
Artificial Intelligence and Law **9:** 29–57, 2001.

Innovative techniques for legal text retrieval

MARIE-FRANCINE MOENS
Interdisciplinary Centre for Law & IT, Katholieke Universiteit Leuven, Tiensestraat 41, B-3000 Leuven, Belgium
E-mail: marie-france.moens@law.kuleuven.ac.be

Abstract. Legal text retrieval traditionally relies upon external knowledge sources such as thesauri and classification schemes, and an accurate indexing of the documents is often manually done. As a result not all legal documents can be effectively retrieved. However a number of current artificial intelligence techniques are promising for legal text retrieval. They sustain the acquisition of knowledge and the knowledge-rich processing of the content of document texts and information need, and of their matching. Currently, techniques for learning information needs, learning concept attributes of texts, information extraction, text classification and clustering, and text summarization need to be studied in legal text retrieval because of their potential for improving retrieval and decreasing the cost of manual indexing. The resulting query and text representations are semantically much richer than a set of key terms. Their use allows for more refined retrieval models in which some reasoning can be applied. This paper gives an overview of the state of the art of these innovative techniques and their potential for legal text retrieval.

Key words: case retrieval model, information discovery, legal text retrieval, machine learning.

1. Introduction

Automated retrieval from large document collections was one of the earliest applications of computer science to law. Today text retrieval remains an important component in legal information systems. However, the legal field provides one of the most difficult test cases for automated text retrieval. Legal text retrieval is primarily based upon concepts, not upon the explicit wording in the document texts. Traditionally, legal text retrieval relies upon external knowledge sources such as thesauri and classification schemes, and an accurate indexing of the documents is manually done. This is expensive, laborious specialist work. As a result only documents in a restricted domain or a selection of documents can be effectively retrieved. Other documents rely upon a rather unreliable full-text search for their retrieval. In addition, legal document collections are ever growing and their users want to have access to the relevant information in an ever faster way.

Legal professionals to a large extent use documents that are electronically stored in databases. In fact, they often use them as a primary source of information. Consequently, it is important that all documents – not just a selection – can be effectively retrieved.

Currently, there are a number of innovative techniques that have a large potential for legal text retrieval. The aim of this paper is to give an overview of these techniques including their problems and potentials. We start with describing the process of retrieval of information in natural language and the characteristics of legal texts. These insights explain why text retrieval and especially legal text retrieval are such difficult tasks. In a next section we give a concise overview of fundamental text retrieval techniques. Then, we discuss current innovative techniques and their potential for legal text retrieval including the representation and learning of an information need, concept learning and classification for text categorization and information extraction tasks, text clustering and text summarization. Finally, we discuss case retrieval and the possibility of integrating legal reasoning in the case retrieval model.

2. Text Retrieval

Automated text retrieval concerns the selection of documents written in natural language from a database that are relevant for a given information need. In a classical *information retrieval (IR) system,* a query composed of key terms is matched with the index terms of a document, and upon matching the documents will be returned. In information retrieval, a rather static document collection is queried by a large variety of volatile queries.

> An example is searching a database of legal cases: the key terms of the query are matched with the key (or indexing) terms of each case in the database.

A variant form of information retrieval is routing or filtering (Belkin and Croft 1992). Here, the information needs are long-lived, with queries applied to a collection that rapidly changes over time. Filtering is usually based on 'users' profiles', which are descriptions of information preferences of an individual or a group of users.

The retrieval process traditionally consists of several probabilistic operations (cf. Blair 1990, p. 319). First, the representation of the information need is often an approximation of the real need of the user or users' group. Second, the automated natural language understanding of the document text is poor, and often yields an incomplete or incorrect characterization of the text and of its content. Finally, the matching between query and document is often restricted to a simple, probabilistic term matching. Documents are usually ranked according to their probability of relevance to the query. As a result it is probable that the whole information retrieval operation does not yield all the documents relevant to the query and/or does supply documents that are not or only marginally relevant to the query. This is called the *information retrieval problem* or, more briefly, the information problem.

The information retrieval problem is also present in newer forms of information selection tools. *Browsing* or navigation *systems* are usually part of hypertext and hypermedia systems and allow users to skim document collections in the search for

valuable information (Conklin 1987; Nielsen 1995; Agosti and Smeaton 1996). In browsing systems, the user does not make his information need explicit. However, the systems exhibit a definite need for adequate and reduced descriptors of their documents' content (e.g., in the form of topic maps, abstracts, and suggested links), which must guide the user in his or her selection of documents. Given the large volume of the collections, these descriptions are preferably automatically made.

> An example is browsing a collection of summaries of legal cases. The summaries help in the selection of cases.

The newest form for information selection regards the difficult task of automatic question-answering. A *question-answering system* (Cowie et al. 2000) is a system that automatically generates the answer to a question from the information found in a document collection. This task requires a good understanding of the natural language question and the passages of the retrieved documents in which the answer to the question might be found.

> An example of this futuristic technique is questioning a database of legislation. An example question could be: "What is the maximum speed of cars that is allowed on a freeway?"

It is generally assumed that future text retrieval systems will be expected to fetch specific facts, answer questions, give advice, or compose reports that satisfy users with even more demanding information needs (Strzalkowski 2000).

Clearly, the straightforward approach to the information problem in text retrieval is a correct understanding and representation of the content of a document, and a matching of this content with a faithful model of the user's interest. The most important problem here is the natural language understanding of document texts and user's preferences.

Natural language is the most elaborate symbolic system that human beings control and is an essential tool in many cognitive processes. It is highly valued as a means of communication and memorizing information (Sperber and Wilson 1995, p. 173). The representation power of natural language is unrivaled. It provides an economical, effective and expressive tool for communication of content (Sparck Jones 1991). The way that thoughts are expressed in natural language is informative in itself. The individual words in a text and their ordering, is part of the content of that text. It is unlikely that natural language will be given up in favor of an artificial language for expressing text content (Coulmas 1989, p. 27).

Communication by natural language is important in the legal field. However, understanding natural language by machine is far from being achieved. The following explains why.

Multiple cognitive studies affirm that the cognitive process of human understanding engages many knowledge sources and sustains multiple inferences. Understanding a text – written or spoken – is not only a process of decoding the code of the creator of the text, whereby coder and decoder have mutual knowledge of every item of contextual information used in interpreting the message (Sperber

and Wilson 1995). Additionally, the creator ostensively provides evidence of his or her intentions which helps focusing the attention of the audience on the relevant information and facilitates a correct understanding. The audience applies inference rules to the recovered semantic representations of the thoughts of the communicator to form a mental interpretation of them. This interpretation even goes as far as inferring a meaning that was not meant by the communicator.

The model of Graesser and Clark (1985, p. 14 ff.) relates four knowledge sources to text understanding. The first source is the explicit linguistic material, including words, syntactic constructions, and linguistic signalling devices that are explicitly manifested in the text. It also includes the linguistic knowledge that the comprehender has about these levels of language analysis. The second source consists of world knowledge structures that are stored in the comprehender's long-term memory. These knowledge structures include both generic knowledge structures and specific knowledge structures. Comprehension suffers when the comprehender's knowledge of the words and topics of the text is inadequate. The third source consists of the goals of the comprehender who reads the text. The meaning of a text varies when a text is accessed for different purposes. The fourth source consists of the pragmatic context of the communication. This includes the social relationship between the reader and the writer, the shared knowledge between the participants of the communicative event, and socially shared attitudes and ideologies. Many inferences are generated during text comprehension, if the comprehender's knowledge base is very rich, and reasoning strategies vary from knowledge domain to knowledge domain. Inferences depend upon knowledge to be found in the text (e.g., the meaning of other, mostly previous sentences), the user's general knowledge system, and upon the purpose of reading a text. Human text understanding involves a huge amount of contextual knowledge. This is especially true when reading legal texts.

The complexity of the cognitive process of understanding natural language text makes the automation of this process a very challenging task. Automatic understanding of texts belongs to the research field of *natural language processing.* Natural language processing aiming at a fully-understood interpretation of texts deals with processing the linguistic coding (vocabulary, syntax, and semantics of the language and discourse properties), the domain world knowledge, the shared knowledge between the creator and user of the text, and the complete context of the understanding at that moment in time, including the ideology, norms, background of the user, and the purposes of using the text. The processing would not only reveal the objective content of the text, it would also clarify the subjective meaning that the text has for its user.

Such a full understanding of texts including its interpretation is far from achieved by automatic means. The problems of automatic text understanding concern both the modelling of the knowledge and the inference mechanism involved, and the computational complexity of the operations. However, a number of promising text-analysis techniques are currently developed that identify information in

texts whereby reducing the amount of knowledge needed or automatically acquiring this knowledge. In addition, a lot of research is rightly devoted to the topic of intelligent agents that know or learn the interests, goals, habits, preferences and/or background (*profile*) or a user of an information system. At last, the matching between information needs and document contents in a retrieval system must be more than a term matching and incorporate reasoning strategies. This topic too has an emerging research interest.

3. Legal Texts and Their Retrieval

Professional settings including the legal field employ their own, distinct text types. *Legal document texts* present themselves in rather conventional forms (Danet 1985). Some of these texts may be part of statute law (treaties, statutes, royal decrees, ministerial decrees, local decrees, etc.). They are officially published and all citizens are supposed to be aware of their content. Other texts relate to the judicial proceedings: police statements, warrants, official pleadings, and court decisions. Each of them indicates a certain step of the procedure, and serves as an official proof thereof. A third kind of text is drawn up as a legal proof in the commercial field, i.e., deeds, contracts and articles of association. In addition, a number of texts are used for administrative reasons (e.g., tax returns). Finally, the texts of legal doctrine are made for scientific or research purposes. Documents most commonly used in retrieval systems are statutes (legislation) and court decisions (cases).

We lack comprehensive descriptions of *legal discourse*, especially studies that aim to support its processing by computer (Moens et al. 1999a). Legal documents generally present themselves as written natural language texts, although documents spoken in natural language, or even multi-media documents (Lederer 1996), might gain in importance in the future.

It has been demonstrated that legal documents are highly structured and that there is a relation between content and structure. To the extreme there are the fill-in forms (for instance in tax returns). Most legal documents follow a conventional schema, also called superstructure:[1] The discourse – even when it has the form of fluent natural language text – is composed of parts or segments, either all obligatory or some obligatory and some optional, which occur in a fixed or partially fixed order. A segment often has a specific function in the overall communicative purpose of the discourse or discusses a specific topic. Legal texts employ specific rhetorical cues, i.e., specific formulations that cue their readers into a correct understanding of the text or act as cohesive connectives referring to certain relationships between the parts of the content.

A more detailed level of discourse description concerns the vocabulary, syntax and semantics of individual sentences, clauses and phrases. Descriptions of legal texts (Moens et al. 1996) indicate that the legal language cannot be considered as

[1] This superstructure or part of it might be tagged (e.g., in XML (eXtensible Markup Language)) at text generation, facilitating an identification of broad chunks of information in the texts.

a pure sublanguage. A sublanguage (or linguistic subvariety) usually deals with a specific domain (subfield) and is used for a specific purpose. A sublanguage has a restricted vocabulary, which is distinctive in the set of words which comprise it. It may be more restricted in its syntactic, semantic, and discourse properties than standard language, but it may also exhibit unusual rules (Kittredge and Lehrberger 1982). However, the vocabulary of legal texts is diverse. Legal language employs a number of domain specific concepts besides the concepts of standard language. Studies of the syntax seem to confirm some differences with ordinary language. Judges and law makers tend to express themselves in exceptionally long sentences and in crucial subclauses. In addition, many terms have a specific semantic meaning in legal context. Hence the proven usefulness of legal thesauri en classification structures in traditional legal information retrieval (e.g., Winkels et al. 2000). Archivists of legal documents traditionally rely upon thesauri and classification concepts.

There is the difference in the language of statutes and of legal cases, the former being more technical, more structured and standardized (at least in theory) and also more studied, the latter being more close to complex prose. Especially in civil law countries, the texts of statute law change dynamically as new versions of its parts (e.g., articles) become valid. In the case of legislation the semantics of the objects retrieved are often clear, and the relationship between the attributes of objects and the typical information needs of users are better understood (for instance a user wants to see the original version of a specific article). However, maintenance and version management of the databases are tedious manual tasks, which could be resolved by processing and tagging the texts with intelligent tools at the time of creation. Retrieval of legal decisions is more complicated: users usually want to retrieve an instance of the application of some abstract concept or want to find similar cases to the one they have at hand. This makes retrieval of cases without human intervention (e.g., in indexing the cases) an especially difficult task, hence the need for intelligent tools for processing query and documents and their matching.

4. Fundamental Techniques

A typical *information retrieval* (IR) *system* selects documents from a collection in response to a user's query, and ranks these documents according to their relevance to the query (Salton 1989, p. 229 ff.). This is usually accomplished by matching a text representation with a representation of the query. It was Luhn (1957) who suggested this procedure. This same procedure is followed in legal information retrieval (Bing 1984; Turtle 1995).

A *search request or query*, which is a formal representation of a user's information need as submitted to a retrieval system, usually consists either of a single term from the indexing vocabulary or of some logically or numerically weighted combination of such terms. When the search request is originally formulated in natural language, a formal representation can be derived by applying simple in-

dexing techniques or by analyzing the request with natural language processing techniques.

The majority of existing *automatic indexing methods* select *natural language index terms* from the document texts (Salton 1989, p. 303 ff.; Moens 2000, p. 77 ff.). The index terms selected concern single words and multi-word phrases, and are assumed to reflect the content of the text. They can be directly extracted from the title, abstract, and full-text of a document. A prevalent process of selecting natural language index terms from texts that reflect its content is composed of the following steps:

1. *Lexical analysis:* The text is parsed (usually with the help of a *finite state automaton*) and individual words are recognized.
2. The removal of *stopwords*: A stoplist contains terms in the subject domain that are insufficiently specific to represent content and is built with terms that very frequently occur in the document corpus or with words that belong to the syntactic classes of function words (e.g., articles, prepositions).
3. The optional reduction of the remaining words to their stem form, called *stemming*: There are different methods of stemming, many of which rely upon linguistic knowledge of the collection's language.
4. The optional formation of *phrases* as index terms: Techniques of phrase recognition employ the statistics of co-occurrences of words, or rely upon linguistic knowledge to detect the syntactic parts of sentences.
5. The optional replacement of words, word stems, or phrases by their *thesaurus class terms*; a thesaurus replaces the individual words or phrases of a text by more uniform concepts.
6. The computation of the importance indicator or *term weight* of each remaining word stem or word, thesaurus class term, or phrase term. It is generally assumed that terms that occur frequently in a text and infrequently in the complete document corpus are good index terms. Therefore the product of the term frequency and the inverse document frequency is regularly used as weight function. The former is the number of times that a term occurs in the text. The latter is a factor that is inversely proportional to the number of documents of a reference collection in which the term occurs. The term frequency can be normalized by a length normalization factor in order to account for differences in text length.

The above regards automatic indexing of document texts. *Full-text* search is a simplified variant and considers all terms (except for stopwords) as index terms (Blair and Maron 1985, 1990).

A *manual indexing* of legal documents is still common, especially when controlled language index terms, such as thesaurus class terms and conceptual terms, are assigned. Manual indexing also includes the manual assignment of content mark-up for instance in SGML (*Standard Generalized Markup Language*) or XML (*eXtensible Markup Language*). The manual process is very time-consuming and thus costly and, when confronted with many documents, is not a realistic task. As

a result not all texts are properly indexed for a reliable retrieval. In the legal field we are confronted with an overload of legislative and jurisprudential documents, which in many cases are only accessible by a full-text search.

The abstract representations of both document text and query make a comparison possible. The texts, the representations of which best match the query representation, are retrieved. There are a number of *retrieval models* that are defined by the form used in representing document text and query and by the matching procedure. Both text and information need representations are uncertain and additionally do not always exact match. Querying an information retrieval system is not like querying a classical database. The matching is not deterministic. Retrieval models often incorporate this element of uncertainty. Moreover, retrieval models generally rank the retrieved documents according to their potential relevancy to the query.

The common retrieval models are the Boolean retrieval model, the vector space model and the probabilistic model.

In the *Boolean retrieval model* (Salton 1989, p. 235 ff.), a query has the form of an expression containing index terms and Boolean operators (e.g., 'AND', 'OR' and 'NOT') defined upon the terms. The retrieval model compares the Boolean query statement with the term set used to identify document content. A document the index terms of which satisfy the query is returned as relevant. In the pure Boolean model, no ranking of the documents according to relevance is provided. Variants of the model provide ranking based upon partial fulfillment of the query expression.

In the *vector-space retrieval model* (Salton 1989, p. 313 ff.; Wang et al. 1992), documents and queries are represented as vectors in a vector space with the relevance of a document to a query computed as a distance measure. Both document and query are represented as term vectors in an n-dimensional vector space of the form $(D_m = (a_{m1}, a_{m2}, \ldots, a_{mn})$ and $Q_k = (q_{k1}, q_{k2}, \ldots, q_{kn})$, where the coefficients a_{mi} and q_{ki} represent the values of index term i in document D_m and Q_k respectively ($i = 1 \ldots n$ where n = number of distinct terms in the index term set of the collection). The coefficients take on binary values (1 when term i appears in document D_m or query Q_k or 0 otherwise) or numeric values indicating the weight or importance of the index terms. Comparing document and query vectors is done by computing the similarity between them (Jones and Furnas 1987). Common similarity functions are the cosine function, which computes the cosine of the angle between two term vectors, and the inner product, which computes the scalar product between the term vectors. The result of the comparison is a ranking of the documents according to their similarity with the query. The model has been criticized because it does not accurately represent queries and documents (Raghaven and Wong 1986). The vector space model uses terms as the basis of an orthogonal basis of the vector space. It adapts a simplifying assumption that terms are not correlated and term vectors are pair-wise orthogonal. However, many useful and interesting retrieval results have been obtained despite the simplifying assump-

tions. There is no consensus whether more sophisticated vector-based models that take into account term correlations and that are computationally more expensive give better retrieval results (Carbonell et al. 1997). The generalized vector space model (Wong et al. 1985) translates the term vectors of query and documents into a vector space of which the documents are the basis for representing terms. The Latent Semantic Indexing (LSI) model (Deerwester et al. 1990) assumes that neither terms nor documents are the optimal choice for the orthogonal basis of a semantic space, and that a reduced vector space consisting of the most meaningful linear combinations of documents would be a better representation for the content of documents.

The classic *probabilistic retrieval model* views retrieval as a problem of estimating the probability that a document representation matches or satisfies a query (Baeza-Yates and Ribeiro-Neto 1999, p. 30 ff.). Often, this probability is computed by learning the weight of query terms from the documents that are judged relevant or non-relevant for the query and that contain or do not contain the terms (Maron and Kuhns 1960; Robertson and Sparck Jones 1976; Fuhr 1992; Sparck Jones et al. 2000). The current models use more refined statistical techniques, such as 2-Poisson distributions (Robertson and Walker 1994) and logistic regression (Gey 1994) for estimating this probability. When estimating the probability of the relevance of a document to a query, again term independence is assumed.

Another useful probabilistic model that has been implemented in the legal domain (Turtle and Croft 1992; Turtle 1995) is the inference network model. In this model document and query content are represented as linked networks (e.g., representing the probabilistic results of different indexing techniques, representing the probabilistic relation between a term and a concept, or representing the weighted preferences of a user). The networks thus incorporate knowledge that reflects the properties of the subject domain, possibly upon linguistic knowledge, and/or knowledge of the supposed retrieval strategies of a user. Inference is based upon the combination of evidence as it is propagated in linked networks. Documents are ranked according to this belief of relevance.

The most common measures to evaluate retrieval results are recall and precision (Salton 1989 p. 248; Baeza-Yates and Ribeiro-Neto 1999, p. 74 ff.). *Recall* is measured in terms of number of correct responses by the system upon the number of correct responses by an expert. *Precision* is the number of correct responses by the system upon the total number of responses by the system. The *F-measure* combines recall and precision values:

$$F = \frac{(\beta^2 + 1)\,P\,R}{\beta^2 P + R}, \tag{1}$$

where P = precision, R = recall, and β = a factor that indicates the relative importance of recall and precision.

Recall, precision and F-measure ideally are close to 100%. Precision is often measured (possibly by means of some interpolations) at 11 recall levels (at 0%,

10%,. . . , 100% recall). The values are placed in the recall-precision graph. Recall and precision are usually inversely related: when recall is low, precision is high and vice versa. The breakeven point is the point in the graph at which precision equals recall.

5. Innovative Techniques

Important causes of the information retrieval problem are an incorrect automatic understanding of the information need and of the content of document texts, and a poor strategy for matching query and document content. There are a number of current interesting approaches to information retrieval that employ a limited amount of knowledge or automatically learn this knowledge. A first approach regards techniques for the representation and learning of the user's interest. Other approaches focus upon concept learning and classification in text categorization and information extraction tasks, text summarization, and automatic reasoning with legal cases. The techniques are valuable for classical retrieval systems and for novel information selection tools such as browsing and question-answering systems.

5.1. REPRESENTATION AND LEARNING OF THE INFORMATION NEED

Intelligent interfaces can be provided for querying the document base. A variety of kinds of knowledge are potentially of use. One can use the knowledge of an 'intelligent agent'. There are many definitions of the concept '*agent*' (we refer here to Bradshaw 1997, p. 3 ff.). A crude definition is that an agent is software that through its imbedded knowledge and/or learned experience can perform a task continuously and with a high degree of autonomy in a particular environment, often inhabited by other agents and processes. An information agent supplies a user with relevant information that is for instance drawn from a collection of documents.

The main goal of employing an information agent in information selection and retrieval is to determine the user's real need (*user's profile or model*) and to assist in satisfying this need (Bradshaw 1997). The agent knows the user's interests, goals, habits, preferences and/or background, or gradually becomes more effective as it learns this profile. In retrieval, a user model can include information about the preferences of a user for particular types or sources of documents, their expertise with IR systems and with domains, and knowledge about how they view a particular domain. The knowledge in the profile is intellectually acquired (from the user and experts), implemented and maintained by knowledge engineers. Or, the agent learns the knowledge based on good positive (and negative) training examples. Learning a user's profile has multiple advantages, including the avoidance of costly implementation and maintenance, and easy adaptations to changing preferences.

Learning of users' preferences is closely related to the technique of relevance feedback (Rocchio 1971; Salton and Buckley 1990; Buckley and Salton 1995). Such an approach assumes the relevancy of documents that are similar to pre-

viously retrieved documents found relevant. Since document content and user information needs are imperfectly represented by current retrieval systems, an initial retrieval rarely results in exactly the documents desired by a user. Several iterations modifying the query are often necessary to achieve acceptable results. The technique of relevance feedback automatically expands the terms of the query with related terms that co-occur with the query terms in the documents retrieved, or learns better term weights of the query from the relevant and non-relevant documents (cf. below: techniques of concept learning and classification). In the technique of pseudo-relevance feedback no interaction with the user is needed, but queries are expanded with terms from the top-ranking documents in the list of retrieval results.

> An example is automatically informing legal professionals about new statutes or amendments of statutes in their domain of interests. The profile of the professional could be automatically learned from his or her previous consultations of legislation in the database.

5.2. CONCEPT LEARNING AND CLASSIFICATION

5.2.1. *The methods*

Traditional information retrieval in the legal field has demonstrated the usefulness of indexing documents with fixed concepts, which is often called conceptual information retrieval (Dick 1987; Hafner 1987). Classifications according to legal concepts are common in legal science. Law itself employs a classification. We can assign classification codes to complete legal documents. We discuss this task under the heading text categorization. We can classify certain information in legal texts (e.g., classify facts of a legal case into more abstract concepts). We discuss this task under the heading information extraction.

Humans perform a text classification task by skimming the text and inferring the classes from specific expressions or word patterns and their context. Automatic text classification simulates this process and recognizes the classification patterns as a combination of text features. These patterns must be general enough to have a broad applicability, but specific enough to be consistently reliable over a large number of texts. The knowledge of the patterns and their corresponding class is manually acquired and implemented in a knowledge base, or is automatically learned from classified example texts. Systems that automatically sort patterns into categories are called pattern classifiers or shortly classifiers (Nilsson 1990, p. 2).

The majority of text classifiers involve the construction of a classification procedure from a set of example texts for which the true classes are known (e.g., manually classified or annotated). This form of *supervised learning* is called pattern recognition or discrimination. The aim is to detect general, but high-accuracy classification patterns and rules in the training set, which are highly predictable to correctly classify new, previously unseen texts. The set of new texts is called

the test set. Because text classes are not mutually exclusive, it is convenient to learn a binary classifier for each class, rather than to formulate the problem as a single multi-class learning problem. More specifically, the archetypal supervised classification problem is described as follows (Bishop 1995, p. 1 ff.; Hand 1997, p. 5 ff.). Each object is defined in terms of a vector of features (often numerical, but also possible nominal such as color, presence or absence of a characteristic): $x = (x_1, x_2, \ldots, x_n)$, where x_j = the value that the feature j takes for object x and $j = 1 \ldots n$ (n = number of features measured). The features together span a multivariate space termed the measurement space or feature space. For each object of the training set, we know both the feature vector and the true classes. The features of texts are commonly the words and phrases, but other features can be selected (e.g., length of a sentence, presence of a cue term). When selecting feature words or phrases, their number can be very large, creating the necessity of effective feature selection and extraction when training text classifiers. A text classifier learns from a set of positive examples of the text class (texts relevant for the class) and possibly from a set of negative examples of the class (texts non-relevant for the class). From the feature vectors of the examples, the classifier typically learns a classification function, a category weight vector, or a set of rules that correctly classifies the positive examples (and the negative examples) of the class. Each new text is equally represented as a feature vector, upon which the learned function, weight vector, or set of rules is applied to predict its class. Because, there are usually many classes and only few of them are assigned to a given example text, the number of negative examples in a training set exceeds the number of positive ones.

Three broad groups of common training techniques can be distinguished for the pattern recognition problem (see Michie et al. 1994): *statistical approaches*, *learning of rules and trees*, and *neural networks.* The most common and successful techniques for text classification are the statistical naive Bayes method, nearest neighbor classifiers, and Support Vector Machines, and the techniques that learn rules and trees (Yang and Liu 1999). Learning patterns that occur in a sequence can be done by the statistical technique of a Hidden Markov Model (HMM).

The *naive Bayes algorithm* (Mitchell 1997, p. 154 ff.) is a simple and common approach for classifying textual objects. The probability that an object belongs to a certain class given the features of the object is based on the probabilities that these individual features are related to the class. The probability estimates of the individual features are based on the co-occurrence of classes and the selected features in the training corpus (maximum likelihood estimation). In a naive Bayes classifier the computations are simplified by the assumption that the feature values are conditionally independent. As a consequence the probability of a set of features can be computed as the product of the probabilities of the individual features. To classify a new object, the probability of class membership for each class is computed. The classes corresponding with the top k (some constant) probabilities are assigned to the object or a class is assigned when the probability of its membership is higher than a pre-set threshold.

> An example is training a text classifier on an example corpus of cases that automatically detects an appeal case. Different features or parameters might be accounted for (e.g., length of the case, reference to a previous decision, occurrence of cue word patterns). A new case is classified as an appeal case when the joined probability of its features exceeds a certain threshold. The probability of an individual feature is counted (maximum likelihood estimation) in the training corpus of appeal cases.

Nearest Neighbor (*NN*) (Masand et al. 1992) classifiers only store the positive examples of a class. To classify a new object the nearest neighbor classifier compares the feature vector of the new object with the feature vector of each example stored by using a similarity or distance function (e.g., inner product). The classifier finds the k closest examples or the examples for which the similarity exceeds a certain threshold and picks up the classes of these for the new object.

> An example in a civil law country like Belgium is automatically detecting the crime concepts of a criminal case by comparing its crime descriptions with classified statute texts on the crimes. Upon a sufficient match of the descriptions (e.g., based on matching terms or by means of a vector comparison of the term vectors) the corresponding classification of the crime can be assigned to the case.

A promising current technique for text classification is the technique of *Support Vector Machines* (*SVM*) (Joachims 1998). When two classes (in text classification the positive and negative examples of a class) are linearly separable, the technique finds the hyperplane in the n-dimensional feature space that maximizes the margin between the examples of the classes (e.g., decision surface in two dimensions). A new example is classified by computing the side of the hyperplane it belongs to. The technique can be generalized to examples that are not linearly separable.

> An example is automatically informing legal professionals about new statutes or amendments of statutes in their domain of interests. Support Vector Machines have the property of finding the function of a hyperplane with maximum margins in a feature space with a very large amount of features. Learning the domain of interest of a legal professional involves classifying texts as relevant or non-relevant based upon the terms (words and phrases) that occur in the training set. The term or feature set is typically large. By applying the learned function upon a new text, it can easily be classified as relevant or non-relevant.

Hidden Markov Models (*HMMs*) are probabilistic finite state automata to model the probabilities of a linear sequence of events (Manning and Schütze 1999, p. 317 ff.). A Markov Model is defined by a set of states, a set of transitions, each going from an initial state to a final state and the probabilities of the transitions, and a set of output symbols that can be emitted when in a state (or transition). For each state (or transition), a probability distribution is defined for all symbols that

can be emitted (it is also possible to allow null transitions that do not produce any symbol). In a Hidden Markov Model, you don't know the state sequence that the model passes through, but only some probabilistic function of it. Given a training corpus in which the information is sequentially structured and which is manually annotated, efficient algorithms learn the probabilities of all transitions and emissions. Typical emissions are the information segments of a text and transitions refer to the sequential structure of the segments. The technique is useful for extracting information that is highly sequentially structured. There is a current research interest in automatically learning the topology of the model from the training data with manually annotated examples (Craven et al. 1999). Once the model is trained, the most probable path, i.e., the most probable sequence of information segments, of a new text can be computed.

> An example is automatically learning the structure of legal cases. Legal cases are composed of segments that occur in a fixed or partially fixed order. From a training corpus that is manually annotated, the different allowable structures of the cases can be learned as a Hidden Markov Model. When a new case is automatically structured, the most probably structure (most probable path in the model) is computed.

Another promising technique is inducing classifying expressions in the form of *decision rules and trees* from example texts (Mitchell 1997, p. 20 ff.). The training examples are represented as a set of features or as a set of relations between features. The learned expressions have the form of decision rules in propositional logic or in first-order logic. Each rule is associated with a particular class. When a tree is learned it consists of nodes and branches, whereby each node, except for terminal nodes or leaves, represents a test or decision and branches into subtrees for each possible outcome of the test. Each leave of the tree is associated with a particular class. To classify a new object a rule that is evaluated as true is an indication of its class. In case of a learned decision tree, one starts at the root of the tree and moves through it (evaluating each node) until a leaf (class of the object) is encountered.

In case of rule induction a rule is often learned that covers all (or most) of the positive examples and none (or fewest) of the negative examples. The rule is added to the rule set. Objects that satisfy the rule are removed from the training set for further consideration. This process is repeated until no more example cases remain to be covered. In many cases the rule is learned in a greedy way, for instance by adding (general-to-specific) or deleting (specific-to-general) the best feature at each step. In case of tree induction, a tree is often constructed general-to-specific (top-down) fashion with a greedy algorithm (e.g., C4.5: Quinlan 1993). The attribute that alone best classifies the training examples (e.g., largest information gain or difference in entropy) is selected and used as the test at the root node of the tree. A descendant of the root node is then created for each possible value of this attribute. The training examples are sorted to the appropriate descendant node and the process is repeated using the training examples associated with each descendant node in order to select the best attribute to test at this point of the tree. When the learned

rules or tree overfits the training data, i.e. when the rules or tree perfectly fits the training data, but is not general enough to predict new cases, the rules or tree can be pruned when the pruning does not increase the error on a validation set.

> An example is automatically learning rules that are associated with a legal concept from annotated passages that treat the concept (e.g., a set of classified statutes). The rules formulate the variant textual patterns that express the concept (e.g., IF 'medicin* AND ('*prescri** ' *OR 'administ** ' *OR 'deliver** ') *AND ('sleep induc** ' *OR 'narcotic** ' *OR 'anaesthetic** ') *AND 'dependen** ' *THEN special_*concept = 'maintenance of toxicomania' AND general_concept = 'drugs and narcotics', where '*' stands for wild card characters).

Learning rules and trees gives good results when the attribute (feature) set is rather accurate and limited (e.g., small texts). It is a useful method when there are conditional dependencies among the features, which is often the case in texts. Finally, the learned rules can be complemented by ones manually implemented or verified. However, the method often suffers from practical and computational limitations. It is often impossible to test all possible rules and trees given the number of attribute combinations. In this case one might use a greedy algorithm (see above) that considers a single best feature for inclusion or exclusion at each stage of building a rule without backtracking. Because each new choice depends on the previous choices, a good but not always optimal rule or tree is obtained. Promising classification results are obtained by using the technique of adaptive resampling (Weiss et al. 1999). In this technique multiple decision trees are generated from different selections of the training examples. In such a set, examples that were previously erroneously classified by the learned tree, can be redundantly selected for learning the next tree. A new text is classified by taking votes from a pool of decision trees.

5.2.2. *Text categorization in the legal field*

We mean here by text categorization the assignment of subject descriptors or classification codes to complete texts. There are a large number of text categorization applications in the legal domain. Statute law is often categorized into domains and topics. Court decisions can be classified for more effective retrieval. Conceptual retrieval of case law is important and is extensively studied (e.g., Dick 1987; Hafner 1987), but the retrieval often relies upon a manual classification of the cases. There is the interesting work of the research group of Kevin Ashley at the University of Pittsburgh (Brüninghaus and Ashley 1997, 1999), which regards the classification of cases according to factors with text classifiers that are trained upon example texts. Legal cases can be represented in terms of their values on domain dependent factors. Factors are derived features recognized by domain experts as strongly influencing a case's outcome.

There are a number of challenges in applying traditional learning algorithms to text categorization and especially to categorization of legal texts.

Text categorization has to cope with a large and poorly defined feature set (there are many words and phrases of a text) and an often low density of positive training examples. Effective feature selection techniques before and during training (e.g., χ^2 test: Moens and Dumortier 2000) are important to avoid overfitting of the trained classifier. If legal domain knowledge is available (e.g., in the form of a thesaurus or ontology), it can be used for selecting effective features.

Compared to other texts, legal texts are often very complex and text categorization often involves the assignment of abstract concepts. The terms of the concepts do not always occur in the texts and might be expressed in a large variety of phrasal expressions. Especially translating factual descriptions into abstract concepts is very difficult, because of the many variant expressions of a concept and their formulations in natural language.

> An example is the concept of "negligence" with might be present in many different fact descriptions in a legal case (e.g., motorcycle carelessly driven, nurse forgets the medication of her patient, etc.) each of which can be expressed in many equivalent natural language expressions.

This makes that the good categorization results of applying the above classifiers upon more "easy" collections (e.g., breakeven point of recall and precision up to almost 88% on the benchmark collection: Reuters-21578: Weiss et al. 1999) cannot yet be duplicated for legal texts.

5.2.3. *Information extraction in the legal field*

Information extraction (IE) is a technique for extracting domain-specific information from texts (Cowie and Wilks 2000). Text fragments (e.g., noun phrases, sentences, segments) are mapped to fields or template slots that have a definite semantic meaning. In other words the technique regards classifying text fragments. Influenced by the work of Schank (1975) successful information extraction exclusively relied upon domain-specific knowledge and some linguistic knowledge of the language and the text type captured in production rules and frames (see Jacobs 1992 for an overview). Currently, there is a large interest in training text classifiers (see above) that learn the extraction patterns from example texts (Riloff 1996; Craven et al. 1998; McCallum et al. 1999; Soderland 1999). The contextual patterns of the information are automatically acquired often in a bootstrapping manner. The context considered can vary from adjacent words in sentences, the complete text or the document collection. As a result the information extraction capability continuously improves, both in qualitative and quantitative terms, while more documents of heterogeneous subject domains can be processed. Especially, the classifiers that learn rules and trees are very promising (e.g., Soderland 1999). In addition, information extraction supports a more accurate indexing and retrieval, and is useful as a first step in feature selection for text categorization and clustering, and in information selection for text summarization.

The results in terms of recall and precision of the extraction tools that have been trained on example texts approach the results of systems that use knowledge patterns that were manually implemented. However, it seems that in both types of systems, some information is much easier to extract than others and extraction from semi-structured texts is easier than from unstructured free text. Current systems rather successfully identify named entities (locations, organizations and persons), numeric quantities (money and percentages) and expressions involving times (date and time) without relying upon a manually encoded lexicon of entities (F-measures ($\beta = 1$) $< 94\%$ in MUC-7). Many of the named entities can be recognized because of the specific context they appear in. More sophisticated systems identify with less success the relations between entities in the texts (e.g., a person as the employee of a company) and the events that an entity takes places in (F-measures ($\beta = 1$) $< 51\%$ in MUC-7). A plausible explanation is that equivalent concepts in natural language as well as their contexts are expressed in a variety of ways.

Legal texts that are often well-structured are good test cases for learning extraction patterns.

> An example is learning the extraction patterns of case citations, relevant statutes that are applied, and cue patterns that signal argumentation segments from a training set of cases.

5.3. TEXT CLUSTERING

Although techniques of supervised learning are especially researched for classifying texts, *unsupervised learning techniques* receive a great deal of attention. In unsupervised learning the classes are not a-priori defined, but inferred from the data. The techniques include the discovery of groupings, relationships, and other patterns in text data. Unsupervised learning techniques do not need example texts that are manually classified or annotated.

A major group of techniques are the clustering of objects that share common features. *Cluster analysis* is a multivariate statistical technique that automatically generates groups in data (Kaufmann and Rousseeuw 1990). Non-hierarchical methods partition a set of objects into clusters of similar objects. Hierarchical methods construct a tree-like hierarchy of objects with the root representing one big cluster containing all objects, the leaves representing the individual objects, and the nodes containing the intermediate groupings. The technique of clustering supposes:

1. an abstract representation of the object to be clustered, containing the features for the classification;
2. a function that computes the relative importance (weight) of the features;
3. a function that computes a numerical similarity between the representations.

Clustering is employed to group terms if they often co-occur in documents or to group documents if they discuss the same topic terms.

The technique of *term clustering* groups words or phrases based on the document texts in which they co-occur in order to detect related terms. Here, the selected features are the selected documents in which the terms occur. It is assumed that words that are contextually related, i.e., often co-occur in the same sentence, paragraph, or document, are semantically related and hence should be classified in the same class. The exact nature of a term relationship (e.g., synonyms, related terms) is difficult to identify, nevertheless the related terms are very useful in retrieval to expand the key terms or to expand query terms with related terms (Baeza-Yates and Ribeiro-Neto 1999, p. 123 ff.). The general process of constructing a "thesaurus" of related terms consists of selecting a representative document set and its content terms. The pairwise association or correlation between two terms can be computed from this set based on their co-occurrence in the same document, their physical distance in the document or based upon their occurrence in the same neighbourhood of terms.

> An example is automatically detecting related terms from a domain-specific corpus (e.g., legislation on the topic of education) by means of term clustering techniques. Such an automated tool can suggest terms when building a thesaurus or ontology in the domain of education law.

Clustering techniques have been developed to *group document texts* based on common words that they contain (see Willett 1988 for an overview). Here, the features are selected words and phrases from the texts.

One of the earliest applications of the technique is an efficient implementation of the vector space model. The *cluster retrieval model* groups similar documents (i.e., that share many terms). For each cluster of documents, a representation is made (e.g., the average vector (centroid) of the cluster). In the retrieval model a query is ranked against the representative of a group of documents and upon matching, the cluster of documents is returned (van Rijsbergen 1979, p. 45 ff.; Croft 1980; Griffiths et al. 1986; Salton 1989, p. 341 ff.; Hearst and Pedersen 1996). Although the cluster retrieval model has sometimes been criticized (Voorhees 1985), it still is a popular model for organizing a document collection.

Clustering of documents is also useful for generating a topical overview of a complete document collection. The clusters are labelled with the document titles or main key terms. Such a topical overview is often integrated in interfaces for browsing.

A nice example is the Scatter/Gather system (Cutting et al. 1992). Initially, the user is presented the content of k large clusters that represent the broad topics of the collection ("scatter" documents). The user can "gather" the contents of one or more clusters and the system "rescatters" this document subset to form k new clusters. This process can be iterated several times. The result is an overview of topic and topic combinations at different levels.

> An example is automatically generating a topic map of a database of legal cases, which must facilitate the selection of the cases.

Clustering with unsupervised neural networks (Kohonen's self-organizing maps) is used for classification (Merkl and Schweighofer 1997) and categorization of legal texts (Schweighofer and Merkl 1999). The resulting topic maps are adequate when the features of the classification are identified based on a knowledge base of legal concepts and rules.

It is also useful to cluster a list of retrieved documents for a more efficient consultation (Leuski and Allan 2000).

Research is currently devoted to finding computationally efficient algorithms that find natural groupings in a set of objects without imposing constraints such as a fixed number of clusters or a threshold similarity value between the objects of a cluster (e.g., Moens et al. 1999b; Aslam et al. 2000). Unless very specific text features can be selected for the groupings, the results of a clustering might be deceptive. Texts contain a lot of features: often the words that are shared between texts (after elimination of stopwords) are not relevant for the classification that is sought in the collection.

Better results are obtained when learning the topics by the grouping of sentences or paragraphs that share common content terms within one text. The vocabulary within a single text is much less diverse. Some promising work has been done in this field (Hearst 1997; Salton et al. 1997; Moens et al. 1999b).

> An example is clustering the paragraphs of the alleged offences and opinion of the court part of criminal cases according to the topics that they contain (Moens et al. 1999b). Here, clustering algorithms based on the selection of representative objects not only generate a natural clustering, they also find a representative paragraph in a group of paragraphs, that is useful to include in the summary of a case.

5.4. TEXT SUMMARIZATION

Except for some initial interest in the subject in the late 50s and 60s, automatic text summarization has never received a large attention up until recently. Text summarization consists of three steps (Sparck Jones 1993). The text analysis step identifies the essential content of the source text resulting in a source text representation. In the transformation step the content of the source text is condensed either by selection or generalization of what is important in the source. The selected and generalized information is captured in a summary representation. Finally, the synthesis step involves drafting and generation of the summary text based upon the summary representation.

The ultimate goal of the *text analysis* is the complete natural language understanding of the source text, whereby each sentence is processed into its propositions representing the meaning of the sentence, and whereby the sentence representations are integrated in the global meaning representation of the text. Then, in the transformation step, this representation could be pruned and generalized according to

the focus of the summary. Current text summarization systems (for an overview see Moens 2000, p. 133 ff.) are not so sophisticated, especially in the text analysis step. They often combine text analysis with *information selection* and extract certain sentences in the text that are relevant for the abstract.

A first group of techniques for text analysis and information selection relies heavily upon knowledge sources to interpret the surface features of a text. These methods find their origin in natural language processing, are adequate to generate good abstracts, but are restricted with regard to the application domain. The symbolic knowledge mainly concerns linguistic, domain-world, and contextual knowledge. The linguistic knowledge commonly deals with lexical, syntactical, and semantic properties. It also includes knowledge of discourse structures, especially as flagged by lexical and other surface cues. Domain world knowledge deals with the representation of the domain dependent content of the text and is often of semantic nature. The knowledge representation generally integrates the linguistic and the domain modeling. When the text analysis also integrates information selection, contextual knowledge that models the communicative preferences of the users of the abstract is also needed.

The earliest systems especially relied upon domain knowledge (e.g., DeJong 1982; Jacobs 1992): such as the words and phrases typically employed in texts about terrorists attacks. Still today this technique yields valuable results for summarizing texts in a limited subject domain (an example: Hahn and Reimer 1999).

There is a growing interest in using discourse structures in automatic text summarization (e.g., Marcu 1999; Boguraev and Neff 2000; Moens 2000, p. 142 ff.). The structures of a discourse (schematic, rhetorical, and thematic) are explicitly used by the writer of a text (or in speech by the speaker) to signal certain semantic content, so that a reader maximally discovers the text's message. The patterns also guide the reader in interpreting the passages of the text. A reader can likewise approach a text with various structural expectations. It is this shared knowledge that is highly valued in automatic summarization of texts and in identifying certain information in it.

> An example is automatically detecting an argumentation passage based on a rhetorical cue in the text of a legal case. The phrasing "this conclusion follows from" probably follows the text of a conclusion and introduces the followed argumentation.

The knowledge is usually captured in production rules and frames, which are organized into conceptual graphs and semantic networks of frames. The structured knowledge representations are also called content schemes, scripts, templates, or text grammars. A knowledge representation in the form of a content scheme does not only guide the parsing of the text, but can also prevail as a target representation of the abstract, which is often generated from the instantiated frames in the schemes or scripts.

A second group of techniques for text analysis and information selection takes advantages of the word distributions in texts and consists of more shallow statistical techniques. They originated in information retrieval research (indexing with natural language index terms), are weaker in terms of general results, but are more general with regard to the application domain. The statistical techniques are shallow, in the sense that they severely reduce the domain and linguistic knowledge needed for the analysis of the source text or learn this knowledge. Consequently, these techniques are more independent from the subject domain and text genre, and can be broadly applied. A major approach concerns the identification of important topic terms and the extraction of contextual sentences that contain them. A simple, but common approach extracts sentences that contain highly weighted terms possibly in close proximity. So, clusters of significant words within sentences are located and sentences are scored accordingly.

A promising technique for topic recognition and text summarization is the technique of lexical chains. A lexical chain is a chain of the same term, a synonym or related words or phrases as they occur in a document text. The synonyms and related words are detected with a thesaurus (e.g., WordNet).

> An example is having multiple occurrence of the phrase "cock fight" and its synonyms in an opinion part of a case, which might suggest that "cock fight" is an important concept in the motivation of the judge.

Anaphor resolution helps in detecting the true distribution of the terms. Strong chains (e.g., strength computed as a function of the length of the chain) represent the main topics of a text. The sentences of a text in which a representative chain member occurs the first time in the text is often useful in a summary extract (Barzilay and Elhadad 1999).

Another current approach to the selection of informative content regards the techniques that classify the discourse parameters involved in text summarization based on example abstracts of example source texts (e.g., Kupiec et al. 1995; Hovy and Lin 1999). Given a training corpus of source texts and their summaries (or important sentences annotated in the source texts) a number of summary-worthiness features are selected: length of a sentence, sentences containing indicator phrases, the first, final or medium sentences, sentences with high-weighted content words, and sentences with proper names that occur more than once, etc. Supervised training algorithms learn the importance of the features for the text type or subject domain of the training corpus, or for the user-specific summary (cf. above: the techniques for learning extraction patterns). The learning algorithms that are most commonly used are naive Bayes and learning of rules and trees.

A more difficult task to perform automatically is the *generalization* of the selected information. Generalization is condensing the information to a more abstract form. This task requires a bulk of semantic information. The generalization includes translating a single proposition into a more general one and to translate a sequence of propositions into a more general one. Thesauri and ontologies with semantic classifications of the lexical items are certainly needed (cf. Hahn 1990;

Hahn and Reimer 1999; McKeown and Radev 1999), but probably not sufficient to describe the information at a more general level.

> An example is generalizing a factual description of a legal case into a more general concept, such as "motorcycle carelessly driven" into "negligence", and to translate a sequence of propositions into a more general one, such as "entering a building", "sitting down", "looking at the menu", "ordering food", "eating and paying" into "restaurant visit".

The *synthesis step* is especially concerned with the organization of the content and its presentation in the summary. Function and audience of the summary determine relevancy in the source text and the format of the output. This step often bears upon the reformulation of the text in order to generate a summary that forms a complete, coherent, and comprehensible text, which is comparable to most manually created abstracts. This requires additional, mainly linguistic knowledge and techniques developed in the field of text generation. In between a profile and a perfectly coherent text, there are the many practical systems that perform some kind of editing of the sentences or of other text units extracted from the source text. But lawyers and legal professionals feel uncomfortable when the original text is changed and prefer to see the original phrasing of the text, when information is extracted (Uyttendaele et al. 1998).

In the legal field, summarization of cases is important for their easy selection and retrieval. Especially summarization of the cases' opinions is of great interest. Currently, abstracts are manually built as for instance the case abstracts of the JURIS data base in France. In legal documents there is a strong relationship between document structure and content. Knowledge of the discourse structures is important to automatically locate information in legal texts. There is the schematic structure that is important to recognize the broad passages of a text, there are the rhetorical connectives that indicate the relation between text segments or subclauses and there are the typical thematic patterns (Moens et al. 1999a). It is recognized that more studies about legal discourse and the signaling linguistic cues are useful for automatically processing the texts. The difficult task of generalization of the content is very important when summarizing legal texts and represents a very challenging research topic. One step further is generalizing the similarities between multiple document texts and detecting their differences. Such summaries are very useful for the retrieval of legal cases (see below).

5.5. A MODEL FOR CASE RETRIEVAL

Finding relevant cases and relevant kernels of information in the opinions of the cases is important when solving a legal problem. Lawyers reason with legal cases. They look for similar cases and for relevant similarities and differences between cases. They are also interested in these cases wherein the judge gives general abstract information on the application and interpretation of legislation. A promising

research direction is defining a retrieval model for legal cases that allows for a limited form of legal reasoning with semantically rich representations of the content of cases and information need.

As it is discussed in the foregoing, there are currently techniques to recognize important topics, to assign concepts to texts and to find detailed information in texts. The most advanced techniques use machine-learning techniques to acquire the knowledge involved and are actively researched. The result is the automatic generation of semantically rich representations of query and document text.

Research into retrieval models that reason with query and document representations when matching them, is very actual. The *logic-based retrieval model* (van Rijsbergen 1986; Nie 1989; van Rijsbergen 1989; Chiaramella and Nie 1990; Chiaramella and Chevallet 1992; Nie 1992; Lalmas 1998) assumes that queries and documents can be represented by logical formulas. The retrieval then is inferring the relevance of a document for a query. The above Boolean model is logic-based. But, the typical logic-based model will use the information in query and document in combination with extra knowledge that will be used by the matching function as part of proving that the document implies the query. Boolean logic is too restricted for this task. It cannot deal with temporal and spatial relationships, and especially not with contradictory information or uncertain information. Research into logic-based retrieval models is still in an early stage (Lalmas 1998).

Case-Based Reasoning (CBR) is an artificial intelligence technique in which the machine uses knowledge acquired from previous experiences for solving new, analogue problems. Upon input of a legal problem, a typical CBR-system in the legal field selects the most relevant decisions from a case database and automatically generates argumentation fragments based on the precedents in order to solve the new problem (Aleven 1999). Because such a case database is still manually implemented and indexed with a set of factors and a solution, this technology is not yet commercially used.

Techniques for information retrieval and Case-Based Reasoning might mutually complement when selecting relevant cases from a database. This is a hypothesis that is sustained by different authors in the literature (Ashley 1992; Rissland and Daniels 1996; Aleven 1999). However, few studies investigate an effective integration of these technologies. The current progress in an automatic identification of the content of texts and the development of retrieval models that reason with document and query allow for a cross-fertilization of information retrieval and Case-Based Reasoning. The retrieval of legal cases is a first-class example of such an application.

6. Conclusions

Legal information is mostly found in the form of natural language texts. When automatically retrieving this information, a large amount of linguistic, domain, and communicative knowledge is needed in order to correctly identify the content

of document texts and information need and in finding their correspondence. The growth of the document collections and the costs of indexing by specialized people in the legal field provoke a large interest into better automatic techniques.

In the near future we may expect real progress in legal text retrieval. It then might be possible that queries and documents are better understandable by machine. There is the explosion of machine learning techniques that acquire the knowledge involved in retrieval, such as the learning of user's profiles and content attributes of texts. The techniques regard both supervised and unsupervised approaches. Innovative techniques for text categorization, information extraction, text clustering and text summarization are being developed. In addition, case retrieval can be improved by a cross-fertilization of Case-Based Reasoning techniques and current retrieval models.

References

Agosti, M. and Smeaton, A. F. (eds.) (1996). Information Retrieval and Hypertext. Boston: Kluwer Academic Publishers.

Aleven, V. (1999). Case-Based Reasoning. In Oskamp, A. and Lodder, A. R. (eds.) Informatietechnologie voor juristen. Handboek voor de jurist in de 21ste eeuw, 211–228. Deventer: Kluwer

Ashley, K. D. (1992). Case-Based Reasoning and Its Implications for Legal Expert Systems. Artificial Intelligence and Law 1: 113–208.

Aslam, J., Reiss, F., and Rus, D. (2000). Scalable Information Organization. In Proceedings RIAO'2000 Content-Based MultiMedia Information Access Collège de France, Paris, France 12–14 April 2000. Paris: CID – CASIS.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). Modern Information Retrieval. Harlow, UK: Addison Wesley.

Barzilay, R. and Elhadad, M. (1999). Using lexical chains for text summarization. In Mani, I. and Maybury, M. T. (eds.) Advances in Automatic Text Summarization, 111–121. Cambridge, MA: MIT Press.

Belkin, N. J. and Croft, W. B. (1992). Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of the ACM 35(12): 29–48.

Bing, J. (ed.) (1984). Legal Information Retrieval. Butterworths: North Holland.

Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford: Clarendon Press.

Blair, D. C. (1990). Language and Representation in Information Retrieval. Amsterdam: Elsevier Science Publishers.

Blair, D. C. and Maron, M. E. (1985). An Evaluation of Retrieval Effectiveness for a Full-text Document-retrieval System. Communications of the ACM 28(3): 289–299.

Blair, D. C. and Maron, M. E. (1990). Full-text Information Retrieval: Further Analysis and Clarification. Information Processing & Management 26: 437–447.

Boguraev, B. K. and Neff, M. S. (2000). Lexical Cohesion, Discourse Segmentation and Document Summarization. In Proceedings RIAO'2000 Content-Based MultiMedia Information Access Collège de France, Paris, France 12–14 April 2000. Paris: CID – CASIS.

Bradshaw, J. M. (ed.) (1997). Software Agents. Menlo Park, CA: AAAI Press.

Brüninghaus, S. and Ashley, K. D. (1997). Finding Factors: Learning to Classify Case Opinions under Abstract Fact Categories. In Proceedings of the Sixth International Conference on Artificial Intelligence and Law, 123–131. New York: ACM.

Brüninghaus, S. and Ashley, K. D. (1999). Toward Adding Knowledge to Learning Algorithms for Indexing Legal Cases. In Proceedings of the Seventh International Conference on Artificial Intelligence and Law, 9–17. New York: ACM.

Buckley, C. and Salton, G. (1995). Optimization of Relevance Feedback Weights. In Fox, E. A., Ingwersen, P., and Fidel, R. (eds.) Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 351–357. New York: ACM.

Carbonell, J. G., Yang, Y., Frederking, R. E., Brown, R. D., Geng, Y., and Lee, D. (1997). Translingual Information Retrieval: A Comparative Evaluation. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 708–728. San Francisco, CA: Morgan Kaufmann.

Chiaramella, Y. and Chevallet, J. P. (1992). About Retrieval Models and Logic. The Computer Journal 35(3): 233–242.

Chiaramella, Y. and Nie, J. (1990). A Retrieval Model Based on Extended Modal Logic and Its Application to the RIME Experimental Approach. In Vidick, J.-L. (ed.) Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 25–43. New York: ACM.

Conklin, J. (1987). Hypertext: an Introduction and Survey. IEEE Computer 20(9): 17–41.

Coulmas F. (1989). The Writing Systems of the World. Oxford, UK: Basil Blackwell.

Cowie, J. and Wilks, Y. (2000). Information Extraction. In Dale, R., Moisl, H., and Somers, H. (eds.) Handbook of Natural Language Processing, 241–260. New York: Marcel Dekker.

Cowie, J., Ludovik, E., Molina-Salgado, H., Nirenburg S., and Scheremetyeva. S. (2000). Automatic Question Answering. In Proceedings RIAO'2000 Content-Based MultiMedia Information Access Collège de France, Paris, France 12–14 April 2000. Paris: CID – CASIS.

Craven, M. et. al. (1998). Learning to Extract Symbolic Knowledge from the World Wide Web. In Proceedings of the Tenth Conference on Innovative Applications of Artificial Intelligence (IAAI-98). Menlo Park, CA: AAI Press/The MIT Press

Croft, W. B. (1980). A Model of Cluster Searching Based on Classification. Information Systems 5: 189–195.

Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. (1992). Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In Belkin, N. J., Ingwersen, P., and Pejtersen, A. M. (eds.) Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 318–329. New York: ACM.

Danet, B. (1985). Legal Discourse. In van Dijk, T. A. (ed.), Handbook of Discourse Analysis 1, 273–291. London: Academic Press.

Deerwester, S., Dumais, S. T., Furnas, G.W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science 41(6): 391–407.

DeJong, G. (1982). An Overview of the FRUMP System. In Lehnert, W. G. and Ringle, M. H. (eds.) Strategies for Natural Language Processing, 149–176. Hillsdale: Lawrence Erlbaum.

Dick, J. P. (1987). Conceptual Retrieval and Case Law. In Proceedings of the First International Conference on Artificial Intelligence and Law, 106–114. New York: ACM.

Fuhr, N. (1992). Probabilistic Models in Information Retrieval. The Computer Journal 35(3): 243–255.

Gaizauskas, R. and Humphreys, K. (2000). A Combined IR/NLP Approach to Question Answering Against Large Text Collections. In Proceedings RIAO'2000 Content-Based MultiMedia Information Access Collège de France, Paris, France 12–14 April 2000. Paris: CID – CASIS.

Gey, F. C. (1994). Inferring Probability of Relevance Using Methods of Logistic Regression. In Croft, W. B. and van Rijsbergen, C. J. (eds.) Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 222–231. London: Springer.

Graesser, A. C. and Clark, L. F. (1985). Structures and Procedures of Implicit Knowledge (Advances in Discourse Processes, XVII). Norwood, NJ: Ablex Publishing Corporation.

Griffiths, A., Luckhurst, H. C, and Willett, P. (1986). Using Interdocument Similarity Information in Document Retrieval Systems. Journal of the American Society for Information Science 37(1): 3–11.

Hafner, C. D. (1987). Conceptual Organization of Case Law Knowledge Bases. In Proceedings of the First International Conference on Artificial Intelligence and Law, 35–42. New York: ACM.

Hahn, U. (1990). Topic Parsing: Accounting for Text Macro Structures in Full-text Analysis. Information Processing & Management 26(1): 135–170.

Hahn, U. and Reimer, U. (1999). Knowledge-based Text Summarization: Salience and Generalization Operators for Knowledge Base Abstraction. In Mani, I. and Maybury, M. T. (eds.) Advances in Automatic Text Summarization, 215–232. Cambridge, MA: MIT Press.

Hand, D. J. (1997). Construction and Assessment of Classification Rules. Chichester: John Wiley & Sons.

Hearst, M. A. (1997). TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. Computational Linguistics 23 (1): 33–64.

Hearst, M. A. and Pedersen, J. O. (1996). Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In Frei, H.-P., Harman, D., Schaüble, P., and Wilkinson, R. (eds.) Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 76–84. New York: ACM.

Hovy, E. and Lin, C.-Y. (1999). Automated Text Summarization in SUMMARIST. In Mani, I. and Maybury, M. T. (eds.) Advances in Automatic Text Summarization, 81–94. Cambridge, MA: MIT Press.

Jacobs, P. S. (ed.) (1992). Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval. Hillsdale, NJ: Lawrence Erlbaum.

Jones, W. P. and Furnas, G. W. (1987). Pictures of relevance: a geometric analysis of similarity measures. Journal of the American Society for Information Science 38(6): 420-442.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In European Conference on Machine Learning (ECML).

Kaufman, L. and Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & Sons.

Kittredge, R. and Lehrberger, J. (eds.) (1982). Sublanguage: Studies of Language in Restricted Semantic Domains. Berlin: W. de Gruyter.

Kupiec, J., Pedersen, J., and Chen, F. (1995). A Trainable Document Summarizer. In Fox, E. A., Ingwersen, P., and Fidel, R. (eds.) Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 68–73. New York: ACM.

Lalmas, M. (1998). Logical Models in Information Retrieval: Introduction and Overview. Information Processing & Management 34(1): 19–33.

Lederer, F. I. (1996). Technology Augmented Litigation. In Proceedings of the First European Conference on Law, Computers and AI Exeter April 15–16, 1996, 70-81.

Leuski, A. and Allan, J. (2000). Improving Interactive Retrieval by Combined Ranked Lists and clustering. In Proceedings RIAO'2000 Content-Based MultiMedia Information Access Collège de France, Paris, France 12–14 April 2000. Paris: CID – CASIS.

Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. IBM Journal of Research and Development 1(4): 309–317.

Manning, C. D. and Schütze, H. (1999). Foundations of Statistical Natural Language Processing. Cambridge, MA: MIT Press.

Marcu, D. (1999). Discourse Trees Are Good Indicators of Importance in Text. In Mani, I. and Maybury. M. T. (eds.) Advances in Automatic Text Summarization, 123–136. Cambridge, MA: MIT Press.

Maron, M. E. and Kuhns, J. L. (1960). On Relevance, Probabilistic Indexing and Information Retrieval. Journal of the Association for Computing Machinery 7(3): 216–244.

Masand, B., Linoff, G., and Waltz, D. (1992). Classifying News Stories Using Memory Based Reasoning. In Proceedings of the Fifteenth SIGIR Conference, 59–65. New York: ACM.

McCallum, A., Nigam, K., Rennie, J., and Seymore, K. (1999). A Machine Learning Approach to Building Domain-specific Search Engines. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 662–667. San Mateo, CA: Morgan Kaufmann.

McKeown, K. and Radev, D. R. (1999). Generating Summaries of Multiple News Articles. In Mani, I. and Maybury, M. T. (eds.) Advances in Automatic Text summarization, 381–399. Cambridge, MA: MIT Press.

Merkl, D. and Schweighofer, E. (1997). The Exploration of Legal Text Corpora with Hierarchical Neural Networks: A Guided Tour in Public International Law. In Proceedings of the Sixth International Conference on Artificial Intelligence and Law, 98–105. New York: ACM.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (eds.) (1994). Machine Learning, Neural and Statistical Classification. New York: Ellis Horwood.

Mitchell, T. M. (1997). Machine Learning. Boston, MA: McGraw-Hill.

Moens, M.-F. (2000). Automatic Indexing and Abstracting of Document Texts (The Kluwer International Series on Information Retrieval 6). Boston: Kluwer Academic Publishers.

Moens, M.-F., Gebruers, R., and Uyttendaele, C. (1996). SALOMON: Final Report. Technical Report ICRI, K.U. Leuven.

Moens, M.-F., Uyttendaele, C., and Dumortier, J. (1999a). Information Extraction from Legal Texts: The Potential of Discourse Analysis. International Journal of Human-Computer Studies 51: 1155–1171.

Moens, M.-F., Uyttendaele, C., and Dumortier, J. (1999b). Abstracting of Legal Cases: The Potential of Clustering Based on the Selection of Representative Objects. Journal of the American Society for Information Science 50(2): 151–161.

Moens, M.-F. and Dumortier, J. (2000). Text Categorization: The Assignment of Subject Descriptors to Magazine Articles. Information Processing & Management 36, 841–861.

MUC-7 (1999). Proceedings of the Seventh Message Understanding Conference. San Mateo: Morgan Kaufmann.

Nie, J. (1989). An Information Retrieval Model Based on Modal Logic. Information Processing & Management 25(5): 477–494.

Nie, J.-Y. (1992). Towards a Probabilistic Modal Logic for Semantic Based Information Retrieval. In Belkin, N. J., Ingwersen, P., and Pejtersen, A. M. (eds.) Proceedings of the Fifteenth ACM SIGIR Conference on Research and Development in Information Retrieval, 140–151. New York: ACM.

Nielsen, J. (1995). Multimedia and Hypertext: The Internet and Beyond. Boston: AP Professional.

Nilsson, N. J. (1990). The Mathematical Foundations of Learning Machines. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.

Raghaven, V. V. and Wong, S. K. M. (1986). A Critical Analysis of Vector Space Model for Information Retrieval. Journal of the American Society for Information Science 37(5): 279–287.

Riloff, E. (1996). An Empirical Study for Automated Dictionary Construction for Information Extraction in Three Domains. Artificial Intelligence 85: 101–134.

Rissland, E. L. and Daniels, J. J. (1996). The Synergistic Application of CBR to IR. Artificial Intelligence Review 10(5/6): 441–475.

Robertson, S. E. and Sparck Jones, K. (1976). Relevance Weighting of Search Terms. Journal of the American Society for Information Science 27(3): 129–146.

Robertson, S. E. and Walker, S. (1994). Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In Croft, W. B. and van Rijsbergen, C. J. (eds.)

Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 232–241. London: Springer.

Rocchio, J. J. (1971). Relevance Feedback in Information Retrieval. In Salton, G. (ed.) The SMART Retrieval System: Experiments in Automatic Document Processing, 313–323. Englewood Cliffs, NJ: Prentice Hall.

Salton, G. (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Reading, MA: Addison-Wesley.

Salton, G. and Buckley C. (1990). Improving Retrieval Performance by Relevance Feedback. Journal of the American Society for Information Science 41(4): 288–297.

Salton, G., Singhal, A., Mitra, M., and Buckley, C. (1997). Automatic Text Structuring and Summarization. Information Processing & Management 33(2): 193–207.

Schank, R. C. (1975). Conceptual Information Processing. Amsterdam: North Holland.

Schweighofer, E. and Merkl, D. (1999). A Learning Technique for Legal Document Analysis. In Proceedings of the Seventh International Conference on Artificial Intelligence and Law, 156–163. New York: ACM.

Soderland, S. (1999). Learning Information Extraction Rules for Semi-structured and Free Text. Machine Learning 34(1/3): 233–272.

Sparck Jones, K. (1991). The Role of Artificial Intelligence in Information Retrieval. Journal of the American Society for Information Science 42(8): 558–565.

Sparck Jones, K. (1993). What Might Be in a Summary? In Knorz, G., Krause, J., and Womser-Hacker, C. (eds.) Information Retrieval '93: Von der Modellierung zur Anwendung 9–26. Konstanz: Universitätsverlag.

Sparck Jones, K., Walker, S, and Robertson, S. E. (2000). A Probabilistic Model of Information Retrieval: Development and Comparative Experiments. Information Processing & Management 36(6): 779–840.

Sperber, D. and Wilson, D. (1995). Relevance: Communication and Cognition (2nd edition). Oxford, UK: Basil Blackwell.

Strzalkowski, T., Stein, G. C., Bowden, G., and Bagga, A. (2000). Towards the Next Generation Information Retrieval. In Proceedings RIAO'2000 Content-Based MultiMedia Information Access Collège de France, Paris, France 12–14 April 2000. Paris: CID – CASIS.

Turtle, H. (1995). Text Retrieval in the Legal World. Artificial Intelligence and Law 3: 5–54.

Turtle, H. R. and Croft, W. B. (1992). A Comparison of Text Retrieval Models. The Computer Journal 35(3): 279–290.

Uyttendaele, C., Moens, M.-F., and Dumortier, J. (1998). SALOMON: Abstracting of Legal Cases for Effective Access to Court Decisions. Artificial Intelligence and Law 6: 59–79.

Van Rijsbergen, C. J. (1979). Information Retrieval (2nd ed.). London: Butterworths.

Van Rijsbergen, C. J. (1986). A Non-classical Logic for Information Retrieval. The Computer Journal 29: 111–134.

Van Rijsbergen, C. J. (1989). Towards an Information Logic. In Belkin, N. J. and van Rijsbergen, C. J. (eds.) Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 77–86. New York: ACM.

Voorhees, E. (1985). The Cluster Hypothesis Revisited. In Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 95–104. New York: ACM.

Wang, Z. W., Wong, S. K. M., and Yao, Y. Y. (1992). An Analysis of Vector Space Models Based on Computational Geometry. In Belkin, N. J., Ingwersen, P., and Pejtersen, A. M. (eds.) Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 152–160. New York: ACM.

Weiss, S. M. et al. (1999). Maximizing Text-mining Performance. IEEE Intelligent Systems July–August 1999: 63–69.

Willett, P. (1988). Recent Trends in Hierarchic Document Clustering: A Critical Review. Information Processing & Management 24(5): 577–597.

Winkels, R., Bosscher, D., Boer A., and Hoekstra, R. (2000). Extended Conceptual Retrieval. In Legal Knowledge and Information Systems: Jurix 2000: The Thirteenth Annual Conference, 85–97. Amsterdam: IOS Press.

Wong, S. K. M., Ziarko, W., and Wong, P. C. N. (1985). Generalized Vector Space Model in Information Retrieval. In ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '85), 18–25. New York: ACM.

Yang, Y. and Liu, X. (1999). A Re-examination of Text Categorization Methods. In Proceedings of the 22nd International Conference on Research and Development in Information Retrieval, 42–49. New York: ACM.