



SALOMON : FINAL REPORT



K.U. Leuven - November 1996

Preface

PREFACE	ii
1. INTRODUCTION	1
1.1. Purpose and Scope of the Project	1
1.2. General Approach	3
1.3. Organisational Issues	4
2. PART 1: LINGUISTIC ASPECTS	6
2.1. Background	6
2.1.1. Overview	6
2.1.2. The 'Open Texture' of Law	6
2.1.3. The Need for Automated Legal Support	7
2.1.4. Computer Meets Language Barrier	8
2.1.5. Sublanguages and Controlled Languages	9
2.1.6. The Language of Legal Documents	10
2.2. Linguistic Module	13
2.2.1. Introduction	13
2.2.2. Measuring the Vocabulary of the SALOMON Corpus	13
2.2.3. Sentence Delimiting	16
2.2.4. Part-of-Speech Tagging	18
2.2.5. Term Conflation	19
2.3. Overall Conclusion	20
3. PART 2: AUTOMATIC GENERATION OF A CASE PROFILE	22
3.1. Background	22
3.1.1. Introduction	22
3.1.2. Research in Automatic Abstracting	23
3.1.3. Approaches Relying on the Surface Features of the Texts (Shallow Source Representation)	24
3.1.4. Approaches Relying on Additional Knowledge (Deep Source Representation)	25
3.1.5. Important Research Strategies in Automatic Abstracting	27
3.1.6. Research in Automatic Categorisation of Text	29
3.1.7. Generation of Document Profiles in the Legal Field	29
3.2. Methods	30
3.2.1. Legal Relevance of the Corpus Analysis	30
3.2.2. Starting Point: the Manual Practice	31
3.2.3. Potential and Limits of Automatic Abstracting	32
3.2.4. Initial Structuring of the Cases Based on a Text Grammar	32
3.2.5. Recognition of the Topic Structure and Representative Text Units Based on Statistical Methods	36
3.2.6. Categorisation of the Alleged Offences	39
3.3. Results and Discussion	39
3.3.1. Global Architecture of the Demonstrator	40
3.3.2. Initial Structuring	40
3.3.3. Identification of Representative Text Units of the Alleged Offences and Opinion of the Court	45

3.3.4. Contributions of the Research	48
3.3.5. Examples of Case 'Index Cards'	49
3.3.6. Towards the Future	53
4. GENERAL CONCLUSIONS	55
5. LITERATURE	57
6. TECHNICAL REPORTS	64
7. LECTURES	67
8. PUBLICATIONS	68

1. Introduction

This report covers the research conducted for the SALOMON project from October 1993 till October 1996. The SALOMON project is a *contribution to the automatic processing of legal textual information*. This introductory chapter further reveals the purpose and scope of the project, describes the general approach, and deals with organisational issues. The body of this report treats two basic topics: the linguistic aspects of the SALOMON project and the automatic generation of a case profile. The two major parts each conclude with an overall appreciation of the work performed and with some recommendations for future research. Readers interested in further details are referred to the relevant technical reports and publications listed at the end of the report.

1.1. Purpose and Scope of the Project

The SALOMON project *initially* concerned a *computer-assisted analysis into the characteristics of legal language*, the results of which would be directly relevant to e.g. legal information retrieval, legal knowledge representation and machine translation of documents. The legal texts on which the analysis would bear are court decisions.

The feasibility of this analysis largely relied on the initial hypothesis that the legal language can be considered as a *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 & Lehrberger, 1992). At a higher level of analysis, it may be seen that the grammar of this sublanguage reflects the information structure of discourse in the subfield, while the semantic classes of words used and the semantic relations between these classes reflect the knowledge structure of the subfield. The fact that a sublanguage deals with a restricted subject domain and is used for communication of specific information results in some rather helpful restrictions on the range of linguistic data that needs be accounted for in a sublanguage analyser (Liddy, Jörgenson, Sibert, & Yu, 1991)

As it is further extensively explained (see *Linguistic Aspects / Background*) the language of the court decisions can not be considered as a sublanguage. The language of the court decisions has many characteristics of ordinary language. Only, court decisions usually exhibit a specific text structure in the organisation of the broad semantic units of which the case is composed. These semantic units are characterised by a limited set of linguistic cues and by a limited set of relations between them (see *Automatic Generation of a Case Profile / Methods / Legal Relevance of the Corpus Analysis*).

So, returning to the initial goal of SALOMON, it would be an enormous task to make a computer-assisted analysis of the legal language. This would also entail to explore in much more detail what is about the legalese of court decisions that makes automatic understanding of these 'decisions' such a hard problem. Such a goal would be long-range research, involving a substantial body of linguistic description and with no immediate prospect of capitalising on the research results. However, considering an *urgent need for automated legal support* and the possibilities for additional financing, it seemed advisable to explore other research avenues.

With the advent of computers in law courts and in offices of public prosecutors, massive volumes of information are now available at low cost in free text form. Yet, people cannot read and digest this information any faster than before; in fact, for the most part they can digest even less. On the other hand, the availability of court decisions in electronic form opens up new vistas for a more comprehensive overview of existing jurisdiction. Up to the recent past, only a fraction of all pronounced decisions was made available through specialised journals and legal information services (e.g. JUSTEL, CREDOC, ...), and the principles for selection are far from transparent. The disposition of a full overview of jurisprudence has great additional value for policy purposes. It allows an immediate evaluation and, if necessary, re-orientation by the government of its policy.

Most current *legal Information Retrieval (IR) systems* allow the user to search for keywords using a syntax that is largely based on Boolean queries. These systems provide little or no assistance to the user in

forming a query; rather, they expect the user to fully master a complicated syntax, which may not be equally acceptable to all potential users. Also, the search output often results in the retrieval of many irrelevant cases. This amount of 'output overload' may be reduced, and the amount of precision increased, by refining the query. However, each refinement may in turn increase the probability that some relevant documents will be excluded by the reformulated query (Blair & Maron, 1985). The inherent shortcomings of Boolean retrieval sparked off research into alternative retrieval models. Most prominent are Salton's vector space model, developed in the Smart project (Salton, 1971, 1989) as well as various proposals for probabilistic IR (Fuhr, 1992), which provide a ranking of the documents according to their degree of relevance to the query. These alternative methods are no doubt significant improvements to the Boolean model. Yet, the final determination of what is or is not relevant in the system's output can only be done by the user, taking into account his information needs and current background knowledge. An electronic search may point to a very large set of lengthy documents which often must be examined while the clock is ticking. Hence, it is of paramount importance to supply the user with tools to skim through the information in order that the user can decide fairly quickly whether it would be worthwhile to look at the full text of the document. Moreover, Croft, Krovetz, and Turtle (1990) demonstrate that users often query documents in terms they are familiar with, and these terms are frequently not the terms in the document itself, which prohibits the users from retrieving the relevant documents. A survey of Allen (1990) confirms the importance of consulting *document 'profiles' in the search for relevant documents*.

Being able to make efficient use of information from court decisions requires that the key information is accessible in some sort of structured format. Of course, examples of such structured information already exist, in the form of case headnotes, abstracts, indexes and digests as provided by legal information publishers. However, there are numerous problems with 'document surrogates' manually drafted: they require extensive and expensive human time and expertise to prepare; they always come with some delay; they are biased by the viewpoint of their authors; and they are rarely consistent both between authors and with respect to time. Hence on-line court decisions create a need for automatic text processing methods to automatically extract the essence of court decisions.

The *purpose* of the SALOMON project, then, was *to investigate and select existing methods for identifying key information in texts, to possibly refine the methods and create new methods, and to test and evaluate the application of such methods upon legal court decisions in a demonstrator*. Such a goal would necessarily, although selectively, entail an analysis into the characteristics of legal language of the court decisions.

More specifically, the SALOMON project assesses the feasibility of automatically *mapping relevant sections from the full text of court decisions onto structured document profiles*. The decisions were drawn from a corpus of criminal law, consisting of more than 3000 decisions pronounced by the correctional court in Leuven between January 1992 and June 1994. There were several reasons for choosing this corpus:

1. it was one of the few Belgian legal corpora (written in Dutch) available in machine-readable form: criminal law is a test case for the computerisation of Belgian law courts;
2. (Belgian) criminal law cases have a typical and explicit text structure;
3. (Belgian) criminal law is clearly structured.

The document profiles produced in the mapping process should resemble the manually constructed headnotes of printed law reports. In particular, they should contain the following information:

1. the name of the court that pronounced the decision;
2. the date of the decision;
3. the type of the criminal offence that is the object of the decision;
4. the key paragraphs and key concepts that appear to express the essence of the opinion of the court¹;
5. references to the statutory provisions which the court deems applicable to the case at hand;
6. a pointer to the full text of the court decision.

Of course, many court decisions contain additional information that may be of potential interest to the jurist, such as the factual details of the offence, the presence of aggravating circumstances (e.g. recidivism), and the penalty which the court considers to be appropriate punishment. However, in order not to exceedingly complicate matters, the research team decided to ignore such additional issues for the time being.

¹ Equivalent nomenclature of the opinion of the court is *ratio decidendi* and *considerans*.

An evaluation of the results of the demonstrator also reveals the benefits and shortcomings of the methods chosen. The latter may be alleviated when the legal world establishes *standards for text creation of legal cases*. As a useful *side-effect* the SALOMON research gives insight when such standardisation may be useful.

1.2. General Approach

The initial intention of the SALOMON project was to examine a variety of techniques that could be used to effect the desired generation of case profiles. To begin with, a profound analysis of the correctional cases was carried out. First, the language of legal texts was investigated (see *Linguistic Aspects / Background*). Statistical methods were employed for measuring the vocabulary of the SALOMON corpus (see *Linguistic Aspects / Measuring the Vocabulary of the SALOMON Corpus*). Also, the correctional cases were extensively manually studied, revealing the typical text structure of the semantic units of which a case is composed of and the specific textual cues that characterise these units (see *Automatic Generation of a Case Profile / Methods / Legal Relevance of the Corpus Analysis*).

Further, it was useful to consider the manual process of abstracting legal cases, not only for defining the desired output, but also in finding appropriate techniques, which may be automated. Studies of human abstracting in general and in the legal field are found in *Automatic Generation of a Case Profile / Methods / Starting Point: the Manual Practice*.

The research did not overlook the problems of understanding natural language texts (see *Linguistic Aspects / Background / Computer Meets Language Barrier*). Current and past research in automatic abstracting, indexing and categorisation of text was extensively studied (see *Automatic Generation of a Case Profile / Background*).

A *summary* is a condensed derivative of the source text, i.e. a content reduction to either *selection or generalisation* on what is important in the source (Sparck Jones, 1993). The general process of automatic abstracting can be described as the transformation of an abstract representation of the source text, containing the necessary attributes for summarisation into a summary representation embodying the organised content of the summary. Document abstracts, generated automatically, generally belong to two types. Firstly, the abstract is constructed for easy and fast determination of relevance: it indicates at a single glance whether or not the complete text version is of interest (*indicative abstract*). Secondly, the abstract is a document surrogate expressing the main contents of the document: its components may be used for text search and linking (*informative abstract*). In this way abstracting has some *relation with indexing*. A very brief summary may serve as a complex structured index description (Sparck Jones, 1993). The summary components (e.g. words, phrases and sentences) can be used as indices or keys for accessing the information of the text. At present the majority of abstracts automatically generated are document extracts. It has been shown that document extracts consisting of 20% or less of the original may be as informative as the full text (Kupiec, Pedersen, & Chen, 1995).

The SALOMON team realised that *complete understanding of the text* of the court decision and generation of a coherent case abstract, was technically *out of reach*. This would involve an interpretation of the source document based on a formal representation of the significance of entire sentences and integration of sentence analyses into an overall source meaning representation, condensation of the source meaning by selection and/or generalisation on important content; and generation of the headnote. Such a goal would also imply more detailed studies of discourse comprehension, especially that involving abstracting, which provide ideas for identifying and organising the requisite knowledge in such a manner that it can be effectively brought to bear on the problems of text understanding and text generation. Moreover, such an approach would require a substantial additional effort to build the necessary knowledge bases. The problem is not only that of storing increasingly more knowledge in computer memories, but also that of having the computer mobilise the appropriate knowledge at the appropriate time (Jacobs & Rau, 1993, p. 173 ff.).

However, it was recognised that *document profiles containing relevant text units extracted from the case and containing case categories* automatically attributed make court decisions more accessible. Moreover, such profiles broadly correspond with the headnotes currently generated in a manual way, and thus their automatic generation corresponds to the goal of the project. These document profiles have indicative as well as informative properties. So the research objective in this phase was to concentrate on techniques that identify relevant text units. This also includes establishing criteria how to define relevancy of the textual parts of the

case, which in the legal field entails its specific problems (see *Automatic Generation of a Case Profile / Methods / Potential and Limits of Automatic Abstracting*).

The research moved along two strategies, cited by Sparck Jones (1993) as worthwhile pursuing in making progress in automatic abstracting. First, *text structure* is important when accessing the content of a text. For modelling the text structure of the different text types and for relying on it for text processing tasks such as text generation, abstracting and retrieval, we may build on realisations in natural language text processing (see *Automatic Generation of a Case Profile / Methods / Initial Structuring of the Cases Based on a Text Grammar*). Secondly, the progress made in *information retrieval*, especially the current refinement and sophistication of *statistical techniques* developed in this domain, may be fertile for automatic abstracting (see *Automatic Generation of a Case Profile / Methods / Recognition of the Topic Structure and Representative Text Units Based on Statistical Methods*). Since the end of the 1960's, not much attempts have been made to incorporate statistical techniques in automatic abstracting. Nevertheless since then, new techniques have been developed and are still developing. For categorising text, a well-established method regarding the automatic learning of categories from examples was adapted (see *Automatic Generation of a Case Profile / Methods / Categorisation of the Alleged Offences*). Finally, it was realised that following these strategies some of the sub-tasks of human abstracting were automated.

To realise the above goals, a demonstrator was built in the programming language C on a Sun™ SPARC station 5 (32 Mb RAM, 1 Gb HD, 85 MHz) under Solaris® 2.3 (see *Automatic Generation of a Case Profile / Results and Discussion / Global Architecture of the Demonstrator*). In addition to robustness of the software, portability of the techniques to other legal domains and beyond, and portability of the software modules to other hardware platforms, the main requirement of this demonstrator was to seek a high recall and precision² of the information figuring on the document profiles. To test and evaluate this requirement, a test set and appropriate evaluation procedures were established (see *Automatic Generation of a Case Profile / Results and Discussion / Initial Structuring & Identification of Representative Text Units of the Alleged Offences and Opinion of the Court*). The test set has been carefully selected in a way that the set is homogeneous and representative for the complete corpus of correctional cases. The evaluation procedures are borrowed from the fields of text categorisation and text extraction. This evaluation involves an assessment of the methods used, gives guidelines for further sophistications of the methods (see *Automatic Generation of a Case Profile / Results and Discussion / Contributions of the Research*), and gives insight into which future processes may be useful during the creation of the electronic text, facilitating a subsequent electronic use of the texts (see *Automatic Generation of a Case Profile / Towards the Future*).

It was clear that extraction of relevant text excerpts and categorisation of the cases may be improved by using *natural language processing tools*. Several tools were tested and adapted for the SALOMON project (see *Linguistic Aspects / Linguistic Module*). These include the tools for the recognition of syntactic word categories (see *Linguistic Aspects / Linguistic Module / Part of Speech Tagging*) and the reduction of word variants to a common root (see *Linguistic Aspects / Linguistic Module / Term Conflation*), which have a definite potential for index term selection and computation of the relative importance of index terms. Additionally, the application of the natural language processing tools may give insight into the legal language, the initial research goal of SALOMON.

1.3. Organisational Issues

The SALOMON project was performed at the *Interdisciplinary Centre for Law and Information Technology*, by a team consisting of a *computational linguist* (Dr. Rudi Gebruers), a *legal expert* (Caroline Uyttendaele), and a *computer scientist* (Marie-Francine Moens). The project was supervised by Prof. Dr. Jos Dumortier. The division of labour was as follows:

1. The computational linguist studied the properties of legal language. He compiled frequency lists for the words contained in the SALOMON text corpus, and studied the corpus-specific distribution of frequency classes. Furthermore, he examined and developed techniques for identifying and normalising valuable content indicators. More specifically, he designed and implemented a tokenisation module for marking sentence boundaries, separating punctuation marks from 'ordinary'

² Recall and precision are metrics employed in the domains of information retrieval, text categorisation, and text extraction. Generally, recall measures the proportion of relevant materials retrieved and precision measures the proportion of retrieved materials that are relevant (Salton, 1989, p. 248).

words, and for identifying 'fixed' syntagms; a module for annotating words with part-of-speech tags that reflect the syntactic status of these words and which provide clues for the automatic identification of word groups; and a module for reducing word variants to a common root by intelligently removing affixes.

2. After assessing the needs of potential user groups, the legal expert defined input/output-behaviour expected from the demonstrator and provided feedback on the methods used by human abstractors when drafting legal headnotes. She analysed and described the structure and content of the correctional cases in a detailed and exhaustive way. She charted the text features that were used for dividing the texts of the court decisions into legally relevant segments and for identifying routine-like passages. Also, she manually developed a classification of criminal law cases. Finally, she was in charge of evaluating the overall juridical accuracy of the output of the demonstrator.

3. As part of the preparation of her doctoral dissertation, the computer scientist studied the topics of automatic abstracting and indexing. She was in charge of the global system architecture of the demonstrator. She designed a domain independent formalism for representing text structure. She designed and implemented a parser to identify text structure based upon the knowledge coded in the formalism. Both the legal expert and she formally implemented the knowledge related to the text structure of the correctional cases and designed new techniques for identifying informative text units in texts. More specifically, the computer scientist designed and implemented modules for statistical weight calculation of index terms, for constructing and manipulating vector representations of texts, for comparing and clustering text representations, for extracting text components, and for categorising text fragments based on examples. She also was in charge of defining procedures for evaluating the output of the demonstrator.

The results of the demonstrator were evaluated by Tine Bouwen, a student entering her final year of law school. The SALOMON team could also count on the intellectual support of Dr. Jerry Hobbs (Stanford Research Institute, CA, USA) and Prof. Dr. John Zeleznikow (La Trobe University, Australia).

The SALOMON project was financed by the 'Onderzoeksfonds K.U. Leuven' and the 'Nationaal Fonds voor Wetenschappelijk Onderzoek'.

2. PART 1: Linguistic Aspects³

2.1. Background

2.1.1. Overview

This background chapter starts from the observations that law and ordinary language are intimately related and that there is an urgent need for automated legal support. It is argued that, while ordinary language as a whole is a nightmare for computers, certain linguistic subvarieties are more amenable to computational processing than others, in the sense that only the former can be subjected to a radical semantic analysis which should enable computer systems to generate appropriate responses. Unfortunately, the language of court decisions does not seem to be one of these subvarieties. One possible research avenue would be to explore in much more detail what it is about the legalese of court decisions that makes automatic understanding of these decisions such a 'hard' problem. Arguably, this would be long-range research, involving a substantial body of linguistic description and with no immediate prospect of capitalising on the research results. Considering the urgent need for automated legal support and the possibilities for additional financing, it was decided that the SALOMON team should explore other research avenues.

2.1.2. The 'Open Texture' of Law

Law and language are intimately related. To begin with, legal sources, i.e. legislation, statutes, cases and doctrine, are expressed in a natural language like English and Dutch. Someone who wishes to consult these sources obviously must have a good command of the language in which they are expressed. However, certain characteristics of natural language are essential to legal transactions in a much deeper sense. Arguably, any attempt at representing legal issues without taking account of these characteristics is seriously misguided.

Legal control is primarily control by rules referring to a general *type* of conduct, applying to a *class* of persons, and limited to a *class* of circumstances. Hence the communication of such a rule requires the use of general classifying words, fixing the necessary conditions which anything must satisfy if it is to be within the scope of the rule. However, the legislator cannot anticipate what the future may bring. Therefore, the rule should not be so detailed that the question whether it applies or not to a particular case would *always* be settled in advance. Of course, a legal rule would be useless unless there were plain cases where its applicability is indisputable. Still, one might want to maintain the rule also for situations resembling the plain cases in some respects, but differing from them in others – situations which the legislator did not, and perhaps could not, initially envisage. In short, it should be possible to formulate general standards of conduct without blindly prejudging what is to be done on all particular occasions. To borrow a term originally due to Waissman (1945/65, p. 126) and introduced in jurisprudence by Hart (1961, p. 124), the standards should have an 'open texture'.

Now, unlike 'artificial' languages⁴, ordinary language is irreducibly open-textured. As we know at least since Wittgenstein (1953), a criterial definition of the meaning of ordinary-language concepts, in terms of necessary and sufficient conditions, is rarely available. Practically all the plausible examples of word definitions in terms of necessary and sufficient conditions come from jargon vocabularies (e.g. *highball*), kinship vocabularies (e.g. *brother*), and axiomatised systems (e.g. *triangle*), as observed by Fodor et al. (1980). The majority of words, however, defies a definition in such terms. Consider Hart's example of the word *vehicle* in the context of a hypothetical law that proscribes using vehicles in public parks (1961, p. 124). Should the word be interpreted to include such things as bicycles, toy automobiles, wheelchairs, etc.? Or is it intended to include

³ This part of the report concerns the contributions of Dr. Rudi Gebruers to the SALOMON-project.

⁴ An 'artificial' language is a constructed language, possibly with features of natural languages but not functioning as the native idiom of its users.

only motor vehicles? As Jackendoff (1983, p. 121) and others have pointed out, we seem to rely on notions of typicality and centrality in deciding whether to include a thing or event in a category. Ordinary language concepts do enable us to identify instances of those concepts, but most of the time they also leave room for disagreement in the case of atypical or non-central instances. Indeed, it could be argued that, if ordinary language concepts were precise, we would have to possess infinitely many of them. It will be evident, then, that the 'semantic indeterminacy' (Susskind, 1987) inherent to ordinary language concepts is fundamental for the proper functioning of a legal system that intends to deal not only with a hard core of plain cases but also with 'problems of the penumbra', where there will always be some controversy about the applicability of a concept to the situation at hand. In sum, open texture is an ineliminable feature of ordinary language, to be welcomed rather than deplored by the legal practitioner.

2.1.3. The Need for Automated Legal Support

Though the cynic might observe that the Law often opts for the medium of yesterday, jurists have been excited by the prospect of intelligent machines since the earliest days of computing. Given that court decisions may be highly controversial and may affect everyone, it is crucial to know what it is for a decision to be rationally justified. Computers being paragons of rational behaviour, it is but a small step to the questions of how far legal reasoning is mechanizable and to what extent legal decisions are, or ought to be, computable. On the other hand, the increasing bulk of legal material makes the problem of manually retrieving relevant information more and more difficult. Already in 1946, the year in which the first electronic computer became operational, voices were heard demanding the creation of automatic retrieval systems to assist legal research (Kelso, 1946). The first more or less successful demonstration of a legal information retrieval (IR) system by John Harty took place in 1960, and since then numerous systems have been developed both by commercial ventures, professional bodies and governments (Bing, 1986). Furthermore, apart from its use for retrieving legal source materials and for analysing the leeways available to a judge in deciding a new case, the computer is increasingly used in the legal office for drafting documents, keeping diaries, managing cases, circulating messages, and so on (Mital & Johnson, 1992).

However, whereas the advantages of legal office automation are undisputed, neither legal IR nor legal expert systems have met with unqualified success and approval. Despite considerable work around the world, legal expert systems have rarely advanced beyond the theoretical stage. And where they have, they have been restricted to *recurrent* problems in some narrow area of Law for which the required expertise is manageable in size and complexity (Capper & Susskind, 1988). Legal expert systems are not yet able to provide wholly *unique* and accurate conclusions for any particular situation. This is true today and will be true for some generations to come.

Computer-assisted legal IR has fared a little better. The basic method of most commercial systems stems from the work of Harty and involves storing the keywords of statutes, court decisions and other documents to be retrieved. Law retrieval then becomes a two-stage process of searching the keyword files for a Boolean combination of keywords, followed by the display of (references to) texts that match the Boolean query (Bing & Fjeldvig, 1984). Some systems provide sophistications such as stem-stripping, automatic thesaurus look-up and various display modes, but the search techniques employed can never hope to explore legal concepts beyond the occurrence of specific words. On the whole, then, conventional legal retrieval systems often retrieve much irrelevant material and fail to identify all the relevant documents, either due to the lack of skill on the part of the user or, more likely, due to poor search strategies (Blair & Maron, 1985; Dabney, 1986).

In order to make the Harty retrieval machine intelligent enough to recognise relevant similarity under variation in grammar, terminology, or perspective, it could be given a more or less elaborated superstructure reflecting the *contents* of statutes, decisions, and other source materials (Hafner, 1981; Karlgren & Walker, 1983; deBessonnet, 1991; Dick, 1991). However, the fact that such superstructures always make implicit assumptions about the way in which a legal text should be interpreted, and about the kinds of questions that might be asked, lends support to the contention that the design of such an 'intelligent' legal document retrieval system may remain a challenging exercise for computer scientists rather than become an accepted tool in the legal profession. On the other hand, many users feel they are getting into a second breathing with the more conventional legal IR systems, once they have learned to live with the computerised austerity. Their tolerance towards inadequate system responses may in part be explained by the fact that computers and software have become more reliable, more accessible, much more convenient to use and, most importantly, much cheaper. But the main explanation is that, without the deficient Harty machine, the workload imposed by the ever-growing size of legal literature would be overwhelming: the user prefers a deficient technology over no technology at all.

2.1.4. Computer Meets Language Barrier

What are the reasons for the lack of success of legal expert systems and Horty retrieval machines? The fundamental difficulty seems to be that both types of conventional systems are designed to suit the computer, not the Law.

In general, the Law is based on concepts. Legal expert systems can deal with concepts, but only within a manually *predetermined* framework of conceptual objects and conceptual relationships. Unfortunately, it proves very difficult to build a conceptual framework which is *rich* enough to express all the important facts in a particular legal context, yet *abstract* enough to suppress the irrelevant detail. In short, aprioristic definition of relevant concepts with reference to some artificial context-model appears to be both too exhausting and never exhaustive enough. By contrast, the Horty machine, chasing words using string matching and Boolean operators, obviously remains at the linguistic 'surface' and can never penetrate far enough into the conceptual level. On the other hand, it seems that none of the formalisms proposed to represent legal knowledge (Zelevnikow & Hunter, 1994, p. 95-164) can rival the representation power of ordinary language. Ordinary language, with its open-textured concepts, avoids the difficulty of absolute and invariant definitions. Furthermore, ordinary language words carry with them a range of context-models: a red traffic light, for instance, does not have the same colour as a red tomato or a red salmon. Whatever the context in which a concept is first acquired, there may be infinitely many other contexts which are sufficiently similar for us to recognise that the same concept applies in these contexts. In short, ordinary language concepts are inherently context-dependent. By reducing concepts to a single context, to the exclusion of other possible contexts, we would no longer be able to cover the infinite domain of things we may need those concepts for (Bosch, 1983).

If ordinary language is indeed the best medium for storing and conveying legal knowledge, and given that no lawyer can still afford to be without automated legal support, the question to ask is how well computers get along with ordinary language. Unfortunately, the very properties which make language such a powerful tool for communication make it a nightmare for the computer. When communicating through natural language⁵, we tend to economise on words. We try to give just the right combination of words which will allow the receiver of our message to 'get the right picture', without overwhelming him with what we may reasonably believe he already knows or takes for granted. Successful communication, then, requires some amount of problem solving at both ends of the communication channel. The sender must determine which pieces of information must be presented explicitly to allow the receiver to understand. The receiver must combine these bits of information with background knowledge and make the appropriate inferences. Accordingly, there is agreement that a fully satisfactory language processing system should be based not only on linguistic techniques (such as lexical analysis, syntactic parsing, and the application of semantic rules), but also on techniques for accommodating the whole range of facts entering into linguistic discourse, as well as on automated means for reasoning about such elusive matters as plans, motives, beliefs, and emotions, to say nothing of social conventions. Unfortunately, there is much less agreement on how to solve most of the issues involved, or even on the general approach to be taken.

One school of thought, which is more interested in human cognition than in natural language *per se*, argues that automatic language processing should be based on the global use of many different types of knowledge at each point in the process (Schank & Riesbeck, 1981). This school attempts to reduce the myriad elements appearing in ordinary discourse to a much smaller set of conceptual subcategories, represented in language-independent 'conceptual dependency' schemes that facilitate drawing elementary inferences. This conceptual knowledge is assumed to be embedded in larger memory structures which organise common-sense knowledge about stereotypical situations, plans, goals, interpersonal relationships, and even emotions (Dyer, 1983). However, significant work remains to be done before these techniques could be applied to tasks of realistic complexity. They have not yet reached the stage where they can handle the tremendous amount of knowledge required for *general* language processing, with the result that systems constructed along these lines tend to be 'pilot' systems that demonstrate the feasibility of an approach, but do not work on more than a handful of carefully selected examples.

Many other observers prefer a more modular view of language processing (Allen, 1987). In this view, our ability to process common-sense knowledge is clearly distinguished from our word building and sentence diagramming skills, a distinction that seems to be corroborated by studies on dysphatic persons. This school of thought tries to divide the knowledge employed by a language user into components which interact only through limited, precisely defined interfaces. One component covers the mapping of linear word sequences into *syntactic* structures showing how the words are constructed out of more basic elements ('morphemes') and how these words are combined into successively larger groupings. Another component concerns the mapping of syntactic structures into *semantic* formulas representing context-independent features of a sentence's

⁵ In the following, we will restrict ourselves to the processing of *written* language.

meaning. These include the semantic roles played by the various entities mentioned in the sentence, as well as quantificational information, semantic dependencies, and temporal relations. Yet another component deals with the mapping of semantic formulas into representations of the *actual messages* conveyed by sentences given the pragmatic contexts in which they are used. This involves knowing how language connects to the 'real' world, how sentences aggregate into coherent discourse segments, and how language fragments ultimately tie up with beliefs, plans, intentions, and standard conventions of language use. There is no single architecture for fitting the various knowledge sources together. Stratified systems observe a strict sequence in applying the knowledge sources. Incremental systems keep the sources separate, but apply them simultaneously, extracting whatever is useful at the moment, given whatever information is available. However, due to the enormous complexity of the issues involved, all relevant work truly integrating syntax, semantics and pragmatics has been limited to very small-scale demonstration systems. What is more, despite oft-repeated claims to the contrary, there is no parsing system that would be able to supply a correct syntactic structure for all randomly-chosen sentences submitted to it (see Black [1993] for some insightful experiments). Also, attempts at 'semanticising' parsers have remained fragmentary, and have fallen short of mapping the semantics of natural language in any complete sense. Finally, work on pragmatics has been largely theoretical, with no immediate prospect of integrating the results in large-scale systems.

2.1.5. Sublanguages and Controlled Languages

Given the unfeasibility of fully automatic manipulation of unrestricted text, research in automatic language processing has tended to focus on linguistically restricted input. Two trends will be distinguished here, one focusing on natural sublanguages, the other on controlled (sub)languages.

When language is used to discuss a restricted domain, and in particular when used by a community of speakers sharing specialised knowledge, it frequently takes on characteristics not shared by the language in its 'standard' use. Such a 'sublanguage' may be more restricted in its syntactic, semantic, and discourse properties, but it may also exhibit rules not normally regarded as part of the standard language (Kittredge & Lehrberger, 1982; Grishman & Kittredge, 1986). Some sublanguages, e.g. the languages used in weather reports and patient summaries by physicians, have properties that increase the feasibility of automatic language processing. Typically, these sublanguages are relatively closed systems in terms of grammatical variability. Practically, this means that for a steadily growing corpus of texts in a given sublanguage, $C_1, C_2, \dots, C_n, \dots$, where each C_i is properly included in C_{i+1} , the corresponding sequence of linguistic descriptions $D_1, D_2, \dots, D_n, \dots$ will eventually 'converge' on a stable set of grammatical choices after some n . Furthermore, many sublanguages have interesting word selection properties. The selection of a word w is the set of words which w commonly co-occurs with in a given linguistic environment. Selection is difficult to state precisely for the words of a whole language, because in many contexts (science fiction and fairy tales, for instance) violation of ordinary selectional constraints may be acceptable. However, when the universe of discourse is limited and highly structured, selectional classes tend to have sharp boundaries, reflecting the division of real-world objects into classes that are sharply distinguished in the domain, and only particular sequences of such classes will be judged 'acceptable' (e.g. in cell biology: *ions enter the cell*, but not *The cell enters ions*). As a result, in many sublanguages selectional constraints have virtually the force of grammatical constraints and significantly reduce the amount of ambiguity (both lexical and structural) to be covered by automatic sublanguage processors.

Unfortunately, many sublanguages suffer from 'leaks', as word meanings and syntactic constructions from the standard language or from neighbouring domains enter the sublanguage dynamically without going through a process of conventionalization (Kittredge, 1982). The desire to automatically manipulate such a sublanguage inevitably leads to the *prescription* of additional constraints beyond those inherent in the sublanguage. The resulting language is called a 'controlled language' (Pulman & Rayner, 1994). The artificial reduction of lexical and structural ambiguity and the prescription of grammatical rules make controlled languages somewhat restricted in expressiveness, but at the same time *guarantee* that certain computational processes will always succeed: the problem is tailored to the solution that can be provided. There are two principal problems with this approach. The first is that any controlled language standard must be validated with real users to determine if it allows them to say what they want to say. Systematic elimination of all important sources of ambiguity and vagueness makes it impossible to write all but the simplest kinds of texts. The second problem is to develop writing aids that help authors conform to a controlled language standard easily and effectively (Adriaens, 1996; Douglas & Hurst, 1996; Van der Eijck, de Koning, & van der Steen, 1996). Writing in a controlled language may require so much thinking about what words and constructions to use or to avoid that authors tend to get frustrated. On the other hand, one cannot expect any writing aid to certify that a text conforms completely to some controlled language standard. The reason is that some rules of any controlled

language require human judgements that are beyond the capability of any current natural language software and may, in fact, never be attainable. For the time being, all we can expect is that at least many of the mechanical errors that writers make in applying a controlled language standard are automatically removed.

2.1.6. The Language of Legal Documents

The question to ask now is whether the language of legal documents shares sufficient characteristics with sublanguages that, possibly after the prescription of artificial constraints, lend themselves to processing by the computer. In order not to complicate matters too much, we will restrict ourselves to the language variants used in the text of court decisions pronounced by Flemish and Dutch courts.⁶ However, much of what will be said may be applicable to the language of similar documents in other legal territories.

To begin with, there is, unfortunately, no comprehensive *description* of the language used in the text of court decisions. Van Ginneken (1914, p. 217-254) was one of the first linguists to discuss linguistic characteristics of legal documents written in Dutch. He notes that such documents are studded with lexical and grammatical peculiarities. The range of vocabulary is extremely wide and includes many archaic expressions, often borrowed from Latin or French, as well as common words in uncommon meanings (e.g. *betekenen, instantie, verzet*) and compounds with a temporal expression as their first member (e.g. *nu-eiser, toen-gedaagde, thans-requestrant*). On the grammatical plane, Van Ginneken observes, amongst other peculiarities, a general tendency to length and complexity, the use of marked word order (e.g. *dat hij gekomen is in verzet* for *dat hij in verzet gekomen is*), a predilection for gerunds (e.g. *het te wijzen vonnis*) and 'dangling' participle constructions (e.g. *zijnde aan ieder een afschrift gelaten*), and a tendency for omitting articles (e.g. *blijkens voormeld verslag*) and verbs (e.g. *het feit waarvan acte* (sc. *genomen werd*)). More than sixty years later, Van Ginneken's work was given a repeat performance in an article (Reinsma & Reinsma, 1976) which produced only few new insights. Apart from such fragmentary descriptive work, there have been countless attempts at purifying the language of legal documents, both by jurists feeling that their colleagues should no longer corrupt the legal jargon, and by philologists obstinately propagating therapies for getting a better and simpler legal language, but rarely appreciating the contingency between language change and legal, i.e. institutional change (see Leliard, 1979, for further references). In the absence of a clear definition of 'legal language' and lacking explicit criteria to distinguish 'good' from 'bad' legal language, most of these *prescriptive* efforts have remained fruitless.

More recent years have witnessed an increased interest in *quantitative* aspects of legal language. De Mulder (1984) reports on earlier work by A. Oskamp and himself in which the vocabulary of Dutch legislation (Criminal Law and Criminal Procedure) was compared with the vocabulary of ordinary Dutch documents. Amongst other things, it was found that in the legal corpora fewer word types are very common, whereas each of these word types has on average a higher frequency than in the non-legal corpora. Also, the 'head' of the frequency lists for the legal corpora turned out to contain many substantives, whereas the 'head' of the frequency lists for the other corpora mainly contained neutral 'function words'. These findings by Oskamp and De Mulder are corroborated in a recent and more detailed study by van Noortwijk (1995), who also surmises that word types are much less evenly spread in legal corpora than in other corpora. This would seem to suggest that the distribution of word types in a legal corpus is a useful indicator of differences in the subject matter dealt with in different clusters of legal documents – an idea already entertained by Boreham/Niblett (1976) and further explored by van Noortwijk (1995, p. 221-265) on the basis of a corpus of statute law.

Apart from work on general, quantitative aspects of legal language, there have also been a few detailed studies concerning genuinely linguistic issues. Hofhuis (1988) deals with some syntactic aspects of court decisions and notes, contrary to what is often suggested, that standard Dutch syntax is predominant in those texts. Maes (1991, p. 217-259) attempts to explain the high frequency of 'nominal', i.e. non-pronominal, anaphors in legal decisions.⁷ Nominal anaphors are used either when a lexically empty pronoun would be insufficient to identify the intended referent or when an informational surplus or special perspective is needed. Contrary to Crystal and Davy (1969, p. 202), Maes argues that identificational needs explain only a minor set of nominal anaphors in legal decisions. Much more relevant are the need to indicate role changes (e.g. legal roles, such as *eiseres/verweerder*, changing into family-roles, such as *moeder/vader*), the need to approximate 'direct'

⁶ This should not be taken to imply, of course, that there are only marginal differences between the language of judicial decisions pronounced by Flemish courts and the language in which Dutch courts write their decisions. As already noted by Bellefroid (1933, p. 3), though the law systems of Belgium and the Netherlands have much in common, they have evolved along different paths and hence also show substantial differences. As a result, either system contains concepts and words which the other lacks, and any attempt at harmonizing the languages of the two systems will be of no avail as long as the legal institutions are separated

⁷ On the wider issue of 'cross-reference' in legal documents, see Studnicki et al. (1992).

discourse in the written report of a decision, and, more generally, the needs imposed by the institutional setting in which court decisions are pronounced. Each court decision being a 'counterfactual reconstruction' of factual reality, involving an artificial *mise en scène*, disindividuation and objectivation of individual and subjective facts (Van den Bergh & Broekman, 1979, p.98 ff.; Loth, 1984, p. 36 ff.; Foqué & 't Hart, 1990, p. 138), it is indeed only plausible to expect linguistic symbols of counterfactual reconstruction in the text of court decisions (see also Soetaert, 1980).

One example of such symbols is indeed the pervasive use of nominal anaphors where pronouns would be the most natural means to establish coreferentiality (see also Zoeppritz, 1989). Nominal anaphors seem to indicate that the 'normal' perception of referential coherence does not apply in court decisions. Moreover, they signal the transformation from factual individuals to objectivated legal subjects, and symbolise that legal relevance and validity only apply to legal roles, not to accidental individuals. But there are many other examples. The institutional function of court decisions is also reflected in an intricate pattern of different forms of modality, especially in the *considerans* and *dictum* parts of the decision. Modal expressions 'project' events and situations against a system of constitutive rules, determining the probability with which 'natural' facts can be validly subsumed under some legal category, and regulatory rules, determining which juristic facts should be present before certain legal consequences can be imposed (see also Loth, 1984, p. 36 ff). Still another linguistic symbol of counterfactual reconstruction would be the unusual and archaic *overwegende*-format in the *considerans* part of decisions. This format signals that the combination of fragments in the *considerans* part should not be regarded as a coherent story of what preceded the decision, but as the grounds adduced for the validity of the legal action taken in the decision. Also the archaism and 'illegal' word order accompanying the link between the *considerans* part and the *dictum* (*om deze redenen, de rechtbank, gelast ..., zegt dat ..., machtigt ...*) symbolises that no factual *causal* relationship is involved, but a relationship justified by *legal* sources of coherence. In cases like these, language apparently performs not only a declaratory but also a justificatory purpose (see also Maley, 1985). Apart from archaic constructions, court decisions also tend to display a syntax of generalisation (e.g. agentless passives, nominalisations, thematisations) whose overall effect is that of creating distance, impersonality, and the possibility of transcending concrete and unique facts in favour of the universals of normative rhetoric.

Thus, studies like that by Maes (1991) seem to corroborate the view that legal language is indeed 'a highly specific 'form of life' wherein characteristic usages bear a meaning or have a legal value not by virtue of corresponding to any determinate set of facts or social processes, but by virtue of the complex normative institutions and rules of an effective legal system' (Goodrich, 1984, p. 524) – a view which resonates in work of a language-reforming bent (Van den Hoven, 1988) and in discussions on the translatability of legal documents (de Groot, 1987; Mincke, 1971). Still, however valuable such studies are, they are but a poor substitute for a comprehensive linguistic description of the language used in court documents. And practically none of the studies that do contain genuinely descriptive material raise the issue of the computer-processability of court language. Again, Maes (1991) is a notable exception, though this author immediately concedes that his findings nowhere come near an algorithm which would enable a computer to interpret nominal anaphors as any experienced legal practitioner would be able to do.

Fortunately, we do not need a comprehensive linguistic description of the language used in court documents in order to answer the question at the beginning of this section. Given the current state-of-the-art in the field of natural-language processing techniques and given the nature of court decisions, the prospects of fully automatic interpretation of such decisions look very dim indeed. In the preceding section, we argued that, under certain conditions, sublanguages may be more amenable to computational processing than language as a whole. More specifically, the computational tractability of sublanguages seems to be correlated with 'closure' and 'selectional' properties. To the extent that a sublanguage tends to converge on a stable set of lexical and grammatical choices more rapidly and to the extent that sublanguage selectional classes tend to correspond more closely to the natural semantic classes that might be identified by an expert in the domain, the chances of computer-processability increase. None of these conditions seems to be fulfilled for the sublanguage used in court decisions.

To begin with, the sublanguage of court decisions suffers from 'leaks', as word meanings and grammatical constructions from the standard language or from other domains enter the court discourse, for instance when factual circumstances are described or when experts' reports are quoted (see also Hofhuis, 1988). In other words, the language of such documents is far from homogeneous. Even if the computer were able to successfully deal with the specifically legal part of that language, there would be no guarantee that it would also be able to handle the non-legal part in a satisfactory manner, for instance the open-textured concepts of ordinary language (we are assuming here that the legal and non-legal parts can be straightforwardly disentangled, but actual discourse may prove much more intricate). In short, the language of court decisions has less than optimal closure properties. But there is more. Whereas most professional vocabularies are of a strongly *denotative* kind and tend to be subsumed under the classification of extra-linguistic data, the legal vocabulary is highly *auto-referential* and places great stress on the legal word as an

entity in itself. It is 'a paramount terminology by virtue of its membership of a lexical and *connotative* system rather than by virtue of any simple denotation' (Goodrich, 1987, p. 177 ff.; italics supplied). Legal semantics is not some statically given correlation between words and things, but dynamically arises in the course of legal discourse. Though often drawn from the standard vocabulary, legal terms acquire a special meaning in a legal context and are embedded in a dynamic legal culture. Hence they presuppose 'legal literacy' on the part of someone who wants to understand exactly what is meant by these terms. It is not at all clear how this 'legal literacy' could be incorporated in a computer. A further problem arises from the way ordinary language and legal discourse are 'mixed' in legal documents. It is instructive to see, for instance, that linguists with no preliminary legal training arrive at sensible interpretations of legal texts which, unfortunately, appear to be utter non-sense from the viewpoint of experienced legal practitioners. The real problem for the former is not so much the presence of all kinds of legal paraphernalia as the perfidious use of words and constructions also found in ordinary discourse (Zoeppritz, 1989). To our knowledge, no one has as yet come up with a computationally tractable solution for such problems.

Up to now we have been concerned with problems posed by *non-controlled* language in legal decisions. Could these problems not be alleviated by the imposition of artificial restrictions on the language used when the text of legal decisions is drafted? The answer to this question cannot be a simple one. To begin with, a *controlled* legal language will most probably be feasible only for highly restricted application areas, where only a limited amount of variation on a limited amount of stereotypes is involved. The question here is when routine-like decisions are still interesting enough to warrant the overhead of building systems for processing such documents. Furthermore, as we noted above, systematic elimination of all important sources of ambiguity and vagueness makes it impossible to write all but the simplest kinds of texts. In practice, it may prove difficult to strike a balance between the control expected by the computer and the freedom required in the drafting process. As shown by Mital and Johnson (1992, p.123-166), current systems for legal document drafting and document assembly, also those based on knowledge templates and hypertext technology, do not really reduce the *level* of involvement of legal practitioners operating the systems. Apart from revealing the complexity of legal document drafting, this would also seem to imply that one cannot expect a system to fully 'understand' legal documents simply by applying 'in reverse' the knowledge, both linguistic and otherwise, which is embodied in state-of-the-art writing aids. Finally, even if niches could be found where artificial controls on legal language are acceptable and workable, there will always be a mass of legal decisions to which the controlled language approach would not apply or for which it would have few, if any, beneficial effects.

2.2. Linguistic Module

2.2.1. Introduction

In considering linguistic analysis it is good to keep in mind that different types of analyses may be appropriate for different types of applications. A limited amount of linguistic sophistication may be sufficient for building a spelling corrector. But much more extensive linguistic abilities will be required for translation and summarisation: arguably, one cannot translate or summarise a document if one cannot make sense of what the author of the document wants to say. Fortunately, not all document processing tasks require full text understanding, and our working hypothesis implies that the task to be accomplished by the SALOMON system is one of these. Data can be extracted from documents without full text understanding, as evidenced by the partial success of certain statistical methods. Classification of texts and text passages has been one of the main topics that IR researchers have been investigating for several decades, and the absence of full text understanding techniques does not seem to have precluded the construction of valuable techniques.

Where knowledge of language is used for IR tasks such as extraction and classification, the marriage seems to be most successful when linguistic knowledge is integrated into a black-box front end to essentially non-linguistic IR processes. Moreover, most of such front ends do not attempt a radical analysis of text meanings, represented in some knowledge representation language, but remain at the 'surface', e.g. by reducing word variants to a common root and by making explicit only those aspects of its linguistic structure which may prove useful for ruling out unlikely word meanings and normalising indexing phrases. The reason why IR and 'deep' language processing do not and will not fit comfortably together has been revealingly discussed by Smeaton (1995) and need not detain us here any longer. It should be obvious that, if IR is based on language, theories of how language works should be of help to IR research. Yet, we do not yet seem to have theories of language which actually do help IR, and developing such a theory was beyond the reach of the SALOMON project for obvious reasons. Until such theories are forthcoming, the role that natural language processing (NLP) techniques could play in an application like SALOMON will have to be a modest one.

Part of the backbone of the SALOMON system is statistics. The underlying hypothesis is that the relative frequencies of words and word chunks in a text corpus provide a gross characterisation of text content which may be sufficient for inferring the main concepts and topics of a text. There are two main problems with this hypothesis:

1. The hypothesis assumes that a useful measure of word significance can be obtained by comparing the relative frequency of a word w in a particular document (w 's document frequency) with the relative frequency of w in the complete document collection (w 's collection frequency). In order to be a good discriminator for a document d , a term should then be neither extremely frequent nor extremely rare in the collection as a whole, and it should be relatively frequent in d , but relatively rare in the rest of the collection. However, this definition assumes a particular distribution of word frequencies across a document collection, which need not be actually present.
2. The hypothesis tacitly assumes that there is a one-to-one correspondence between words or word chunks on the one hand, and concepts on the other. In reality, the correspondence is rather many-to-many: one and the same concept can be expressed in different ways, and one and the same expression can often be used to refer to different concepts.

In the following sections, we will first discuss some quantitative aspects of the language used in the SALOMON corpus of court decisions, in order to check whether the first assumption is problematic. Following on from that, we will report on how knowledge of language was brought to bear on the development of several linguistic submodules which could be integrated as black boxes into the SALOMON prototype in order to make adjustments to the second assumption.

2.2.2. Measuring the Vocabulary of the SALOMON Corpus

2.2.2.1. Purpose

It is reasonable to conjecture that the distinction between 'content' words (such as verbs, nouns, adjectives, and adverbs), and 'function' words, i.e. words which do not directly reveal document content (such as articles, auxiliaries, conjunctions, prepositions, and pronouns) is reflected in their distribution over a collection of documents. Function words will occur with reasonably constant relative frequencies over all the documents, whereas the relative frequencies of content words will vary widely from one document to another. Some content words may be extremely rare, others may be highly frequent, while still others may occupy a region of medium-frequency. Words in the latter region may be potentially good document discriminators, if their occurrence is restricted to documents in certain subject matters.

The purpose of the quantitative analysis of the vocabulary in the SALOMON corpus was twofold:

1. to obtain absolute and relative frequency data for the word types used in the corpus;
2. to ascertain whether there is a sufficiently large area of medium-frequency terms which might be potentially good document discriminators.

2.2.2.2. Experiment 1

The first experiment concerned the analysis of the word frequencies present in a sample of 1,452 court decisions. In order to get relevant results, 14% of the original 'word' material was removed, leaving 1,070,314 word tokens to be examined. A slightly modified version of the `freq` routine (Tuthill, 1981) was used to reduce the word tokens to 24,149 types and to calculate the absolute frequencies for each of these word tokens. The frequencies found range from 1 (*hapax legomena*) to 81,607 (the article *de*). Also, word classes were computed based on the product $R \times F$, where R is the rank of a word and F its relative frequency. The results are summarised in Table 1.

Table 1

<i>R x F in the interval</i>	<i>Number of words in the interval</i>
0.000000 - 0.009999	0
0.010000 - 0.019999	10,229
0.020000 - 0.029999	8,702
0.030000 - 0.039999	2,126
0.040000 - 0.049999	945
0.050000 - 0.059999	559
0.060000 - 0.069999	429
0.070000 - 0.079999	226
0.080000 - 0.089999	198
0.090000 - 0.099999	213
0.100000 - 0.109999	111
0.110000 - 0.119999	83
0.120000 - 0.129999	64
0.130000 - 0.139999	57
0.140000 - 0.149999	56
0.150000 - 0.159999	51
0.160000 - 0.169999	92
0.170000 - 0.179999	8
0.180000 - 0.189999	0
0.190000 - 0.199999	0

More detailed results of this experiment can be found in technical report TRS-12.⁸

2.2.2.3. Experiment 2

The second experiment concerned the analysis of the word frequencies present in the passages describing the alleged offence in a sample of 1,452 court decisions. In order to get relevant results, 11% of the original 'word' material was removed, leaving 223,918 word tokens in total, and 200,676 word tokens not starting with a capital letter. Six subexperiments were performed on the remaining material, each subexperiment defining the notion 'orthographic word' in a slightly different manner.

Table 2 summarises the results obtained in the six subexperiments. The first column states whether all 223,918 word tokens were considered (T), or only those not starting with a capital letter (R). The second column states whether mapping of upper letters to lower letters was disabled (Y) or not (N) before the actual counting started. The third column states whether an orthographic word, according to the definition applied, may contain an internal point (Y) or not (N). The remaining columns give the actual results: the number of word tokens counted (column 4), the number of word types identified (column 5), the size of the set containing the words covering 50% of the total corpus material (column 6), the maximal word frequency observed (column 7), the mean word frequency (column 8), and the maximal logfrequency value (column 9).

Table 2

T/R	U/L	.	#tokens	#types	50 %	freq _{max}	freq _{mean}	logfreq _{max}

⁸ Technical reports concerning linguistic research in the SALOMON project are listed in chapter 6 of this report.

1	T	Y	Y	223,918	9,994	40	14,274	22.4	9.566195
2	T	N	Y	223,918	9,536	36	14,618	23.5	9.590009
3	T	Y	N	226,036	9,443	40	14,278	23.9	9.566475
4	T	N	N	226,036	8,966	37	14,623	25.2	9.590351
5	R	N	Y	200,676	5,308	27	14,274	37.8	9.566195
6	R	N	N	201,291	5,090	27	14,274	39.5	9.566195

In each subexperiment a histogram was prepared in order to reveal how the observed word frequencies are distributed over the word material. These histograms were based on the logfrequencies registered in the respective subexperiments and classified into one of ten logfrequency intervals. On the whole, the six histograms have a remarkably similar asymmetric structure, due to a very large set of low-frequency words (ranging between 57.3% and 66.0% of the total vocabulary). In other words, the frequency patterns differ only slightly w.r.t. the parameter 'orthographic word', and they seem to suggest that the SALOMON corpus contains a relatively rich vocabulary. A small part of this vocabulary occurs very often, another very large part is extremely rare, with only a relatively small group of potentially good document discriminators in between. More detailed results of this experiment can be found in technical report TRS-13. It should be noted, however, that the counts may be a bit misleading as they do hardly take into account the existence of formulation variants, and thus may count separately what in fact should be counted together.

2.2.2.4. Experiment 3

The third experiment compared fragment frequencies with document frequencies and yielded a list of words that only occur in the passages describing alleged offences. The results of this experiment can be found in technical report TRS-13.

2.2.3. Sentence Delimiting

2.2.3.1. Purpose

Whereas large-scale semantic processing is not as yet well understood, syntactic information alone can be helpful for the identification of 'significant' indexing phrases. An important step in the unravelling of syntactic structure is the identification of grammatical word classes (or 'parts-of-speech'). Word classes cannot be given an identity of their own, apart from the sentence patterns in which they occur. Accordingly, most procedures for assigning words of a text to their respective grammatical classes assume that the text is first divided up into sentences. Sentence delimiting, then, is a prerequisite for part-of-speech tagging.

2.2.3.2. Method

Finding sentence boundaries is not as easy as it might seem. In the absence of special pre-editing procedures, automatic sentence identification is at present at best problematic. It is no good to say that a sentence begins with a capital and ends with a period, because abbreviations like *B.T.W.* also satisfy such a criterion and the period symbol also occurs inside numbers. Adding the requirement that two spaces must follow the period is no help either. Errors are much too likely in actual texts, and the fact remains that a string of words is a sentence or not, regardless of the number of spaces between them.

Ideally, therefore, the computer should know at least about the make-up of abbreviations and numbers, and it should know what makes a 'complete' sentence. In fact, one could argue that a 100% foolproof method for marking sentence boundaries presupposes a complete lexicon and grammar of the

language. Unfortunately, grammar is intimately linked up with semantics: very often word sequences allow more than one syntactic grouping, but these are rarely equally acceptable from a semantic point of view. In other words, a grammar would never be complete without the accompanying semantics for ruling out semantically unacceptable groupings. Writing such a grammar would be a truly Herculean task, which far exceeds the material constraints on the SALOMON project. The alternative, therefore, was to settle for a more modest approach, based on crude heuristics, which will inevitably remain somewhat inaccurate, but which yields useful results most of the time.

2.2.3.3. Results

A study of a sample of court decisions revealed the most striking deviations from 'standard' punctuation and abbreviation conventions. Inconsistencies and downright typing errors preclude a 100% foolproof rule-based approach; instead, the following heuristics seemed promising in view of the peculiarities attested in the SALOMON text corpus: assume a sentence boundary when

1. a full stop, colon or semicolon is immediately followed by one or more newline symbols;
2. a newline symbol is immediately followed by at least one other newline symbol;
3. a lower letter is immediately followed by a sequence consisting of a full stop, a blank and an upper letter (in that order), provided the full stop is not part of an abbreviation.

An algorithm incorporating this pragmatic notion of sentence boundary was implemented as a C-program. More details on the algorithm and the program, as well as suggestions for improvements, can be found in technical report TRS-07. On identification of sentence boundaries in a text corpus with pretokenisation, see technical report TRS-15.

2.2.4. Part-of-Speech Tagging

2.2.4.1. Purpose

Part-of-speech tagging concerns transforming a sequence of words into a sequence of labelled words, where each label indicates the grammatical class that the word represents. Such labelling may be useful for several reasons:

1. It allows a more accurate recognition of function words (which may be formally identical to content words).
2. It induces a semantic differentiation between homographs (and thereby enhances precision) provided the homographs represent different word classes.
3. It is a prerequisite for the kind of higher-level analysis needed for the recognition of linguistic groups.

Developing a part-of-speech tagger involves two tasks:

1. development of a word annotation scheme;
2. development of a tagging program.

2.2.4.2. Method

There are basically two approaches to developing a word annotation scheme. In the first approach one first defines word classes and then assigns words to these classes by comparing properties of words with defining characteristics of word classes. This definitional approach is labour-intensive, fragile and error-prone, but on the whole, conceptually fairly clear. The alternative approach consists in deriving word clusters on the basis of the percentage of environments shared by the word in a text corpus. This derivational approach yields the word classes that are actually present in a text corpus, but it is computationally expensive, it lacks perspicuity (it produces no explicit classification criteria) and it may miss certain generalisations because the corpus rarely is a good representative for the whole sublanguage.

Considering the advantages and disadvantages of both approaches, a mixed alternative was explored. First, some clustering experiments were performed in order to find out whether useful word classes could be derived from a corpus by applying very simple distributional techniques. More specifically, the idea was to adopt a very primitive notion of context, defined as the 'immediate successor' of a word, and to examine its power in supporting a tentative word classification schema. The set-up of these experiments is more fully documented in technical report TRS-14. The clustering experiments did not automatically yield a useful set of part-of-speech tags, but were very helpful in eliciting word classes that are salient in the SALOMON corpus. In a second step, therefore, a minimal theoretical framework was developed for formulating the criteria for establishing word class membership. Both the framework and the annotation legislation we formulated in terms of it are fully documented in technical report TRS-18.

Also, various methods have been proposed for actually annotating text with part-of-speech tags, ranging from completely manual methods, over large-scale dictionary lookup, to completely automatic taggers which tag parts-of-speech 'from scratch', without consulting voluminous pre-compiled dictionaries and without extensive human intervention during the tagging process. The latter fall under two main categories, taggers based on a statistical approach and rule-based taggers. In the statistical approach, a manually tagged corpus is used as a training set, from which a model of lexical and contextual probabilities is estimated. Once these probabilities are known to the tagger, it can tag new text by assigning it the tag sequences given the highest probability by the model. Rule-based methods try to guess the part-of-speech of every single word in a text using knowledge of linguistic regularities, instead of inferring the information from lexical idiosyncrasies. Though statistical taggers have often been preferred to rule-based taggers, Brill has shown that also a rule-based tagger can achieve state-of-the-art tagging performance by inferring rules from a training corpus (Brill, 1992, 1993). As retraining Brill's tagger is fairly straightforward and since our resources were insufficient for pursuing the statistical approach, we therefore adopted Brill's approach to part-of-speech tagging. Technical reports TRS-15 and TRS-18 provide more details about the rule-based tagger and about the training cycles in which the tagger was progressively refined by comparing its results with manually tagged samples of the SALOMON corpus.

2.2.4.3. Results

Our work on an automatic part-of-speech tagger produced both theoretical and practical results. Theoretically, it was found that a purely derivational approach to word classification does not automatically yield a useful set of part-of-speech tags, but may be very helpful in eliciting word classes that are salient in a text corpus. Also, it was shown how a minimal grammatical framework – basically referring only to internal word structure, mutual substitutability of words (*in absentia*), structural connections between words (*in presentia*), and (lack of) systematic correspondences between word patterns – may be sufficient for developing a word annotation scheme that provides sufficient knowledge for low-level syntactic analysis of the kind exemplified in the terminology extraction tool LEXTER (Bourigault, 1993, 1995). Practically, retraining Brill's tagger module resulted in tagging performance to up to 97% accurate annotations for the training corpus, and to up to 94% accurate annotations for a new, previously unseen test corpus.

2.2.5. Term Conflation

2.2.5.1. Purpose

A term conflation algorithm is a program that reduces morphological variation by mapping term variants, such as *clusters* and *clustering*, to a base form or stem, such as *cluster*, which is assumed to convey the essential meaning common to all variants. The expectation is that stemming improves the performance of systems for measuring content similarity. This expectation is based on the hypothesis that morphologically similar word forms are often semantically related as well. Hence, reduction of such morphological variants often yields a more reliable starting-point for measuring content similarity.

2.2.5.2. Method

Various methods have been proposed for reducing morphological variation. Affix removal algorithms remove substrings from words under certain conditions, and sometimes also transform the result. Successor variety algorithms identify stem boundaries based on the distribution of letters in a large body of text. N-gram stemmers conflate words based on the number of n-grams they share. Still another way to do stemming is via look-ups in a table storing all words and their corresponding stems.

For an application like SALOMON, it seemed most appropriate to follow a pragmatic approach, combining an affix stripper with a table look-up mechanism. Affix strippers tend to be very efficient as they do not involve morphological analysis or dictionary lookup. However, due to their lack of linguistic knowledge, they frequently introduce mistakes. The idea we explored, then, was to refine an existing affix stripper by providing it with knowledge about word classes and about morphological irregularities. The affix stripper was based on the well-known Porter algorithm as implemented by Frakes (1992). In view of peculiarities of the language used in the SALOMON corpus, several modifications had to be made to Frakes's implementation. In part, these modifications were based on an experiment to assess the amount of morphological variation in a legal text corpus with the help of association measures on the basis of the character structure of words. Both the experiment and the resulting modifications are fully documented in technical report TRS-20.

2.2.5.3. Results

The resulting stemming algorithm, which was implemented as a C program, is documented in technical report TRS-20. A comparison of this stemmer with an earlier version developed by the UPLIFT-team produced the results summarised in Table 3. The improvements over the results obtained with the UPLIFT stemmer are obviously due to the fact that the SALOMON stemmer has recourse to a table of exceptions. However, as there is no reason to assume that the test sample contained exceptionally many morphological irregularities, any stemming device that aspires to be appropriate for the SALOMON corpus should be capable of handling exceptions. Also, even for regular verb groups in the sample, the SALOMON stemmer, missing only 11.2% of the groups, performed significantly better than the UPLIFT stemmer which missed almost 70% of the 357 verb groups.

Table 3

CATEGORY	TOTAL	SALOMON		UPLIFT	
		HITS	MISSES	HITS	MISSES
nouns	850	791	59	464	386
finite verb items	2200	2148	70	1190	1018
finite verb groups	550	494	56	109	441
infinitives	550	515	35	463	87
participles	550	529	20	250	250

However, it should be noted that we have only been concerned with the general accuracy of stemmers in terms of the number of times the stemmed output matches, or fails to match, some pre-defined, linguistically motivated conflation standard. Additional studies should be conducted to assess a stemmer's effect on retrieval performance in precision-oriented and recall-oriented searches, and on the compression rate that can be achieved by storing stems instead of full terms in index files.

2.3. Overall Conclusion

The current practice in IR in general, and automatic text classification and data extraction in particular, is largely based on statistical techniques. These techniques are reasonably successful, but it is believed by many that natural language processing techniques can produce text representations that enable more accurate inferences about document content. By considering previous work on language-based techniques from this perspective, some clear lessons are apparent.

Mainstream NLP techniques derive from applications like machine translation and natural language interfaces which are much more restricted than, and far too different from, IR which has much more degrees of freedom and imprecision. Pending more fundamental research on how IR could be significantly improved as a result of using knowledge of language, the kind of NLP that actually works for large-scale IR applications must remain fairly low-level NLP.

We have been applying these lessons in the SALOMON project and have adopted a philosophy of using NLP resources rather than NLP processes in the IR engines. The tools needed to produce these resources have been adapted from previous work and should themselves be reusable for other applications with only minor modifications. The adaptation process itself merged traditional linguistic work with techniques on semi-automatic knowledge acquisition in an interesting and novel manner.

But of course, much work remains to be done. To begin with, the tagger and stemmer tools can and should be further refined in the course of more extensive trouble-shooting than was possible in the project. Furthermore, we conjecture that a limited amount of structural analysis may, under certain conditions, improve on a purely word-based indexing strategy. This is because word groups tend to be less ambiguous and narrower in meaning than single words, and because many legal concepts in fact correspond to groups of words rather than single words. It should be straightforward to develop a term extraction tool which takes advantage of knowledge about the word class patterns that can or cannot go to make up terminological units (Bourigault, 1993). Besides refining and developing morpho-syntactic tools, one could envisage implementing strategies for the automatic discovery of semantic associations between terms in documents (Hemels, 1994). A more detailed study of the shortcomings of such low-level NLP techniques could provide indications as to how the input to systems like SALOMON should be changed in order to be more amenable to error-free automatic processing. Finally, and most importantly in the context of SALOMON's follow-up, extensive studies should be conducted to assess the effect of low-level NLP techniques on automatic classification, data extraction and, more generally, on retrieval performance in precision-oriented and recall-oriented searches.

3. PART 2: Automatic Generation of a Case Profile⁹

3.1. Background

3.1.1. Introduction

The main purpose of this part of the project is to develop and test several techniques to make a vast corpus of Belgian criminal cases (written in Dutch) easily accessible. The research relates to the fields of automatic abstracting, indexing and text categorisation.

The summary process (Sparck Jones, 1993) is the transformation of the source text in a summary text. The general process of *automatic abstracting* can be described as the transformation of an abstract representation of the source text, containing the necessary attributes for summarisation into a summary representation embodying the organised content of the summary. It is important to have a meaning *representation of the source text*. Also, a *summary representation* embodying the organised content of the summary, from which the actual output summary text is synthesised, has to be formed. The actual summarisation process is the derivation of the summary representation from the source text representation. It is critical to define the *attributes* of the source text representation. These attributes contain information directly *supplied by the input texts* or include information *supplied from knowledge sources* that support the information supplied by the input texts. The nature of the input source text (type of text) determines the attributes of the source text representation. Secondly it is also important to define how the information in the source text representation is exploited for summarising (again allowing for supplement from knowledge sources independent of the text). Here the intended purpose of the summary regarding the function and the audience of the summary (type of abstract) is critical. The summary representation may be edited to obtain a more understandable summary (again allowing for knowledge sources independent of the text).

At present, the majority of abstracts automatically generated are document extracts. Existing methods focus on the identification of relevant information in the source text and selection of this information for inclusion in the text of the abstract. Generation of coherent abstract text is usually restricted to attempts in anaphor resolution (Paice & Jones, 1993) and to other improvements of the readability of the abstracts by considering rhetorical connectives of the extracted sentences, by discarding references to tables, figures, and references to other texts, or material in parenthesis (Paice, 1990), and by merging sentences (Mathis, Rush, & Young, 1973).

Current automatic abstracting of texts mainly serves two purposes.

1. The abstract is constructed for easy and fast determination of relevance: it indicates whether or not the complete text version is of interest (*indicative abstract*). The goal is to generate a concise document description that is more revealing than a title but short enough to be absorbed at a single glance.
2. The abstract is a document surrogate expressing the main contents of the document: its components may be used for text search, text linking etc. (*informative abstract*).

Text indexing is a subtask of information retrieval. Raw documents must be converted into expressions in some text representation. These expressions are sometimes called *document representatives* (Lewis, Croft, & Bhandaru, 1989; Lewis, 1992a), and must have a structure employable by the text retrieval software. The document representative contains the document attributes or descriptors, necessary for subsequent electronic use of the document. The most common way of representing text is by attaching content identifiers in the form of keywords or index terms to the texts and attributing a weight to each index term. This weight reflects the relative importance of the index term for the content of the text (Salton, 1989; Salton & Buckley, 1988).

⁹ This part concerns the contributions of Marie-Francine Moens and Caroline Uyttendaele to the SALOMON-project and is described in Uyttendaele, Moens, and Dumortier (1996), Moens, Uyttendaele, and Dumortier (1996a), and Moens, Uyttendaele, and Dumortier (1996b) (see chapter 8 of this report).

Summarising is a form of information capture. In this way it has some *relation with indexing* (Hutchins, 1987). A very brief summary may serve as a complex structured index description (Sparck Jones, 1993). The components of a summary (e.g. words, phrases and sentences) can be used as indices or keys for accessing the information of the text of the document. Further according to this author another distinction between summarising and indexing is that the indexing role is coarser, so prediction from linguistic elements to content relevancy, is easier and evaluation can be more robust. A summary is an intrinsically more complicated object, standing in a more complicated relation to its source text, than an indexing description of that text.

Text categorisation is an information retrieval task in which one or more category labels are assigned to a document. It is the classification of documents with respect to a set of one or more pre-existing categories (Lewis, 1992a). It usually involves describing the text on a general, abstract level and assumes that the documents contain features from which the general categories may be inferred. Such an approach requires additional knowledge about the documents for interpreting the surface features of the texts. When automatically categorising text, this knowledge may be formally implemented or may be deduced from manually categorised example documents.

Additional details can be found in TRS-39.

3.1.2. Research in Automatic Abstracting

Following Sparck Jones (1993) we will classify approaches in automatic abstracting in two main strategies. A first strategy relies on the surface features of the text and is called *shallow processing of the text*. Although, in this strategy text processing relies on some heuristics, additional knowledge for interpreting surface features is very restricted. A second strategy employs various degrees of additional knowledge to interpret the surface features found in the text and is called *deep processing of the text*. In this model we have a deeper model of the contents of the text prior to processing.

3.1.3. Approaches Relying on the Surface Features of the Texts (Shallow Source Representation)

The *surface features* of a text may not be neglected when identifying relevant information in it. The approaches, discussed in the next sections, will more or less rely on them, while applying some additional heuristics. Applications of these methods are found in abstracting systems for *expository texts*¹⁰. However, one has to be cautious when only relying on surface cues: they may be a partial, and sometimes rather inaccurate reflection of the true content of a document (cf. Paice, 1991).

The *title* (Maeda, Momouchi, & Sawamura, 1980; Paice & Jones, 1993) and *subtitles* are usually recognised as a fruitful source of information. This is based on the hypothesis that an author conceives the title as circumscribing the subject matter of the document. Section headings serve as a table of contents (Bernstein & Williamson, 1984).

Lay-out features, such as italics, bolds, and underlining may identify important information in a text. Some orthographic markers, such as exclamation signs, may draw the attention to significant sentences.

Significant words and phrases in a text reflect the content of the text and may serve well as crude abstracts (keyword abstracts). We may distinguish between *terms occurring in titles, captions and section headings*, and *terms occurring in the body of the text*. The former may receive special treatment based on the location of the terms in the specific document. A weight indicates the relative importance of words and phrases and better discriminates semantically important terms. Salton (1989, p. 438) assigns a higher weight to keywords that occur in the main title than those found only in lesser headings. The terms occurring in the body of the text may be subjected to a standard term-weighting process.

In a text one distinguishes content or *informative words* from *function words*, which express syntactical or operational relationships and which can be disregarded for certain purposes. A stop list¹¹ eliminates non content words from a text. The importance of the remaining words for the text is calculated based on distribution properties of the term in the text or in the complete corpus. Algorithms for automated weighting of index terms, have been developed since the end of the 1950's (Salton & Buckley, 1988). Since then sophisticated weighting scheme's have been developed and are still developing (cf. *supra*).

Phrases, especially noun phrases, are considered as important semantic carriers of the information content of a text (Maeda et al., 1980). Abstracts benefit from the inclusion of key phrases. *Key phrases* may be computed based on a frequency analysis of co-occurrences of the phrase components and/or closeness of the components (*statistical phrases*) (Fagan, 1989; Kupiec et al., 1995).¹²

Identification of key words and phrases is fully applied in information retrieval systems. Hence, their limitations in retrieval systems are well understood (Riloff & Lehnert, 1994). Different words and phrases can express the same concept. This *problem of synonyms* may effect the weighting of terms and produce redundancy in the abstract. Words and phrases may have a different meaning according to the context (*polysemy*), which again may affect weighting, but is considered a less serious problem in abstracting when the words and phrases are extracted from one document context. Depending on the local context, single words or phrases are suitable as index terms. So, it may not always be predictable what kind of index term to use (*problem of local context*). Single words and even phrases do not always reflect well the content of a text. Sometimes the entire context of a sentence, paragraph or even the whole text is required (*problem of global context*).

The foregoing leads to the recognition of important *sentences that reflect the content of a text*. A variety of methods have been developed to compute the value or weight of each sentence of the source text. Sentences with the highest scores (above a pre-defined threshold) are retained for summary purposes.

Again, this technique goes back to Luhn (1958) who proposed that abstracts might be generated automatically by selecting from a source text sentences that *contain strong clusters of 'significant' words*. Each potential cluster would receive a score reflecting the number of significant and non-significant words in it. Each

¹⁰ For the purpose of automatic summarisation usually a *distinction* is made *between expository and narrative text* (Rau, Jacobs, & Zernik, 1989). Narrative texts have a plot. The text is usually constructed in a way the reader can easily follow the actions. Events central to the plot are described in the same way as superfluous events. Expository texts describe topics and subtopics. Here the organisation of the text is important to find the information regarding the topics. Scientific texts are an important part of expository texts. Beside expository and narrative texts, there exist text types related to specific disciplines (e.g. legal court decisions).

¹¹ A list of stopwords is usually constructed based on information about occurrence frequencies: the most frequent words of a corpus are considered as stopwords (Fox, 1992).

¹² Identification of phrases may also be based upon the syntactical relationships between the components (*syntactical phrases*) (Croft, Turtle, & Lewis, 1991; Smeaton & Sharidan, 1991).

sentence would receive the score of the highest scoring cluster in it, if any. Those sentences whose scores exceeded some set threshold would be extracted for inclusion in the abstract. Significant words were recognised by first eliminating common function words, and then accepting as significant the most frequent of the remaining word stems.

Edmundson (1964; 1969) has done some important work on sentence significance. He studied the characteristics of manual document abstracts both in terms of content and machine-recognisability. He developed the cue method (see *infra*), the key method (cf. Luhn), the title method, and the location method for sentence scoring. He developed a system that assigned numerical weights to machine-recognisable characteristics. He weighted keywords according to their frequency in the document, and summed the keyword weights for each sentence. He found that words occurring in title and subheads are good indicators of content. Sentences are given a significance value based on the *number of title and subhead words they contain*. Based on the observations of Baxendale (1958), Edmundson gave various scores to sentences, whether they occurred at the beginning or end of a paragraph, near the beginning or end of the whole document, or below a heading (*location heuristics*). Also, Kupiec et al. (1995) acknowledge *sentences immediately below section headings* describe well the contents of a text.

Prikhod'ko and Skorokhod'ko (1982) stressed the importance of the analysis of links between sentences in automatic abstracting. Each sentence is scored by the *number of links* with the other sentences of the text. Again, those sentences, the score of which passes a threshold are included in the abstract. This approach is based on the assumption that sentences related to a large number of other sentences are highly informative and are prime candidates for extraction. The number of links between sentences is based on common lexical and semantic text units. The number of links is determined by the number of common terms between sentences. These terms are lexically, morphologically (word stems) or semantically (synonyms)¹³ equivalent. Stop words are not accounted for.

Thresholds are empirically determined, based on the outcome of experiments. Prikhod'ko and Skorokhod'ko (1982) compute the threshold on the basis of the desired degree of compression of the source text, or in function of the mean weight of the sentences and/or in function of the distribution of the weights.

For a specific corpus an optimal combination of extraction selection heuristics may be learned from a training set (Kupiec et al., 1995). Given a training set of documents and hand-selected document extracts, a classification function is developed that estimates the probability a given sentence is included in the abstract. New abstracts can then be generated by ranking sentences according to this probability and by selecting a user-specified number of scoring sentences. The text features employed by Kupiec et al. (1995) are length of sentences, sentences in the first ten and the last five paragraphs of the document, the first, final or medium sentences in a document, sentences with frequent content words, sentences with proper names that occur more than once in the document, and sentences containing indicator phrases¹⁴, or sentences following section headings that contain indicator phrases.

3.1.4. Approaches Relying on Additional Knowledge (Deep Source Representation)

This approach entails performing a *semantic analysis of the source text based upon a semantic representation of the type of text under consideration*. The summary relies on common, 'expected' structures in the text, which form the basis for the summary. These approaches build on the accomplishments of the work done in text extraction, a subfield of natural language processing. Text extraction usually relies on text representations of the text corpus that reflect predictable patterns of linguistic context. This research aims at extracting a pre-defined and narrowly defined class of facts.

For extracting purposes there is usually no need to produce a representation of the full meaning of the document. According to Sparck Jones (1993) deep source representations can capture *linguistic information, domain world information, or communicative information*. All three types of information give a very different characterisation of the text as a whole and correspondingly imply different bases for summary formation. Linguistic information deals with rhetorical relations, especially as flagged by lexical and other surface cues. These rhetorical relations may be seen as embodying standard, known ways of organising texts that are conventionally associated with seeking to achieve certain communicative effects. Domain world information deals with the representation of the domain dependent content of the text. Communicative information deals with the representation of the intentional structure recognised by the reader.

¹³ The authors rely on additional knowledge for determining word stems and synonyms.

¹⁴ The use of indicator phrases: see next section.

The approaches have been applied for *expository text* (e.g. Paice & Jones, 1993; Hahn & Reimer, 1986; Hahn, 1990), for *narrative texts* (e.g. Rumelhart, 1977; DeJong, 1979, 1982; Rau *et. al*, 1989; Jacobs & Rau, 1990; Appelt, Hobbs, Bear, Israel, & Tyson, 1993), as well as for specific types of text (e.g. Young & Hayes, 1985). However, they are not without problems.

Because, knowledge-based approaches rely on a set of expectations about the contents of texts, they are successful in restricted domains. The knowledge involved may vary from simple thesauri to more complicated text grammars. The systems usually have in common that they target specific information in the texts and ignore complementary details. A severe *disadvantage* of these systems is that they cannot incorporate unexpected information in the input text into their summaries. Also, there is the danger that the knowledge, initially conceived for the texts, grows larger, and becomes more complex and more difficult to apply and maintain when keeping up with dynamic changes of the input texts.

Index terms, used as crude abstracts, may be transformed in more *uniform or more general concepts* with the help of a thesaurus. A thesaurus offers a precise and controlled vocabulary to describe the content of a document.

Specific linguistic expressions are reliable indicators of the domain or the topic of the text (Riloff & Lehnert, 1992) or of the relevancy of text segments for abstracting purposes. The use of these expressions in automatic text processing is motivated by the observation that human readers can reliably identify relevant texts (or relevant portions of texts) merely by skimming the texts for specific cues.

Cue words and indicator phrases may provide the *context for the semantic role of text fragments or text strings*. These indicator constructs are language dependent. They may be very specific for domain dependent information or they may be specific to certain types (genres) of texts. Cue words and indicator phrases are useful to indicate *significant sentences* in a text or to reject sentences that are without any value in the summarisation process. The use of 'bonus' and 'stigma' words have already been proposed by Edmundson (1969). Bonus words include superlatives and value words, such as 'greatest', 'significant', whilst the stigma words consist of anaphors and denigrating expressions (e.g. 'hardly', 'impossible'). These words receive a positive or a negative weight. The significance value of a sentence is the sum of the weights of the component words of the sentence. Rush, Salvador and Zamora (1971) constructed a 'word control list', which contained short phrases as well as individual words. Each entry was accompanied by a semantic code, indicating whether the word or phrase was a positive or negative indicator for its significance for extraction, or whether it had some other significance. Also Paice (1981) relies on cue words and indicator phrases to identify sentences that are good indicators of what a document is about. Cue words and indicator phrases may also identify *significant concepts* in the text (Riloff & Lehnert, 1992; Paice & Jones, 1993). These cues may further reveal *rhetorical relations* in the text (Miike, Itoh, Ono, & Sumita, 1994) and identify the document structure. In this way the text may be segmented.

Sometimes abstracting relies on more *complex knowledge structures*. Paice (1981) lances the idea to a grammatical approach to the use of cues and indicators. A text grammar may be defined as a system of text features, that deals with the functions and relations of these features in the text. A text grammar may account for text structures and other linguistic, domain-dependent, or communicative structures that occur in the full text of specific text types.

The most frequent formalisms used to represent a text grammar are *frames and scripts*. A set of frames, tailored to the domain of application, is normally used to facilitate the analysis and representation. Concept frames contain characterisations of entities and entity relationships. Scripts describe sequences of events and inferences about additional events that may occur. When analysis is complete, output templates are used to generate a textual summary from the instantiated frames or scripts. Sometimes a *rule base* is used to implement the text grammar.

One of the earliest successful summarisations of *narrative text* was FRUMP (DeJong, 1979, 1982). This research demonstrated that reasonably good summaries could be produced (with great computational efficiency) using predictive text analysis techniques. FRUMP accurately extracted certain conceptual information from texts in pre-selected topic areas. The summaries are all based on a a priori set of scriptal expectations about that domain. FRUMP reads newspaper stories with a purpose, looking for the most important events in the story, and ignoring all other details. The knowledge is organised in sketchy scripts. A sketchy script contains only the important events. FRUMP had sketchy scripts for 60 different situations ranging from earthquakes to a country aiding another militarily or economically, and to labour strikes. FRUMP selects sketchy scripts by looking for clues in the text. The clues determine the sketchy script to be selected. The approach of FRUMP is followed by Tait (1985) in the summarising system 'Scrabble'. This system tries to account for unexpected data in the input texts, based on a variety of heuristics, which are not explained in detail. This article is also unclear about the results.

In the following we mention some other approaches. In narrative text, Lehnert (1981; 1982) constructs the structure of a story by recognising the distinctive plot units. This system accounts for variations in plot structure and does not rely on a pre-defined structure of the text. Plot units are identified by recognition of its component affect states. The composition of a plot unit in terms of affect states and the recognition of affect states is based on a predictive knowledge base. Recognition of affect states is based on knowledge about plans, goals and themes. The focus algorithm, developed by Sidner (1983) and based on linguistic knowledge, is worthwhile to mention. This algorithm finds the focus of an English text and identifies the moving of the focus as the discourse progresses. Tracking the movement of the focus includes a means of distinguishing the presence of more than one focus in the text. The TOPIC system of Hahn (1990) exploits the text thematic structure of *expository text* in conjunction with domain knowledge, but still relies strongly on domain knowledge, embodied in lexical experts and frames. Paice and Jones (1993) elaborate Luhn's idea that important concepts tend to be mentioned many times, in combination with a text grammar containing indicator cues. The text grammar is organised as a rule base. For each semantic role of a concept the rule base contains the context in which the concept occurs. A list of several candidate strings may be compiled for a given conceptual role. Then the best among these candidates is chosen based upon the number of occurrences within the given context and the weight of the context for the conceptual role. In this way semantic roles that are common to papers in a specific field and the relevant text units for abstracting are identified.

Additional details can be found in TRS-39.

3.1.5. Important Research Strategies in Automatic Abstracting

According to Sparck Jones (1993) progress in automatic abstracting may be realised along two directions. First, *text structure* is important when accessing the content of a text. For modelling the text structure of the different text types and for relying on it for text processing tasks such as text generation, abstracting and retrieval, we may build on realisations in *natural language processing*. Secondly, the progress made in *information retrieval*, especially the current refinement and sophistication of *statistical techniques* developed in this domain, may be fertile for automatic abstracting. Since the end of the 1960's, not much attempts have been made to incorporate statistical techniques in automatic abstracting. Nevertheless since then, new techniques have been developed and are still developing.

Text structure

According to Paice (1990) and Pinto Molina (1995) progress in automatic abstract generation depends on the existence of a satisfactory *theory of text structure*. This theory involves the description and explanation of the structure of the different text types and may be based on linguistic, domain, and cognitive paradigms. The text organisation above the sentence level is significant in marking what is important and thus is relevant to summarising. According to Paice (1990) each sentence (or other coherent segment of text) plays a specific role, both in relation to the overall purpose of the text and to other nearby sentences (or segments). The discipline of text linguistics considers the complete text as a superior grammatical unit. The literature on text structure is very heterogeneous, and a satisfactory theory of text structure is lacking. More studies of text syntax and semantics are needed, providing a description of the properties and organisation of different genres of text.

Following van Dijk (1980) and Pinto Molina (1995) text implies the superimposition and interrelationship of two basic structures (surface and deep) and a third one (rhetoric/schematic) that is complementary. The *surface structure (microstructure)* corresponds to the physical reality of the text and to its basic meaningful symbols words. The *deep structure (macrostructure)* of the text represents the topics and sub-topics of the text and is conceived as a hierarchic tree of text components, a finding which is usually confirmed for expository text. The macrostructure of narrative texts can be seen as a configuration of plot units, each of which is composed of smaller entities called, 'affect states', which mark positive events, negative events, and mental events of null or neutral emotionality (Lehnert, 1982). The *rhetoric/schematic structure (superstructure)* can be considered as a transition structure between the surface and the deep structure and reflects specific cognitive, pragmatic and social functions in textual communication. Text structure, especially the rhetoric structure, greatly varies according to the type of text and alters even within scientific texts of different disciplines (Pinto Molina, 1995).

Text structure plays an important role in abstracting. This process implies the transformation of the surface and rhetoric structures of text into the description of its deep (content) structure (Pinto Molina, 1995).

Characterising the functions of text sections or the automatic recognition of their functions is important when searching an automatic method to obtain the information representation of the document. Almost any type of text can be analysed into a small number of major components. In some cases these components may be predictable according to the text type, in other cases there is much more variation.

Important concepts in text linguistics (de Beaugrande & Dressler, 1981), playing a role in automatic abstracting and identification of the text structure, are text cohesion, text coherence and intertextuality. *Text cohesion* concerns the ways in which the components of the surface text, it is the actual words we see, are mutually connected within a sequence. *Text coherence* relates to the global organisation of texts. It concerns the ways in which the configuration of concepts and relations which underlay the surface text, are mutually accessible and relevant. In abstracting, the concept of a sequence of sentences is introduced, as a group of textual units that possess cohesion and coherence. *Intertextuality* concerns the factors which make the utilisation of one text dependent upon the knowledge of one or more texts previously encountered. Intertextuality is responsible for the evaluation of text types as classes of texts with typical patterns of characteristics.

When incorporating text structure theories in automatic abstracting, we may build on natural language processing research for *modelling the knowledge of the text structure* and analysing the texts. The text structure may be modelled for specific text types. This would facilitate not only the *generation of text*, but consequent automatic processing of the text for abstracting or retrieval tasks. Alternatively, when text structure models are known and described, *statistical methods* may be found for automatically recognising these structures in the texts.

Progress made in the domain of information retrieval

In early times automatic abstracting and information retrieval (especially text indexing) were strongly related. This relation is reflected in the work of Luhn (1957; 1958), Baxendale (1958) and Earl (1970). After 1970 this relation has weakened. However, because of the current interest in automatic abstracting, especially for texts of unrestricted domains, researchers explore the potential of statistical techniques, developed and still developing in the field of information retrieval. Parallel, in the information retrieval field there is growing interest in complex indices for document access. This research is beneficial for both retrieval and abstracting.

Most of the statistical techniques do not regard a text in isolation, but consider a complete text corpus for the calculations.

Since the time of Luhn (1957; 1958) there has been some remarkable progress in automatic indexing of texts and identification of index terms. In 1957 Luhn identified a 'significant word' based on its frequency of occurrence in the text (stop words excluded): a word was considered as significant if this frequency (*in-document frequency*) exceeds a certain value. Sparck Jones (1973) argued that words found in many documents were not important as index terms, and developed the *inverse document frequency* as a term score. The importance of an index term is inversely proportional to the number of documents in which the term occurs. Salton and Yang (1973) combined the inverse document frequency with the in-document frequency by taking their product as a measure of term importance. An overview of the methods developed is given by Noreault, McGill, and Koll (1981), Ro (1988), and Salton and Buckley (1988). The *product of the in-document frequency and the inverse document frequency* has become very popular for computing the content value of terms. Recently different variants of this function have been developed to account for differences in text types (Salton & Buckley, 1988; Lee, 1995). Another approach to term weighting is *probabilistic indexing* (Robertson & Sparck Jones, 1976). Here, the weight of an index term is learned from examples. A more recently developed and promising approach to term indexing is *latent semantic indexing* (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990), taking into account terms that commonly appear in documents and documents that jointly share terms for revealing a latent semantic structure of a document and revealing the value of an index term in this semantic structure. The recently held TREC-4 (Text Retrieval Conference-4 November 1995) proves the interest in new term weighting functions.

Also theories developed for *automatic text comparison* involving similarity functions (e.g. inner product and cosinus coefficient) (Jones & Furnas, 1987; Ellis, Furner-Hines, & Willett, 1993) are useful for automatic abstracting, especially for detecting similar and redundant text units. Research into the use of statistical techniques for the automatic determination of text themes and text structure is progressing (Salton & Buckley, 1991; Salton & Buckley, 1992; Salton, Allan, & Buckley, 1993; Hearst & Plaunt, 1993; Salton & Allan, 1994; Salton, Allan, Buckley, & Singhal, 1994; Salton, Allan, & Singhal, 1996).

Additional details can be found in TRS-39.

3.1.6. Research in Automatic Categorisation of Text

Categorisation is the classification of documents with respect to a set of one or more *pre-existing categories* (Lewis, 1992b) and involves describing the text on a general, abstract level. Categorisation assumes that the documents contain features from which the general categories may be inferred. Such an approach requires additional knowledge about the documents for interpreting the surface features of the texts. This knowledge may be formally implemented or may be deduced from manually categorised example documents. Document categorisation is very useful for filtering and routing purposes.

Categorisation often relies on *finding features in the text* (often indicator phrases) that define certain pre-defined categories. The lexicon of features is manually constructed (Hayes & Weinstein, 1991), available from external sources (Liddy & Paik, 1993), or learned from example documents (Riloff & Lehnert, 1992). Approaches relying on a manual implementation of a feature set or on learning from example texts that are a priori categorised only are successfully applied in restricted domains (Hayes, 1992; Riloff & Lehnert, 1992). The work of Liddy and Paik (1993) demonstrates the real feasibility of using semantic information from a machine-readable dictionary in the form of words with their corresponding Subject Field Codes (semantic categories) to represent the contents of documents and queries in a manner that will facilitate automatic filtering of documents. They use an electronic version of Longman's Dictionary of Contemporary English. When constructing this dictionary lexicographers attribute Subject Field Codes to each word. Such a code classifies each content word of the text by its semantic class (e.g. acid in class 'science' and subclass 'chemistry'). Single words belong to different semantic classes. The classes most frequently covered by the text are retained as the categories of the text.

Masand, Linoff, and Waltz (1992) do not rely on a feature dictionary but apply *Memory Based Reasoning (MBR)* (also a sub-field of Machine Learning) to categorise DOW Jones news wire press releases. MBR solves a new task by looking up examples of tasks similar to the new task and using similarity with these remembered solutions to determine the new solution. Using a training set of approximately 50.000 documents, the system was trained to assign one or more of the 350 Dow Jones codes to new press releases. Each new document is compared with each document of the example document base. The categories of the *k*-nearest neighbour documents of the example database are attributed to the new document. The number of nearest documents (*k*) is pre-defined. Text features are single words and capital word pairs which are typical for company and product names in business-oriented news. Text is also compressed by eliminating stop words and common words. The remaining words are then weighted applying the inverse document frequency metric. Given the number of example texts, this approach makes use of a parallel super-computer to make the comparisons feasible. In case of a large example set, similar documents of the example set may be clustered in advance. MBR will then find the most similar centroid cluster objects and subsequently find the most similar objects of these clusters, thus reducing the number of comparisons made (Iwayama & Tokunaga, 1995).

Additional details can be found in TRS-37.

3.1.7. Generation of Document Profiles in the Legal Field

In the legal field the research of the University of British Columbia, Canada relates to the construction of document profiles (Gelbart & Smith, 1991; Gelbart & Smith, 1994). FLEXICON (= Fast Legal Expert Information CONSULTANT) automatically generates structured representations of cases and other legal documents. A case profile consists of: case header information (the parties of the dispute, date, court, jurisdiction, and judges); a classification of the subject of law; a listing, in four quadrants, of the most significant concepts, facts, case citations and statute citations occurring in the case, in decreasing order of system-computed terms' weights; and key paragraphs. The case profiles are created using three major text processing functions. The recognition of case and statute citations in legal text is accomplished by template-matching functions and simple rules. Concept terms are extracted from the text by matching sections of the text with terms contained in the phrase dictionary. Fact terms are the remaining words and phrases following noise word removal. The four types of profile terms are weighted and sorted by factors reflecting their distribution in the processed document and in the data collection. The relative importance of citations is proportional with the frequency of occurrence in the document. The relative importance of concept and factual terms is proportional with the frequency of occurrence in the document and inversely proportional with the number of documents in which the term occurs. In case of case citations other factors intervene with the calculations such as age of the case, level and competence of the court. Important paragraphs are extracted by using word and phrase lexica, templates and rules. The patterns that can be associated with important or useless paragraphs consist of, but

are not limited to: the relative location of the paragraphs in the text; the density and clustering, in specific paragraph categories, of legal concepts, factual terms, case citations, and statute citations; the occurrence of 'positive' or 'negative' cue words and indicator clauses; the interrelation among paragraphs in terms of common words used across paragraph categories; and syntactical elements, such as tense and person. These profiles have the characteristics of indicative as well as informative abstracts. The profiles contain the most important information of the documents and the keywords of document profiles may be compared with user's queries for retrieval. In the future FLEXICON aims to learn the patterns and characteristics of specific paragraphs of the text based on neural network technology.

Additional details can be found in TRS-36 and TRS-38.

3.2. Methods

3.2.1. Legal Relevance of the Corpus Analysis

The choice of a corpus of criminal decisions as a research object, was no coincidence. Only criminal cases are available in machine readable format for the time being. Moreover, criminal law is clearly structured and criminal decisions have a fixed, recurring composition. For the SALOMON project, a corpus was used consisting of all the decisions that the correctional court of Leuven pronounced between January 1992 and June 1994. These are more than 3000 documents altogether, containing more than 5000 offences charged.

A sample of criminal cases was manually studied. The profound analysis of the correctional cases resulted in a detailed description of the text structure and its attributes. Basically, the cases can be classified into 7 main categories, distinguishing general decisions from particular ones. The latter are concerned with appeal procedures, civil interests, refusals to witness, false translations by interpreters, infringements by foreigners or the internment of people.

Although belonging to different categories, all criminal cases have a typical structure. They are made up of 9 elements, some of which are optional (cf. Figure 1):

1. *superscription*, containing the name of the court, the date and the registration numbers of the court administration and of the prosecutor;
2. identification of the *victim*;
3. identification of the *accused*;
4. *alleged offences*, describing the crimes and factual evidence;
5. *transition formulation*, marking the transition to the grounds of the case;
6. *opinion of the court*, containing the arguments of the court to support its decision;
7. *legal foundations*, containing statutory provisions applied by the court;
8. *verdict*;
9. *conclusion*, possibly containing the name of the court and the date.

The SALOMON techniques were developed in order to extract and summarise the most relevant parts of the cases: the alleged offences, the opinion of the court and the legal foundations.

The *alleged offences* give an exact description of the crimes a person is accused of. At least 50 percent of the cases studied judge more than one offence. The accused may have committed two or more offences, or there may be several accused involved in the same case. The *opinion of the court* allows to distinguish three types of cases within the studied corpus: *routine cases* (containing only routine/unimportant grounds in their opinion), *non-routine cases* (containing other than routine-grounds) and *leading cases* (containing more than 5 'principle grounds'). Principle grounds are the paragraphs of the opinion in which the court gives general, abstract information about the application and the interpretation of some statutes. The leading cases only represent 3 to 5% of the total corpus. Finally, the *legal foundations* consist of a complete enumeration of legal texts and articles applied by the court. Several of these foundations (*routine foundations*)

are cited in each case; they have no relevance for the user. The user is only interested in the foundations concerning the essence of the case.

The corpus analysis also yielded a description of the legally most relevant parts of the case. After examining manually constructed headnotes of printed law reports, it was decided that document profiles produced by the demonstrator should contain the following information:

1. the name of the court that pronounced the decision;
2. the date of the decision;
3. the type of the criminal offence that is the object of the decision;
4. the key paragraphs and key concepts that appear to express the essence of the opinion of the court;
5. references to the statutory provisions which the court deems applicable to the case at hand;
6. a pointer to the full text of the court decision.

Additional details can be found in TRS-46, TRS-47, TRS-48, TRS-50, TRS-54, TRS-55, TRS-56, TRS-58, TRS-59, TRS-60, TRS-61, TRS-62, TRS-68, TRS-69, and TRS-73.

3.2.2. Starting Point: the Manual Practice

It is useful to consider the *manual process of abstracting legal cases*, not only for defining the desired output, but also for finding appropriate techniques, which may be automated for automatic abstracting. The intended output for SALOMON is inspired on the abstracts actually preceding every publication of a legal case in magazines or retrieval systems. These abstracts are drawn up manually by specialised staff. They consist of several *keywords* (describing the legal question treated in the case) and a short *summary* of the case (reflecting the legal principles applied by the court).

The drawing of the abstract mostly happens according to the following technique: the summary is composed first by extracting one or more interesting paragraphs from the decision. Consequently, the appropriate keywords are selected, either from a fixed list (related to the classification of the case), either they are copied from the text of the case.

The *manual practice of abstracting in general* is described by Cremmings (1982), Rowley (1988), Lancaster (1991) and Pinto Molina (1995). These guidelines mostly relate to the abstracting of expository text. Nonetheless, they are interesting for abstracting legal texts. The process of manual abstracting is strongly cyclic with alternations over the source text drafting, writing and revising the summary. The summarising can be guided by checklists.

Rowley (1988, p. 24) suggests 5 steps in manual abstracting: reading of the document in order to gain an understanding of its content and an appreciation of its scope; taking notes of the main points of the document; drafting of a rough abstract; editing of the abstract while making spelling corrections and style improvements. According to her a practised abstracter does not 'read' every word in the document, but scans a significant proportion of the document. Often, but not always, much of the significant information can be culled from the latter paragraphs of a document. Paragraphs headed 'Results', 'Conclusions', 'Recommendations', 'Discussion', 'Future work' are often fruitful sources of material for inclusion in the abstract, while introductory paragraphs are generally intended to offer orientation about the subject of the document.

Pinto Molina (1995) suggests that the analyst should first make a quick reading to recognise such fundamental characteristics of the document as form, class, and structure of the information. A second reading concentrates on the various headings of the document and its key sections ('Purpose', 'Methodology', 'Results', and 'Conclusions'), because these generally contain the deep and rhetoric structures of the document. This author also lists important strategies in human documentary abstracting as the selection of relevant text units, which preferably may be achieved by elimination of repeated text items, and items of little or no relevance. This author notes that the surface structure is cognitively important in the abstracting process. The deep structure sought is subjectively variable depending on the knowledge of the abstractor and on documentary demands. Brown and Day (1983), following Kintsch en van Dijk (1978) propose teaching methodology for abstracting based on five rules: removal of insignificant information, deletion of redundant information, concept super-arrangement, thematic sentence selection (if possible representing text), and abstract construction.

Additional details can be found in TRS-39.

3.2.3. Potential and Limits of Automatic Abstracting

Some of the *recommendations for manual abstracting* have a *potential in automatic summarisation*. These recommendations concern the recognition of fundamental characteristics of the document as form, class, and structure of the information, the deletion of insignificant and redundant information, and the selection of thematically important sentences.

Except for very restricted text domains, some aspects of human abstracting are currently *out of reach in automatic summarisation*. Human abstracting also involves interpretation (Pinto Molina, 1995). Here apart from the objectivity of the textual content, certain extra-textual factors intervene, among them the base knowledge of the abstractor, the broad context of the text, and the abstracting objectives. Interpretation goes beyond the merely perceived and requires activation of all kind of extra-linguistic knowledge. Moreover, the comprehension of sequences of sentences in a text must have a cyclical nature, corresponding to the cyclical principle of textual elaboration of information, which joins old (already known) and new pieces of information. Finally, any text has a relatively permanent aboutness but a variable number of meanings in accordance with the particular use that the person can make of the aboutness at a given time. The aboutness of a text may be automatically defined, the variable number of meanings is very hard, if not impossibly, to automatically detect. The interpretation of the abstractor is often reflected when selecting and generalising the main points of the text.

Especially, in the legal field one has to be careful when employing knowledge for interpreting surface features of the legal texts or when employing statistical techniques for identification of content bearing text units. Knowledge bases inevitably reflect a certain *interpretation* of the documents. Different knowledge engineers have different ways of selecting, representing and processing knowledge. Other interpretations may be perfectly valid as well. Likewise, the selection of statistical techniques presumes a certain knowledge about the text corpus and its characteristics reflected in the statistical function parameters. Furthermore, editing the extracted text may entail a certain interpretation of the text. This is exactly the reason why even manual abstracts of legal cases are no more than the extraction of relevant text parts. There is no need to go that far as to re-edit the text, given the danger of misinterpreting or misrepresenting the case. After all, it is not up to the abstractor to make the law.

Additional details can be found in TRS-39 and TRS-70.

3.2.4. Initial Structuring of the Cases Based on a Text Grammar

SALOMON employs *deep text* processing to automatically categorise the cases and to identify their logical structure. All criminal cases can be classified into 7 categories and have a typical structure. The categories concern general cases and specific cases the latter belonging to the categories appeal procedures, civil interests, refusals to witness, false translations by interpreters, infringements by foreigners or the internment of people. They are made up of 9 elements (superscription, identification of the victim, identification of the accused, alleged offences, transition formulation, opinion of the court, legal foundations, verdict, and conclusion), some of which are optional. Some of these components have a typical substructure (e.g. date and name of the court in the superscription). The case category and structure have been recognised relying on a *text grammar*. Identification of text structure is an important first step in automatic abstracting (Spark Jones, 1993). Additionally, irrelevant paragraphs of the alleged offences and opinion of the court, and irrelevant legal foundations are identified.

We designed a domain-independent formalism for representing text structure. Identification of the text structure often relies on linguistic, domain- and communicative information (Spark Jones, 1993). The use of '*super-structure*' schemes or *grammars* (called 'abstract frames' by Paice [1991]) may be promising for elucidating the text or information structure of certain text types. The idea is to anticipate structural schemes that are common to all text of a specific type. Text grammars are a promising representation form. Text grammars are often used for the automatic understanding and generation of texts. Text grammar research in the field of information retrieval is still in its infancy. Rama and Srinivasan (1993) developed a prototype for the representation and content extraction of medical abstracts.

A text is usually composed of different blocks or segments which fulfil a semantic role in the text and which may have a sequential, hierarchical, or causal relation between them. The *domain-independent formalism* designed allows to represent such semantic units of a text and their relations. Segments may be optional. Segments belong to one of the following segment types: text block limited by word pattern(s), paragraph, sentence, or phrase. Each type may be characterised by specific word patterns or a logical combination of these patterns. Patterns consisting of one or several words are the most important features that delimit or classify text. Word patterns with a same delimiting or classifying function are grouped in a semantic class. Patterns consist of one or more words (strings) in a fixed order. Words or pattern elements are separated by spacing, or by punctuation marks and spacing. A pattern element is a word string, number, wild card, or word template. Wild cards represent random text and/or spacing. A word template is composed of fixed and wild card characters. The wild cards were useful to recognize dates and word stems in the text and to ignore the arbitrary use of capitals in word patterns. We did not allow nested patterns in the representation, whereby a pattern is used as the pattern element of another pattern. A delimiting or classifying word pattern may occur in the text in variant formulations (members of a semantic class). The variants are lexical, morphological and/or syntactical, or bear on punctuation marks. It is important to control the number of word pattern variants. We could limit the pattern variants by defining an attribute in the pattern representation that allows facultative neglecting of punctuation marks, and by the use of wild cards as pattern elements or as string characters. The use of wild cards is very advantageous: the knowledge engineer himself defines the degree of fuzzy match between each word pattern and the text processed. More wild cards in the pattern increase the risk of an incorrect interpretation of the text by the system. The formalism allows to represent the structure of specific text types as a coherent text grammar. A grammar for a programming language, once defined, is not altered often in the course of time (Nagy & Seth, 1992). Yet, a document grammar may be subject of multiple changes during its use. This made a separate and maintainable implementation of the grammar necessary.

There is a choice of *forms* to represent a text grammar. *Frames* are well suited to represent document structure in general. Frames offer the possibility to describe complex objects in a detailed way by treating a cluster of information as one entity. Frames can be organized in a network, reflecting document structure and content. There is an increasing interest in representing a document with a *semantic network* (Wang & Ng, 1992). A semantic network is a special type of graph, in which descriptions are applied not only to the nodes in the graph, but also to the lines joining them (Edwards, 1991). The nodes represent the objects with their attributes, the lines the relations between the objects. In the formalism designed the complete text and its segments as well as the semantic classes of word patterns are represented by frames. The segment frames have a hierarchical (*has a*), sequential (*precedes*) or conditional (*if ... then*) relation between them. Segment frames may be optional. A sub-segment inherits from its parents the text positions, between which the sub-segment has to be found. Segment frames attributes describe segment types and features described above. Word patterns and their semantic class are represented by a one level hierarchy of frames. The top frame is connected with the appropriate segment frame(s) (*limiting or classifying relation*).

A parser was implemented to recognise document structure based upon the text grammar. Automatic document parsing based upon a formal grammar determines whether a document is conform with the text grammar and which mark-ups are accordingly attributed. The text grammar of a document is represented by a semantic network of frames. Parsing a document based upon this network aims at recognizing nested segments, ordered segments and segments, the legitimacy of which depends upon the existence of other segments.

The nested structure of segments (*has a relation*) is described by a context-free syntax, represented by a tree structure. The recognition problem of trees can be studied in terms of *push down automata* (PDA's). A push-down stack (last-in-first-out) is needed to recognize central recursivity. The tree is accessed in a depth-first strategy. Segments of a same hierarchic level, possibly but not necessarily follow each other. Recognition of siblings takes place in the order of declaration in the text grammar. The recognition procedure uses the *precedes relation*, when defined in the grammar. The activation of a frame may depend upon the existence of a specific text segment (*if ... then relation*). In this case the frame is activated after positive evaluation of the production rule attached to the frame.

Word patterns delimit (*delimiting relation*) or classify (*classifying relation*) the text and its segments. They are described with a regular syntax. A regular syntax is parsed with a *finite state automaton* (FSA). A fuzzy search or probabilistic ranking of the match between the word pattern and the text is not applied. The knowledge engineer himself defines locations in the pattern where an inexact match is approved.

The parser is *deterministic*: alternative solutions are ordered by priority. Most text characteristics uniquely define the text segments. A backtracking mechanism would not necessarily result in better parsing. When text is processed it is important to detect an ungrammatical situation at the place of occurrence and not interpret this situation as the result of an incorrect previous decision. So the ungrammatical situation may be optimally corrected for further parsing (Charniak, 1983). For instance when only one of the segment limits is

positively identified, the whole segment may be identified at this limit, thus minimally disturbing the processing of following or preceding segments.

The parser may be considered as a '*partial parser*'. A partial parser makes a careful and thorough analysis of the portions of text it is designed for, while skipping over the irrelevant portions (McDonald, 1992). In addition a partial parser is exceptionally careful that what it does is accurate, i. e., that the analysis would not change if it had been able to make sense of additional parts of the text. A partial parser is targeted for a specific kind of information. At one hand there is a set of one or more algorithms and text processing mechanisms that are hopefully very general, and on the other hand a body of rules, a grammar, that embodies the parser's knowledge on the target information and how it appears in the text. The greater part of a topic-specific grammar consists of rules about the words that convey concepts and relationships that can be quite specific for the domain of interest. Such a parser transforms the text into some abstract structured representation of the information it contains.

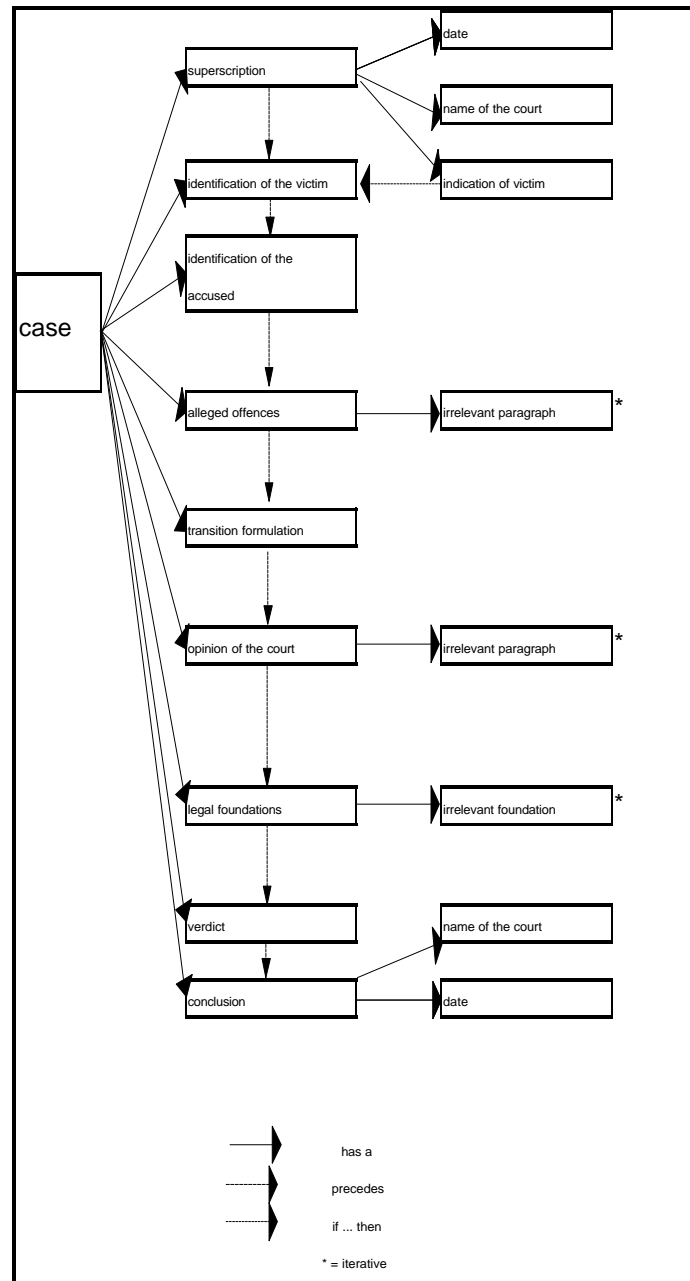
The result of our parser is a document tagged in *SGML*-syntax, indicating the general category of the document and the categories of all text segments and sub-segments defined. A text, marked in *SGML*, is very useful for database storage and retrieval. Except for the attribution of category tags, the parsing does not structurally, lexically, morphologically or syntactically alter the original text.

The *text features* that characterize the structure of the correctional cases were *acquired* by the expert in criminal law, who identified the case categories, their components and attributes. She interviewed other experts in the field and people responsible for the publication and manual summarisation of cases in professional journals.

The knowledge related to the 23 categories, the ca. 300 word patterns (consisting of an average of 3.5 strings, numbers, or templates) and 31 classes, and the more than 100 relations between text segments was acquired and implemented in respectively 11 and 5 man days. Some necessary corrections of and additions to the knowledge base, carried out after processing and evaluating an initial sample of 25 cases, required 3 man days. Easy implementation and maintenance of a complex knowledge base was the first priority. Figure 1 shows a schematic representation of a case and its segments.

Figure 1

Main segments of a correctional case



Machine-learning systems solve problems by examining samples described in terms of measurements or features. Machine learning is useful as an aid rather than a replacement of knowledge acquisition. It has been proven useful only for the acquisition of simple lexico-semantic patterns that classify texts (Jacobs, 1993). These techniques are also handy when a sample base must not be manually tagged, but is readily available.

For SALOMON automatic learning of the text features that classify a case or a case segment did not seem beneficial. The training set should be large to cover a sufficient amount of category examples. Feature variants are lexically and syntactically very divers and could not be learned by approximation. Many morphological variants (for instance different genus or gender, the use of capitals) are anticipated when the knowledge is manually acquired. Further, complex patterns (combinations in propositional logic of simple patterns) classify the texts of the correctional cases. Also, the 'simple' patterns are not restricted to a specific type. They could be single words, noun phrases, consecutive words with no syntactic relation, or whole sentences. Finally, not only the text patterns have to be learned, but also the relations between text segments. It was found that at least an almost similar effort would be needed to sample enough representative examples

and carry out the manual tagging of the categories in these examples, than manually constructing the knowledge base.

Additional details can be found in TRS-26, TRS-30, TRS-31, TRS-50, TRS-51, TRS-52, TRS-53, and TRS-75.

3.2.5. Recognition of the Topic Structure and Representative Text Units Based on Statistical Methods

The texts of the alleged offences and opinion of the court are unpredictable covering an unrestricted domain. The recognition of their topic structure and of representative text units is subject of the next section.

The *alleged offences* give an exact description of the crimes a person is accused of. At least 50 percent of the cases studied judge more than one offence. The accused may have committed two or more offences, or there may be several accused involved in the same case. Such elaborate alleged offences contain a bulk of redundant material. More precisely, a delict description in the alleged offences is disclosed in a separate paragraph of the text. Such a description contains the specific facts of the delict, integrated in the text of the description. The alleged offences may contain several delict descriptions: some of them may be identical, but referring to different facts. SALOMON discriminates distinct delict descriptions (key paragraphs) and eliminates redundant descriptions.

The *opinion of the court* allows to distinguish three types of cases within the studied corpus: *routine cases* (containing only routine/unimportant grounds in their opinion), *non-routine cases* (containing other than routine-grounds) and *leading cases* (containing more than 5 'principle grounds'). Principle grounds are the paragraphs of the opinion in which the court gives general, abstract information about the application and the interpretation of some statutes. Non-routine cases may be long and elaborated, discussing different crimes. Theme progression is not necessarily linear: a theme may be abandoned and resumed (picked up again) during the discourse.

Shallow techniques (Sparck Jones, 1993) are employed to eliminate redundant information in the delict descriptions, to group the paragraphs of the opinion of the court into thematically coherent units, and to identify thematically important text units and key terms. Shallow processing techniques are needed because crimes concern every aspect of society and the linguistic context of the information is not predictable particularly because cases may cover delicts not previously seen as a result of new legislation.

When automatically grouping paragraphs of the alleged offences and opinion of the court, SALOMON builds upon current research in information retrieval regarding automatic structuring of full text. Each paragraph is represented as a vector of weighted index terms.¹⁵ Index terms are selected after elimination of stopwords.¹⁶ The index terms of the alleged offences are weighted with the in-paragraph frequency¹⁷, the index terms of the opinion of the court with the inverse document frequency¹⁸. In different experiments paragraphs are compared with the inner product¹⁹ and cosinus coefficient²⁰. Additionally, preliminary

¹⁵ In information retrieval document and query are often represented as a vector (Salton & Allen, 1994). Document and query are represented as a set of terms (term vectors) in a n -dimensional vector space \mathcal{R}^n (n = number of terms for representing documents and queries). Each component of the vectors corresponds with one term. The value of the vector component indicates the occurrence of the term (Boolean model) or the relative importance or weight of the term (vector model). The vector model is very popular allowing a simple computation of the similarity between two objects (document and query, between documents, or between text fragments).

¹⁶ Stopwords were identified as the most frequent words in the corpus.

¹⁷ In-paragraph frequency computed as the number of times an index term t_j occurs in the text paragraph.

¹⁸ Inverse document frequency (*idf*) computed as: $\log(N/dfj)$

with N = number of documents in the collection

dfj = number of documents in the collection which contain index term t_j .

Computation of *idf* is based upon the complete corpus of cases.

¹⁹ $\sum_{i=1}^n WO1_i \cdot WO2_i$: the similarity between two texts is calculated as the inner product of their vector representations

WO1 and **WO2**.

experiments had indicated the usefulness of selecting index terms based on their syntactic category. Nonhierarchical clustering methods, which provide a clustering around representative objects and which do not rely upon the order of input (except in the case of ties), are employed to thematically group the paragraphs of the alleged offences and the opinion of the court.

Cluster analysis is a multivariate statistical technique that automatically generates groups of similar objects. Clustering methods are usually categorised according to the type of cluster structure they produce. Nonhierarchical methods split a data set of n objects into k clusters. When no overlap is allowed these methods are known as partitioning methods. Hierarchical methods result in a hierarchic structure of the data (usually represented as a dendrogram).

Cluster analysis in text based systems is not new. Documents may be clustered on the basis of co-occurring citations (Small & Sweeney, 1985). Terms may be clustered on the basis of the documents in which they co-occur (Crouch, 1988). Nodes of information in hypertext systems may be grouped based upon the number of independent paths between them (Botafogo, 1993). Hierarchical methods have been extensively used in information retrieval for grouping similar documents on the basis of terms that co-occur in the documents, providing an efficient search structure for large document collections (an overview in Willett, 1988; Voorhees, 1986).

Many of the *nonhierarchical clustering* methods have in common that clustering is performed around cluster representatives. A cluster representative may be a central point in the vector space (centroid), computed as the average vector, or may be the most centrally located object of the cluster. The nonhierarchical methods, used in information retrieval to provide a search structure in a document collection, are the simple pass method (Salton, 1971) and methods based on the construction of central points (Anderberg, 1973; Salton, 1971). A drawback of these algorithms is that the results depend on the order of the objects in the input file (Kaufman & Rousseeuw, 1990, p.114). The cluster centroid against which each document is matched changes after each document processed. None of these methods deal with the search for a optimal natural number of clusters and with the identification of representative objects.

We employ the *covering clustering algorithm* (Kaufman & Rousseeuw, 1990, p. 111) to eliminate redundant paragraphs of the alleged offences.²¹ In this algorithm the number of clusters (k) is not fixed, but

$$^{20} \frac{\sum_{i=1}^n WO1_i \cdot WO2_i}{\sqrt{\sum_{i=1}^n WO1_i^2} * \sqrt{\sum_{i=1}^n WO2_i^2}}$$

: the similarity between two texts is calculated as the cosinus coefficient of their vector representations **WO1** and **WO2**.

²¹ In the covering model (adapted for similarity coefficients) following function is minimised (cf. Kaufman & Rousseeuw, 1990, p. 111):

$$\text{minimise } \sum_{i=1}^n y_i$$

subject to:

- y_i is equal to 1 if and only if object i has been selected as representative object and is equal to 0 otherwise.

- Z_{ij} is equal to 1 if and only if object j is assigned to the cluster of which i is the representative object and is equal to 0 otherwise. Z_{ij} is subject to following constraint:

$$\sum_{i=1}^n Z_{ij} = 1 \quad j = 1, 2, \dots, n \text{ and } j \neq i \text{ which implies that for a given } j \text{ (} \neq i \text{),}$$

one of the Z_{ij} is equal to 1 and all others are 0

- $Z_{ij} \leq y_i \quad i, j = 1, 2, \dots, n \text{ and } j \neq i$. this constraint implies that an object j can only be assigned to an object i if i has been assigned to a single representative object: if $y_i = 0$ (i is not a representative object) all Z_{ij} are 0, if $y_i = 1$ (i is a representative object) then all Z_{ij} can be either 0 or 1.

- $\sum_{i=1}^n s(i, j) Z_{ij} \geq S \quad j = (1, 2, \dots, n) \text{ and } j \neq i$. which expresses that each object j must lie within a minimum similarity S of its representative object.

each object must at least have a given similarity (threshold) with the representative object (medoid) of its cluster. The objective is to minimise the number of representative objects. In SALOMON we implemented an extra constraint: for a certain value of k we do not only search an acceptable clustering, but we also search the best acceptable clustering.

The *k-medoid clustering method* (Kaufman & Rousseeuw, 1990, p. 68 ff.) is used to group the paragraphs of the opinion of the court. The *k-medoid* method searches the best possible clustering in k -groups of a set of objects. A set of objects is automatically divided in k -groups such that the average similarity between paragraphs of the same cluster and their medoid is maximised. The *k-medoid* model is based on the mathematical model proposed by Vinod in 1969 (cited by Kaufman & Rousseeuw, 1990, p. 109).²²

The optimal solution of this problem is the generation of all possible solutions and the choice of the best possible solution for which the total (or average) similarity of each object and its medoid is maximised. The number of clusters (k) is specified a priori or is determined as part of the clustering method. Kaufman and Rousseeuw (1990, p. 83 ff.) define for each object i of the cluster structure a '*silhouette width*' which measures the degree of fitness of an object to its cluster. A variant hereof, applied for similarity measures, defines the silhouette width of each object for a given k (except for $k = 1$ or $k = n$ ²³) as the normalised difference of the average similarity of the object i to all other objects of its cluster and the similarity of i with its second choice cluster.²⁴ For each possible k -value the average silhouette width of all objects involved in the clustering is computed as a parameter for the goodness of the clustering. The best k value is this for which the average silhouette width (*silhouette coefficient*) is maximised. To test whether $k = 1$ or $k = n$ represents a better clustering, we respectively test whether the average similarity between cluster objects and their representatives increases or whether the average similarities between objects of different clusters decreases.

An optimal solution of the *k-medoid* method is computationally possible for relatively small problems with up to 50 or 60 objects. For a large number of objects we developed a variant of the PAM-program developed by Kaufman and Rousseeuw (1990, p. 68 ff.), which finds a good, but not optimal solution. The algorithm is a reallocation algorithm: an initial clustering is improved in consequent steps until a specific criterion

²² In the *k-medoid method* (adapted for similarity values) following function is maximised (cf. Kaufman & Rousseeuw, 1990, p. 68 ff):

$$\text{maximise } \sum_{i=1}^n \sum_{j=1}^n s(i, j) Z_{ij} \text{ with } j \neq i$$

whereby $-s(i, j)$ = similarity between objects i and j

subject to $-Z_{ij}$ is equal to 1 if and only if object j is assigned to the

cluster of which i is the representative object and is equal to 0 otherwise. Z_{ij} is subject to following constraint:

$$\sum_{i=1}^n Z_{ij} = 1$$

$j = 1, 2, \dots, n$ and $j \neq i$ which implies that for a given j ($\neq i$), one

of the Z_{ij} is equal to 1 and all others are 0

$$-Z_{ij} \leq y_i$$

$i, j = 1, 2, \dots, n$ and $j \neq i$. y_i is equal to 1 if and only if object i has been

selected as representative object and is equal to 0 otherwise. This constraint implies that an object j can only be assigned to an object i if i has been assigned as a representative object: if $y_i = 0$ (i is not a representative object) all Z_{ij} are 0, if $y_i = 1$ (i is a representative object) then all Z_{ij} can be either 0 or 1.

-the function is subject to a last constraint:

$$\sum_{i=1}^n y_i = k$$

with $k =$ number of clusters: this means that exactly k clusters are to be chosen as representative objects, implying that the result will be exactly k non empty clusters.

²³ n = number of objects to be clustered.

²⁴ Computation of the degree of fitness ($s(i)$) of an object i to its cluster (adapted for similarity values) (cf. Kaufman & Rousseeuw, p 83 ff.):

$$s(i) = \frac{a(i) - b(i)}{\max\{a(i), b(i)\}}$$

whereby $a(i)$ = average similarity of i to all other objects of its cluster;

$b(i)$ = maximum of the average similarities between object i and the objects of each other cluster where object i doesn't belong (in other words the similarity of i with its second choice cluster).

is met. First, an initial clustering is performed by successive selection of the most centrally located objects as representative objects, until k objects are found. Then all pairs of objects (i,h) for which object i has been selected as representative object and object h has not, will be considered in the search for a better clustering. A better clustering is a clustering which increases the intra-cluster similarities. By considering all pairs, which possibly contribute to the clustering, and choosing the best contribution, the clustering is independent of the sequence of input of the object (Kaufman & Rousseeuw, 1990, p. 104). The whole process is repeated until the intra-cluster similarities can no more be increased.

The *medoid* of each cluster forms a representative description of each crime/topic treated in the alleged offences/opinion of the court. The medoid of the cluster is the object of the cluster that has an maximum average similarity with all other objects of the cluster. We assume that text units that are closely linked by patterns of content words to a number of other text units of the text are informative and thus relevant to include in the case summary. Each cluster of opinion of the court paragraphs containing more than three objects is represented by its most important keywords (different methods are possible among which the highest weighted terms of the average vector of the cluster).

Additional details can be found in TRS-23, TRS-24, TRS-29, TRS-34, TRS-38, TRS-40, TRS-41, TRS-42, TRS-45, TRS-64, TRS-74.

3.2.6. Categorisation of the Alleged Offences

After eliminating redundant delict descriptions, it is useful to describe the delict descriptions of the alleged offences with descriptors that reflect the crime category. These descriptors are especially useful for information retrieval and information filtering purposes.

Crime categories are hierarchically structured. The public prosecutor employs the '*list (book) of qualifications*'. This list reproduces a collection of offences, grouped by theme in a hierarchical way. For each of the offences a numeric code, a keyword, a delict description model, the applicable statutory provisions and the possible penalties are suggested. The prosecutor draws up the indictment with the help of it, just copying the relevant description and completing it with the necessary factual details. At present about 35 head classes of crimes are found, subdivided in subclasses and possibly sub-subclasses. However, the models for delict description are *not binding* for the prosecutors and their administrative staff (unlike the numeric codes assigned to each delict, which are uniform for the whole country). They may always use other terminology, other punctuation marks, other patterns, in another order if they feel like it. At present, the list of qualifications is not complete. Entire fields of criminal law (social law, taxation law, environmental law, etc.) do not occur in it. Moreover, this list will probably never be completely finished, as the legislator passes new statutes every day.

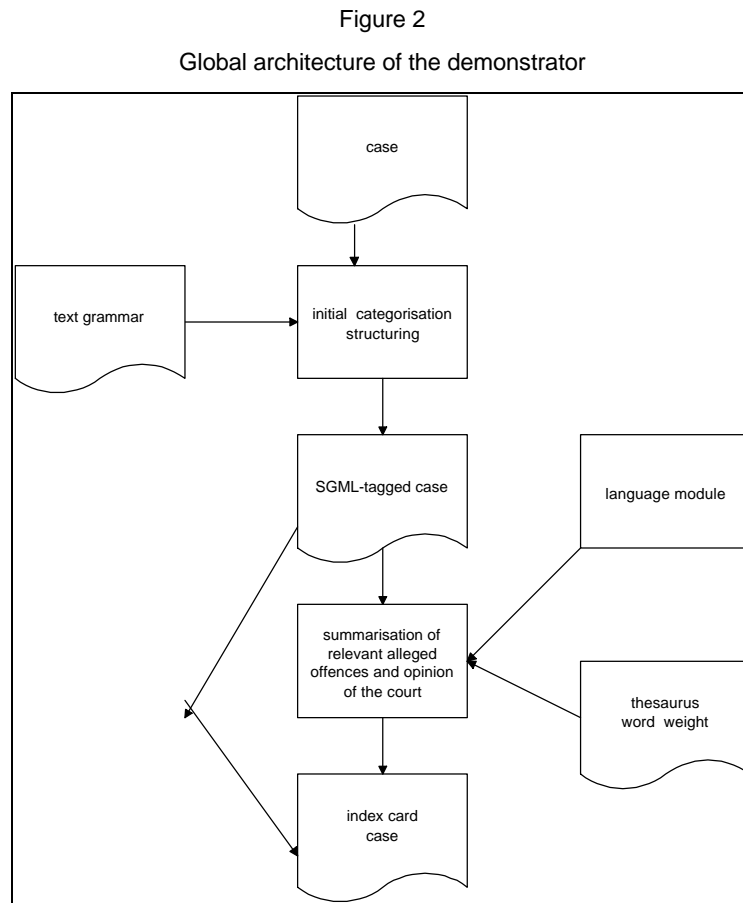
The list of qualifications can be considered as a corpus of categorised example texts (although at present incomplete). The text of the delict description, which is a variant of the standard delict description (because of facts incorporated and small aberrations from the standard text) can be compared with the example texts. The technique of *nearest neighbour search* (cf. Masand et al., 1992) allows to identify the most suitable example text. The crime categories (head categories and sub-categories) attributed to the example text are adopted to describe the text of the delict description.

The selection of this approach has been inspired by the fact that the demonstrator already had functions for text representations as a text vector and had functions for computations of vector similarity. The categorisation module of the demonstrator is only implemented in a prototypical way in order to demonstrate the usefulness of this approach. Hereby, the functions require example texts that are SGML-tagged with the name of crime categories. When no complete list of qualifications is available (as it is the case now), a minimum similarity (threshold value) is required when matching delict description and example text.

Additional details can be found in TRS-37, TRS-40, TRS-41, and TRS-57.

3.3. Results and Discussion

3.3.1. Global Architecture of the Demonstrator (Figure 2)



The initial processing of the case by SALOMON identifies the general category and the logical structure of the case with the help of a text grammar. Irrelevant portions of the alleged offences and the opinion of the court are recognised. The result is a case tagged in SGML-syntax.²⁵ From the tagged case general information about the case such as date, name of the court and relevant legal foundations are easily extracted and placed on the index card. The relevant parts of the alleged offences and opinion of the court need further processing. Key paragraphs and terms are extracted using the above described clustering methods. The index terms, needed for the vector representation, are selected with the help of a thesaurus with index term weights²⁶ and/or with the help of a language processing module.²⁷

An SGML-tagged case and index card are linked to the full text of the case by their file name. They have the same name but have a different extension, indicating the nature of the file.

Additional details can be found in TRS-25 and TRS-35.

3.3.2. Initial Structuring

²⁵ In the future courts may tag case category and logical structure during text generation.

²⁶ *Inverse document frequency* weights based on the complete corpus. For efficiency reasons an individual list of index term weight for each case is inferred from the thesaurus word weight.

²⁷ The language processing module is not yet linked to the demonstrator. Formal specifications of the interface of the demonstrator and the language module are found in TRS-41.

The system realises an *essential categorisation of the correctional cases*: general decisions are distinguished from the particular ones (decisions about appeal procedures, civil interests, refusals to witness, false translations by interpreters, infringements by foreigners, and internment of people). Each of these categories receives a special tag, which marks the whole decision (for instance '<verzet>' in Figure 3). The feature of this category is the pattern 'eiser op verzet'. Also the *structuring of the correctional case in segments and subsegments* is accomplished. The segments 'superscription' (<opschrift>), 'identification of the victim' (<partij1>), 'identification of the accused' (<partij2>), 'alleged offences' (<tenlastelegging>), 'transition formulation' (<overgang>), 'opinion of the court' (<motivering>), 'legal foundations' (<rechtsgronden>), 'verdict' (<beschikkend_gedeelte>), and 'conclusion' (<slot>) of the first level of the hierarchy are tagged when present in the decision. Some segments are structured in the successive level (for instance in the 'superscription' we are interested in the 'date' [<datum>] and 'name of court' [<rechtbank>]).

A sample of 1000 correctional cases was drawn from the original corpus. The test set was composed of 882 general and 118 specific decisions, a proportion representative for the complete corpus. The classification task is a binary one: the system decides whether a case or a case segment belongs to a particular class. The results were manually verified by a law student. The effectiveness metrics most widely used in IR are *recall* and *precision*. These metrics are also employed for text categorisation. For each case and segment category recall and precision were computed respectively as the proportion of correct assignments to the category upon the real existing number of this category in the case base, and as the proportion of correct assignments to this category upon the number of assignments to this category (Jacobs, 1993). However, a distinction must be made between texts or text segments that have fixed limits (for instance the entire document, paragraph or sentence) and those, the limits of which are defined during processing. In the former case following definitions by Lewis (1995) are adopted. Recall is the proportion of class members that the system assigns to the class. Precision is the proportion of documents assigned to the class that really are class members. An alternative to precision is *fallout*, which calculates the proportion of nonclass members that the system assigns to the class. An ideal system would have recall and precision of 1 (100 %) and fallout of 0 (0%). A contingency table summarises the relationship between the system classifications and the expert judgements (Table 4).

Figure 3
SGML-tagged case

```

<verzet>
<opschrift> Griffie Nr.: ...
<rechtbank> Correctionele rechtbank te Leuven </rechtbank> ...
<datum> 20 januari 1993 </datum> ...
In de zaak van het Openbaar Ministerie en van:
</opschrift>
<partij1> ...
</partij1>
<partij2> Tegen ...
Eiser op verzet ...
</partij2>
<tenlastelegging>
<routine_tenlastelegging> ...Beklaagd te: ...
</routine_tenlastelegging>
...
<routine_tenlastelegging> ...Uit hoofde van ...
</routine_tenlastelegging>
...
</tenlastelegging>
<overgang> Gezien de stukken van het onderzoek ...
Gehoord het openbaar ministerie in zijn vordering
</overgang>
< motivering> Overwegende dat ...
<routine_motivering> ...inbreuk ... vaststaat...
</routine_motivering>
...
<routine_motivering> Gezien de beschikking...
</routine_motivering>
...
</motivering>
<rechtsgronden> Op deze gronden en met toepassing van de artikelen ...
<routine_gronden> ...Wetboek van Strafvordering...
</routine_gronden>
</rechtsgronden>
<beschikkend_gedeelte> DE RECHTBANK ...
</beschikkend_gedeelte>
<slot> Aldus gedaan en uitgesproken ...
</slot>
</verzet>
    
```

Table 4
Contingency Table of Classification Decisions

	Expert says yes	Expert says no
System says yes	a	b
System says no	c	d

Effectiveness metrics are defined in terms of the contingency table:

$$\text{recall} = a / (a + c)$$

$$\text{precision} = a / (a + b)$$

$$\text{fallout} = b / (b + d)$$

A useful, single effectiveness measure for classification decisions takes into account both errors of commission (b) and errors of omission (c):

$$\text{error rate} = (b + c) / (a + b + c + d)$$

When the text to be classified has no fixed limits, a calculation of fallout and error rate did not seem appropriate. These metrics bear on the number of nonclass members. In the course of processing the system examines many candidate class members, when searching the limits of a segment. It is difficult to determine the real number of nonclass members. Also, the recall and precision definitions given by Lewis are difficult to apply. It may be, as it was often the case for the correctional decisions, that the system finds the class member, but not correctly defines its limits.

Table 5
Results of the Categorisation of the Entire Correctional Case

Case category	Effectiveness metrics			
	Recall	Precision	Fallout	Error rate
Appeal procedures	1.000000	1.000000	0.000000	0.000000
Civil interests	1.000000	0.916667	0.001011	0.001000
Refusals to witness	0.888889	1.000000	0.000000	0.001000
False translations	1.000000	1.000000	0.000000	0.000000
Infringements by foreigners	0.733333	1.000000	0.000000	0.004000
Internment of people	1.000000	1.000000	0.000000	0.000000
General case	1.000000	0.994363	0.042373	0.005000
Average	0.946032	0.98729	0.006198	0.001571

Table 6
Results of the Categorisation of the Case Segments

Case segment category	Effectiveness metrics			
	General decisions		Special decisions	
	Recall	Precision	Recall	Precision
Superscription	0.970522	0.970522	0.771186	0.784483
Date superscription	0.916100	0.987775	0.866667	0.939759
Name of court superscription	0.987528	0.996568	0.814159	1.000000
Identification of the victim	0.743935	0.862500	0.575000	0.920000
Identification of the accused	0.787982	0.794286	0.745763	0.846154
Alleged offences	0.843964	0.982759	0.696629	0.925373
Irrelevant paragraph offences	0.819536	0.966945	0.812155	0.954545
Transition formulation	0.867347	0.891608	0.500000	0.632184
Opinion of the court	0.871882	0.895227	0.594595	0.687500
Irrelevant paragraph opinion	0.856416	0.991582	0.907143	0.980695
Legal foundations	0.910431	0.931555	0.813084	0.861386
Irrelevant foundation	0.769907	0.793555	0.688679	0.768421
Verdict	0.896825	0.933884	0.703390	0.954023
Conclusion	0.959184	0.998819	0.728814	1.000000
Date conclusion	--	--	0.000000	1.000000
Name of court conclusion	--	--	0.375000	--
Average	0.87154	0.928399	0.662017	0.883635

Note. -- = not defined (the category does not apply).

Table 7
Fallout and Error Rate of the Categorisation of the Segments with Fixed Limits

Case segment category	Effectiveness metrics			
	General decisions		Special decisions	
	Fallout	Error rate	Fallout	Error rate
Irrelevant paragraph offences	0.026942	0.102202	0.030882	0.100572
Irrelevant paragraph opinion	0.006897	0.073438	0.010267	0.040417
Irrelevant foundation	0.099805	0.143136	0.173228	0.236052

Recall and *precision* are calculated for all categories (Tables 5 and 6). *Fallout* and *error rate* are computed for fixed text blocks (entire case, irrelevant paragraph of the alleged offences, irrelevant paragraph of the opinion of the court, and irrelevant foundation) (Tables 5 and 7). For case segments we separated the results of the processing of general and special decisions. In this way the types of errors are illustrated. In general precision is higher than recall. Recall errors are usually the result of lack of knowledge (for instance missing relations or word patterns), whereas precision errors may be due to ambiguities in the knowledge. A substantial number of

errors are caused by typing errors (no space between two words and spelling errors). This type of error also explains the very low recall rate of the category 'date conclusion' for special decisions and the rather low recall rate of the category 'irrelevant foundation'. An error occurring in a parent segment may cause errors in its children. An example of such a cumulated error is found in the category 'irrelevant foundations' for special decisions. The use of wild cards in the representation of the patterns did not cause any misinterpretation by the system. Macroaveraging described by Lewis (1992b) computes effectiveness metrics separately for the set of documents for each category and then takes the mean of the resulting values. In this way average recall and precision were computed for all categories (Tables 5 and 6) and average fallout and error rate for the categories of an entire correctional case (Table 5). For the case category an average recall and precision of respectively 0.95 and 0.99 is achieved. For the case segments an average recall and precision of respectively 0.88 and 0.93 for general decisions and respectively 0.66 and 0.88 for special decisions is obtained. The results are satisfying taking into account the limited time for knowledge acquisition and implementation.

As a consequence of the structuring *routine and non-routine cases* may be identified. A 'routine case' is defined as a case the opinion of the court of which consists exclusively of routine text.

Additional details can be found in TRS-33, TRS-71, and TRS-72.

3.3.3. Identification of Representative Text Units of the Alleged Offences and Opinion of the Court

There is no such thing as a correct summary of a text, even in a given well defined context. If an abstract is manually constructed, different abstracters will produce different abstracts. We can judge if the summaries automatically derived are reasonable or alternatively manifestly inadequate. In SALOMON we may define evaluation of paragraphs extracted from the alleged offences and opinion of the court as how well the extracted text units are representative in reflecting the complete text. The evaluation is performed by one person. Ideally the evaluation should be done by multiple legal professionals thus reducing the impact of a subjective evaluation of the summaries.

For *evaluating the informativeness of the extracted representative paragraphs* of the alleged offences and opinion of the court, we rely on *scoring categories* employed in text extraction systems (Fourth Message Understanding Conference MUC-4) (Chinchor, 1992). Scoring categories are determined based upon the comparison of a response (automatically attributed) with an expected response (key/ template = manually attributed).

We compare representative paragraphs manually attributed (keys) with representative paragraphs automatically generated (responses):

1. the number of correct responses (in the metrics further called 'correct'): a correct response is a response for which an equivalent paragraph that is manually attributed is found;
2. the number of partial correct responses (in the metrics further called 'partial'): a partial correct response is a response for which a partial match with a paragraph that is manually attributed is found;
3. the number of incorrect responses (in the metrics further called 'incorrect'): an incorrect response is a response, for which a paragraph that is manually attributed is available, but no correspondence can be found;
4. the number of spurious responses (in the metrics further called 'spurious'): a spurious response is a superfluous response with no key in the set of paragraphs that are manually attributed;
5. the number of missing responses (in the explanation of the metrics further called 'missing'): a missing response is a key in the set of paragraphs that are manually attributed, which has no response in the set of paragraphs that are automatically attributed.

Each key of the set of paragraphs, manually attributed, can only be matched with one response from the set of paragraphs, automatically attributed, and vice versa.

Following *metrics* measure four different aspects of performance:

$$\text{correct} + (\text{partial} \times 0.5)$$

$$\begin{aligned}
 1. \text{ recall} &= \frac{\text{correct} + (\text{partial} \times 0.5)}{\text{possible}} \\
 2. \text{ precision} &= \frac{\text{correct} + (\text{partial} \times 0.5)}{\text{actual}} \\
 3. \text{ overgeneration} &= \frac{\text{actual}}{\text{incorrect} + \text{spurious}} \\
 4. \text{ fallout} &= \frac{\text{possible incorrect}}{\text{actual}}
 \end{aligned}$$

'Possible' is the sum of the 'correct', 'partial', 'incorrect' and 'missing'. 'Actual' is the sum of 'correct', 'partial', 'incorrect', and 'spurious'.

Recall and *precision* are metrics borrowed from information retrieval. In the above metrics the importance (weight) of a partial match is fixed at 0.5. Deciding upon a partial match in evaluating representative paragraphs is rather subjective. We established following guideline for a partial match: the paragraph treats the topic (represents the concept), manually formulated, but is not the ideal representative. Recall and precision are ideally 1 (100 %) or close to 1.

Overgeneration measures the percentage of automatically generated responses that were spurious. *Fallout* is a measure of the false positive rate for responses that come from a finite set. Fallout is the tendency for a system to choose incorrect responses as the number of possible responses increases. In this metric 'possible incorrect' is the number of candidate representative paragraphs (all paragraphs) minus the number of representative paragraphs, manually attributed. Overgeneration and fallout are ideally 0 (0 %) or close to 0.

These metrics are calculated for each text of the alleged offences and the opinion of the court of each correctional case and are averaged for all texts of alleged offences and all opinions of the court of the test corpus. The results are obtained by evaluating the abstracting results of a test set of 700 correctional cases. The test set has been carefully selected by an expert in criminal law in a way that the set is homogeneous and representative for the complete corpus of correctional cases. The extracted paragraphs of the alleged offences and opinion of the court were evaluated both 'methodologically' and 'legally'.

A *methodological evaluation* (Table 8) aims at testing the methods (shallow techniques) we use for identifying text structure, themes of the text and representative text units. The evaluation takes into account the text paragraphs that are actually clustered and evaluates the informativeness of the extracted paragraphs. The rather high recall (about 97 % and 85 % for respectively alleged offences and opinion of the court) and precision (about 95 % and 81 % for respectively alleged offences and opinion of the court) values indicate that the techniques employed are suitable in recognising the theme structure of our texts and in identifying representative text units. These results may be improved when the language module is linked to the demonstrator. Then, index terms may be selected and weighted in a more refined way. The main errors of the output were due to the fact that paragraph boundaries were incorrectly and inconsistently attributed during text generation. The results of structuring the alleged offences are better than these of the opinion of the court. This can be explained by the often standardised naming of legal concepts and a by stereotypical way of describing these concepts in the alleged offences, making a thematical grouping and recognition of redundant material very effective. Overgeneration of responses is very low (about 3 % and 8 % for respectively alleged offences and opinion of the court), indicating that the system rather robustly identifies the number of themes in the text. Fallout (about 28 % and 24 % for respectively alleged offences and opinion of the court) is restricted. Fallout computes the proportion of faulty responses (incorrect and spurious) given the number of incorrect responses that the system could generate.

Table 8

Average results of a 'methodological' evaluation of the abstracting of alleged offences and opinion of the court

	recall	precision	overgeneration	fallout
Alleged offences	0.972664	0.954161	0.037021	0.282887
Opinion of the court	0.847502	0.810143	0.085496	0.239101

A *legal evaluation* (Table 9) aims at recognising the extracted paragraphs upon their legal relevancy. Here relevancy relates to informativeness of the delict descriptions (alleged offences) and the value in indicating legal principles (in the opinion of the court). This evaluation takes into account all paragraphs of the alleged offences (routine, non routine, factual, and delict description paragraphs) or all paragraphs of the opinion of the court (routine, non routine, factual, and principle paragraphs). It gives insight into the combined use of deep and shallow techniques and into the effect of the faults of the initial structuring upon the recognition of relevant paragraphs. It also gives insights in how well the system performs in extracting principle paragraphs considering the noise of routine paragraphs and factual considerations.

In case of the alleged offences the errors of the initial structuring influenced the results. These errors concern an incorrect structuring, a failure of recognising routine paragraphs due to a lack of knowledge, and an incorrect recognition of routine paragraphs due to ambiguous knowledge. Some of these errors may be avoided when automatically structuring text during text generation. However, the overall recognition of relevant delict descriptions is good. The very low fallout rate indicates that the system chooses correct responses even with a high number of possible responses.

In case of the opinion of the court similar errors of the initial structuring phase influence the results. Moreover, case abstracts ideally contain only legal principles applied and ignores factual considerations. Recognition of relevant principle grounds was not the objective of the research. Recognition of the principle grounds involves interpretation and is a strongly subjective operation. The system finds an important part of the legally relevant principle paragraphs that were manually attributed (almost 75 %). However, the system generates too many paragraphs (overgeneration of more than 50 %), which are often factual considerations. Such a large overgeneration necessarily decreases precision: precision is computed as the proportion of correct answers in all the answers generated. Sometimes when a legal principle paragraph is extracted, it is often not the most interesting one. It must be noted that recall and precision of the extracted paragraphs of the opinion of the court with a correct initial case structuring will be increased (cf. alleged offences by about 10 %). Nonetheless, even a legal evaluation scores satisfying compared to the test scores of text extraction systems (Final Test Score Summaries Appendix G in *Proceedings Fourth Message Understanding Conference (MUC-4)*, 1992).

Table 9

Average results of a 'legal' evaluation of the abstracting of alleged offences and opinion of the court

	recall	precision	overgeneration	fallout
Alleged offences	0.817327	0.817422	0.117243	0.091463
Opinion of the court	0.746371	0.330498	0.545124	0.21461

In a limited experiment (sample set of 17 cases with very long alleged offences) we compared the results of the inner product and cosinus coefficient for eliminating redundant paragraphs of the alleged offences. The cosinus coefficient obviously performed better and was consequently chosen for evaluating the complete test set. Redundancy in the alleged offences often concerns a redundant partial delict description, mentioning only some essential concepts. In such cases length normalisation (see TRS 34) of the cosinus coefficient is advantageous. Additional experiments regarding weight and similarity function selection are planned.

It was our aim to test and evaluate different methods for attributing *key terms* to elaborate texts of the opinion of the court based on the above metrics. Because of lack of time and man power this evaluation was not carried out.

The topic structure of the texts of the alleged offences and opinion of the court is automatically recognised building on techniques, recently developed in the domain of information retrieval. Comparably to the TextTile system (Hearst & Plaunt, 1993) and the research by Salton (Salton & Buckley, 1991; Salton &

Buckley, 1992; Salton et al., 1993; Salton & Allan, 1994; Salton et al., 1994; Salton et al., 1996), the SALOMON-system groups these paragraphs around the subject treated. In contrast with the text processed by the TextTile system, in our legal texts paragraphs treating the same subject do not necessarily follow each other. Also, in SALOMON it is necessary to identify significant paragraphs to include them in the summary.

Algorithms based on the *selection of representative objects* are new in the context of text based systems. These algorithms do not rely upon the order of input. The algorithms also provide the possibility to identify highly informative text units that through their lexical patterns are linked to other text units (cf. Prikhod'ko and Skorokhod'ko, 1982). As a result redundant information is deleted from the delict descriptions and thematically coherent text pieces of the argumentation of the judge are identified. In order to obtain a summary of the opinion of the court that contains representative paragraphs, a cluster algorithm that does not rely on threshold values is employed. In this way we obtain a natural clustering, necessary to generate a balanced summary.

Additional details can be found in TRS-43 and TRS-77.

3.3.4. Contributions of the Research

By following two research directions suggested by Sparck Jones (1993), the SALOMON-project contributes to the research in automatic abstracting. The project proves that *recognition of the text structure is an important first step in automatic abstracting* and that *progress in automatic indexing can be successfully applied in automatic abstracting*. Moreover, the research contributes in finding *new and interesting techniques for automatic text structure recognition and theme identification*, which may not only be applied in automatic indexing and abstracting, but also in automatic linking of texts even beyond the legal field.

Text structure is especially prominent in Belgian legal cases. The use of this structure for automatic abstracting fits the current research interest in using text structure for abstracting and indexing purposes. A substantial part of the text structure is identified based on knowledge about the text type. This knowledge is organised as a text grammar, incorporating not only the attributes of the text type, but also the relations between them. In this way a more elaborated semantic model of the text type is created and a refined identification of relevant information in the cases is possible. The formalism designed is domain independent and may be used to model other text types. Such an approach may be more advantageous than tailored applications such as FLEXICON (see *supra*), which also relies on a number of text cues. Document grammars have a well-known potential for modelling multi-media documents.

The topic structure of elaborated offences and opinions of the court is automatically recognised building on techniques, recently developed in the domain of information retrieval. In this way redundant information is deleted from the delict descriptions and thematically coherent text pieces of the argumentation of the judge are identified.

The use of cluster algorithms based on the *selection of representative objects* is new in the context of text-based systems. These algorithms, while not relying on the sequence of input, provide the possibility to identify patterns of lexical connectivity between text units and to distinguish informative text units. FLEXICON (*supra*) extracts relevant text units based on techniques developed by Luhn (1958), Baxendale (1958), Edmundson (1969) and Earl (1970) such as locational heuristics, frequency occurrences of index terms, and the use of indicator phrases. We successfully proposed another approach. In order to obtain a balanced summary of the opinion of the court that contains a representative paragraph and key terms regarding each topic treated, a cluster algorithm, which produces a natural clustering, is employed.

There is no such thing as one abstracting algorithm. The variety of texts and the subsequent variety in text structure and content can influence the success of abstracting programs. Even in the case of a shallow source representation, it is *hard to obtain complete domain independence* for constructing a text source representation. The SALOMON research indicates that the choice of statistical techniques makes assumptions about the parameters used in these functions. Nevertheless, the methods employed in SALOMON have a potential for automatic abstracting and information retrieval. Full text retrieval of long texts may benefit by the structuring of the text according to topics and subtopics. In this way the user may efficiently query portions of the text (Brown, Foote, Jones, Sparck Jones & Young, 1995). Methods for exploiting the discourse structure of large texts may be useful in identifying which terms are central to the content of a text (Lewis & Sparck Jones, 1996).

In SALOMON we started from the manual practice of abstracting legal cases. Part of this process can be automatically simulated. This includes the identification of the case type, the structure of the information,

deletion of redundant and insignificant information, and selection of thematically relevant text units and key terms. In this way we obtain a summary of the case, which is about 20% of the size of the full text of the case.

However, part of the manual process seems out of reach (Paice, 1990; Pinto Molina, 1995). It would be wrong to overestimate the possibilities of legal text extraction systems, they are still far from ideal. The legal field is not straightforward in the way that there is only one unique solution possible to a problem. This subjectivity of the law causes severe problems in designing legal extraction systems such as SALOMON, mainly due to the use of knowledge bases on the one hand and of statistical techniques on the other.

Knowledge bases inevitably reflect a certain *interpretation* of the cases, a problem the SALOMON team was confronted with when implementing the knowledge of the text grammar. In order to avoid too much subjectivity in identifying the irrelevant paragraphs of the alleged offences and of the opinion of the court, the knowledge base contains a limited amount of patterns indicating the irrelevancy of the respective paragraph. A more elaborated knowledge base would increase the risk of subjective interpretations. Different knowledge engineers have different ways of selecting, representing and processing knowledge. Other interpretations may be perfectly valid as well.

Human abstracting always involves interpretation (Pinto Molina, 1995). Here apart from the objectivity of textual content, certain extra-textual factors intervene, among them the base knowledge of the abstractor, the broad context of the text and the abstracting objectives. Since these aspects of human abstracting are still out of reach in automatic summarisation, SALOMON cannot for the time being automatically select leading cases by way of statistical techniques. SALOMON generates a large part of the principle grounds of the opinion of the court, but can not make a distinction between principle and factual grounds. The identification of the principle grounds in the opinion of the court is typically a subjective operation, not only because of the interpretation involved, but also because of the need for *contextual information*, to be found within as well as beyond the text of the case: other statutory provisions, legal principles, and multiple social customs and norms. It is up to the user himself - with the help of the full-text of the decision- to situate the SALOMON summary in a general contextual framework.

The SALOMON summaries have features of indicative as well as of informative abstracts. They enable the user to judge the relevancy of a case within seconds, without having to read the full text of it. Additionally, components of the summary can be useful as index terms in a search engine.

Systems like SALOMON can simplify the lawyer's job a great deal. Unable to provide the user with ready-made answers to complicated legal cases, they can at least direct him towards documents where the answer must be found, and even represent them on the screen (Moles & Dayal, 1992). *The legal text extraction system is no more than a lawyer's tool*, like a book or a library, telling him what the law is in a certain case and where to find it (Zeleznikov & Hunter, 1992, 1994). It is not intended to perform the interpretation of cases or legislation itself, only to assist the user in his own interpretation process. The purposes of the system should be clearly specified in order to make it immediately clear for which type of users the system was intended, and what they can reasonably expect of it (Susskind, 1986). The intervention of domain experts is indispensable, methods for building the knowledge base should be well-considered (Susskind, 1986; Wang & Ng, 1992).

Additional details can be found in TRS-22, TRS-39, TRS-49, and TRS-70.

3.3.5. Examples of Case 'Index Cards'²⁸

SUMMARY OF CORRECTIONAL CASE	
NAME OF CASE =	/users/sien/pc/testset/algemeen/gg/g2
DATE =	10 november 1992.
COURT =	CORRECTIONELE RECHTBANK TE LEUVEN
REPRESENTATIVE PARAGRAPHS OF THE OFFENCES=	
REPRESENTATIVE PARAGRAPHS OF THE OPINION OF THE COURT=	
Overwegende dat beklaagde in zijn verklaring dd. 18.11.91 toegaf dat hij geen arbeidsreglement opstelde, en evenmin schriftelijke arbeidsovereenkomsten opstelde voor de deeltijdse werknemers;	
(6) ²⁹	

²⁸ The 'index cards' are at random selected in the test corpus. Categorisation of each offence (type of crime) is implemented as a prototype and not yet included in the general output of the demonstrator.

REPRESENTATIVE KEY TERMS OF THE OPINION OF THE COURT=

arbeidsovereenkomsten deeltijdse

REPRESENTATIVE GROUNDS =

OP DEZE GRONDEN en met toepassing van de artikelen 1384 van het Burgerlijk Wetboek; 38-40-65 van het Strafwetboek; 1-4-25.1-27-28 van de wet van 8 april 1965, tot instelling van de arbeidsreglementen; 2 van de CAO van 27.2.1981, gesloten in de Nationale Arbeidsraad, betreffende sommige regelingen van het Arbeidsrecht ten aanzien van de deeltijdse arbeid, algemeen bindend verklaard bij K.B. van 21.9.1981; 11 bis 2 van de wet van 3.7.78, betreffende de arbeidsovereenkomsten; 56.1-57-59-60 van de wet van 5.12.68, betreffende de CAO's en PC's;

²⁹ This number (only for opinion of the court paragraphs) indicates the number of paragraphs in the text that the extracted paragraph represents.

SUMMARY OF CORRECTIONAL CASE

NAME OF CASE = /users/sien/pc/testset/algemeen/pp/p27

DATE = 7 april 1992

COURT = CORRECTIONELE RECHTBANK TE LEUVEN

REPRESENTATIVE PARAGRAPHS OF THE OFFENCES=

Ten nadele van ... een geldsom van ongeveer 17.500 fr. zijnde de recette van verschillende tourritten, die hem niet toebehoorde bedrieglijk weggenomen te hebben,

verdachte een dienstbode of een loondienaar zijnde en het feit gepleegd hebbende tegenover zijn meester voornoemd. (onderfarde 3).

1. als eigenaar of als bestuurder van een motorrijtuig, dit rijtuig in het verkeer te hebben gebracht of toegelaten te hebben dat het in het verkeer gebracht werd op de openbare weg, op terreinen die toegankelijk zijn voor het publiek of voor een zeker aantal personen die het recht hebben om er te komen, zonder dat de burgerrechtelijke aansprakelijkheid waartoe het aanleiding kan geven, gedekt was door een verzekering welke aan de bepalingen van de wet van 1 juli 1956 beantwoordt (art. 1, 2 § 1 en 18 §§ 1 en 3 van de wet van 1 juli 1956).

een motorvoertuig, dat vooraf niet ingeschreven was bij de Dienst van het Wegverkeer, op de openbare weg in het verkeer te hebben gebracht (art 3§1 van het KB van 31-12-1953, art 29 van de wet betreffende de politie over het wegverkeer, KB tot coördinatie van 16-3-1968).

REPRESENTATIVE PARAGRAPHS OF THE OPINION OF THE COURT=

Dat hij bij vonnis van 9.8.1989 werd veroordeeld tot 12 maanden gevangenisstraf met 5 jaar probatieuitstel voor 2/3de wegens 17 zware diefstallen en 2 pogingen daartoe; dat hij de opgelegde probatievoorwaarden niet naleefde; dat hij onderdak vond bij de genaamde ... en diens vertrouwen beschaamde door diens TV-toestel te stelen; dat hij hetzelfde deed bij zijn werkgever die hem met reden controleerde vermits beklagde toegaf dat hij hem bedroog bij de taxiritten;

(1)

Overwegende dat beklagde duidelijk onbetrouwbaar is; dat het niet uitgesloten is dat zijn vaste relatie, de middenstandsopleiding en de betere verstandhouding met zijn vader een stabiliserende invloed zullen hebben doch een voorbeeldige bestraffing zich opdringt; dat art. 464 SWB een minimum gevangenisstraf van drie maanden verplichtend stelt; dat het niet behoort verzachtende omstandigheden in te roepen of louter een geldboete op te leggen; dat dergelijke handel- en leefwijze maatschappelijk niet wordt aanvaard bij een voortdurend stijgende criminaliteit.

(1)

REPRESENTATIVE GROUNDS =

OM DEZE REDENEN en met toepassing van de artikelen 25-26-38-40-65-461-463-464 van het Strafwetboek; art. 1, 2 §1 en 18 §§ 1 en 3 wet 1.7.1956; art. 22-24 wet 21.11.1989; art. 3 § 1 K.B. 31.12.1953; art. 29 K.B. 16.3.1968;

SUMMARY OF CORRECTIONAL CASE

NAME OF CASE = /users/sien/pc/testset/algemeen/k/k11

DATE = 24 juni 1992.

COURT = CORRECTIONELE RECHTBANK TE LEUVEN

REPRESENTATIVE PARAGRAPHS OF THE OFFENCES=

In overtreding van art. 4 c en art. 15 van de wet van 30 juli 1979, betreffende de radioberechtiging, zonder schriftelijke vergunning van de Minister in het Rijk, aan boord van een zeeschip, binnenschip, luchtvaartuig of enige andere drager onderworpen aan het Belgisch recht, radioverbindingen die niet voor hem bestemd zijn, te hebben opgevangen of te hebben getracht ze op te vangen namelijk een radardetector van het merk BEL en type Micro-Eye;

ROUTINE CASE

REPRESENTATIVE GROUNDS =

OP DEZE GRONDEN en met toepassing van de artikelen 38-40-42-43-65 van het Strafwetboek; 4 c en 15 van de wet van 30 juli 1979, betreffende de radioberechtiging;

SUMMARY OF CORRECTIONAL CASE

NAME OF CASE = /users/sien/pc/trainmotiv3/verli

DATE = 16 september 1992.

COURT = CORRECTIONELE RECHTBANK TE LEUVEN

REPRESENTATIVE PARAGRAPHS OF THE OFFENCES=

met behulp van geweld of bedreiging, vernieling of beschadiging van andermans roerende eigendommen, namelijk deuren, flessen, glazen, stoelen, tafels, bakken bier en cola toebehorende aan de ... en ... te hebben gepleegd;

met de omstandigheid dat het feit gepleegd werd in vereniging of in bende, en dat ... het hoofd of aanstoker was.

Opzettelijk verwondingen of slagen te hebben toegebracht aan ... die voor deze een ziekte of ongeschiktheid tot het verrichten van persoonlijke arbeid ten gevolge hadden;

Opzettelijk verwondingen of slagen te hebben toegebracht aan ...;

Door gebaren of zinnebeelden ... te hebben bedreigd met een aanslag op personen of op eigendommen, waarop een criminele straf gesteld is.

REPRESENTATIVE PARAGRAPHS OF THE OPINION OF THE COURT=

Dat ...derhalve zonder grond voorhoudt dat hij wel verantwoordelijk kan zijn voor wat geschiedde bij wat hij een eerste vechtpartij heet, in de nabijheid van de discobar, maar niets te maken zou hebben met wat een weinig nadien nabij de toog voorviel, als het duidelijk een voortgezet groepsgebeuren uitmaakte, met eerste beklaagde wel als hevigst optredende persoon, derwijze dat hij op dat ogenblik als hoofdaanstoker dient beschouwd te worden;

(6)

REPRESENTATIVE KEY TERMS OF THE OPINION OF THE COURT=

groepsgebeuren vechtpartij

REPRESENTATIVE GROUNDS =

OP DEZE GRONDEN en met toepassing van de artikelen 1382 van het Burgerlijk Wetboek; 38-40-44-50-65-66-528-529-79-80-84-327-329-392-398/lid1-399/lid1 van het Strafwetboek;

3.3.6. Towards the Future

If the law is to remain the principal means of social control it must be manageable, available, realistic, workable, and interwoven easily with all aspects of social life. Our current methods for managing legal materials (dominated by print and paper) are not capable of coping with the quantity and complexity of law that now governs us (Susskind, 1996, p. 12-13). Our ability to use computer technology to capture, store, and reproduce data wildly surpasses our ability to use technology to help analyse, refine, and render more manageable the mass of data which data processing has spawned. We are great in getting information in, but not so good at extracting the information we want.

In this respect, the SALOMON project contributed to a better accessibility of correctional cases. In the long run users of text extraction and summarisation systems require results that have a recall and precision value of 100%. *Further progress in automatic text understanding and indexing* is absolutely necessary, but at the same time *parallel research in how to make the information in texts more accessible* seems also very fruitful. A combination of both approaches may result in achieving the above goals in the foreseeable future.

In this respect Susskind (1996, p. 83) warns against a too great fixation both with computerising existing tasks and with the organisation structures which have developed to discharge such tasks. Instead, he recommends that there should be a greater focus on processes, which are sets of tasks and activities which together give rise to an end result, a product or service which is of direct value to a customer, client or user.

In the case of automatic generation of document profiles maybe we have a natural temptation to find automated techniques for understanding the texts as they are now printed on paper and forget that the conventions we use for printed texts have been established based on the experience and constraints of the printed medium of centuries. Critics of full text retrieval systems believe that the addition of various complementary and supplementary techniques is simply tinkering at the edges of the greater problem, which is that text held within these systems is insufficiently structured ever to yield useful, valuable and directly applicable information (Susskind, 1996, p. 186). To make efficient use of information technology and of the electronic medium will put forward new conventions and consequent constraints on the way we communicate through this medium, but on the other hand will open new possibilities.

Communication with the electronic medium starts with document creation. Already a tool such as word processing can enable considerable change when it is not just confined to glorified typewriting, but when deployed as drafting tool which helps structure and organise materials. The presentation and publication of legal materials in conventional forms has evolved and become relatively standardised over centuries, it is probably worth to invest and standardise the presentation and publication of electronic legal documents. Document assembly systems could be build that hold a large number of standard templates of text, together with a representation of the expert's knowledge of how, where, and when these various standard words, sentences, and paragraphs should be used. According to Susskind (1996, p. 99) future judges will draft their judgements in electronic form, which will gradually be transmitted automatically to some repository and subjected to analysis and synthesis to lawyers trained as legal knowledge engineers, which analyse, interpret and repackage the formal sources of law and articulate it in structured format suitable for implementation as part of the legal information service.

Of course such a radical shift in approach is not for the immediate future. However, improvements in text creation relying on a better structuring and standardisation together with the refinement of natural language processing and statistical indexing techniques may one day result in correct and exhaustive extraction of information applicable for multiple text based tasks.

Especially the *aspect of text structure* is beneficial in incorporating in document generation systems. Text structure of specific text types usually is straightforward and not subject of different interpretations of uses of a text. More research and description of specific text types is useful for building text grammars on the basis of which text is organised and structured automatically during text creation. Of course the interfaces that such drafting tools employ much be user friendly and transparent.

At present different document representation standards are available (André, Furuta, & Quint, 1989), among which the most important ODA, SGML and HTML. SGML allows to represent documents in terms of their logical structure. Document accessibility and information accessibility in text is an important current research topic. It is probable that document representation standards will be refined in the near future and will be further integrated in word processing packages.

Despite the very good results of the phase 'Initial Structuring' based on document parsing, it would be wise to consider alternative paths because of widespread trends in document management. Contacts with the

Document Architecture Research Group confirmed the feasibility of incorporating a text grammar during text creation.³⁰ Anyway, the point is that indisputable structure and content attributes (e.g. the different components of a case, names of parties, date of case, name of court, etc.) may be defined at text generation or after generation of draft texts, refining a consequent use of the text in retrieval, abstracting, and extracting systems. In this respect the text grammar we developed for the phase 'Initial Structuring' offers useful information. However, more studies that describe and explain text and content structure of legal text are necessary to apply such an approach on a broad scale. It may be wise to invest in studies that describe, explain and ideally standardise the structure of legal document types (e.g. court decisions) on a national and even European scale (cf. Magnusson Sjöberg, 1996). There is a growing international electronic information flow, of which legal information has a considerable share. Such an approach contributes to an improved theoretical framework regarding structurally and semantically important components of legal documents and provides an important basis for document related task such as browsing, information extraction, information linking and retrieval.

The SALOMON-research also contributed to the *automatic recognition of topic structure and relevant text units in text of which the text structure is not a priori defined*. In this way the research contributes in automatically identifying the permanent aboutness of natural language texts. The techniques proposed may be further refined in the future, especially for identifying other discourse models.

A vital aspect of successful document management is reaching a *consensus on terminology*. The development of a taxonomy of, especially legal, concepts is a central task for the future. This would allow for writing aids automatically or interactively correct and standardise the spelling of these concepts in the texts. Consequent electronic uses of the text such as indexing, abstracting, and text linking will be ameliorated.

As our research showed any approach (deep or shallow) in varies degrees relies on a priori expectations of the occurrence of surface features in the corpus. So, in future commercial applications it is necessary *to inform users* of applications that rely on indexing descriptions, *about the techniques used*.

Additional details can be found in TRS-76.

³⁰ Filip Evenepoel personal communication.

4. General Conclusions

Our research contributes to the study of legal language and to the disciplines of automatic indexing and abstracting. On the practical side, we demonstrated that case profiles, automatically generated are of great help to the lawyer.

Our research learns that full text understanding, especially of legal texts, is currently out of reach. This would not only entail storing and applying a huge amount of linguistic, domain-specific, and common world knowledge about the information structures and concepts used in the texts. It would also implicate full sentence analysis and integrating the significance of entire sentences into an overall source meaning representation. Moreover, the research indicates that legal language has many characteristics of ordinary language and often deals with 'open texture' concepts. Such concepts are left explicitly vague by the creator of the legal texts, making different interpretations possible. In addition of understanding the communication structures that creator and user share on a cognitive level, and of applying the inferential structures that creator and user normally share for adding meaning to the surface features of a text, complete understanding must be able to deal with the different uses a person makes of a text at a given time or within certain circumstances. Thus, complete understanding of the texts would entail a user's specific background knowledge, emotions, goals, plans, etc. We lack the appropriate models to efficiently implement such an approach. Besides, as our investigation shows, such an approach is not desired by legal professionals.

However, the growth of texts electronically available, the urgent need to keep law manageable, workable and accessible, and the globalisation of specific types of legal texts all point to the need to make information in legal texts, especially this information that creator and user of the text deal in a straightforward manner, easily accessible. This goal of the project has been studied for a specific text genre (court decisions) in the prospect of a specific useful application (automatic construction of case profiles).

A text has a relatively permanent aboutness. This aboutness is sometimes defined by the creator of the electronic document by adding content attributes to the document. When the content is variable and not restricted to a specific narrow domain, such an attribution is often subjective and inconsistent. So, methods for automatic attribution of content attributes are investigated.

SALOMON designed and developed techniques for automatically identifying and extracting relevant information from the cases. The extracted text units are used to form a case profile, which is indicative and informative of the content of the full text of the case.

We extensively studied and analysed the information structure and its appearance in linguistic structures of correctional cases. The results thereof were successfully applied to extract some general information from the case texts and to determine the main themes of the alleged offences and opinion of the court. The insights gained by this analysis have a definite potential for current and future text based applications.

Research into the application of linguistic tools contributed in finding techniques that enable more accurate inferences about document content. For this purpose existing natural language processing tools were adapted in an interesting and novel manner by merging traditional linguistic work with techniques on semi-automatic knowledge acquisition. Legal concepts are often conveyed by word groups. The foundations for automatic recognition of word groups have been laid.

Moreover, our research succeeded in finding interesting techniques for automatic text structure recognition and theme identification in natural language texts, which not only may be applied in automatic indexing and abstracting, but also in automatic linking of texts even beyond the legal field. The use of cluster algorithms based on the selection of representative objects is new in the context of text-based systems. These algorithms provide the possibility to identify highly informative text units that through their lexical patterns are linked to other text units.

Our research confirms that identification of the text structure is an important first step in automatically identifying the text's content. We argue that, when text structure of a specific text type is simple, straightforward, unambiguous, and largely disseminated, it is advisable to standardise it and to impose it on document drafting, allowing for an automatic control and for feedback by the writer.

The results of the demonstrator indicate that we could simulate part of the manual practice of abstracting legal cases. This includes the identification of the case type, the structure of the information, deletion of redundant and insignificant information, and selection of thematically relevant text units and key terms. As it requires a subjective interpretation and a large amount of contextual knowledge, recognition of

legal principle rules in the opinion of the court is more difficult. Nonetheless, a tool like the SALOMON demonstrator can simplify the lawyer's job a great deal. It does not solve any legal questions, but guides the user effectively towards relevant texts.

The research and expertise of SALOMON is further explored and applied in the Media-On-Line project, sponsored by the 'Vlaams Actieprogramma Informatietechnologie'. This project aims at automatically attributing content attributes to Belgian magazine articles (written in Dutch and in French), making an automatic routing of the articles according to users' profiles possible. The research has also implications for automatic linking. More specifically this project aims at recognising index terms of specific semantic categories (person and company names) and computing the relative importance of these index terms in the texts. The project also attempts to automatically attribute general category labels to the magazine articles. The expertise of partially parsing texts and targeting specific information in texts is further explored in a separate part of the Media-On-Line project, which aims at extracting specific information in classified newspaper ads.

5. Literature

- Adriaens, G. (1996). SECC: Using Text Structure Information to Improve Checker Quality and Coverage. In *Proceedings CLAW-1996* (Leuven) (pp. 226-232).
- Allen, J. (1987). *Natural Language Understanding*. Benjamin/Cummings, Menlo Park, CA.
- Allen, R. B. (1990). User Models: Theory, Method and Practice. *International Journal of Man Machine Studies*, 32, pp. 511-543.
- Anderberg, M. R. (1973). *Cluster Analysis for Applications*. Academic Press, New York - London.
- André, J., Furuta, R., & Quint, V. (1989). *Structured Documents*. Cambridge University Press, Cambridge - Sydney.
- Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D., & Tyson, M. (1993). FASTUS: A Finite-state Processor for Information Extraction from Real-world Text. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 1172-1178). Morgan Kaufmann Publishers, San Mateo, CA.
- Baxendale, P. B. (1958). Machine-made Index for Technical Literature - an Experiment. *IBM Journal of Research and Development*, 2 (4), pp. 354-361.
- Bellefroid, J. (1933). *Beschouwingen over de Nederlandsche Rechtstaal in Vlaanderen*. Het Kompas, Mechelen.
- Bernstein, L. M., & Williamson, R. E. (1984). Testing of a Natural Language Retrieval System for a Full Text Knowledge Base. *Journal of the American Society for Information Science*, 35 (4), pp. 235-247.
- Bing, J., & Fjeldvig, T. (1984). *Handbook of Legal Information Retrieval*. North Holland, Amsterdam.
- Bing, J. (1986). Legal Text Retrieval Systems: The Unsatisfactory State of the Art. *Journal of Law and Information Science*, 2 (1), pp.1-17.
- Black, E. (1993). Statistically-based Computer Analysis of English. In E. Black, R. Garside, & G. Leech (Eds.), *Statistically-driven Computer Grammars of English: The IBM/Lancaster Approach* (pp. 1-16). Rodopi, Amsterdam - Atlanta.
- Blair, D. C., & Maron, M. E. (1985). An Evaluation of Retrieval Effectiveness for a Full-text Document-retrieval System. *Communications of the ACM*, 28 (3), pp. 289-299.
- Boreham, J., & Niblett, B. (1976). Classification of Legal Texts by Computer. *Information Processing & Management*, 12, pp. 125-132.
- Bosch, P. (1983). "Vagueness" is Context-dependence: A Solution to the Sorites Paradox. In T. T. Balmer & M. Pinkal (Eds.), *Approaching Vagueness* (pp. 189-210). North-Holland, Amsterdam.
- Botofago, R. A. (1993). Cluster Analysis for Hypertext Systems. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 116-125). ACM, New York.
- Bourigault, D. (1993). Analyse Syntaxique Locale pour le Repérage de Termes Complexes dans un Texte. *Revue T.A.L.*, 34 (2).
- Bourigault, D. (1995). LEXTER, a Terminology Extraction Software for Knowledge Acquisition from Texts. *Proceedings KAW-1995* (Calgary).
- Brill, E. (1992). A Simple Rule-based Part of Speech Tagger. In *Proceedings ANLP-1992* (Trento) (pp. 152-155).
- Brill, E. (1993). *A Corpus-based Approach to Language Learning*. PhD dissertation, University of Pennsylvania.
- Brown, A. L., & Day, J. D. (1983). Macrorules for Summarizing Text: The Development of Expertise. *Journal of Verbal Learning and Verbal Behaviour*, 22, pp. 1-14.
- Brown, M. G., Foote, J. T., Jones, G. J. F., Sparck Jones, K., & Young, S. J. (1995). Automatic Content-based Retrieval of Broadcast News. In *Proceedings ACM Multimedia '95* (pp. 35-43).
- Capper, P., & Susskind, R. (1988). *Latent Damage Adviser – The Expert System*. Butterworths, London.
- Charniak, E. (1983). A Parser with Something for Everyone. In M. King (Ed.), *Parsing Natural Language* (pp. 117-149). Academic Press, London.
- Charrow, V. R., Crandall, J. A., & Charrow, R. P. (1982). Characteristics and Functions of Legal Language. In R. Kittredge & J. Lehrberger (Eds.), *Sublanguage: Studies of Language in Restricted Semantic Domains* (pp. 175-190). W. de Gruyter, Berlin - New York.
- Chinchor, N. (1992). MUC-4 Evaluation Metrics. In *Fourth Message Understanding Conference (MUC-4). Proceedings of a Conference Held in McLean, Virginia June 16-18, 1992* (pp. 22-29). Morgan Kaufmann Publishers, San Mateo, CA.

- Cremmings, E. T. (1982). *The Art of Abstracting*. ISI Press, Philadelphia.
- Croft, W. B., Krovetz, R., & Turtle, H. (1990). Interactive Retrieval of Complex Documents. *Information Processing & Management*, 26 (5), pp. 593-613.
- Croft, W. B., Turtle, H. R., & Lewis, D. D. (1991). The Use of Phrases and Structured Queries in Information Retrieval. In A. Bookstein, Y. Chiaramella, G. Salton, V. V. Raghaven (Eds.), *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval* (pp. 32-45). ACM, New York.
- Crouch, C. J. (1988). A Cluster Based Approach to Thesaurus Construction. In *11th International Conference on Research and Development in Information Retrieval* (pp. 309-320). ACM, New York.
- Crystal, D., & Davy, D. (1969). *Investigating English Style* (Chapter 8: The Language of Legal Documents). Longman, London.
- Dabney, D. P. (1986). The Curse of Thamus: An Analysis of Full-text Legal Document Retrieval. *Law Library Journal*, 78, pp. 5-40.
- De Beaugrande, R.-A., & Dressler, W. U. (1981). *Introduction to Text Linguistics*. Longman, London - New York.
- DeBessonet, C. G. (1991). *A Many-valued Approach to Deduction and Reasoning for Artificial Intelligence*. Kluwer, Dordrecht.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41 (6), pp. 391-407.
- De Groot, G. R. (1987). Het vertalen van juridische teksten. In J. P. Balkema & G. R. de Groot (Eds.), *Recht en vertalen* (pp. 13-36). Kluwer, Deventer.
- DeJong, G. (1979). *Skimming Stories in Real Time: An Experiment in Integrated Understanding*. PhD dissertation, Yale University.
- DeJong, G. (1982). An Overview of the FRUMP System. In W. G. Lehnert & M. H. Ringle. (Eds.), *Strategies for Natural Language Processing* (pp. 149-176). Lawrence Erlbaum Associates, Hillsdale-London.
- De Mulder, R. V. (1984). *Een model voor Juridische Informatica*. Vermande, Lelystad.
- Dick, J. P. (1991). Representation of Legal Text for Conceptual Retrieval. In *Proceedings ICAIL-1991* (Oxford) (pp. 244-253).
- Douglas, S., & Hurst, M. (1996). Controlled Language Support for Perkins Approved Clear English (PACE). In *Proceedings CLAW-1996* (Leuven) (pp. 93-105).
- Dyer, M. (1983). *In-depth Understanding*. MIT Press, Cambridge.
- Edmundson, H. P. (1964). Problems in Automatic Abstracting. *Communications of the ACM*, 7 (4), pp. 259-263.
- Edmundson, H. P. (1969). New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16 (2), pp. 264-285.
- Edwards, J. S. (1991). *Building Knowledge-based Systems: Towards a Methodology*. Pitman Publishing, London.
- Earl, L. L. (1970). Experiments in Automatic Extracting and Indexing. *Information Storage and Retrieval*, 6 (6), pp. 313-334.
- Ellis, D., Furner-Hines, J., & Willett, P. (1993). Measuring the Degree of Similarity between Objects in Text Retrieval Systems. *Perspectives in Information Management*, 3 (2), pp. 128-149.
- Fagan, J. L. (1989). The Effectiveness of a Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval. *Journal of the American Society for Information Science*, 40 (2), pp. 115-132.
- Fodor, J. *et al.* (1980). Against Definitions. *Cognition*, 8, pp. 263-367.
- Foqué, R., & 't Hart, A. C. (1990). *Instrumentaliteit en rechtsbescherming. Grondslagen van een strafrechtelijke waardendiscussie*. Gouda Quint, Arnhem.
- Fox, C. (1992). Lexical Analysis and Stoplists. In W. B. Frakes & R. Baezo-Yates (Eds.), *Information Retrieval. Data Structures & Algorithms* (pp. 102-130). Prentice Hall, Englewood Cliffs, NJ.
- Frakes, W. (1992). Stemming Algorithms. In W. B. Frakes & R. Baezo-Yates (Eds.), *Information Retrieval: Data Structures & Algorithms* (pp. 131-160). Prentice Hall, Englewood Cliffs, NJ.
- Fuhr, N. (1992). Probabilistic Models in Information Retrieval. *The Computer Journal*, 35, pp. 243-255.
- Gelbart, D., & Smith, J. C. (1991). Beyond Boolean Search: FLEXICON, A Legal Text-based Intelligent System. *Third International Conference on Artificial Intelligence & Law. Proceedings of the Conference* (pp. 225-234). ACM, New York.
- Gelbart, D., & Smith, J. C. (1994). Automating the Process of Abstracting Legal Cases. *International Journal of Law and Information Technology*, 1 (3), pp. 332-334.
- Goodrich, P. (1984). The Role of Linguistics in Legal Analysis. *The Modern Law Review*, 47, pp. 523-534.

- Goodrich, P. (1987). *Legal Discourse. Studies in Linguistics, Rhetoric and Legal Analysis*. St. Martin's Press, New York.
- Grishman, R., & Kittredge, R. (Eds.) (1986). *Analyzing Sublanguage in Restricted Domains: Sublanguage Description and Processing*. Lawrence Erlbaum, Hillsdale.
- Hafner, C. D. (1981). *An Information Retrieval System Based on a Computer Model of Legal Knowledge*. UMI Research Press, Ann Arbor.
- Hahn, U. (1990). Topic Parsing: Accounting for Text Macro Structures in Full-text Analysis. *Information Processing & Management*, 26 (1), pp. 135-170.
- Hahn, U. & Reimer, U. (1986). Semantic Parsing and Summarizing of Technical Texts in the TOPIC System. In R. Kuhlen (Ed.), *Informationslinguistik* (pp. 153-193). Niemeyer, Tübingen.
- Hart, H. L. A. (1961). *The Concept of Law*. Clarendon Press, Oxford.
- Hayes, P. J. (1992). Intelligent High-volume Text Processing Using Shallow, Domain-specific Techniques. In P. S. Jacobs (Ed.), *Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval* (pp. 227-241). Lawrence Erlbaum Associates, Hillsdale, NJ.
- Hayes, P. J., & Weinstein, S. P. (1991). Construe/TIS: A System for Content Based Indexing of a Database of News Stories. *2nd Annual Conference on Innovative Applications of Artificial Intelligence* (pp. 49-64). AAAI Press, Menlo Park, CA.
- Hearst, M. A., & Plaunt, C. (1993). Subtopic Structuring for Full-length Document Access. In R. Korfhage, E. Rasmussen, & P. Willett, *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 59-68). ACM, New York.
- Hemels, F. (1994). *REMINDER: Automatic Discovery of Semantic Associations in Text Corpora*. Thesis, University of Twente.
- Hofhuis, H. F. M. (1988). De taal van de rechter. In W. M. J. Bekkers et al. (Eds.), *Meesterlijke taal* (pp. 51-64). Tjeenk Willink, Zwolle.
- Hutchins, J. (1987). Summarization: Some Problems and Methods. In K. P. Jones (Ed.), *Meaning: The Frontier of Informatics (Informatics 9)* (pp. 151-173). Aslib, London.
- Iwayama, M. & Tokunaga, T. (1995). Hierarchical Bayesian Clustering for Automatic Text Classification. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence* (pp. 1322-1327). Morgan Kaufmann Publishers, San Mateo, CA.
- Jackendoff, R. (1983). *Semantics and Cognition*. MIT Press, Cambridge, MA.
- Jacobs, P. S. (1993). Using Statistical Methods to Improve Knowledge-based News Categorization. *IEEE Expert*, 8 (2), pp. 13-23.
- Jacobs, P. S., & Rau, L. F. (1990). SCISOR: Extracting Information from On-line News. *Communications of the ACM*, 33 (11), pp. 88-97.
- Jacobs, P. S., & Rau, L. F. (1993). Innovations in Text Interpretation. *Artificial Intelligence*, 63, pp. 143-191.
- Jones, W. P., & Furnas, G. W. (1987). Pictures of Relevance: A Geometric Analysis of Similarity Measures. *Journal of the American Society for Information Science*, 38 (6), pp. 420-442.
- Karlgren, H., & Walker, D. E. (1983). The Polytext System: A New Design for a Text Retrieval System. In F. Kiefer (Ed.), *Questions and Answers* (pp. 273-294). Reidel, Dordrecht.
- Kelso, L. O. (1946). Does the Law Need a Technological Revolution? *Rocky Mountain Law Review*, 18, pp. 378-392.
- Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data. An Introduction to Cluster Analysis*. John Wiley & Sons, New York.
- Kintsch, W., & van Dijk, T. (1978). Toward a Model of Text Comprehension and Production. *Psychological Review*, 85 (5), pp. 363-394.
- Kittredge, R. (1982). Variation and Homogeneity of Sublanguages. In R. Kittredge & J. Lehrberger (Eds.) *Sublanguage: Studies of Language in Restricted Semantic Domains* (pp. 107-137). W. de Gruyter, Berlin.
- Kittredge, R., & Lehrberger, J. (Eds.) (1982). *Sublanguage: Studies of Language in Restricted Semantic Domains*. W. de Gruyter, Berlin.
- Kupiec, J., Pedersen, J., & Chen, F. (1995). A Trainable Document Summarizer. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 68-73). ACM, New York.
- Lancaster, F. W. (1991). *Indexing and Abstracting in Theory and Practice*. The Library Association, London.

- Lee, J. H. (1995). Combining Multiple Evidence from Different Properties of Weighting Schemes. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 180-188). ACM, New York.
- Lehnert, W. G. (1981). Plot Units and Narrative Summarization. *Cognitive Science*, 5, pp. 293-331.
- Lehnert, W. G. (1982). Plot Units: a Narrative Summarization Strategy. In W. G. Lehnert & M. H. Ringle (Eds.), *Strategies for Natural Language Processing* (pp. 375-412). Lawrence Erlbaum Associates, Hillsdale - London.
- Leliard, J. (1979). *Het kleed van Themis. Beschouwingen over de rechtstaal in het Nederlandse taalgebied*. Kluwer, Antwerpen.
- Lewis, D. D. (1992a). *Representation and Learning in Information Retrieval*, PhD dissertation, University of Massachusetts.
- Lewis, D. D. (1992b). An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 37-50). ACM, New York.
- Lewis, D. D. (1995). Evaluating and Optimizing Autonomous Text Classification Systems. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 246-254). ACM, New York.
- Lewis, D. D., Croft, W. B., & Bhandaru, N. (1989). Language-oriented Information Retrieval. *International Journal of Intelligent Systems*, 4, pp. 285-318.
- Lewis, D. D., & Sparck Jones, K. (1996). Natural Language Processing for Information Retrieval. *Communications of the ACM*, 39, 1, pp. 92-101.
- Liddy, E., Jörgenson, C. L., Sibert, E., & Yu, E. S. (1991). Sublanguage Grammar in Natural Language Processing for an Expert System. In *RIA0 91 Conference Proceedings Intelligent Text and Image Handling* (pp. 707-717). C.I.D.-C.A.S.I.S., Paris.
- Liddy, E. D., & Paik, W. (1993). Document Filtering Using Semantic Information from a Machine Readable Dictionary: Preliminary Test Results. *Proceedings of the ACL Workshop on Very Large Corpora* (15 pp.).
- Loth, M. A. (1984). *Recht en taal: een kleine methodologie*. Gouda Quint, Arnhem.
- Luhn, H. P. (1957). A Statistical Approach to the Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1, pp. 309-317.
- Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2, pp. 159-165.
- McDonald, D. D. (1992). Robust Partial-parsing through Incremental, Multi-algorithm Processing. In P. S. Jacobs (Ed.), *Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval* (pp. 83-99). Lawrence Erlbaum Associates, Hillsdale-London.
- Maeda, T., Momouchi, Y., & Sawamura, H. (1980). An Automatic Method for Extracting Significant Phrases in Scientific or Technical Documents. *Information Processing & Management*, 16, pp. 119-127.
- Maes, A. A. (1991). *Nominal Anaphors and the Coherence of Discourse*. PhD Dissertation, Katholieke Universiteit Brabant.
- Magnusson Sjöberg, C. (1996). Corpus Legis - A Legal Document Management Project. In M. Brinnen (Ed.), *Telekommunikation -rättsliga aspekter* (pp. 160-190). Norstedts Publishing Company, Stockholm.
- Maley, Y. (1985). Judicial Discourse: The Case of the Legal Judgment. In J. E. Clark (Ed.), *The Cultivated Australian: Festschrift for Arthur Delbridge* (pp. 159-173). Buske, Hamburg.
- Masand, B., Linoff, G., & Waltz, D. (1992). Classifying News Stories using Memory Based Reasoning. *Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval* (pp. 59-65).
- Mathis, B., Rush, J. E., & Young, C. E. (1973). Improvement of Automatic Abstracts by the Use of Structural Analysis. *Journal of the American Society for Information Science*, 24, pp. 101-109.
- Miike, S., Itoh, E., Ono, K., & Sumita, K. (1994). A Full-text Retrieval System with a Dynamic Abstract Generation Function. In W. B. Croft & C. J. van Rijsbergen (Eds.), *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* (pp. 152-161). Springer-Verlag, London - Budapest.
- Mincke, W. (1971). Die Problematik von Recht und Sprache in der Übersetzung von Rechtstexten. *Archiv für Rechts- und Sozialphilosophie*, 57, pp. 446- 465.
- Mital, V., & Johnson, L. (1992). *Advanced Information Systems for Lawyers*. Chapman and Hall, London.
- Moles, R. N., & Dayal, S. (1992). There is More to Life than Logic. *Journal of Law and Information Science*, pp. 188-218.
- Nagy, G., & Seth, S. (1992). A Prototype Document Image Analysis System for Technical Journals. *Computer*, 25 (7), 10-22.

- Noreault, T., McGill, M., & Koll, M. B. (1981). A Performance Evaluation of Similarity Measures, Document Term Weighting Schemes and Representations in a Boolean Environment. In R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, & P. W. Williams (Eds.), *Information Retrieval Research* (pp. 57-76). Butterworth & Co., London - Toronto.
- Paice C. D. (1981). The Automatic Generation of Literature Abstracts: An Approach Based on the Identification of Self-indicating Phrases. In R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, & P. W. Williams (Eds.), *Information Retrieval Research* (pp. 172-191). Butterworth & Co, London - Toronto.
- Paice, C. D. (1990). Constructing Literature Abstracts by Computer: Techniques and Prospects. *Information Processing & Management*, 26 (1), pp. 171-186.
- Paice, C. D. (1991). The Rhetorical Structure of Expository Text. In K. P. Jones (Ed.), *Informatics 11. The Structuring of Information* (pp. 1-25). Aslib, London.
- Paice, C. D., & Jones, P. A. (1993). The Identification of Important Concepts in Highly Structured Technical Papers. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 69-78). ACM, New York.
- Pinto Molina, M. (1995). Documentary Abstracting: Toward a Methodological Model. *Journal of the American Society for Information Science*, 46 (3), pp. 225-234.
- Prikhod'ko, S. M., & Skorokhod'ko, E. F. (1982). Automatic Abstracting from Analysis of Links between Phrases. *Nauchno-Tekhnicheskaya Informatsiya, Seriya 2*, 16 (1), pp. 27-32.
- Pulman, S. G., & Rayner, M. (1994). *Computer Processable Controlled Language*. Technical Report, SRI International, Cambridge Computer Science Research Centre.
- Rama, D. V., & Srinivasan, P. (1993). An Investigation of Content Representation Using Text Grammars. *ACM Transactions on Information Systems*, 11 (1), pp. 51-75.
- Rau, L. F., Jacobs, J. S., & Zernik, U. (1989). Information Extraction and Text Summarization Using Linguistic Knowledge Acquisition. *Information Processing & Management*, 25 (4), pp. 419-428.
- Reinsma, M., & Reinsma, R. (1976). De Vrouw in wier Lichaam zich eerstbedoeld Leven ontwikkelt of zestig jaar Nederlandse rechtstaal. *Nederlands Juristenblad*, 26, pp. 857-872.
- Riloff, E., & Lehnert, W. (1992). Classifying Texts Using Relevancy Signatures. In AAAI-92. *Proceedings of the Tenth National Conference on Artificial Intelligence* (pp. 329-334). AAAI Press, Menlo Park, CA.
- Riloff, E., & Lehnert, W. (1994). Information Extraction as Basis for High-precision Text Classification. *ACM Transactions on Information Systems*, 12 (3), pp. 296-333.
- Ro, J. S. (1988). An Evaluation of the Applicability of Ranking Algorithms to Improve the Effectiveness of Full-text Retrieval. II. On the Effectiveness of Ranking Algorithms on Full-text Retrieval. *Journal of the American Society for Information Science*, 39 (3), pp. 147-160.
- Robertson, S. E., & Sparck Jones, K. (1976). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27, pp. 129-146.
- Rowley, J. E. (1988). *Abstracting and Indexing*. Clive Bingley, London.
- Rumelhart, D. (1977). Understanding and Summarising Brief Stories. In D. LaBerge & S. J. Samuels (Eds.) *Basic Processes in Reading: Perception and Understanding* (pp. 265-303). Lawrence Erlbaum, Hillsdale.
- Rush, J. E., Salvador, R., & Zamora, A. (1971). Automatic Abstracting and Indexing. Production of Indicative Abstracts by the Application of Contextual Inference and Syntactic Coherence Criteria, *Journal of the American Society for Information Science*, 22 (4), pp. 260-274.
- Salton, G. (1971). *The Smart Retrieval System*. Prentice Hall, Englewood Cliffs, NJ.
- Salton, G. (1989). *Automatic Text Processing. The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Publishing Company, Reading, MA.
- Salton, G., & Allan, J. (1994). Automatic Text Decomposition and Structuring. In *RIAO 94 Conference Proceedings Intelligent Multimedia Information Retrieval Systems and Management* (pp. 6-20). C.I.D.-C.A.S.I.S, Paris.
- Salton, G., Allan, J., & Buckley, C. (1993). Approaches to Passage Retrieval in Full Text Information Systems. In R. Korfhage, E. Rasmussen, & P. Willett (Eds.), *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 49-58). ACM, New York.
- Salton, G., Allan, J., Buckley, C., & Singhal, A. (1994). Automatic Analysis, Theme Generation, and Summarization of Machine-readable Texts. *Science*, 264, pp. 1421-1426.
- Salton, G., Allan, J., & Singhal, A. (1996). Automatic Text Decomposition and Structuring. *Information Processing & Management*, 32 (2), pp. 127-138.

- Salton, G., & Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24 (5), pp. 513-523.
- Salton, G., & Buckley, C. (1991). Automatic Text Structuring and Retrieval Experiments in Automatic Encyclopedia Searching. In A. Bookstein, Y. Chiaramella, G. Salton, & V. V. Raghaven (Eds.). *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval* (pp. 21-30). ACM, New York.
- Salton, G., & Buckley, C. (1992). Automatic Text Structuring Experiments. In P. S. Jacobs (Ed.), *Text-based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval* (pp. 199-210). Lawrence Erlbaum Associates, Hillsdale, NJ.
- Salton, G., & Yang, C. S. (1973). On the Specification of Term Values in Automatic Indexing. *Journal of Documentation*, 29, pp. 361-372.
- Schank, R., & Riesbeck, C. (Eds.) (1981). *Inside Computer Understanding*. Lawrence Erlbaum, Hillsdale, NJ.
- Sidner, C. L. (1983). Focusing in the Comprehension of Definite Anaphora. In M. Brady & R. C. Berwick (Eds.), *Computational Models of Discourse* (pp. 267-330). The MIT Press, Cambridge, MA - London.
- Small, H., & Sweeney, E. (1985). Clustering the Science Citation Index Using Co-citations. *Scientometrics*, 7, pp. 391-409.
- Smeaton, A. F. (1995). Low Level Language Processing for Large Scale Information Retrieval: What Techniques Actually Work. In *Proceedings of the Workshop on Terminology, Information Retrieval and Linguistics* (Rome), 9 pp.
- Smeaton, A. F., & Sheridan, P. (1991). Using Morpho-syntactic Language Analysis in Phrase Matching. In *RIA0 91 Conference Proceedings Intelligent Text and Image Handling* (pp. 414-429). C.I.D.-C.A.S.I.S., Paris.
- Sparck Jones, K. (1973). Index Term Weighting. *Information Storage and Retrieval*, 9, pp. 619-633.
- Sparck Jones, K. (1993). What Might Be in a Summary. In G. Knorz, J. Krause, C. Womser-Hacker (Eds.), *Information Retrieval 93: Von der Modulierung zum Anwendung* (pp. 9-26).
- Soetaert, R. (1980). Rechtstaal. *Nederlands van nu*, 28 (2), pp. 54-62.
- Studnicki, F. et al. (1992). Introduction to Cross-reference Clauses in Legal Texts. *Informatica e Diritto*, 18(1-2), pp. 213-237.
- Susskind, R. E. (1986). Expert Systems in Law: a Jurisprudential Approach to Artificial Intelligence and Legal Reasoning. *The Modern Law Review*, 49, pp. 168-194.
- Susskind, R. E. (1987). *Expert Systems in Law*. Clarendon Press, Oxford.
- Susskind, R. (1996). *The Future of Law. Facing the Challenges of Information Technology*. Clarendon Press, Oxford.
- Tait, J. I. (1985). Generating Summaries Using a Script-based Language Analyser. In L. Steels & J. A. Campbell (Eds.), *Progress in Artificial Intelligence* (pp. 312-318). John Wiley & Sons, New York -Toronto.
- Tuthill, W. (1981). HUM – A Concordance and Text Analysis Package. Technical Paper, Comparative Literature Department, University of California, Berkeley.
- Van den Bergh, G. C. J. J., & Broekman, J. M. (1979). *Recht en taal. Preadvies*. Kluwer, Deventer.
- Van den Hoven, P. J. (1988). Rechtszekerheid, rechtvaardigheid, verstaanbaarheid. *Tijdschrift voor Taalbeheersing*, 10 (3), pp. 209-219.
- Van der Eijck, P., de Koning, M., & van der Steen, G. (1996). Controlled Language Correction and Translation. In *Proceedings CLAW-1996* (Leuven) (pp.64-73).
- Van Dijk, T. A. (1980). *Macrostructures. An Interdisciplinary Study of Global Structures in Discourse, Interaction, and Cognition*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Van Ginneken, J. (1914). *Handboek der Nederlandsche Taal. Deel II De Sociologische Structuur*. Malmberg, Nijmegen.
- Van Noortwijk, C. (1995). *Het woordgebruik meester. Een vergelijking van enkele kwantitatieve aspecten van het woordgebruik in juridische en algemeen Nederlandse teksten*. Vermande, Lelystad.
- Voorhees, E. M. (1986). Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval. *Information Processing & Management*, 22, pp. 465-476.
- Waissman, F. (1945). Verifiability. In *Proceedings of the Aristotelian Society* (Supplement 19) (pp. 119-150). Cited from the reprint in A. Flew (Ed.) (1965) *Logic and Language* (pp. 122-151). Anchor Books, Garden City.
- Wang, J. T. L., & Ng, P. A. (1992). TEXPROS: An Intelligent Document Processing System. *International Journal of Software Engineering and Knowledge Engineering*, 2 (2), 171-196.
- Willett, P. (1988). Recent Trends in Hierarchic Document Clustering: A Critical Review. *Information Processing & Management*, 24, pp. 577-597.
- Wittgenstein, L. (1953). *Philosophical Investigations*. Blackwell, Oxford.

- Young, S. R., & Hayes, P. J. (1985). Automatic Classification and Summarization of Banking Telexes. In *The Second Conference on Artificial Intelligence Applications. The Engineering of Knowledge Based Systems* (pp. 402-408). IEEE Computer Society Press, Washington, DC.
- Zeleznikow, J., & Hunter, D. (1992). Rationales for the Continued Development of Legal Expert Systems. *Journal of Law and Information Science*, pp. 94-110.
- Zeleznikow, J., & Hunter, D. (1994). *Building Intelligent Legal Information Systems*. Kluwer Law and Taxation Publishers, Deventer - Boston.
- Zoeppritz, M. (1989). Zu einigen sprachlichen Formen und deren Funktion in juristische Texten. In F. Haft (Ed.) *Das LEX-Projekt: Entwicklung eines juristischen Expertensystems* (pp. 9-32). Attempto, Tübingen.

6. TECHNICAL REPORTS

- TRS-1** Gebruers, R., *Rechtstaal: Voer voor de computer?*, November 1993, 23 pp.
- TRS-2** Gebruers, R., *Automatische excerptering van rechterlijke beslissingen: Taakomschrijving en Algemene Aanpak*, February 1994, 8 pp.
- TRS-3** Gebruers, R., *Basisingrediënten voor praktijkgerichte natuurlijke-taalverwerking: Een rijk en dynamisch lexicon*, March 1994, 23 pp.
- TRS-4** Gebruers, R., *Te plannen onderzoeksactiviteiten*, April 1994, 4 pp.
- TRS-5** Gebruers, R., *Globale planning en taakverdeling*, May 1994, 6 pp.
- TRS-6** Gebruers, R., *Automatische segmentering van vonnissen*, June 1994, 3 pp.
- TRS-7** Gebruers, R., *Automatische annotatie en ontleding van documenten*, July 1994, 7 pp.
- TRS-8** Gebruers, R., *Automatische afbakening van zinseenheden met het oog op inhoudsanalyse*, October 1994, 14 pp.
- TRS-9** Gebruers, R., *Tussentijdse evaluatie van het SALOMON-project*, November 1994, 9 pp.
- TRS-10** Gebruers, R., *SALOMON: Automatische excerptering van rechterlijke beslissingen*, December 1994, 15 pp.
- TRS-11** Gebruers, R., *Het SALOMON-project: De taaltechnologische inbreng*, January 1995, 26 pp.
- TRS-12** Gebruers, R., *Woordfrequenties in correctionele vonnissen*, March 1995, 30 pp.
- TRS-13** Gebruers, R., *Woordfrequenties in delictomschrijvingen*, March 1995, 79 pp.
- TRS-14** Gebruers, R., *Part-of-Speech Tags for Legal Documents*, June 1995, rev. October 1995, 26 pp.
- TRS-15** Gebruers, R., *A Practical Part-of-Speech Tagger for Legal Documents*, August 1995, 12 pp.
- TRS-16** Gebruers, R., *Part-of-Speech Tagging Guidelines*, December 1995, rev. January 1996, 19 pp.
- TRS-17** Gebruers, R., *SALOMON – Legal Document Profiling*, January 1996, 16 pp.
- TRS-18** Gebruers, R., *Word Tagging: Linguistic Motivation, Practical Guidelines, and Experimental Results*, April 1996, 42 pp.
- TRS-19** Gebruers, R., *Juridische databanken en taaltechnologie*, May 1996, 20 pp.
- TRS-20** Gebruers, R., *Conflating Related Terms in Legal Documents*, August 1996, 104 pp.
- TRS-21** Gebruers, R., Uyttendaele, C., & Moens, M.-F., *Definitie van de gewenste performantie (functionaliteit)*, May 1994, 26 pp.
- TRS-22** Moens, M.-F., *Juridische kennissystemen*, December 1993, 22 pp.
- TRS-23** Moens, M.-F., *Overzicht van een aantal statistische methoden gebruikt in computerlinguïstiek en Information Retrieval*, January 1994, 8 pp.
- TRS-24** Moens, M.-F., *Information Retrieval*, February 1994, 15 pp.
- TRS-25** Moens, M.-F., *Analyse van de gewenste performantie*, May 1994, 7 pp.
- TRS-26** Moens, M.-F., *Een voorstel voor een kennisrepresentatietaal voor SALOMON*, October 1994, 22 pp.
- TRS-27** Moens, M.-F., *Verslag van het onderhoud met Dr. J. R. Hobbs, Program Director, Natural Language, SRI International, Menlo Park, CA*, October 1994, 3 pp.
- TRS-28** Moens, M.-F., *Werkplan SALOMON-systeem*, January 1995, 10 pp.
- TRS-29** Moens, M.-F., *Cluster analyse en het potentieel nut van deze techniek voor SALOMON*, January 1995, 20 pp.
- TRS-30** Moens, M.-F., *Logisch ontwerp van de preprocessor, SALOMON*, June 1995, 20 pp.
- TRS-31** Moens, M.-F., *Fysiek ontwerp van de preprocessor, SALOMON*, June 1995, 19 pp.
- TRS-32** Moens, M.-F., *Werkplan SALOMON-systeem update*, July 1995, 6 pp.

- TRS-33** Moens, M.-F., *Exhaustiviteit- en precisieberekeningen van de resultaten van de preprocessor*, September 1995, 2 pp.
- TRS-34** Moens, M.-F., *Tekstvoorstelling, gewichts- en similariteitsberekeningen in tekstgebaseerde systemen met een voorstel voor SALOMON*, December 1995, 48 pp.
- TRS-35** Moens, M.-F., *Requirement Analysis and Specifications of the SALOMON Abstracting Module*, December 1995, 18 pp.
- TRS-36** Moens, M.-F., *Information Extraction in SALOMON*, December 1995, 10 pp.
- TRS-37** Moens, M.-F., *Opmerkingen over SALOMON betreffende doelstelling, methode en verder verloop*, February 1996, 9 pp.
- TRS-38** Moens, M.-F., *Information Retrieval*, part of course 'Juridische Informatica' of Prof. J. Dumortier, March 1996, 36 pp.
- TRS-39** Moens, M.-F., *Automatic Text Summarisation: an Overview*, May 1996, 32 pp.
- TRS-40** Moens, M.-F., *Logical Design Abstracting Machine SALOMON*, February 1996, rev. September 1996, 13 pp.
- TRS-41** Moens, M.-F., *Physical Design Abstracting Machine SALOMON*, February 1996, rev. September 1996, 31 pp.
- TRS-42** Moens, M.-F., *Cluster Analysis of Text Fragments: Building Clusters Based on the Selection of Representative Objects*, February 1996, rev. September 1996, 20 pp.
- TRS-43** Moens, M.-F., *Evaluation of the SALOMON Abstracting Machine: Procedures*, September 1996, 8 pp.
- TRS-44** Moens, M.-F., *SALOMON: Automatic Abstracting of Legal Cases for Effective Access to Court Decisions*, October 1996, 8 pp.
- TRS-45** Moens, M.-F., & Uyttendaele, C., *De toepasbaarheid van cluster analyse informeel getest op de vonnissen van SALOMON*, February 1995, 84 pp.
- TRS-46** Moens, M.-F., & Uyttendaele, C., *SALOMON: A System for Summarising Legal Texts*, March 1995, 34 pp.
- TRS-47** Moens, M.-F., & Uyttendaele, C., *SALOMON: Automatische informatie-extractie uit correctionele vonnissen*, August 1995, 49 pp.
- TRS-48** Uyttendaele, C., *Belgisch strafrecht, een algemeen overzicht*, March 1994, 20 pp.
- TRS-49** Uyttendaele, C., *Praktisch belang van het SALOMON-project*, April 1994, 6 pp.
- TRS-50** Uyttendaele, C., *De structuur van strafrechtelijke vonnissen*, May 1994, 15 pp.
- TRS-51** Uyttendaele, C., *Manuele kennisextractie uit vonnissen en arresten*, June 1994, 5 pp.
- TRS-52** Uyttendaele, C., *Correctionele vonnissen met een afwijkende structuur*, June 1994, 19 pp.
- TRS-53** Uyttendaele, C., *Vonnissen met een afwijkende structuur schematisch voorgesteld*, July 1994, 52 pp.
- TRS-54** Uyttendaele, C., *De tenlastelegging: onderlinge samenhang en structuur*, August 1994, 19 pp.
- TRS-55** Uyttendaele, C., *De tenlastelegging: onderdelen*, August 1994, 19 pp.
- TRS-56** Uyttendaele, C., *Tenlastelegging: output*, August 1994, 17 pp.
- TRS-57** Uyttendaele, C., *Verslag van het onderhoud met de heer Dobbelaere*, August 1994, 3 pp.
- TRS-58** Uyttendaele, C., *De relevante onderdelen van de tenlastelegging*, September 1994, 14 pp.
- TRS-59** Uyttendaele, C., *De motivering van een strafvonnis: analyse*, October 1994, 41 pp.
- TRS-60** Uyttendaele, C., *De rechtsgronden van een strafvonnis: analyse*, October 1994, 16 pp.
- TRS-61** Uyttendaele, C., *De relevante onderdelen van de tenlastelegging 2*, November 1994, 52 pp.
- TRS-62** Uyttendaele, C., *Analyse van het strafvonnis betreffende algemeen strafrecht*, November 1994, 16 pp.
- TRS-63** Uyttendaele, C., *Arresten van het Hof van Beroep: poging tot een eerste analyse*, November 1994.
- TRS-64** Uyttendaele, C., *Model Salton getoetst aan de realiteit van de vonnissen*, January 1995, 7 pp.
- TRS-65** Uyttendaele, C., *Weergave van het rechtscollege in correctionele vonnissen*, March 1995, 1 pp.

- TRS-66** Uyttendaele, C., *Nederlandstalige vonnissen van de correctionele rechtbank te Brussel: Preliminaire studie*, April 1995, 8 pp.
- TRS-67** Uyttendaele, C., *De weergave van routinemotiveringen in correctionele vonnissen*, May 1995, 3 pp.
- TRS-68** Uyttendaele, C., *Routinepatronen in de tenlastelegging*, May 1995, 6 pp.
- TRS-69** Uyttendaele, C., *Weergave van de datum in correctionele vonnissen*, June 1995, 7 pp.
- TRS-70** Uyttendaele, C., *SALOMON. Automatische extractie van relevante informatie uit correctionele vonnissen. Opmerkingen van de jurist*, June 1995, 30 pp.
- TRS-71** Uyttendaele, C., *Evaluatie van de preprocessor: werkwijze en timing*, August 1995, 5 pp.
- TRS-72** Uyttendaele, C., *Evaluatie van de preprocessor: analyse van de fouten in de kennisbank*, October 1995, 10 pp.
- TRS-73** Uyttendaele, C., *De rol van wetcitataten in de tenlastelegging*, November 1995, 7 pp.
- TRS-74** Uyttendaele, C., & Moens, M.-F., *Kwantitatieve analyse van het termgebruik in de delictomschrijving*, December 1994, 57 pp.
- TRS-75** Uyttendaele, C., & Moens, M.-F., *Analyse van de preprocessor*, June 1995, 21 pp.
- TRS-76** Uyttendaele, C., & Moens, M.-F., *Verslag van vergadering met de Document Architecture Research Group*, K.U. Leuven, April 1996, 4 pp.
- TRS-77** Uyttendaele, C., & Moens, M.-F., *Evaluation of the SALOMON Abstracting Machine: Results*, October 1996, 13 pp.

7. Lectures

- LRS-1** Gebruers, R., *Rechtstaal: voer voor de computer?* Lecture delivered at the Faculty of Law, K.U. Leuven, May 1994.
- LRS-2** Gebruers, R., *Taalcomputer-kunde alias computer-taalkunde: Een overzicht.* Lecture delivered at the Interdisciplinary Centre for Law and Information Technology, K.U. Leuven, November 1994.
- LRS-3** Gebruers, R., *SALOMON: Naar automatische categorisering en excerptering van rechtsdocumenten.* Lecture delivered at the Centre for Computational Linguistics, K.U. Leuven, March 1995.
- LRS-4** Gebruers, R., *Het SALOMON-PROJECT.* Lecture delivered at the Faculty of Law, K.U. Leuven, May 1995.
- LRS-5** Gebruers, R., *SALOMON: An Exercise in Legal Information Extraction. Linguistic Aspects.* Lecture delivered at the Faculty of Law, K.U. Leuven, December 1995.
- LRS-6** Gebruers, R., *SALOMON – Legal Document Profiling.* Lecture delivered at the Faculty of Psychology, K.U. Leuven, January 1996.
- LRS-7** Gebruers, R., *Juridische databanken en taaltechnologie.* Lecture delivered at the Faculty of Law, K.U. Leuven, May 1996.
- LRS-8** Moens, M.-F., *Methoden voor informatie-extractie uit full text toegepast in het juridisch domein.* Lecture delivered at the JURIX-bijeenkomst, Faculty of Law, K.U. Leuven, May 1995.
- LRS-9** Moens, M.-F., *SALOMON: An Exercise in Legal Information Extraction. Information Technology Aspects.* Lecture delivered at the Workshop *Information Technology and Law*, Faculty of Law, K.U. Leuven, December 1995.
- LRS-10** Moens, M.-F., *Information Retrieval.* Lecture delivered at the Faculty of Law, K.U. Leuven, March 1996.
- LRS-11** Moens, M.-F., *Informatie-ontsluiting en automatisch samenvatten van teksten.* Lecture delivered at the Centrum voor Recht, Bestuur en Informatisering, Katholieke Universiteit Brabant, Tilburg, Nederland, September 1996.
- LRS-12** Moens, M.-F., *SALOMON: Automatic Abstracting of Legal Cases for Effective Access to Court Decisions.* Lecture to be delivered at *JURIX '96 Ninth International Conference on Legal Knowledge-based Systems*, Katholieke Universiteit Brabant, Tilburg, Nederland, December 1996.
- LRS-13** Uyttendaele, C., *SALOMON. Automatische extractie van relevante informatie uit correctionele vonnissen. Opmerkingen van de jurist.* Lecture delivered at the JURIX-bijeenkomst, Faculty of Law, K.U. Leuven, May 1995.
- LRS-14** Uyttendaele, C., *SALOMON: An Exercise in Legal Information Extraction. Legal Aspects.* Lecture delivered at the Workshop *Information Technology and Law*, Faculty of Law, K.U. Leuven, December 1995.

8. Publications

Uyttendaele, C., Moens, M.-F., & Dumortier, J. (1996). SALOMON : Abstracting of Legal Cases for Effective Access to Court Decisions. In *Proceedings of JURIX '96 Ninth International Conference on Legal Knowledge-based Systems* (pp.47-58). Tilburg University Press, Tilburg.

Moens, M.-F., Uyttendaele, C., & Dumortier, J. (1996a). Automatic Text Structuring and Categorization as a First Step in Summarizing Legal Cases (submitted).

Moens, M.-F., Uyttendaele, C., & Dumortier, J. (1996b). Automatic Text Structuring and Abstracting: Building Clusters Based on the Selection of Representative Objects (submitted).