# Abstracting of Legal Cases: The SALOMON Experience

Marie-Francine Moens, Caroline Uyttendaele, Jos Dumortier
Interdisciplinary Centre for Law and IT (ICRI)
K.U. Leuven
Tiensestraat 41
B-3000 Leuven
Belgium
{marie-france.moens, caroline.uyttendaele, jos.dumortier}@law.kuleuven.ac.be

**Abstract**

The SALOMON project automatically summarises Belgian criminal cases in order to improve access to the large number of existing and future court decisions. SALOMON extracts relevant text units from the case text to form a case summary. Such a case profile facilitates the rapid determination of the relevance of the case or may be employed in text search. Techniques are developed for identifying and extracting relevant information from the cases. A broader application of these techniques could considerably simplify the work of the legal profession.

A double methodology was used when developing SALOMON. First, the case category, the case structure and irrelevant text units are identified based on a knowledge base represented as a text grammar. Consequently, general data and legal foundations concerning the essence of the case are extracted. Secondly, SALOMON extracts informative text units of the alleged offences and of the opinion of the court based on shallow statistical techniques. The application of cluster algorithms based on the selection of representative objects has a potential for automatic theme recognition, text abstracting and text linking, even beyond the legal field.

Evaluation of the results demonstrates that the SALOMON techniques do not themselves solve any legal questions, but they do guide the user effectively towards relevant texts.

## 1 Introduction

Computers become prominent in law courts and offices of public prosecutors. As a result a huge amount of electronic texts is available. There is an urgent need for intelligent tools that make the information in legal texts manageable (Susskind, 1996, p. 107 ff.).

The SALOMON project (Uyttendaele, Moens, & Dumortier, 1996) developed and tested several techniques to make a vast corpus of Belgian criminal cases (written in Dutch) easily accessible. SALOMON automatically extracts relevant information from the full text of a case, and uses it to compose a summary of each decision. Each criminal case is represented by a case profile ('index card'), which facilitates the rapid determination of the relevance of the case. The user is informed of the name of the court that issued the decision, the decision date, the offences charged, the relevant statutory provisions disclosed by the court, and the important legal principles applied. Moreover, the summary can act as a case surrogate in text search. Additionally, SALOMON wants to contribute to the study of more general methods for text classification, extraction and summarisation.

To realise these goals, a demonstrator was built in the programming language C on a Sun™ SPARC station 5 under Solaris® 2.3.

## 2 Background

A major part of the SALOMON research concerns automatic abstracting of text. Document abstracts generated automatically generally belong to two types (Paice, 1990; Sparck Jones, 1993). Firstly, 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*). 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 is related to *indexing*. A brief summary may serve as a complex structured index description. At present the majority of abstracts automatically generated are document extracts.

The automatic generation of document abstracts has early been recognised as a potential area for automation (Luhn, 1958). At that time automatic *text* abstracting and *indexing* were strongly related. Attempts have been made to extract words, phrases, or sentences that reflect the content of the text. Index terms are weighted depending upon the occurrence in titles and headings (Salton, 1989, p. 439) or upon occurrence frequencies in the text and/or text corpus (Salton & Buckley, 1988). Sentence scores are based on the number of significant and non significant words in it (Luhn, 1958), on location heuristics (Baxendale, 1958), or on the occurrence of positive or negative indicator phrases (Edmundson, 1969), or are computed as the sum of term weights after eliminating stop words (Earl, 1970). Sentences, the score of which surpasses a certain threshold value, are retained for summary purposes.

Until recently, automatic abstracting has never received special attention, apart from the application of artificial intelligence techniques in restricted text domains, which build on the accomplishments of *text extraction*, a subfield of natural language processing. Text extraction aims at extracting a narrowly defined class of facts and relies on representations of the text corpus that reflect predictable patterns of linguistic context. Successful applications perform a detailed semantic analysis of the source text based upon a semantic representation of the text type under consideration (e.g. an overview in Jacobs, 1992). Here, recognition of themes and representative text units usually relies on linguistic and domain knowledge, or solely on linguistic knowledge (Miike, Itoh, Ono, & Sumita, 1994).

With the current information overload, automation of text summarisation receives renewed interest. An example of the automatic generation of case summaries in the *legal field* and their use for information retrieval is FLEXICON (Fast Legal EXpert Information CONsultant) (Gelbart & Smith, 1995). FLEXICON extracts relevant text units based on location heuristics, occurrence frequencies of index terms, and the use of indicator phrases.

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. Secondly, the progress made in *information retrieval*, especially the current refinement and sophistication of *statistical techniques* developed for the identification of index terms and text structure, may be fertile for automatic abstracting of texts of unrestricted domains. Parallel, in the information retrieval field there is a growing interest in complex indices for document access. It is along the two directions, suggested by Sparck Jones that we developed the SALOMON project.

## 3 Methods

It is useful to consider the intellectual process of abstracting, not only for defining the desired output, but also for finding appropriate techniques, which may be automated. 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) 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.

Some of the *recommendations for intellectual abstracting* (Kintsch & van Dijk, 1978; Pinto Molina, 1995) have a *potential in automatic abstracting*. These recommendations concern the recognition of fundamental characteristics of the document as form, class, and structure of the information, deletion of insignificant and redundant information, and selection of thematically important sentences.

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 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. Sparck Jones (1993) distinguishes two strategies in automatic abstracting. A first strategy relies on the surface structure of the text and is called *shallow processing* of the text. Although, in this strategy text processing relies on some heuristics, the knowledge about the text is very restricted. A *deeper processing* employs additional knowledge to interpret the surface features found in the text.

Both strategies are applied in SALOMON. This was necessary because part of the text to be summarised is predictable (logical structure and category of the case, irrelevant paragraphs of the offences and the opinion of the court, irrelevant legal foundations), while other text parts treat unrestricted subject matter (delict descriptions of the alleged offences and the argumentation in the opinion of the court). The former is processed based upon a text grammar, the latter is summarised based upon shallow statistical techniques.

SALOMON employs *deep text processing* to automatically categorise the cases and case segments. Additionally, irrelevant text segments are identified. The knowledge involved in the processing regards the structure of the criminal cases and the text cues that identify the case and its segments. The knowledge is implemented as a text grammar. The use of 'superstructural' schemes or grammars is promising for elucidating the information structure of certain text types (Paice, 1990). The idea is to anticipate structural schemes that are common to all texts of a specific text type. A text is usually composed of different blocks or segments, which fulfil a semantic role in the text and which are combined according to specific semantic relations. The segments may be classified and/or delimited by linguistic and domain clues, which are whitespace characters or punctuation marks, and/or word patterns. For instance in a criminal case the text segment of the legal foundations follows the segment that bears on the opinion of the court and the first paragraph of the legal foundations might be introduced with the word pattern 'On these grounds'.

We designed a domain-independent formalism, which allows to represent the semantic units of a text, their attributes, and relations in the form of a text specific grammar. The formalism represents the text grammar as a semantic network of frames. *Frames* are well suited to represent document structure in general. The nodes of the network represent the objects with their attributes, the lines the relations between the objects. 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, sequential,* or *conditional relation* between them. Segments are paragraphs, sentences, or more informal text blocks of varying length. Word patterns that delimit or classify segments are grouped according to their semantic role and represented by a one level hierarchy of frames. Word pattern frames are connected with the appropriate segment frame(s) (*limiting or classifying relation*).

A parser was implemented to process the case based upon the text grammar. The parser can be considered as a '*partial parser*', which targets specific information in the text, while skipping over other text parts (cf. McDonald, 1992).

SALOMON employs *shallow techniques* to eliminate redundant information in the alleged offences, 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 the linguistic context of the information is not predictable. Crimes concern every aspect of society and crime descriptions are infiltrated with unpredictable facts.

Here, SALOMON builds upon current research in information retrieval. Full text retrieval of long texts may benefit by structuring the text according to topics and subtopics. The research of Salton and his co-researchers (Salton, Allan, Buckley, & Singhal, 1994; Salton, Allan, & Singhal, 1996) focuses on *finding subparts of a large document that are very similar in context*. Small text units (e.g. sentences, paragraphs) are represented as vectors of weighted index terms[1] and similarities between text vectors are computed.[2] Text units are thematically grouped, when the similarity between them exceeds a threshold value. Hearst and Plaunt (1993) used patterns of lexical connectivity between text units to identify the subtopics of a text. Here, only similarity values between adjacent text units are computed and placed in a graph. Ruptures in the topic structure of the text are identified as valleys in the graph. This approach automatically structures linear text, and thus allows a user to efficiently query portions of the text or to identify relevant text excerpts.

In SALOMON each paragraph of the alleged offences and opinion of the court is represented as a vector of weighted index terms (single words). Index terms are selected after elimination of stopwords[3] and proper names[4], and are currently not stemmed. The index terms of the alleged offences are weighted with the in-paragraph frequency.[5] Considering the stereotypical way in describing the crimes committed, also less important content words contribute in identifying redundancy. Discriminating the index terms of the opinion of the court is done with inverse document frequency weights.[6] Here numbers are not considered as index terms. Paragraphs are compared with the cosine coefficient[7]. Nonhierarchical clustering methods based on the selection of representative objects are employed to thematically group the paragraphs of the alleged offences and the opinion of the court.[8]

*Cluster analysis* is a multivariate statistical technique that automatically generates groups of similar objects. *Clustering* methods *based on the selection of representative objects* consider possible choices of representative objects (also called *centrotypes* or *medoids*) and then construct clusters around them.

We implemented the *covering clustering algorithm* (Kaufman & Rousseeuw, 1990, p. 111) to eliminate redundant delict description paragraphs disturbed by different facts in the alleged offences. In this algorithm possible representative paragraphs (medoids) are considered for a potential grouping, but each paragraph must at least have a given similarity (threshold) with the representative paragraph of its cluster. The objective is to minimise the number of representative paragraphs. The threshold value is useful to define the degree of redundancy allowed and was set after several trials. We added an extra constraint: for a given number of representative paragraphs a best solution is found for which the total (or average) similarity between each paragraph and its representative paragraph is maximised.

The *k-medoid clustering method* (Kaufman & Rousseeuw, 1990, p. 68 ff.) was implemented to group the paragraphs of the opinion of the court according to the topic treated. The k-medoid method searches the best possible clustering in *k* groups of a set of objects. A set of paragraphs is automatically divided in *k* groups. 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. Additionally, the best number of clusters (*k*) may be determined as part of the clustering method. Kaufman and Rousseeuw define for each object of the cluster structure a '*silhouette width*' that measures the degree of fitness of an object to its cluster. The average silhouette width taken over all objects of the clustering is a parameter for the goodness of the clustering. We implemented a variant hereof, applied for similarity measures. In this way we compute a natural clustering for which the similarity between an object and its representative object is maximised and the similarity with its second choice cluster is minimised.

An optimal solution is computationally possible for relatively small problems. For a large number of objects we developed a *good, but not optimal solution* for the k-medoid method.[9] The algorithm is a reallocation algorithm: an initial clustering is improved in consequent steps until a specific criterion is met. For our legal texts there was no need to implement a good solution for the covering algorithm.

The *medoid of the cluster* is the object of the cluster that has a maximum average similarity with all other objects of the cluster. It forms a representative description of each crime or topic treated in respectively the alleged offences or opinion of the court. We assume that text units that are closely linked by patterns of content words to a number of other text units are informative (cf. Prikhod'ko & Skorokhod'ko, 1982) and thus relevant to include in the case summary. Additionally, each cluster of opinion of the court paragraphs containing more than three objects is represented by its most important keywords. Different methods (Jardine & van Rijsbergen, 1971; Willett, 1980) are possible for keyword selection. Currently, we select the highest weighted terms of the average vector of the cluster.

At present we limit ourselves to the extraction of information from the case text. No attempt is made to re-edit this information.

## 4 Results

[1] An overview of index term weighting functions is given by Salton and Buckley (1988).

[2] An overview of similarity functions used in text-based systems is given by Jones and Furnas (1987).

[3] Stopwords are identified as the most frequent words in the corpus of legal cases.

[4] Proper names are recognized with heuristic rules.

[5] In-paragraph frequency computed as the number of times an index term *tj* occurs in the text paragraph.

[6] Inverse document frequency (*idf*) of index term *tj* computed as: *log (N/dfj)*

with *N* = number of documents in the collection
*dfj* = number of documents in the collection which contain index term *tj*.

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

[7] $$\frac{\sum_{i=1}^{n} WO1_i * WO2_i}{\sqrt{\sum_{i=1}^{n} WO1_i^2} * \sqrt{\sum_{i=1}^{n} WO2_i^2}}:$$

the similarity between two text paragraphs is calculated as the cosine coefficient of their vector representations *WO1* and *WO2*.

[8] A detailed description of the algorithms and their benefits is described in Moens (1996).

[9] For reasons of efficiency we employed a good solution for 16 and more paragraphs to be clustered.

### 4.1 Corpus analysis

The choice for a corpus of criminal decisions as a research object, was no coincidence. Criminal cases were available in machine readable format at the start of the project. Moreover, criminal law is clearly structured and the decisions have a fixed, recurring composition. For the SALOMON project, a corpus was used consisting of the decisions that the correctional Court of Leuven pronounced between January 1992 and June 1994.

The cases can be classified into 7 main categories, distinguishing general decisions from particular ones. The latter concern appeal procedures, civil interests, refusals to witness, false translations by interpreters, infringements by foreigners or the internment of people.

All criminal cases have a typical structure. They are made up of 9 elements, some of which are optional:

- superscription with name of the court and date;
- identification of the victim;
- identification of the accused;
- alleged offences;
- transition formulation;
- opinion of the court;
- legal foundations;
- verdict;
- conclusion.

The SALOMON techniques were developed in order to extract and summarise the most relevant case components: alleged offences, opinion of the court and legal foundations. The texts of the offences charged and the opinion of the court are often long and elaborated. They are characterised by especially long sentences of an average length of 3 to 5 text lines. Because the sentences are separated by a new line character, we call them paragraphs.

The *alleged offences* contain a description of the crimes committed, as well as routine paragraphs. A delict description is disclosed in a separate text paragraph. Such a description contains the specific facts of the delict, integrated in the text of the description. The offences may contain several delict descriptions: some of them may be identical, but referring to different facts or different accused. Delict concepts are usually described in a fixed, stereotypical way.

The *opinion of the court* contains the argumentation of the judge regarding the crimes committed. It 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 statute application and interpretation. The leading cases only represent 3 to 5% of the total corpus. The opinion of the court often discusses different crimes. A theme may be abandoned and resumed during the discourse.

Finally, the *legal foundations* consist of a complete enumeration of statutory texts and articles applied by the court. Several of them (*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.

### 4.2 Architecture of the demonstrator (Figure 1)

A demonstrator was built for assessing the value of the methods employed. The result of the initial categorisation and text structuring is a case tagged in SGML-syntax (Figure 2).[10] A head tag marks the general category of the case. The identified text segments are marked with the appropriate category tags. From the tagged case general information about the case (date, name of the court and relevant legal foundations) is easily extracted and placed on the index card. The relevant parts of the alleged offences and opinion of the court are further processed. Key paragraphs and terms are extracted using the

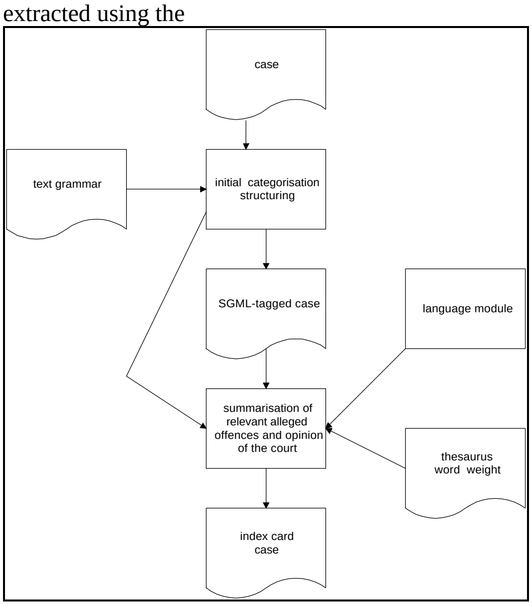


Figure 1: Architecture of the demonstrator

```
<appeal_procedure>
<superscription> Court Administration number: ...
<court> Correctional Court Leuven </court> ...
<date> January 20, 1993 </date> ...
In the case of the Public Prosecutor and of:
</superscription>
<victim> ...
</victim>
<accused>  Against ...
Defendant in opposition ...
</accused>
<alleged_offences>
<routine_alleged_offences> ... Accused: ...
</routine_alleged_offences>
...
</alleged_offences>
<transition_formulation> Given the documents in the case ...
Given the Public Prosecutor’s case for the prosecution
</transition_formulation>
<opinion_of_the_court> Whereas ...
<routine_opinion> ... offence ... is certain...
</routine_opinion>
...
<routine_opinion>  Given the enactment...
</routine_opinion>
...
</opinion_of_the_court>
<legal_foundations> On these grounds and in application of
the following statutory provisions  ...
<routine_foundations> ... Code of criminal procedure...
```

[10] In the future courts may tag case category and logical structure during text generation.

```
</routine_foundations>
</legal_foundations>
<verdict>  THE COURT ...
</verdict>
<conclusion> Thus given  ...
</conclusion>
</appeal_procedure>
```

Figure 2: Example of a SGML-tagged case: the word patterns in italic classify or delimit the case or its segments.

above clustering methods. The index terms, needed for the vector representation, are selected with the help of a thesaurus with index term weights and/or with the help of a language processing module.[11]

### 4.3 Results of the output of the demonstrator

A test set of criminal cases was carefully chosen in a way that it is representative for the complete corpus. The output of the initial categorisation and structuring and of the subsequent abstracting of the alleged offences and opinion of the court, was evaluated in terms of effectiveness metrics.

Evaluation of text abstracts is a difficult task. An intuitive approach is to compare the abstract automatically generated with the abstract intellectually produced by an expert. Our expert was not a member of the research team, but an outsider, namely a student entering her final year of law school.

There is a real need for appropriate evaluation procedures for text abstraction. Developing evaluation procedures was not the aim of our project. Since our abstracting procedures are related to text categorisation and text extraction, we borrowed metrics that have been successfully applied in these fields.

| | recall | precision |
|---|---|---|
| case category | 0.95 | 0.99 |

Table 1: Average results of the initial case categorisation

| | general cases | | special cases | |
|---|---|---|---|---|
| | recall | precision | recall | precision |
| segment category | 0.88 | 0.93 | 0.66 | 0.88 |

Table 2: Average results of the initial structuring of the cases

The expert intellectually categorised the test criminal cases and their segments. She also intellectually marked paragraphs of the offences charged and of the opinion of the court relevant for inclusion in the case summary. The results were compared with the output of SALOMON. We realised that the identification of paragraphs in the opinion of the court that reflect the topics of the argumentation of the judge is sometimes a subjective operation and is ideally repeated by different experts. Due to a limited timing and tight financial circumstances, it has until now been impossible to have the evaluation of the paragraphs of the opinion of the court repeated by other experts.

The *categorisation of the case and case segments* is evaluated based on definitions of recall and precision in the field of *text categorisation* (Jacobs, 1993; Lewis, 1995). *Recall* and *precision* are computed respectively as the proportion of correct assignments to the category upon the real existing number of this category, and as the proportion of correct assignments to this category upon the number of assignments to this category. Recall measures completeness and precision measures accuracy of the category assignments. The metrics are computed for 1000 cases. For the case category an average recall and precision of respectively 95% and 99% are achieved (Table 1). For the case segments an average recall and precision of respectively 88% and 93% for general decisions and respectively 66% and 88% for special decisions is obtained (Table 2). Recall errors are usually the result of lack of knowledge, whereas precision errors may be due to ambiguities in the knowledge. A substantial number of errors are caused by typing errors.

For *evaluating the informativeness of the extracted paragraphs* of alleged offences and opinion of the court we compared representative paragraphs intellectually identified with paragraphs automatically generated. We used the metrics recall, precision, fallout and overgeneration as applied in the field of *text extraction* (Chinchor, 1992; for a detailed discussion of the usefulness of these metrics in evaluating text extraction systems see Chinchor, Hirschman, & Lewis, 1993). The comparison of paragraphs automatically generated with the ones intellectually attributed relies on *scoring categories*. Scoring categories are determined based upon the comparison of a response (here paragraph) automatically attributed with an expected response (here paragraph) intellectually attributed. We compute the number of:

1. correct responses (in the metrics further called 'correct'): a correct response is a response for which a corresponding paragraph that is intellectually attributed is found;

2. partial correct responses ('partial'): a partial correct response is a response for which a partial match with a paragraph that is intellectually attributed is found; we defined a partial correct response as a paragraph that represents the crime (in the alleged offences) or topic (in the opinion of the court) but that is not the ideal paragraph to include in the case summary;

3. incorrect responses ('incorrect'): an incorrect response is a response, for which a paragraph that is intellectually attributed is available, but no correspondence can be found;

4. spurious responses ('spurious'): a spurious response is a superfluous response with no key in the set of paragraphs that are intellectually attributed; a spurious response is a superfluous response that is automatically generated;

5. missing responses ('missing'): a missing response is a key in the set of paragraphs that are intellectually attributed, which has no response in the set of paragraphs that are automatically attributed.

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

The scoring categories allow to compute following metrics, which measure different aspects of performance:

recall= (correct + (partial x 0.5)) / possible

[11] The language module, which allows to stem words and to identify part-of-speech categories (e.g. verbs and nouns) is currently not yet linked to the system.

precision= (correct + (partial x 0.5)) / actual

overgeneration= spurious / actual

fallout= (incorrect + spurious) / possible incorrect

'Possible' is the sum of 'correct', 'partial', 'incorrect' and 'missing'. 'Actual' is the sum of 'correct', 'partial', 'incorrect', and 'spurious'. 'Possible incorrect' is the number of candidate representative paragraphs (all paragraphs) minus the number of representative paragraphs intellectually attributed.

*Recall* is computed as the proportion of correct responses upon the number of responses intellectually attributed. Recall is the degree of completeness of the paragraphs automatically generated. *Precision* is computed as the proportion of correct responses upon the number of responses automatically generated. Precision is the degree of accuracy of the paragraphs automatically generated. A partial correct response, the weight of which is fixed at 0.5, represents the concept, intellectually formulated, but is not the ideal representative. *Overgeneration* measures the percentage of spurious responses upon the number of responses automatically generated. *Fallout* computes the proportion of faulty responses (incorrect and spurious) given the number of incorrect responses that the system could generate.

The metrics are calculated for each text of the offences and the opinion of the court of 700 criminal case of the test set and are averaged.

A '*methodological' evaluation* (Table 3) aims at evaluating extracted paragraphs in representing the topics of the abstracted text. The high recall (about 97% and 85% for respectively offences and opinion) and precision (about 95% and 81% for respectively offences and opinion) values indicate that the techniques employed are suitable in recognising the theme structure of our texts and in identifying representative text units. The main output errors are due to morphological variants of related and identical concepts and incorrect orthographic boundaries. The better results of structuring the alleged offences are explained by the standardised naming and description of legal concepts in the offences, making a thematic grouping and recognition of redundant material very effective. Overgeneration of responses is low, indicating that the system almost correctly identifies the number of themes in the text.

| | recall | precision | overgeneration | fallout |
|---|---|---|---|---|
| Alleged offences | 0.97 | 0.95 | 0.04 | 0.28 |
| Opinion of the court | 0.85 | 0.81 | 0.09 | 0.24 |

Table 3: Average results of a 'methodological' evaluation of the abstracting of alleged offences and opinion of the court

| | recall | precision | overgeneration | fallout |
|---|---|---|---|---|
| Alleged offences | 0.82 | 0.82 | 0.12 | 0.09 |
| Opinion of the court | 0.75 | 0.33 | 0.55 | 0.21 |

Table 4: Average results of a 'legal' evaluation of the abstracting of alleged offences and opinion of the court

A '*legal' evaluation* (Table 4) aims at detecting the limits of our approach. Here relevancy relates to the identification of distinct delict descriptions (offences) and to the value in indicating legal principles (in the opinion). This evaluation takes into account all paragraphs of the alleged offences (routine, non routine, factual, and (redundant) delict description paragraphs) and 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. It also evaluates 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 influence the results. 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 the errors of the initial structuring phase influence the results. The system finds an important part of the legally relevant principle paragraphs that were intellectually attributed (almost 75%), but generates too many paragraphs (overgeneration of more than 50%). Such a large overgeneration necessarily decreases precision: precision is computed as the proportion of correct answers in all the answers generated. The overgeneration concerns some routine grounds and many factual considerations.

We plan to evaluate different methods for attributing *key terms* to elaborate texts of the opinion of the court based on the above metrics.

SUMMARY OF CRIMINAL CASE

NAME OF CASE = /users/sien/testset/verli

DATE = September 16, 1992.

COURT = CORRECTIONAL COURT LEUVEN

REPRESENTATIVE PARAGRAPHS OF THE OFFENCES=

by use of violence or threat, to have destroyed or damaged others movable property, namely doors, bottles, glasses, chairs, tables, crates of beer and coke belonging to ... and ... .

under the circumstances that the facts were committed in association or in gang, and that ... was the leader or the fomenter of the gang.

To have committed assault and battery to ..., causing illness or inaptitude for the accomplishment of personal work;

To have committed assault and battery to ...

By way of gestures or symbols to have threaten ... with offences against his person or property, punishable with an imprisonment imposed by a Crown Court.

REPRESENTATIVE PARAGRAPHS OF THE OPINION OF THE COURT=

Whereas ... claims, without foundation, that he may be responsible for what happened during the so-called first brawl, close to the disco, but that he was not involved with the incident that occurred a little later close to the bar; whereas this clearly was a continued group incident, with the first accused acting as the most violent person, to the extent that he should be considered at that moment as the main fomenter;

REPRESENTATIVE KEY TERMS OF THE OPINION OF THE COURT=

group incident brawl

REPRESENTATIVE GROUNDS =

ON THESE GROUNDS and implementing the articles 1382 of the Civil Code; 38-40-44-50-65-66-528-529-79-80-84-327-329-392-398/1-399/1 of the Criminal Code;

Figure 3: Example of a case profile

## 5 Discussion

SALOMON yields relevant extracts of the case that are indicative and informative about the content of the case. In a first step it employs knowledge-based techniques to identify the category and logical structure of the case. It has been shown that a knowledge representation as a *text grammar,* possibly employable during text drafting, is useful. In a second step shallow (statistical) techniques are employed to recognise the topic structure of the alleged offences and opinion of the court and to extract relevant text units from them. *Cluster algorithms based on the selection of representative objects* provide the possibility to identify informative text units that through their lexical patterns are linked to other text units. As a result, redundant information is deleted from the delict descriptions and thematically coherent text pieces of the argumentation of the judge are identified.

According to Sparck Jones (1993) progress in automatic abstracting might be realised by considering the structure of the text type of the documents in the corpus and by experimenting with indexing techniques currently being developed in the field of information retrieval. We successfully implemented both strategies.

Text structure is especially prominent in Belgian legal cases. 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, which may be more advantageous than the use of thesauri of indicator phrases as in FLEXICON (Gelbart & Smith, 1995). Moreover, we designed a domain-independent formalism that allows, when needed, to focus on other text attributes or to represent other text types. An easy portability of the techniques to other domains is essential when text-based systems employ knowledge about the texts (Chinchor et al. 1993).

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. However, the use of cluster algorithms based on the selection of representative objects is new in this context. These algorithms have advantages when automatically abstracting. They allow for producing a natural clustering. This was important 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. These algorithms also provide the possibility to identify informative text units relevant for abstracting purposes. So, we proposed a new technique that may complement other existing techniques for identification of significant text units such as location heuristics, frequency occurrences of index terms, and the use of indicator phrases employed for instance in the FLEXICON system (Gelbart & Smith, 1995).

As a result, part of the *intellectual abstracting* practice, involving the identification of the case type, the structure of the
information in the case, deletion of redundant and insignificant information and selection of thematically relevant text units and key terms, is automatically *simulated*. In this way we obtain a summary of the case, which is about 20 % of the full text of the case (e.g. Figure 3). However, part of the intellectual process, which relates to the subjectivity of the law, seems presently out of reach. Abstracting is always a reduction of the text content to its summary. This reduction may involve interpretation of the original text. As it is illustrated in the following paragraphs, such an interpretation is not always wanted by the user of the abstract or is not always possible with the current techniques.

The knowledge employed in the text grammar always reflects a certain interpretation, since it is structured in a user-specific way, according to the problem-solving tasks to be supported. This is called the interaction problem (Visser, 1996; van Kralingen, 1996, pp. 6-7). Recognition of the category and the segments of the case is straightforward and poses no problems. However, recognition of routine, irrelevant paragraphs in the case text sometimes entails a subjective element. When building the text grammar we avoided to implement knowledge that would lead to a subjective interpretation of the text segments.

The shallow techniques employed for theme recognition in the alleged offences and opinion of the court do not allow to discriminate principle from factual grounds. The identification of principle grounds not only involves interpretation, but also implicates 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.

SALOMON does not intend to perform the interpretation of cases or legislation itself, but only to assist the user in his own interpretation process (Moles & Dayal, 1992; Zeleznikow & Hunter, 1992, 1994, p. 73). It is one of the tools at the disposal of the ignorant lawyer for retrieving possibly relevant documents (Bing, 1995).

For retrieval and filtering purposes the case profile preferably contains *crime category descriptors*. In a limited experiment we demonstrated the usefulness of machine learning techniques that learn text categories based on example texts. We compared case delict descriptions with example texts with the technique of a *nearest neighbour search* and adopted the categories of the nearest texts (cf. Masand, Linoff, & Waltz, 1992). The 'list(book) of qualifications', which is employed by the public prosecutor and which is a collection of categorised offences grouped by crime concept in a hierarchical way, served as training set. In this way the crime concepts of a criminal case can be learned. Conceptual crime descriptors facilitate the retrieval of the cases (cf. De Mulder, Wildemast, & van den Hoven, 1993).

## 6 Conclusions

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, our research contributes in finding techniques for automatic text structure recognition and theme identification, which may be applied in automatic indexing, abstracting and automatic text linking also beyond the legal field.

A system like SALOMON can simplify the lawyer's job a great deal. It does not provide the user with ready-made answers to complicated legal cases. But, it directs him towards documents where the answer must be found. SALOMON is a tool telling the lawyer about the crimes, the law and the topics of the argumentation of the judge in a certain case.

**Acknowledgements**

We thank Tine Bouwen for the verification of the results. We are grateful to Prof. Dr. L. Verstraelen, Dr. J. Leysen and Prof. Dr. J. Zeleznikow for helpful comments.

**References**

Baxendale, P. B. (1958). Machine-made Index for Technical Literature - an Experiment. *IBM Journal of Research and Development*, 2 (4), pp. 354-361.

Bing, J. (1995). Legal Text Retrieval and Information Services. In J. Bing & O. Torvund, *25 Years Anniversary Anthology in Computers and Law* (pp. 525-585). TANO, Oslo.

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.

Chinchor, N., Hirschman, L., & Lewis, D. D. (1993). Evaluating Message Understanding Systems: An Analysis of the Third Message Understanding Conference (MUC-3). *Computational Linguistics*, 19 (3), pp. 409-449.

De Mulder, R. V., Wildemast, C., & van den Hoven, J. (1993). Conceptuele geautomatiseerde juridische documentatie-systemen. *Computerrecht*, 1993 (2), pp. 69-77.

Earl, L. L. (1970). Experiments in Automatic Extracting and Indexing. *Information Storage and Retrieval*, 6, pp. 313-334.

Edmundson, H. P. (1969). New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16 (2), pp. 264-285.

Gelbart, D., & Smith, J. C. (1995). FLEXICON: An Evaluation of a Statistical Ranking Model Adapted to Intelligent Legal Text Management. In *The Fourth International Conference on Artificial Intelligence and Law: Proceedings of the Conference* (pp. 142-149). ACM, New York.

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.

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

Jacobs, P. S. (1993). Using Statistical Methods to Improve Knowledge-based News Categorisation. *IEEE Expert*, 8 (2), pp. 13-23.

Jardine, N., & van Rijsbergen, C. J. (1971). The Use of Hierarchic Clustering in Information Retrieval. *Information Storage and Retrieval*, 7, pp. 217-240.

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.

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. A. (1978). Toward a Model of Text Comprehension and Production. *Psychological Review*, 85 (5), pp. 363-394.

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.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2, pp. 159-165.

Masand, B., Linoff, G., & Waltz, D. (1992). Classifying News Stories Using Memory Based Reasoning. In *Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval* (pp. 59-65). ACM, New York.

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.

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.

Moles, R. N., & Dayal, S. (1992). There is More to Life than Logic. *Journal of Law and Information Science*, 3 (2), pp. 188-218.

Moens, M.-F. (1996). *Cluster Analysis of Text Fragments: Building Clusters Based on the Selection of Representative Objects*. Technical Report SALOMON No. 42, ICRI, K.U. Leuven.

Paice, C. D. (1990). Constructing Literature Abstracts by Computer: Techniques and Prospects. *Information Processing & Management*, 26 (1), pp. 171-186.

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.

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., 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.

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).

Susskind, R. (1996). *The Future of Law: Facing the Challenges of Information Technology*. Clarendon Press, Oxford.

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.

Van Kralingen, R. W. (1996). *Frame-based Conceptual Models of Statute Law*. Kluwer Law International, 's-Gravenhage.

Visser, P. R. S. (1996). *Knowledge Specification for Multiple Legal Tasks.* Kluwer Law International, 's-Gravenhage.

Willett, P. (1980). Document Clustering Using an Inverted File Approach. *Journal of Information Science*, 2, pp. 223-231.

Zeleznikow, J., & Hunter, D. (1992). Rationales for the Continued Development of Legal Expert Systems. *Journal of Law and Information Science,* 3 (1), pp. 94-110.

Zeleznikow, J., & Hunter, D. (1994). *Building Intelligent Legal Information Systems*. Kluwer Law and Taxation Publishers, Deventer - Boston.