Pergamon

Information Processing & Management, Vol. 33, No. 6, pp. 727–737, 1997
© 1997 Elsevier Science Ltd
All rights reserved. Printed in Great Britain
0306-4573/97 $17+0.00

PII: S0306-4573(97)00035-6

AUTOMATIC TEXT STRUCTURING AND CATEGORIZATION AS A FIRST STEP IN SUMMARIZING LEGAL CASES

MARIE-FRANCINE MOENS and CAROLINE UYTTENDAELE
Katholieke Universiteit Leuven, Leuven, Belgium

(Received 30 April 1996; accepted 6 May 1997)

Abstract—The SALOMON system automatically summarizes 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. In a first important abstracting step SALOMON performs an initial categorization of legal criminal cases and structures the case text into separate legally relevant and irrelevant components. A text grammar represented as a semantic network is used to automatically determine the category of the case and its components. In this way, we are able to extract from the case general data and to identify text portions relevant for further abstracting. It is argued that prior knowledge of the text structure and its indicative cues may support automatic abstracting. A text grammar is a promising form for representing the knowledge involved. © 1997 Elsevier Science Ltd

1. INTRODUCTION

In the legal field there is an urgent need for intelligent tools that make the information in legal texts manageable. The SALOMON (Summary and Analysis of Legal texts FOR Managing Online Needs) project 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. Its 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. In this way, the totality of criminal jurisprudence is comparable on a national level, increasing the value of the information in the cases (Uyttendaele *et al.*, 1996, 1997).

In a first step the case category, the major semantic case components, some general data (e.g. date, court name, relevant legal foundations) and non relevant paragraphs of the text are identified using a text grammar approach. In a second step relevant paragraphs of the text of the offences charged and of the opinion of the court are further abstracted. Because their content is unpredictable and relates to a broad subject domain, the theme structure, key paragraphs, and key terms of these texts are identified with shallow statistical techniques (Moens *et al.*, 1997). It is the first step of the project that is the subject of this paper.

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

1.1. Background

It is not uncommon to incorporate a manually constructed knowledge base for a particular subject area in automatic text analysis. A knowledge base is an abstract representation of a topic

area, or a particular environment, including the main concepts of interest in that area, and the various relationships between the entities. Knowledge bases have been proven successful for *classifying documents* in office environments (Chang & Leung, 1987; Eirund & Kreplin, 1988; Pozzi & Celentano, 1993). Knowledge based technology is also applied in *text extraction* and *information retrieval*. Here domain knowledge and/or linguistic knowledge enhance the precision of the operation. Systems such as FRUMP (DeJong, 1982), TESS (Young & Hayes, 1985), SCISOR (Jacobs & Rau, 1990), CONSTRUE (Hayes, 1992), JASPER (Hayes, 1992), and FASTUS (Appelt *et al.*, 1993) accurately and successfully extract certain conceptual information from texts. Such systems require a restricted text domain and rely on representations of the text corpus that reflect predictable patterns of linguistic context.

Even when texts cover unrestricted subject areas, it may be useful to identify where in the text significant information is to be found. Human readers can reliably identify relevant texts or relevant portions of texts merely by skimming the texts for cues. *Cue words*, indicator phrases and context patterns have been employed to identify significant sentences and concepts in texts for abstracting and classifying purposes (Edmundson, 1969; Paice, 1981; Riloff & Lehnert, 1992; Paice & Jones, 1993). When skimming a text, knowledge of the *text structure* of the text type is also advantageous. Text structure refers to the organization and interconnections between textual units, such that text conveys a meaningful message to the reader (Rama & Srinivasan, 1993). Automatically simulating a first rough skim of a document text, while employing knowledge about text cues and structure, has multiple application potentials including automatic categorization, indexing and abstracting.

The use of knowledge based techniques in text analysis is severely impaired by the lack of a justified theoretical model: we lack a document representation model (Salton & Buckley, 1991). However, the use of *'superstructural' schemes* or *grammars* is promising for abstracting text (Paice, 1990; Paice, 1991). A text grammar is often used when structuring text during text generation. This is very useful for highly structured documents (e.g. forms), but the structuring of the content of free text by the document author or engineer may be subjective and may overlook a document user's interest. Especially in a retrieval environment it is essential that a document representation can be tailored to the specific needs of a user or a group of users.

Text grammar research in the field of information retrieval is still in its infancy. A *text grammar* may be defined as a system of text features such as text structure and word arrangements, which deals with the functions and relations of these features in the text. Rama & Srinivasan (1993) developed a prototype for the representation and content extraction of medical abstracts. There is a choice of forms to represent a text grammar. *Frames* are well suited to represent the hierarchy of topics and subtopics (Hayes, 1992; Rama & Srinivasan, 1993), as well as document structure. There is an increasing interest in representing a document with a *semantic network* of frames (Wang & Ng, 1992).

It was our aim to design a domain-independent formalism, which allows to represent text structure including the major semantic units of a text, their attributes, and relations in the form of a text specific grammar. Parsing of the criminal cases based upon a case specific grammar results in categorization of the cases, identification and categorization of relevant case components, and identification of insignificant case text components. This procedure is a first, important step towards automatically abstracting legal cases.

2. DOMAIN KNOWLEDGE

2.1. Knowledge acquisition

A sample of Belgian criminal cases was investigated by an expert in criminal law, who identified the case categories and their relevant components. She interviewed other experts in the field and people responsible for the publication and manual summarization of cases in professional journals. When intellectually abstracting, an initial step regards the identification of the case category, of semantically relevant components and of insignificant text segments.

All criminal cases can be classified into seven categories and have a typical structure. The categories concern general cases and special cases, the latter being cases concerning appeal procedures, civil interests, refusals to witness, false translations by interpreters, infringements by foreigners or the internment of people. They are made up of nine components or segments (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 an interesting substructure (e.g. date and name of the court in the superscription, irrelevant paragraphs in the alleged offences, irrelevant paragraphs in the opinion of the court, irrelevant foundations in the legal foundations). In total we defined 14 different case components or segments relevant for abstracting purposes, some of them being subsegments of larger text segments. These segments present themselves in the text as: text blocks delimited or categorized by typical word patterns (e.g. the transition formulation), texts blocks preceding and/or following another text segment (e.g. identification of the victim), text paragraphs delimited or characterized by typical word patterns (e.g. irrelevant paragraphs in the alleged offences), text sentences delimited or characterized by typical word patterns (e.g. irrelevant foundations), or plain word patterns (e.g. name of the court). A word pattern is a combination of one or more text strings.

2.2. *Knowledge representation*

A text is usually composed of different components or *segments* which fulfil a semantic role in the text and which are combined according to specific semantic relations. Text segments relevant for categorizing, indexing or abstracting purposes can be shaped differently. They may concern paragraphs, sentences, or more informal text blocks of varying length. The text segments may be classified and/or delimited by linguistic and domain clues, which are white-space characters or punctuation marks, and/or word patterns.

The formalism that we designed allows to represent the broad semantic units of a text, their attributes, and relations in the form of a text 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 (Edwards, 1991). Frames offer the possibility to describe complex objects in a detailed way by treating a cluster of information as one entity. Frames can be reused. Frames can be organized in a network, reflecting document structure and content.

A *segment frame* defines a text segment: its slots describe the segment and its attributes. Each segment has a name (category). Segments belong to one of the following segment types: text block limited by word pattern(s) (indicator words or phrases) or other segments (type 'limits'), paragraph (type 'paragraph'), sentence (type 'phrase'), or word pattern (type 'pattern'). A 'limits' segment is a text block delimited by word patterns or by other segments and possibly characterized by word patterns. The complete text is a special case of this segment type and may be delimited by the begin and the end of the text file. A 'paragraph' segment is delimited by one or more new line characters, and possibly defined by delimiting or classifying word patterns. A 'phrase' segment is a complete sentence or subclause delimited by predefined punctuation marks, and possibly defined by delimiting or classifying word patterns. A 'pattern' segment consists of a defined word pattern (possibly template of text strings). A segment may have an interesting substructure: then the segment contains pointers to the subsegment frames. When the occurrence of a segment depends on other non adjoining segment(s) of the text, a rule specifying this dependence is attached to the frame. Segments have flags indicating whether they are optional or repetitive. When representing our criminal cases, we did not allow overlapping segments except in the case of nested segments.

The segment frames are organized as a semantic network (e.g. Fig. 1). The segment frames have a hierarchical (*has a*), sequential (*precedes*), or conditional relation (*if...then*) between them. The head segment frame defines the complete text or one major text component, and its possible subsegments. Such a representation is based upon a 'top down' interpretation of the text: its global concept is broken into more primitive concepts. Segments of a same hierarchical level may have a sequential relation: they follow one another in the text. The conditional relation

is needed when the legitimacy of a segment depends upon the existence of another segment. The resulting scheme is an abstraction of the structure of the text as it is conceived by the class of users of the text. It is possible to define different views (schemes) of the same text each defining different text uses. Or, as it was the case for the criminal cases, to define different text categories, described by different text grammars and discriminated by different classifying word patterns of their head segment frames.

A text string or sequence of strings (word patterns, indicator words or phrases) is an important indicator of the limits and/or category of a text segment. A segment may be characterized by a specific word pattern or by a logical combination of word patterns. Word patterns with a same delimiting or classifying function are grouped in a semantic class. A *word pattern frame* represents a semantic class and its member word patterns. This frame is connected with the appropriate text segment frame(s) (*limiting or classifying relation*).

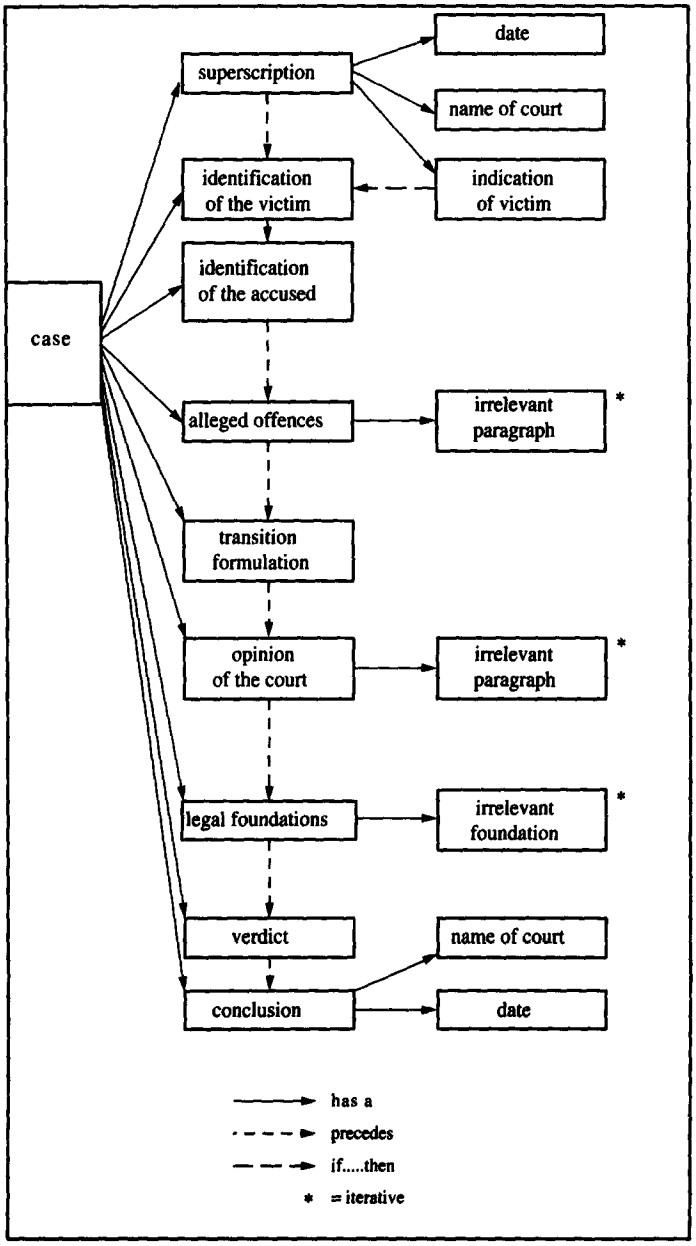

Fig. 1. Example of a representation of the segments of a criminal case.

Given the documents of the preliminary inquiry;
Given the documents of the judicial inquiry;
The court has examined
The Court has examined
Since the plaintiff does not master the Dutch language
Since the plaintiffs do not master the Dutch language
Given ... grounds

Fig. 2. Example of some pattern variants of the semantic class 'begin_transition'.

Word patterns are regular expressions and consist of one or more strings in a fixed order. 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 (e.g. the template '?laintiff?' representing 'Plaintiff', 'plaintiff', 'plaintiffs', etc.). The wild cards of the templates allow for a selective normalization of text strings. In the representation of the criminal cases such templates were useful to represent dates, word stems and the arbitrary use of capitals.

A delimiting or classifying word pattern may occur in the text in variant formulations. The variants are lexical, morphological and/or syntactical, or bear on punctuation marks. It is important to control the number of word pattern variants (Fig. 2) in the knowledge base. 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.

3. PARSING AND TAGGING OF THE TEXT

A parser was implemented to identify the category of the document text and/or to recognize its components based upon the text grammar. 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 parser focuses on finding the segments defined in the text grammar, while neglecting the remainder of the text.

The nested structure of segments (*has a* relation) is described by an extended context-free grammar, represented by a tree structure. The parsing starts with the triggering of the head segment frame. When a segment is identified, its subsegments (siblings) will be searched. A sibling inherits from its parents the text positions between which it is to be found. The segment tree is accessed with a depth-first strategy: subsegments are identified before other segments on a same hierarchical level are searched. The parsing employs a push down stack in order to remember segment frames still to be processed (cf. a *push down automaton*). Segments of a same hierarchic level, possibly but not necessarily follow each other in the text. The recognition of segments on a same hierarchical level takes into account 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), already found in the text. In this case the frame is activated after positive evaluation of the production rule attached to the frame.

The recognition of a segment takes into account its type and its *classifying* and *delimiting* patterns. When categorization of a segment depends upon a word pattern or a logical combination of word patterns, the parser employs a separate module, a *finite state* automaton, that recognizes regular expressions in the text in an efficient way. 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 a segment is not optional and only one of the segment limits is positively identified, the whole segment may be identified at this limit, thus minimally disturbing the processing of other segments.

After a segment is found, its begin and end positions in the text are marked with the segment name. Tags in *SGML* (*Standard Generalized Markup Language*)-syntax are attributed. Except for the attribution of category tags, the parsing does not structurally, lexically, morphologically or syntactically alter the original text. Figure 3 shows an example of a tagged criminal case.

4. EVALUATION PROCEDURE AND RESULTS

The SALOMON system was applied upon Belgian criminal cases issued by the correctional court of Leuven, dating from 1992–1994. The system realizes an essential categorization of the

<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>
<irrelevant_paragraph_alleged_offences> ...*Accused:* ...
</irrelevant_paragraph_alleged_offences>
...
<irrelevant_paragraph_alleged_offences> ...*By reason of*...
</irrelevant_paragraph_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* ...
<irrelevant_paragraph_opinion> ...*offence* ... *is certain*...
</irrelevant_paragraph_opinion>
...
<irrelevant_paragraph_opinion> *Given the enactment*...
</irrelevant_paragraph_opinion>
...
</opinion_of_the_court>
<legal_foundations> *On these grounds and in application of the following statutory provisions* ...
<irrelevant_foundations> ...*Code of criminal procedure*...
</irrelevant_foundations>
</legal_foundations>
<verdict> *THE COURT* ...
</verdict>
<conclusion> *Thus given* ...
</conclusion>
</appeal_procedure>

Fig. 3. Example of a SGML-tagged case: the word patterns in italic classify or delimit the case or its segments.

criminal cases. Also the structuring of the criminal case in relevant and irrelevant segments and subsegments is accomplished.

The text grammar knowledge related to the 23 categories, the *ca.* 300 word patterns (consisting of an average of 3.5 strings, numbers, or templates) organized in 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.

The result of the parsing of a criminal case is a case text indicating the general category: general decisions are distinguished from the special ones (decisions about appeal procedures, civil interests, refusals to witness, false translations by interpreters, infringements by foreigners, and internment of people). Moreover, the case segments defined in the text grammar are identified and tagged including the superscription, identification of the victim, identification of the accused, alleged offences, transition formulation, opinion of the court, legal foundations, verdict, conclusion, date, name of the court, irrelevant paragraphs of the alleged offences and of the opinion of the court, and irrelevant foundations (Fig. 3). From the tagged case general information about the case such as the date, the name of the court and relevant legal foundations are easily extracted and placed in the case abstract. The remaining relevant parts of the alleged offences and opinion of the court are ready for further abstracting with shallow statistical techniques (Moens *et al.*, 1997).

A sample of 1000 criminal cases (test base) was drawn from the original corpus. This test set is distinguished from the case set employed for knowledge acquisition. It is composed of 882 general and 118 special decisions, a proportion representative for the complete corpus. A student entering her final year in law school intellectually categorized the test cases and their components. The results were compared with the output of SALOMON.

The effectiveness of automatic category assignments to the case and its segments was computed in terms of a contingency table (Table 1) (Lewis, 1995), which summarizes the relationship between the system classifications and the expert judgments. Following effectiveness measures are defined in terms of the parameters 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):error rate$=(b+c)/(a+b+c+d)$

So, for each case and segment category *recall* and *precision* were computed respectively as the proportion of correct automatic assignments to the category upon the real existing number of this category in the test base assigned by the expert, and as the proportion of correct automatic assignments to this category upon the number of automatic assignments to this category (cf. Jacobs, 1993). Recall is the proportion of class members that the system assigns to the class. Precision is the proportion of members assigned to the class that really are class members. An alternative to precision is *fallout*, which calculates the proportion of non class members that the system assigns to the class. An ideal system would have recall and precision of 1 and fallout of 0.

Recall and *precision* are calculated for all categories (Tables 2 and 3). For some text segment categories the total number of non class members (b+d) can not be computed. When, for instance, a text segment is a text block delimited by typical word patterns, a non class member, then, may be any text fragment of any length. So, we only computed *fallout* and *error rate* for segments with fixed limits (entire text case, 'paragraph' and 'phrase' segments) (Tables 2 and

Table 1. Contingency table of classification decisions

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

Table 2. Results of the categorization of the entire criminal case

Case category	Effectiveness measures			
	Recall	Precision	Fallout	Error rate
Appeal procedures	1.000,000	1.000,000	0.000,000	0.000,000
Civil interests	1.000,000	0.916,667	0.001,011	0.001,000
Refusals to witness	0.888,889	1.000,000	0.000,000	0.001,000
False translations	1.000,000	1.000,000	0.000,000	0.000,000
Infringements by foreigners	0.733,333	1.000,000	0.000,000	0.004,000
Internment of people	1.000,000	1.000,000	0.000,000	0.000,000
General case	1.000,000	0.994,363	0.042,373	0.005,000
Average	0.946,032	0.98,729	0.006,198	0.001,571

4). 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 such as missing relations or word patterns (e.g. a zero recall of the category 'name of court conclusion' for special decisions), whereas precision errors may be due to ambiguities in the knowledge. A substantial number of errors are caused by typing errors. For instance in the category 'date superscription' 90% and 57% of the errors for respectively general decisions and special decisions responsible for the non identification of

Table 3. Results of the categorization of the case segments

Case segment category	Effectiveness measures			
	General decisions		Special decisions	
	Recall	Precision	Recall	Precision
Superscription	0.970,522	0.970,522	0.771,186	0.784,483
Date superscription	0.916,100	0.987,775	0.866,667	0.939,759
Name of court superscription	0.987,528	0.996,568	0.814,159	1.000,000
Identification of the victim	0.743,935	0.862,500	0.575,000	0.920,000
Identification of the accused	0.787,982	0.794,286	0.745,763	0.846,154
Alleged offences	0.843,964	0.982,759	0.696,629	0.925,373
Irrelevant paragraph offences	0.819,536	0.966,945	0.812,155	0.954,545
Transition formulation	0.867,347	0.891,608	0.500,000	0.632,184
Opinion of the court	0.871,882	0.895,227	0.594,595	0.687,500
Irrelevant paragraph opinion	0.856,416	0.991,582	0.907,143	0.980,695
Legal foundations	0.910,431	0.931,555	0.813,084	0.861,386
Irrelevant foundations	0.769,907	0.793,555	0.688,679	0.768,421
Verdict	0.896,825	0.933,884	0.703,390	0.954,023
Conclusion	0.959,184	0.998,819	0.728,814	1.000,000
Date conclusion	–	–	0.375,000	1.000,000
Name of court conclusion	–	–	0.000,000	–
Average	0.87,154	0.928,399	0.662,017	0.883,635

Note.–=not defined (the category does not apply or division by zero).

Table 4. Fallout and error rate of the categorization of the segments with fixed limits

Case segment category	Effectiveness measures			
	General decisions		Special decisions	
	Fallout	Error rate	Fallout	Error rate
Irrelevant paragraph offences	0.026,942	0.102,202	0.030,882	0.100,572
Irrelevant paragraph opinion	0.006,897	0.073,438	0.010,267	0.040,417
Irrelevant foundations	0.099,805	0.143,136	0.173,228	0.236,052

this category regard spelling errors (no space between the date and a foregoing word). For instance in the category 'irrelevant foundations' 88% and 83% of the errors for respectively general decisions and special decisions responsible for the non correct identification of this category regard the improper use of punctuation marks (no space between the punctuation mark and the following word). The non identification of a parent segment sometimes explains a low recall of its subsegment (e.g. the categories 'date conclusion' and 'irrelevant foundations' for special decisions). The use of wild cards in the representation of the patterns did not cause any misinterpretation by the system. The overall results are satisfying taking into account the limited time for knowledge acquisition and implementation.

5. ADDITIONAL CONSIDERATIONS AND FUTURE WORK

In the future extra tools, which assist in acquiring and implementing the knowledge, can be designed. For instance, because the knowledge implementation is based upon a graph, a graphical interface may be provided to implement the knowledge (Edwards, 1991). Alternatively, an interactive interface could be designed to facilitate the acquisition and implementation of the text grammar. Additionally, controls for detecting loops in the text grammar are useful.

Although some of the knowledge of the text grammar may be acquired with machine learning techniques from example texts, presently such an approach did not seem beneficial. Machine-learning systems solve problems by examining samples described in terms of measurements or features. They have been proven useful for the acquisition of simple lexico-semantic patterns that classify texts (Jacobs, 1993). When the technique of learning from examples is applied to acquire the text characteristics that classify documents, a representative sample of documents must be found and manually classified. More specifically, we need the following steps (Apté *et al.*, 1994). A dictionary of text features is created from the sample of manually classified texts. Then each new document is mapped into the training sample using the dictionary and a label, which identifies the category, is associated. Decision rules that distinguish one category from another may be induced. In case of several alternatives the best rule set is chosen, based on minimizing classification error or cost. Machine learning is useful as an aid rather than a replacement of knowledge acquisition.

For SALOMON automatic learning of the text features that classify a criminal case or case segment did not seem beneficial. Apart from the difficulty of learning the complete text structure from example texts, including all relevant and irrelevant text segments and their relations, there are the complications in automatically acquiring the word patterns that delimit or classify texts. First, these word patterns are expressed in many variants which are morphologically, lexically and syntactically very divers. 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 criminal cases. Also, the 'simple' patterns are not restricted to a specific type. They could for instance be single words, phrases, consecutive words with no syntactic relation, or whole sentences. Apart from the reasonable chance of an incorrect learning of the patterns, 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, as the effort needed for manually constructing the knowledge base.

We plan to extend the text grammar approach to the domain of magazine articles. More specifically, we want to use this approach to extract clippings from the articles. The clippings are to be included in a preview presentation of the articles on-line.

6. CONCLUSION

A growing amount of electronically available free text enlarges the need for an initial automatic categorization and structuring of the texts. When the text characteristics that

discriminate the different categories are possibly complex, but their number is limited, a knowledge based approach is useful. However, a powerful document representation is needed. This paper hopes to be a contribution to the search for a theoretical model of a document content. It has been shown that a representation as a text grammar is very promising.

An initial text categorization and structuring is useful for many purposes including automatic abstracting. The recognition of the text category, and of relevant and insignificant text components is an important first step when intellectually abstracting. Automating this process was especially useful for controlling the overload of present and future court decisions.

Courts may mark the content of legal cases with mark-ups at the time of text generation. This approach is beneficial for marking objective content attributes (e.g. date, name of the court). However, the marking of text content at the time of text generation may be subjective and restricts the possibilities of the user or class of users to define themselves the relevancy of texts or text components. Here, a text grammar defined for text utilization is advantageous.

REFERENCES

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). San Mateo, CA: Morgan Kaufmann.

Apté, C., Damerau, F., & Weiss, S. M. (1994). Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems, 12*(3), 233–251.

Chang, S. K., & Leung, L. (1987). A knowledge-based message management system. *ACM Transactions on Office Information Systems, 5*(3), 213–236.

Charniak, E. (1983). A parser with something for everyone. In *Parsing natural language*, ed., M. King, (pp. 117-149). London: Academic Press.

DeJong, G. (1982). An overview of the FRUMP system. In *Strategies for natural language processing*, eds., W. G. Lehnert & M. H. Ringle, (pp. 149-176). Hillsdale, NJ: Lawrence Erlbaum Associates.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the Association for Computing Machinery, 16*(2), 264–285.

Edwards, J. S. (1991). *Building knowledge-based systems: Towards a methodology.* London: Pitman Publishing.

Eirund, H., & Kreplin, K. (1988). Knowledge-based document classification supporting integrated document handling. In *Conference on Office Information Systems*, ed., R. B. Allen, (pp. 189-196). New York: ACM.

Hayes, P. J. (1992). Intelligent high-volume text processing using shallow, domain-specific techniques. In *Text-based intelligent systems: Current research and practice in information extraction and retrieval*, ed., P. S. Jacobs, (pp. 227-241). Hillsdale, NJ: Lawrence Erlbaum Associates.

Jacobs, P. S. (1993). Using statistical methods to improve knowledge-based news categorization. *IEEE Expert, 8*(2), 13–23.

Jacobs, P. S., & Rau, L. F. (1990). SCISOR: Extracting information from on-line news. *Communications of the ACM, 33*(11), 88–97.

Lewis, D. D. (1995). Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, eds., E. A. Fox, P. Ingwersen, & R. Fidel, (pp. 246-254). New York: ACM.

Moens, M.-F., Uyttendaele, C., & Dumortier, J. (1997). Abstracting of legal cases: The SALOMON experience. In *Proceedings of the Sixth International Conference on Artificial Intelligence & Law* (pp. 114–122). New York: ACM.

Paice, C. D. (1981). The automatic generation of literature abstracts: An approach based on the identification of self-indicating phrases. In *Information Retrieval Research*, eds., R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, & P. W. Williams, (pp. 172-191). London, Toronto: Butterworth & Co.

Paice, C. D. (1990). Constructing literature abstracts by computer: Techniques and prospects. *Information Processing & Management, 26*(1), 171–186.

Paice, C. D. (1991). The rhetorical structure of expository text. In *Informatics 11. The Structuring of Information*, ed., K. P. Jones, (pp. 1-25). London: Aslib.

Paice, C. D., & Jones, P. A. (1993). The identification of important concepts in highly structured technical papers. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, eds., R. Korfhage, E. Rasmussen, & P. Willett, (pp. 69-78). New York: ACM.

Pozzi, S., & Celentano, A. (1993). Knowledge-based document filing. *IEEE Expert, 8*(5), 34–45.

Rama, D. V., & Srinivasan, P. (1993). An investigation of content representation using text grammars. *ACM Transactions on Information Systems, 11*(1), 51–75.

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). Menlo Park, CA: AAAI Press.

Salton, G., & Buckley, C. (1991). Global text matching for information retrieval. *Science, 253*, 1012–1015.

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, The Netherlands: Tilburg University Press.

Uyttendaele, C., Moens, M.-F., & Dumortier, J. (1997). SALOMON: Abstracting of legal cases for effective access to court decisions. *Artificial Intelligence and Law* (in press).

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.

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). Washington, DC: IEEE Computer Society Press.

Marie-Francine Moens, researcher at the Interdisciplinary Centre for Law & IT (ICRI), Katholieke Universiteit Leuven (Belgium).

Caroline Uyttendaele, researcher at the Interdisciplinary Centre for Law & IT (ICRI), Katholieke Universiteit Leuven (Belgium).

The research is part of the SALOMON project. SALOMON automatically generates a synopsis of a legal criminal case.

We thank Tine Bouwen for the verification of the results. We are grateful to Prof. Dr. Jos Dumortier for his precious advice. We thank the anonymous reviewer for the useful comments.

Correspondence concerning this article should be addressed to the authors: Marie-Francine Moens or Caroline Uyttendaele, Katholieke Universiteit Leuven, Interdisciplinary Centre for Law & IT (ICRI), Tiensestraat 41, B-3000 Belgium. Electronic mail may be sent via Internet to: e-mail: {marie-france.moens, caroline.uyttendaele}@law.kuleuven.ac.be.