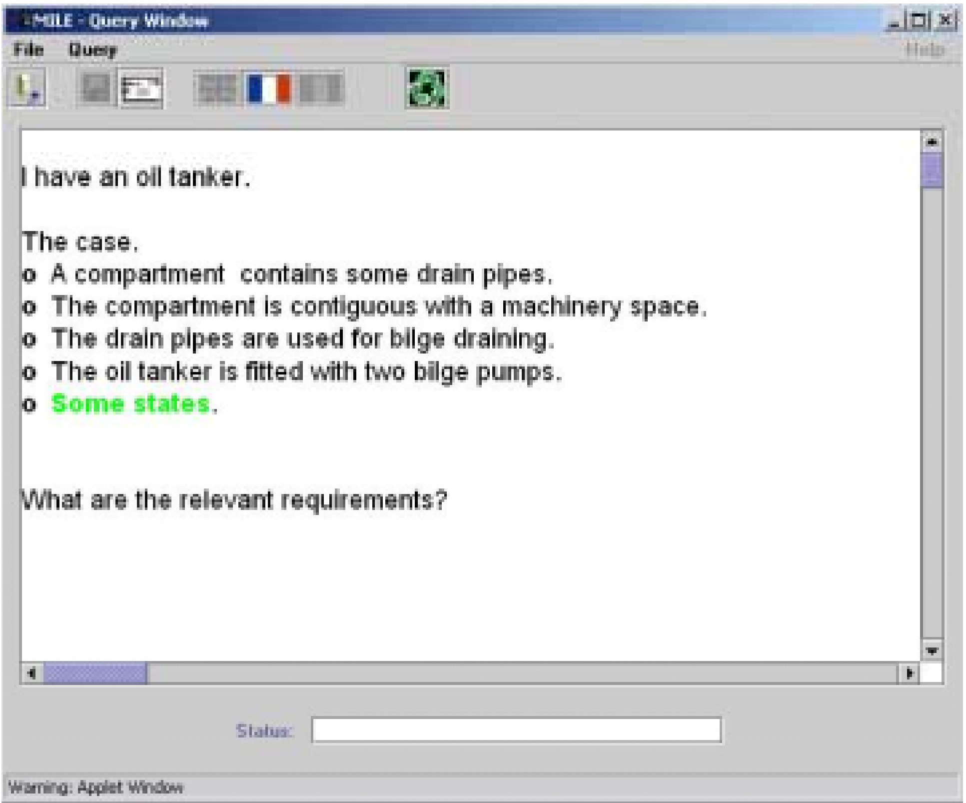


Figure 2: Formulating input for CR as a case

3.1 Formulating FSAs

The CR query system is a generalisation of typical search techniques used for search engines and more experimental retrieval techniques. The expressiveness of finite automata languages has been characterised as a "programming language without variables"; an apt characterisation if one wants to understand both the flexibility and the limitations of finite automata.

Let's assume that the question of interest is the following: "I have an oil tanker. Part of it is a fire extinguishing system. What requirements are relevant to this fire extinguishing system?". This query contains the concepts OIL-TANKER and FIRE-EXTINGUISHING-SYSTEM. Both concepts are used to compose a CR query, but they clearly have different roles in the query. In principle one can use a very complex automaton combining all features in the LKR for all questions. For some concepts, like types of ship, this is prohibitive because it returns too many rules, but it is important to realise that this is not a flaw in the system: If someone asks: "I have an oil tanker. What requirements are relevant?" he is asking for trouble. If the oil tanker is supplied as a detail, but the *focus* of the question is on another concept like a fire extinguishing system, then a detailed automaton (using all relevant relations like the one in Figure 4) is used for the concept in focus, and a simple super-type automaton is used for the details (cf. Figure 3). Application of a set intersection method on the results makes sure that the most relevant rules returned are relevant to the focus of the question, not just some minor fact of the case.

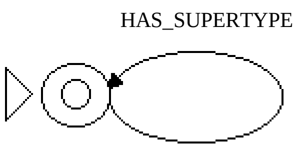


Figure 3: A simple FSA

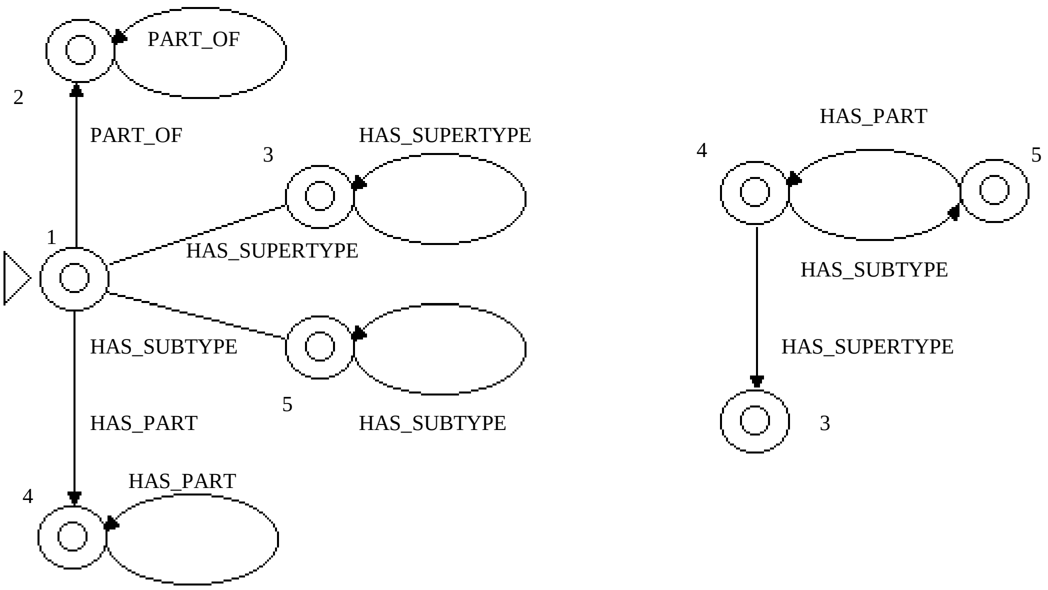


Figure 4: A complex FSA (for reasons of space split into two parts)

3.2 Output of CR

Output of CR is a set of (references to) relevant rules with a 'trace' of how concepts in the user's query are related to the concepts referred to in those rules (i.e. the concepts in the extended set). An example can be seen in Figure 6.

In practice, CR may cause a large number of 'hits', even more than plain text retrieval through keyword matching. In one sense this is a good thing, since it will increase *recall*;

i.e. return more (or most) relevant rules. On the other hand, the user may no longer see the forest for the trees. There are several ways we can support the user:

1. relevance ranking
2. advance ordering
3. filtering

The first two only affect the order in which the CR results are presented to the user, the third affects the amount of results presented. We will discuss each of these solutions in turn.

Relevance Ranking

One parameter in relevance ranking is the size of the intersection of the number of concepts the rules refer to and the concepts in the user's query (either the initial set of the extended set). Let us take an example for the representation in Figure 1. Concept 3 and 4 are in the user's initial set. CR extends this set with Concept 6, 7 and 1. Rules 1, 2 and 3 are returned in the extended set. Rule 1 refers to concepts 1 and 3, rule 2 and rule 3 to concepts 3 and 4. We can say these last two rules may be more relevant for the user since they refer to (all) two concepts from the user's query, while rule 1 only refers to one concept from the initial query. In the case that rule 2 also referred to concept 1, it would be the most relevant rule.

Another parameter is the relevance of concepts referred to 'in the domain'; i.e. rules that are about *central* concepts could be considered more or less relevant than rules about more peripheral concepts. How do we know what concepts are more central or relevant in a domain? One way would be to ask domain experts, but first we would like to find ways of computing (potential) centrality or relevance automatically. Domain experts can later judge the results from this computation.

When thinking about centrality or relevance of domain concepts, two viewpoints emerge:

- A central concept will be linked to many other concepts in the domain
- A central concept will be referred to in many rules that govern the domain

We decided to introduce two different measures as attributes of domain concepts to capture these two notions:

1. *Centrality*: the number of relations a concept has with other concepts in the domain. The more relations, the more central the concept is.
2. *Reference Count*: the number of modelling links a concept has with rules about the domain.

Both measures can be computed at modelling time, they are *static* relevance measures, unlike the size of the intersection of the set of concepts a rule refers to and the set of the initial query discussed above; that is a *dynamic* measure.

It remains to be seen whether rules about 'central' concepts are more or less relevant for MILE-users. On the one hand rules about central concepts are more important, since the concepts are more important. On the other hand, those rules are probably well known and too general for a specific user query. We start from the assumption that more specific rules, about less central concepts, will be considered more relevant in most cases. So there is an *inverse* relation between the relevance of norms and the centrality and reference count of the concepts they refer to.

Central concepts are probably also central in the hierarchy of concepts in the LKR. Concepts at the top of the domain ontology are not referenced at all in norms (concepts like 'object', or 'agent'), concepts at the bottom of hierarchies are only mentioned in few rules (concepts like 'still water bending moment'). The core of the domain ontology of a particular application domain of CLIME, in this case the domain of ship classification, reflects the level at which domain experts like those people who draft the regulations view the domain (cf. Figure 5).

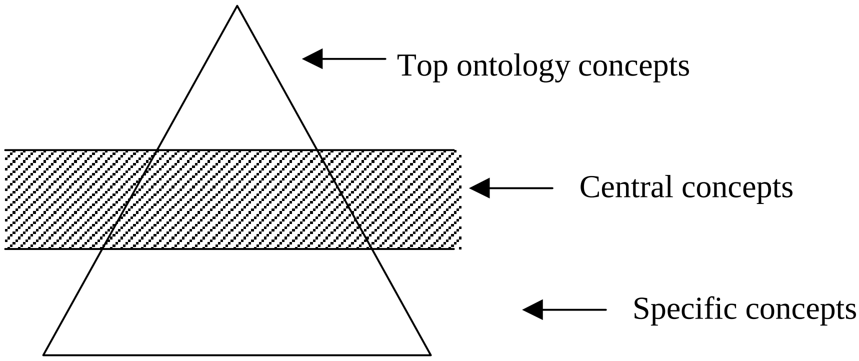


Figure 5: Hierarchy of domain concepts

Advance Ordering

Another way of supporting the user in dealing with the set of potentially relevant rules that CR returns is by organising the rules in advance in some way. We can for instance distinguish subsets of rules on the basis of the concepts they refer to, e.g. separate rules that are about cargo ships from rules that are about passenger ships. This would enable us to present things like: "We have found 25 rules; 12 are about cargo ships, 13 are about passenger ships.", and for instance let the user decide which subset of rules he or she wants to access first.

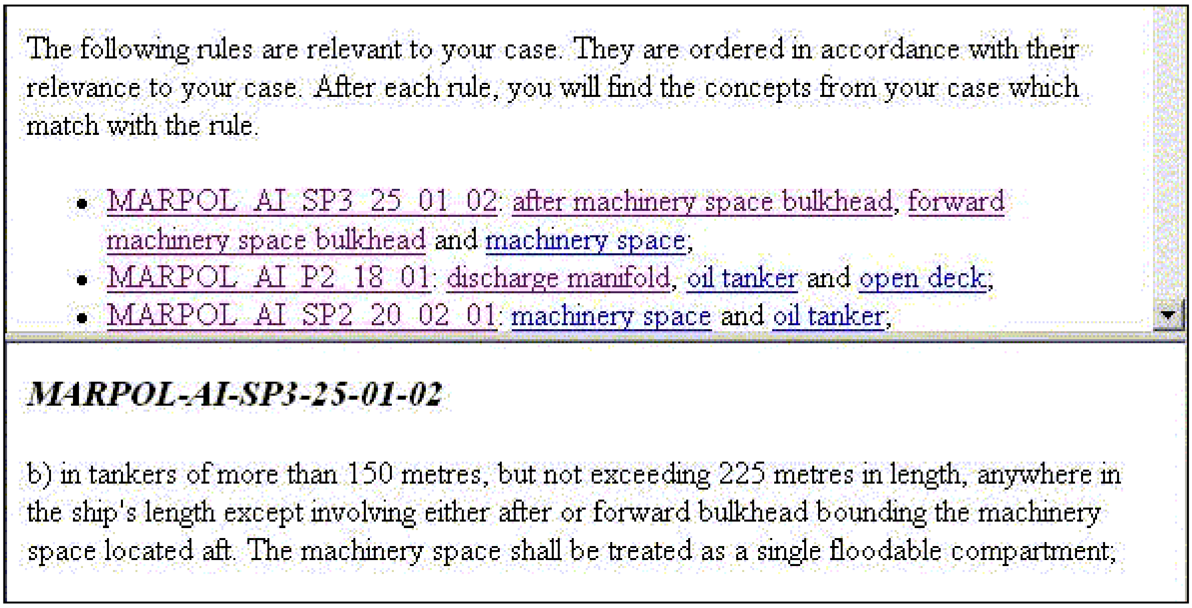
Advance ordering has to do with the pragmatics of discourse; it is not related to the relevance of CR results.

Filtering

Finally, the ordered set of relevant norms can be filtered on a user's request based on the values of attributes of:

1. *concepts*: the notion of centrality and reference count
2. *modelling links*: the author/creator of a link between a concept and the source
3. *norms*: issuing date, source, position in hierarchy of norms, domain specific attributes (like life cycle phase of a ship in our target domain; one of design, construction, in service, repair)

The last type of filter is the most important.


The following rules are relevant to your case. They are ordered in accordance with their relevance to your case. After each rule, you will find the concepts from your case which match with the rule.

- MARPOL_AI_SP3_25_01_02: after machinery space bulkhead, forward machinery space bulkhead and machinery space;
- MARPOL_AI_P2_18_01: discharge manifold, oil tanker and open deck;
- MARPOL_AI_SP2_20_02_01: machinery space and oil tanker;

MARPOL-AI-SP3-25-01-02

b) in tankers of more than 150 metres, but not exceeding 225 metres in length, anywhere in the ship's length except involving either after or forward bulkhead bounding the machinery space located aft. The machinery space shall be treated as a single floodable compartment;

Figure 6: Presenting output of CR

3.3 *Additional functionality for Explanation*

The basic output of CR contains: All the concepts from the user's initial set; the concepts of the extended set; all links to rules that refer to one or more of the concepts in either set; plus *a* path from a concept in the initial set to a concept in the extended set using the relations specified in the FSA. Take the example from Figure 7. Suppose the user's initial set contains the concept C1. The FSA extends it to concepts C2-C5 using relations 1 and 2 (solid black and dotted lines). Now the user wants to know why concept C5 is relevant. The trace in the CR output will give the path C1-C2-C4-C5. As it turns out, there is a different relation in the LKR that immediately connects C1 and C5 (the grey line in Figure 7). Obviously, that single step may sometimes be more convenient for the user. In other cases, we may want to use different paths to explain the connection between concepts for different reasons. These paths may in fact be longer than the one(s) given in the CR output, but may for instance use relations that are easier to explain. It may also be necessary to explain relations between concepts irrespective of previous CR queries.

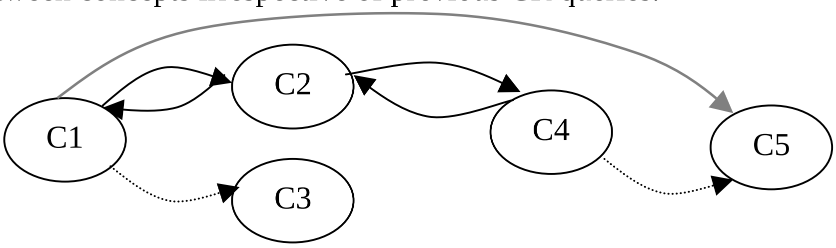


Figure 7: Example CR path

For these purposes we provide a so-called *minimal path* function. It computes linear paths between an input node and an output node in the LKR using all possible relations. Note that although the name seems to suggest otherwise, the paths computed are not (only) the shortest paths; they are paths that do not contain loops. We will describe the algorithm with the aid of the example graph in Figure 8. Minimal paths should be computed between input node Cin and output node Cout. First we follow all outgoing edges from Cin and collect the connecting nodes: In this case only C1. Should Cout be in the collection, we have found a minimal path. Then we do the same for Cout and find C5. Next we proceed with the nodes we found from Cin and find C2 and C5. We have seen that last node before, it is a connecting node and we have found a ‘minimal path’: Cin – C1 – C5 – Cout. In the same way we find C1 and C4 from C5; C3 from C2; C3 from C4 and we have found a second ‘minimal path’: Cin – C1 – C2 – C3 – C4 – C5 – Cout. In this case there are no more paths to be found.

As one can easily see, this algorithm is computationally expensive, so we use a ‘maximal depth’ parameter that cuts of search at a certain depth.

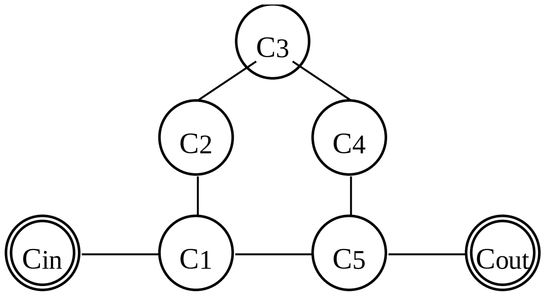


Figure 8: Minimal Path Example

Please note that the minimal path algorithm treats all relations in the LKR the same. It will be up to other modules to decide which paths make more sense then others. As already stated above, it is not always the shortest path that makes the most sense. An example of the use of these minimal paths in explanation for the case in Figure 2 and answer in Figure 6, can be seen in Figure 9.

4. The Demonstrator MILE

The set of rules of the classification society in CLIME comprises an estimated 15,000 articles. An article varies from a few sentences to several paragraphs, sometimes with tables and figures. This means we are dealing with several tenths of thousands of norms. It was clear from the start that it is not feasible to model all these norms within the life span of the project, given the effort we can spend on it, even using the Legal Encoding Tools (LET) we are building simultaneously. We therefore decided to focus on a subset of about 25% of the rules.

after machinery space bulkhead

The concept *after machinery space bulkhead* is related as follows to the concept *compartment*, which was part of the case that you submitted:
A compartment is a space, a space can be a machinery space and a machinery space has as a part an after machinery space bulkhead.

There is 1 connection of the concept *after machinery space bulkhead* to other concepts.

Figure 9: Example use of minimal path in explanation

Of this subset we have now modelled a third, resulting in a knowledge base of ca. 3000 concepts and some 10,000 relations of the types introduced earlier, between them. These concepts will most likely also occur in the remainder of the rules. An interesting question is how much *coverage* of the entire domain already has been achieved with the modelling done so far. In theory it is possible that we have all the concepts we need and that we only need to add the reference links between these concepts and the remainder of the BV rules. In that case we would already have full coverage. The other extreme would be that we exactly cover the part we have modelled, but no rules outside of this scope. Both extremes are not very likely, but where are we in between?

We can use the 'automatic concept recognition' facility of the LET to scan the entire source for known concepts, and, if desired, link these automatically to the text elements in the source. If we do that for the whole set of rules, we find 94,067 links for 14,837 articles in the legal source, an average of about 6,34 links per article. Given the estimate of about 15,000 requirements in the BV rules, this means that most likely all elements in the BV rules can already be accessed using only the concepts identified thusfar. We can check whether this is indeed the case by looking whether they all have at least one reference link to a concept in the LKR. This minimal coverage can then be defined as:

$$\textit{Minimal coverage} = \frac{\text{number of elements in LSR with RefLink}}{\text{number of elements in LSR}} \times 100\%$$

At the moment this minimal coverage is 92%.

A better notion of coverage would be the percentage of all concepts and relations in the domain we have modelled so far. For that we need a good estimate of the total number of concepts in the domain, or the average number of concepts in a rule. The first estimate is not easy to get; the problem with the second is that the variation in length of rules is considerable. Moreover, concepts may be hidden in tables and figures.

We have discussed the notion of 'centrality' of concepts, i.e. the number of relations a concept has with other relations. The most central concepts at present are given in Table 2. The top four (plus 'metric') are concepts from the top-ontology that is given in the encoding tools. The others are clearly from the ship classification domain.

Table 2: Top 10 Central Concepts

Concept	Nr of Relations
ACTION	60
SYSTEM	48
STATE	40
ENTITY	36
EXAMINATION	34
METRIC	31
SHIP	30
DOCUMENTATION	29
ADDITIONAL-CLASS-NOTATION	28
SURVEY	26

We can also have a look at the average number of reference links per concept (32,08 at the moment). Of course, not all concepts will have the same amount of references to it from the rules. Some concepts will be referenced a lot, others only in a specific section or a few requirements. This is what we have called the *reference count* of concepts (see above). Table 3 gives a listing of the ten most referenced concepts in the present LKR, for the manually linked version and the automatically linked version respectively. The concepts that appear in both top 10's are no real surprise, which is a good sign. The concepts of ‘ship’ and ‘society’ appear in both top 10’s and are obviously central in the domain of the BV rules. A concept like ‘surveyor’ or ‘special survey’ appears in the top 10 of the manually linked LKR because it plays a central role in the chapters we modelled, but not so much in the remainder of the rules. Therefore they drop from the top 10 of the automatically linked knowledge base. The concept of ‘section’ in Table 3 is probably an artefact the automatic linking; it is not likely that so many rules in fact are *about* the concept of section. More likely, the refer to a particular section.

Table 3: Top 10 of referenced concepts

Concept	Nr of links manually	Concept	Nr of links automatically
SHIP	227	SECTION	1127
SOCIETY	130	THICKNESS	960
SURVEYOR	85	CARGO	939
REQUIREMENT	71	SOCIETY	911
SERVICE-NOTATION	69	DECK	867
CLASS	56	SHIP	776
RULE	52	PLATING	772
SURVEY	50	REQUIRED	749
OWNER	49	ARTICLE	721
SPECIAL-SURVEY	45	SIDE	712

5. A First Evaluation of MILE

An extensive evaluation of the MILE system with end-users is planned for the near future. A problem is that since we have only linked part of the entire rule set, it is difficult for domain experts to evaluate the outcome of CR. If an expected rule is not returned for a CR query, is that due to a fault in the model, in the query (FSA), in the CR algorithm, or because it has not been modelled yet? Therefore we are in the process of completely modelling and linking a different and smaller set of rules for the same domain, that of the international MARPOL convention, concerning the prevention of maritime pollution (an estimated 500 rules).

As a first small and formative evaluation of MILE with the classification society set of rules, we took a set of 20 test queries that were taken from actual queries presented to domain experts in their daily activities. We tried to formulate FSAs for them and compared the CR results with the answers provided by the experts.

In 18 out of the 20 cases the relevant rule(s) given by the experts were among the first ten returned by CR. In one of the failing cases, the correct rules were not modelled yet, in the other we got too many hits and filtering did not help enough. Choosing a right FSA is very important, but the ordering and filtering certainly help in moving relevant rules to the top of the search result.

6. Conclusions and Discussion

We have presented part of the advanced legal information retrieval system we are building in the CLIME project, a module for Extended Conceptual Retrieval. It should be noted that the original goal of the CR technology in CLIME is to augment a full normative assessment system. The less precise CR can be used to search sources for which norms are not yet modelled and present this information as an extension to the assessment results. It is different from and extends other ways of CR in several ways. First of all, the concepts are of a symbolic nature and form an ontology of the domain (unlike statistical approaches in neural networks or other association based approaches). Secondly, the conceptual graphs are used to both abstract and specialise user queries using type hierarchies. Thirdly, user queries can be extended over other types of relations in the domain. And finally, we have introduced measures for relevance ranking, advance ordering and filtering to better present the search results.

Performance of Extended Conceptual Retrieval is fine. Even though it is implemented in Java, it is fast and reliable. From the way it is programmed it should have a less than quadratic complexity both in space and time requirements in the size of the knowledge base and the number of states of the finite automaton, which determines the query. The filtering part also has a less than quadratic complexity in the size of the filter. Since the finite automata and filters are in practice very small it is reasonable to assume that the practical complexity is linear in the size of the knowledge base. Minimal path computation is a more costly algorithm because of the breadth-first search. Minimal path has shown long computations when tested with great depth as a stand-alone program. This is not surprising, since the complexity is exponential in the depth of the search and the size of the knowledge base. However, for minimal paths to be useful in explanation, they will only be computed for small depths.

Looking at the model we have built so far, the static measures of centrality and reference count we use for relevance ranking of search results, are according to expectations.

Whether the results of Extended CR really help users remains to be seen. It may turn out that the ordering information is too crude and we need more refined ones. One could for instance relate the reference count of concepts to the centrality: non-central concepts that are referred to a lot deserve a different approach than central ones that are hardly referred to etc. Also, the formation of FSAs for extending initial concept sets needs further research. Initial ideas were outlined in this paper, but they need further testing.

Acknowledgement

CLIME is sponsored by the EC ESPRIT programme with project number P25.414. The CLIME partners are: British Maritime Technologies (UK), University of Brighton (UK); Bureau Veritas (France); TXT (Italy), and University of Amsterdam (Netherlands).

References

[1] Turtle, H. (1995). Text retrieval in the legal world. *Artificial Intelligence and Law*, 3(1-2):5-54, 1995.

[2] Hafner, C.D. (1981). *An Information Retrieval System Based on a Computer Model of Legal Knowledge*. PhD thesis, University of Michigan, 1981. Republished by UMI Research Press, Ann Arbor, Michigan, 1981.

[3] Woods, W.A. (forthcoming). Conceptual Indexing: A Better Way to Organize Knowledge, SUN Technical Report.

[4] Valente, A. (1995). *Legal Knowledge Engineering, a modelling approach.* PhD University of Amsterdam. IOS Press, Amsterdam.

[5] Visser, P.R.S. (1995). *Knowledge Specification for Multiple Legal Tasks. A case study of the interaction problem in the legal domain.* PhD University of Leiden. Kluwer Law International.

[6] Breuker, J.A. (1992). On Legal Information Serving. In *Proceedings of JURIX'92, Information Technology and Law*. Koninklijke Vermande, Lelystad, pages 93-102.

[7] Winkels, R.G.F. (1998). CLIME: Legal Information Serving Put to the Test. In: *Pre-proceedings of the Second French-American Conference on AI and Law*. Available on-line at: http://www.lri.jur.uva.nl/~winkels/papers/AIL98_CLIME_Paper.html

[8] Piwek, P., Evans, R. and Power, R. (1999). Editing Speech Acts: A Practical Approach to Human-Machine Dialogue. In preproceedings of Amstelogue'99. Workshop on the Semantics and Pragmatics of Dialogue. University of Amsterdam 7-9 May, 1999.

[9] Winkels, R.G.F., Bosscher, D.J.B., Boer, A.W.F. and Breuker, J.A. (1999). Generating Exception Structures for Legal Information Serving. *Proceedings of ICAIL-99*, ACM, New York, pp. 182-189.

[10] Belew R.K. (1987). A Connectionist Approach to Conceptual Information Retrieval. *Proceedings of ICAIL-87*, ACM, New York, pp. 116-126.

[11] Mulder, R.V. de, Hoven, M.J. van den, and Wildemast, C. (1993). The Concept of Concept in 'Conceptual Legal Information Retrieval'. *Law Technology Journal, 3,* 1.

[12] Osborn, J. and Sterling, L. (1999). JUSTICE: A Judicial Search Tool Using Intelligent Concept Extraction. *Proceedings of ICAI -99*, pp. 173-181.