Text Clustering to Support Knowledge Acquisition from Documents

Stéphane Lapalut

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Text Clustering to Support Knowledge Acquisition from Documents

Stéphane Lapalut

N° 2639

Août 1995

PROGRAMME 3

Intelligence artificielle,
systèmes cognitifs
et interaction homme machine

ISSN 0249-6399

Text Clustering to Support Knowledge Acquisition from Documents

Stéphane Lapalut*

Programme 3 : Intelligence artificielle, systèmes cognitifs
et interaction homme-machine

Projet ACACIA

Rapport de recherche n°2639 - Août 1995

24 pages

Abstract: At the earlier stage of the knowledge acquisition process, interviews of experts produce a large amount of rich but ill-structured texts. Knowledge engineers need some tool to help them in the exploitation of all these texts. We propose the use of a statistical method, the top-down hierarchical classification and a new interpretation of its results. The initial statistical analysis proposed by M. Reinert (Reinert, 1979 and 1992) gives two kinds of results: first a segmentation of texts that reflects their «semantic contexts» that we use to raise structures of texts, and second, classes of significant terms belonging to these contexts, which can be related to the experts or to these specialities. In this paper, we describe the method, its empirical validity and its comparison with similar approaches, its uses with examples and results. We conclude with some research directions to deal with so-called "ontologies" on expert's domains.

Key-words: hierarchical top-down classification, statistical text analysis, text segmentation, text structure discovery, semantic context.

* Email: Stephane.Lapalut@sophia.inria.fr

Agrégation de segments de texte pour l'aide à l'acquisition de connaissances à partir de documents

Résumé : Dans les premières étapes du processus d'acquisition des connaissances, une grande quantité de textes riches en expertise, mais sans structure est produite. Le cogniticien a alors besoin d'un outil pour exploiter tous ces textes. Nous proposons l'utilisation d'une méthode statistique, la classification descendante hiérarchique et une nouvelle interprétation de ses résultats. Cette analyse statistique telle qu'elle a été proposée par Max Reinert (Reinert, 1979 and 1992) donne deux sortes de résultats : premièrement une segmentation des textes qui reflète leurs «contextes sémantiques», que nous utilisons pour mettre en évidence la structure des textes, et deuxièmement, un ensemble de classes de termes attachés à ces contextes, qui peuvent servir à la caractérisation des experts ou de leurs spécialités. Dans ce rapport, nous décrivons la méthode, sa validité empirique, les approches similaires, ainsi que son utilisation avec quelques exemples et résultats. Nous concluons sur des directions de recherche pour traiter les «ontologies» sur des domaines d'expertise.

Mots-clé : classification descendante hiérarchique, analyse statistique de texte, segmentation de texte, découverte de la structure de texte, contexte sémantique.

Text Clustering to Support Knowledge Acquisition from Documents

Stéphane Lapalut
ACACIA project, INRIA Sophia Antipolis,
BP 93, 06 902 Sophia Antipolis cedex, France
[Stéphane.Lapalut@sophia.inria.fr]

1 Introduction

In case of domain without established theory, such as complex accident analysis, the only way to obtain a significant and useful amount of data is to observe and interview experts working on various selected cases. This produces ill-structured text interviews. When reading all these interviews, the knowledge engineer lacks guidelines to model the domain, to distinguish and characterize the different approaches of the experts and to produce useful knowledge bases.

Given a huge corpus of expert's interviews, we propose the use of a statistical method called "top-down hierarchical classification", to handle both self contained texts and sets of chosen texts. It detects *groups* of terms of the corpus (a set of one or more texts) strongly distinguishable, according to the statistical occurrence of meaningful terms or pairs of them in small text units, such as sentences. These *groups* are called classes and have been identified as relevant *semantic contexts* (Benzecri, 1973 and Reinert, 1979). They lead to a partition of the corpus that reflects its structure. After interpreting each class with the help of related terms, the knowledge engineer knows the subject of each part of the structure of the corpus. With these results, he is able to select parts of given expert's interviews relevant to his purpose and to focus his work.

In this paper, we first start with a precise description of the method and the associated statistical analysis. The second part introduces the bases of the tool and method established by Max Reinert, and the way we propose to extend it for the purpose of knowledge acquisition. The third part deals with examples from the domain of road safety expertise and we show the validity of both the text clustering approach and our method. The two last parts deal

with related work, conclude on the realized work and propose some ideas for further research.

2 The top-down hierarchical classification method and its extension

This section describes the statistical classification method with theoretical and practical details. This method has been developed by Max Reinert since 1984 and implemented in a tool called ALCESTE (Reinert, 1992). This tool presently processes only French texts, even if an English extension is planned. The current version is a commercial one. The initial goal was to help the dissection of questionnaire answers. The principle has been generalized and applied to several kinds of text, from interviews to books. The purpose of Reinert was to use semantics contexts to help psychologists in their analysis and research of models. Given the statistical analysis results, we have found text clustering as another application.

2.1 Principle of the initial method

The initial objective addressed was to discover semantic contexts characterized by groups of terms from a given corpus. A principle derived from the Huyghens decomposition formula is used: "to find a partition of a set that minimizes the intra-class variance, it is sufficient to find dichotomies that maximize the inter-class variance".

The overall process is drawn in figure 1. The investigator wants to test some hypothesis. He builds the appropriate questionnaire, submits it to some people and gets the answers. He puts all the answers in a corpus, which is naturally structured by the questionnaire grid. These first roughly distinguished pieces of the corpus are called ICU (*initial contextual units*). Then an automated process cuts each ICU into regular *elementary contextual units* (ECU). The classification, called *top-down hierarchical classification* is done on this set of ECU with the help of the set of words from the corpus. The result consists of a set of classes, defined by exclusive sets of ECU and sets of preponderant words appearing in them. Automated steps use statistical criteria as described in the next sections.

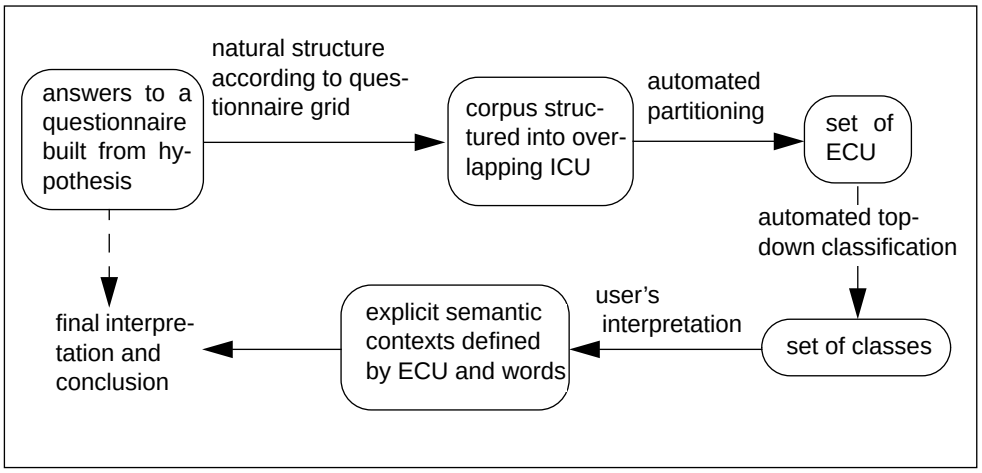


Figure 1: Initial method for questionnaire analysis.

2.2 Our extension of the initial method

We refine the initial cycle of figure 1 to adapt it for our purpose of discovering the structure of a corpus. We have found a correlation between natural articulations of the corpus and the structure given by the classes. We can sketch our method by extending figure 1 as shown in figure 2. The classes give a structure that is refined by the knowledge engineer with the help of class interpretations. Formally, we can consider the expert's documents processing as the research of correspondences between sets from a triplet $\{U_1, S, TU_1\}$ where:

- U_1 is the elementary segmentation of the corpus (the set of ECU),
- TU_1 the objective clustering of U_1 into classes (found by the top-down classification) and
- S the implicit organization of the corpus according to the interrelated topics in interviews ; this organization is to be discovered.

The whole analysis permits the clarification of S, which is the correspondence between the corpus viewed as a sequence of ECU and semantics contexts defined by classes in TU_1. These different stages in the splitting of the corpus are depicted in figure 3.

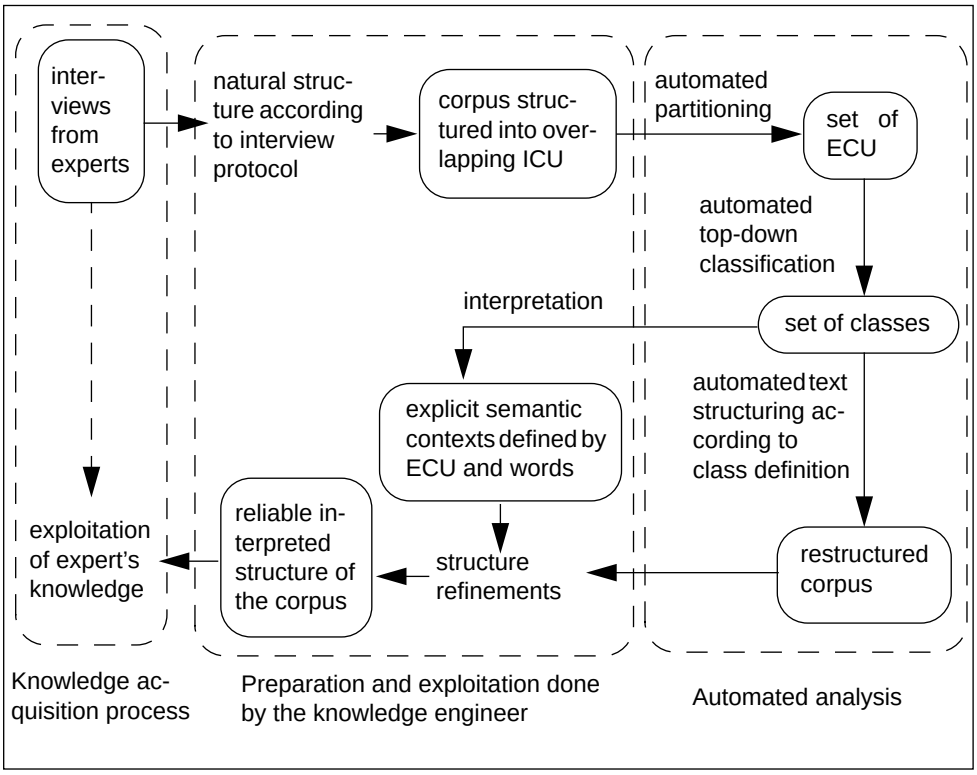


Figure 2: Our extension of the method towards knowledge acquisition process.

Figure 3: Evolution of the corpus structure (TU1 is not represented ; it groups ECU into clusters arranged in a hierarchy, as the one in figure 6 page 8).

Now, let us describe in detail the main point of the method: the top-down classification algorithm. Firstly we explain the format of the inputs, secondly the theory that underlies the algorithm and thirdly the outputs relevant for our method.

2.3 Input data

In this section we highlight the data format and the way they are extracted from the initial corpus. ALCESTE takes two inputs: the corpus, which can be composed of one or more texts, and a set of parameters, called the analysis plan, which the user gives to conduct the analysis. The corpus is organized by the user as a sequence of ICU. These units delimit the different texts in the corpus or the natural articulations of a unique text, like a chapter in a book (figure 4). In the case of single expert interview, the text interview is mostly a unique ICU. For each ICU, a label (like *yve_int in figure 4) and a number of special markers, (keywords starting with *), are used to type each ICU. By these keywords, the user specifies the kind of information each ICU is supposed to contain according to the protocol followed to obtain texts. The same keywords can be used in several ICU, such as "*INT" to type each ICU obtained from interviews of one expert. This feature is very useful when we treat several texts from several experts (see section 3.3).

After ICU typing, the knowledge engineer sets the analysis plan parameters (ECU length, maximum number of classes) and starts the automated analysis. Before the main algorithm processing, the statistical analysis called "top-down hierarchical classification", a morphological reduction of the terms of the corpus is done. Two lists are then extracted from the corpus: one contains all the words from the corpus in alphabetical order, the other one contains the sequence of ECU that composed the corpus in the order they appear in the corpus. Some words, called toolwords (noisy words), such as prepositions or pronouns are recognized and typed according to dictionaries. Other words, such as nouns, verbs, adjectives, are considered as meaningful terms and called plain-words (non-noisy words). An ECU is a word sequence that integrates a fixed number of *plain-words*, as specified by the user in the analysis plan. ECU useful length ranges from 10 to 20 plain-words. ECU can be sketched as sentences from the corpus. The segmentation is performed according to the punctuation with a priority order of the signs (. > ? > ! > ; > : > , > space). The fixed length is needed to validate the statistical algorithm and is not a strong constraint for the purposed context identification. The mostly used analysis plan allows a double classification with ECU of two different lengths. It permits an adjustment of this length to obtain a better classification. Also, the cross between the two hierarchies of classes found according to a $\chi 2$ criteria determines a stable classification (see figure 7, page 10).

Example of a corpus composed of ten texts that define ten ICU with the keyword codes:

the first keywords name the ICU with the name of the text file it contains
the following keywords describe the types with the codes:

*Y states the name of the expert Yve
*P " " Pie
*M " " Man
*J " " Jlo
*F " " Fra

*INT means interview
*DUO means case study by two experts
*TRIO means case study by three experts
*SOLO means case study by a single expert

*002 *003 *24 are the index numbers of the studied cases

corpus

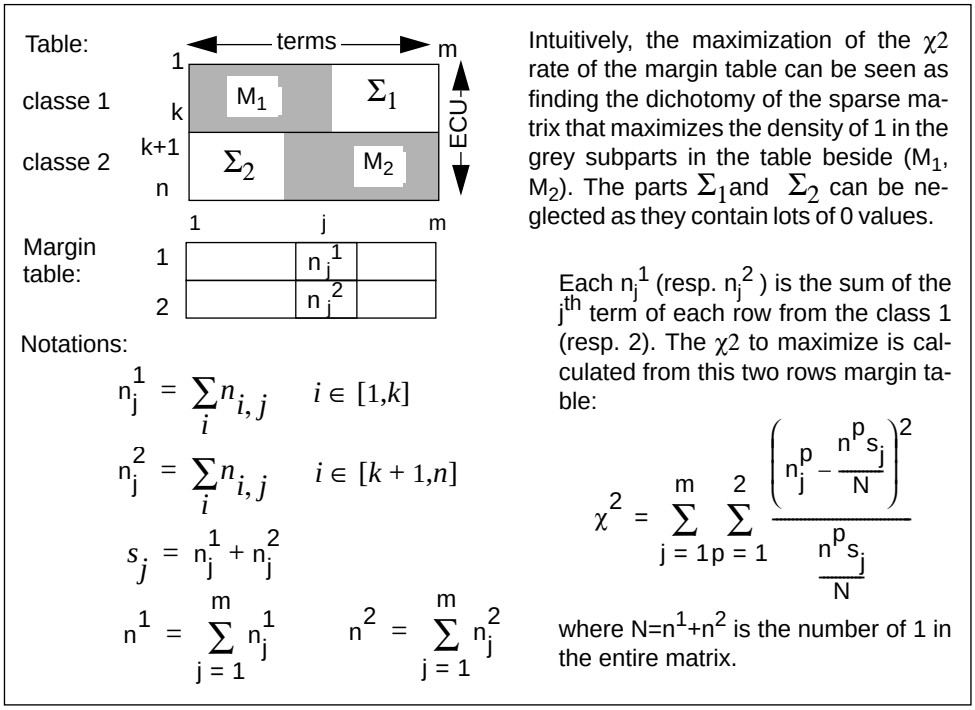
```
*yve_int *Y *INT
        "ASCII text from interview of the expert Yve."
*pie_int *P *INT
        "interview of the expert Pie."
*man_int *M *INT
        " interview of the expert Man."
*jlo_int *J *INT
        "interview of the expert Jlo."
*fra_int *F *INT
        "interview of the expert Fra."
*dan_man_002 *D *M *DUO *002
        "conversation between Dan and Man during case 002 resolution."
*dan_jlo_24 *D *J *DUO *24
        "conversation between Dan and Jlo during case 24 resolution."
*dom_fra_24 *E *F *DUO *24
        " conversation between Dom and Fra during case 24 resolution."
*man_pie_jlo_003 *M *J *P *TRIO *003
        "conversation between Man, Jlo, Pie during case 003 resolution."
*man_003 *M *SOLO *003
        "discourse of Man during his case 003 resolution."
```

Figure 4: Headers of the ten ICU in a corpus composed of ten expert's texts.

The program uses the two above lists as rows and columns of a double entry boolean table. The presence of a term in ECU is noted with 1 and its absence with 0 (figure 5). The completed table, which is a sparse matrix, is used as the input of the main algorithm. We describe it in the next section.

2.4 The Algorithm

The applicability of the underlying statistical theory used by this algorithm was proven in (Benzecri, J.P., 1973; Kendall, M.S., 1967; Reinert, M., 1986). In this section we expose the key points of the algorithm to have an overview of the whole process. Given the boolean table, which use terms as column



$$n_j^1 = \sum_i n_{i,j} \qquad i \in [1,k]$$

$$n_j^2 = \sum_i n_{i,j} \qquad i \in [k+1,n]$$

$$s_j = n_j^1 + n_j^2$$

$$n^1 = \sum_{j=1}^{m} n_j^1 \qquad n^2 = \sum_{j=1}^{m} n_j^2$$

$$\chi^2 = \sum_{j=1}^{m} \sum_{p=1}^{2} \frac{\left(n_j^p - \frac{n^p s_j}{N} \right)^2}{\frac{n^p s_j}{N}}$$

Figure 5: Principle of the sparse matrix dichotomy using $\chi 2$ distance.

entries and ECU as row entries (m_{ij}=1 means that the i^{th} term belongs to the j^{th} ECU), we define classes as particular sets of rows ; a given row belongs to a single class. The final set of classes is an incomplete partition of the initial set of ECU. To distinguish classes, the algorithm uses a $\chi 2$ distance between the margins of two given sub-tables, determined by successive dichotomy as describes in figure 5. A margin is a row vector formed by the sum of all values in each column (see figure 5). At each step of the algorithm, the dichotomy that maximizes the $\chi 2$ rate of the margin tables is found. In a simpler form, it can be stated as:

- first step: find the dichotomy that maximizes the $\chi 2$ association rate between the margins of the two determined classes.

- other steps: until the specified number of classes is reached, do:

  1- pick the class with the greatest number of rows,

  2- among all row combination, find the dichotomy that splits this class into two subtables and maximizes the $\chi 2$ of their margins,

  3- replace the picked class with the two found classes.

An example for a four terminal classes partitioning is drawn on figure 6. At the end of this process, each class is defined by an exclusive set of ECU and a set of terms ; some terms can be found in several classes. Then for each class, two steps enable to determine the sets of the best correlated ECU and terms according to a $\chi 2$ rate. Those results are used to draw the segmentation in our method.

The length of ECU in the corpus determines the quality of the found hierarchy of classes. The quality of a hierarchy refers to the partitioning of the corpus and to the related sets of terms that defined each class. The higher the ratio of terms that mostly belongs to the same class (occurrence number of a term in the ECU of the same class against the total number of his occurrence in all ECU), the better the classification. To improve this ratio and obtain a good hierarchy, we have to choose the appropriate length for ECU. The tool gives a percentage of terms that mostly appear in a single class to estimate the classification correctness, the average ranges from 50% to 70%.

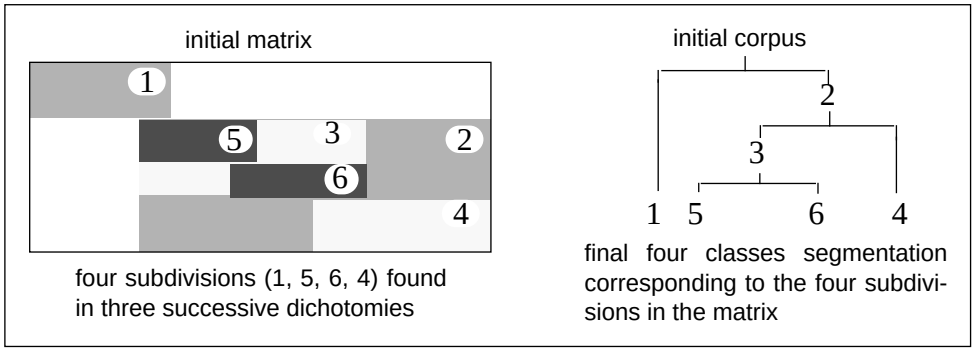


Figure 6: Example of a classification in four terminal classes.

2.5 Statistical analysis outputs

Many productions result from the automated analysis. About ten times the size of the corpus distributed into fourteen files is generated. A lot of information about the corpus can be obtained with appropriate interpretations. We

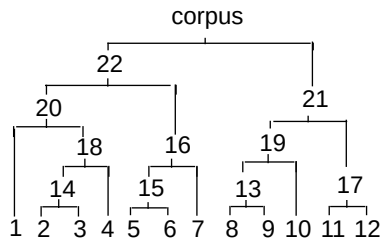
focused our effort on a small but relevant part of these outputs for our purpose. We briefly describe the format of files we have used, the interpretation and the results they lead to.

The first result we exploit is the double classification and the selected stable classes. The corpus structured as in figure 4 (page 6) gives the hierarchies in figure 7 (page 10). In this example, the analysis plan asks for 12 classes and the comparison between the two classifications enlightens 11 stable classes. For each of them, the number of correlated ECU is given with $\chi 2$ rates. The rate of correctness of 54% empiricaly validates the choice of the ECU lengths made (12 and 14 plain-words per ECU).

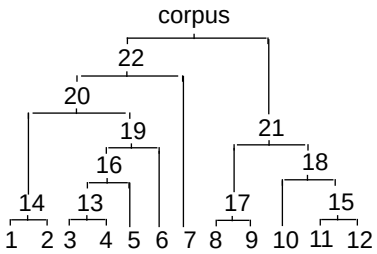
To be useful, a meaning must be attached to each class. This work is the burden of the user as the classification is not supervised nor guided with a predetermined lexicon. Each class is characterized by the list of its ECU and four files of correlated terms. These files permit the class labelling and for each class, consist of two lists of *terms* and two lists of *couples of terms*. For both couples of lists, one list concerns the best correlated terms to the class (*profile*) and the other the less correlated terms (*anti-profile*). An excerpt from a term profile is shown on figure 8. As few words are preponderant in profiles for each class, knowledge engineers are led to the same semantics context interpretations. These interpretations require some domain knowledge and lead to terminological choices. This point will be discussed in section 3.3, as they do not influence the main result, i.e. the enlightenment of the implicit structure of the corpus. For each class, the knowledge engineer cross-checks the correctness of his four interpretations. In most cases, no contradiction occurs and one expression results to characterize each class. These class tags are used to guide the knowledge engineer in the next step, the labelling of each part of the corpus structure.

The outcome of our method arises in this last stage, the structure refinement. The set of preponderant ECU attached to the classes permits the clustering of the initial set of ECU, e.g. the corpus. With the help of a graphic distribution of ECU from the corpus for each class according to their $\chi 2$ association rate, the codification is able to locate the small corpus parts (less than a page) that contain articulations of the implicit structure of the corpus. This stage is detailed in section 3.1 with examples.

① Hierarchy with ECU length of 12:



② Hierarchy with ECU length of 14:



Stable classification in 11 classes obtained by the crossing of the two hierarchies:

classes	crosses ① ②	χ^2
1	1 and 1	3923
2	12 and 9	3930
3	3 and 3	1853
4	11 and 8	4127
5	6 and 4	911
6	8 and 12	2369
7	5 and 7	1149
8	9 and 11	3285
9	10 and 10	3820
10	2 and 6	685
11	4 and 5	1265

Crosses between classes (from 1 to 12) and intermediate classes (from 13 to 22) of above hierarchies ① and ② determine 11 stable classes (classes 7 from ① and 2 from ② give no significante crossing).

For each pair of class (C1, C2) with C1 belonging to hierarchy ① and C2 to ②, the program calculates χ^2 distance between margins of their associated submatrix. Pairs with higher χ^2 are retained and determined stable classes.Results reported in the table reports come from the cross of the two above hierarchies. Class numbers with associated pairs and χ^2 are given.

Figure 7: Example of stable classification resulting from crossing between two hierarchies of classes obtained from the corpus sketched in figure 4 with two different ECU lengths (correctness of 54.08%).

3 Results and validity of our extension

The algorithm is blind with respect to the semantics of the word correspondences it makes. It only gives us an abstract result of all word associations that the reader can outline from the text. Of course, these associations reflect a feature of natural language, called *semantic context*. We describe here the way to interpret the analysis results and all the information given about the corpus. As six kinds of different analysis plan are relevant for common kinds of corpus and user's purposes, we focus on the most useful plan that is per-

formed on both dialogues and book texts. This plan performs a cross between two hierarchies built with different lengths of ECU given by the user.

Our method addresses two goals, firstly to retrieve the natural articulations and the structure of the corpus, which is the main goal when the corpus contains only one text, and secondly to have qualitative information about differences and similarities between components of the corpus, especially in the case of corpus composed of several texts that come from different experts. First we describe class interpretation, which is common to all kinds of analysis, then the relations between classes and corpus, considering both single text and multiple text corpora. All examples used from now are translated from french.

3.1 Class interpretation

3.1.1 Class description

Each class is characterized by four lists of terms. Two lists concern the most representative terms and pair of terms, the two others represent the less significant terms and pair for terms of the considered class. In all these lists, each term is characterized by five numbers (surrounded numbers refer to the columns in figure 8):

1- the place number in the dictionary of terms built from the corpus (column ①),

2- the number of occurrences in the ECU of the class (column ②),

3- the number of occurrences in the whole ECU (column ③),

4- ratio of ECU in the corpus in which the term appears (column ④),

5- $\chi 2$ association rate between the term and the class (column ⑤),

6- the term in its reduced form with an optional mark (column ⑥).

The lists are sorted from the greater $\chi 2$ rate to the lower one ; keywords are treated apart (figure 8). Terms with low $\chi 2$ rate and a 100% belonging rate are as representative as the one with the highest $\chi 2$ rate (such a term exclusively belongs to one class). The list of terms from class 7 of the previous classification and its interpretation are given in figure 8.

List of terms correlated to the class 7 in the previous classification, with $\chi 2$ superior to the average $\chi 2$ (62.96):

①	②	③	④	⑤		⑥
409	50.	75.	66.67	1057.00	X	mesur+
72	25.	30.	83.33	667.29		grip+
122	20.	25.	80.00	510.39		characteris+
88	17.	20.	85.00	462.62		equipment+
57	13.	13.	100.00	420.22	0	join+
134	24.	51.	47.06	342.56		road+
593	12.	14.	85.71	329.12		surface+
121	5.	7.	71.43	112.47		countr+
247	5.	8.	62.50	97.20		light+
368	7.	19.	36.84	74.82		itinerar+
475	6.	15.	40.00	70.56		tyre+
934 *	91.	448.	20.31	503.27	*	*fra-ent
945 *	91.	996.	9.14	158.69	*	*F
944 *	150.	2653.	5.65	130.33	*	*ENT

The tagging of this class has been stated as:
"interview of Fra dealing with grip of tyre on roads".

Figure 8: Example of a class tags, according to its term *profile*.

3.1.2 Class interpretation

For each list, the user tags all classes with an expression constituted by preponderant terms, such as "the kinematics analysis process" or "the grip of tyre on road under various conditions", according to the terms with the highest $\chi 2$ rates and percentages. For each class, two expressions say what the class *is* and two others say what the class *is not*. One can think that such a tag is a subjective task that will lead to different results with different users. It is not the case, as there are not so many terms to decide the right description, i.e. terms with the highest $\chi 2$. Tags attach two expressions describing what a class is about and two additional expressions relating what the class is not about. So, four pieces of information are sufficient to check the rightness of the whole class description and then, to give a unique meaningful expression.

In this process, useful data that help the user are the explicit domains to which the subdomains evoked in the corpus belong (such as kinematics calculus, which is a subdomain of kinematics analysis). We use them as sorts to type each expression. They allow a simple way to check the coherence of the four expressions that describe classes. Comparisons between classes seem to

be easier too with the help of these sorts. As we have experimented this way, it is only useful to improve the coherence between all class interpretations.

In the example of figure 9, a single text corpus results in five classes. In the car accident analysis field, there are two important stages in the diagnosis: *collecting* all relevant information after the car crash (noted C) and *analyzing* every document to search all kinds of data needed (noted A). The experts currently make another distinction, between three specialities: analysis of the infratructure in the car crash area (noted I), analysis of the *driver*'s behavior (noted D) and analysis of *kinematics* aspects (noted K). Types C and A are

classes	profile	anti-profile
1	A	I or C
2	C	A
3	crash	I, C
4	I or D	A, C
5	car1 crash	C

Typing of five classes from a "single text" corpus to crosscheck coherence of profiles according to anti-profiles ("crash" belongs to type A).
Types are consistent w.r.t. profile and anti-profiles for each class (dealing with crash belongs to kinematics analysis).

Figure 9: Example of class interpretation crosschecking.

exclusive since these are two distincts phases in the expert's activity. For the sake of the analysis, we consider that types I, D, K are exclusive, according to the focus of expert's analysis in regard of there own specialities. We remark that a link exists between types A and K.

For each class, the comparisons between tags related to both term lists and pair of term lists for class profiles and anti-profiles result directly in a single expression. If for a class, the profile leads to A and the anti-profile to A, there is a misleading interpretation for this class. Then the knowledge engineer has to revise his tags according to profiles and anti-profiles.

3.2 Single text analysis

In this part, we detail the results of a single text corpus study, the text entitled "*dan-jlo-24" in figure 4. Two experts, Dan and Jlo, of the same specialities (kinematics), are dealing with the case 24, which was unknow to them, with lots of comments about their activities according to a thinking aloud protocol. The text analysis leads to five classes. We use their related sets of ECU

determined by ALCESTE to retrieve the implicit structure of the conversation between the experts during this case study.

3.2.1 From classes to text structure

Each set of ECU related to classes contains ECU with their $\chi 2$ rate association and with the class they belong to. Each ECU is indexed with a number according to the order in which they appear in the corpus. These two kinds of information are used as cartesian coordinates to build the graphical correlation between all ECU and each class. Figure 10 represents graphics built with results from the previous single text corpus. Each graph relate ECU distribution (X axis) according to their $\chi 2$ distance to this class (Y axis). ECU range from 1 to 1340. The five graphs comprise the final splitting (vertical dotted lines) refined by the knowledge engineer. For each graph, each bar above the X axis relates the $\chi 2$ rate for one ECU of the related class. Bars under the X axis relates $\chi 2$ of ECU from the four others classes (that does not means that these $\chi 2$ are negative, but these oppositions allow a good visual appreciation of relative importance of each class). This is a simple visualization of ECU distribution according to classes. We observe that for a given class, the distribution is not a random one. Clusters of ECU appear and the hypothesis we made is that they reflect subjects dealt with in distinguishable parts of the corpus, in this case, one text. The gaps between clusters intend to be parts of the text where knowledge engineer can find articulations and swaps between one subject to another. Articulation locations give the final splitting drawn by dotted lines in figure 10.

To test this hypothesis, we submitted texts to a reader. His goal was to locate parts and transitions between them. For a 60 pages long conversation between two experts, the reader was able to recognize parts after three or four readings, without having certitude of their relevancy. In the same time, we analyzed the text with our method and we found a partitioning. The two results are compared in figure 10. According to precise locations of transitions in the pages of the text, the match between the program and the reader sounds accurate. For 12 parts suggested by the analysis, only four conflicts occur. In the next section, we investigate in depth the differences between the qualitative and quantitative information given by the reader in respect of the data provided by the analysis.

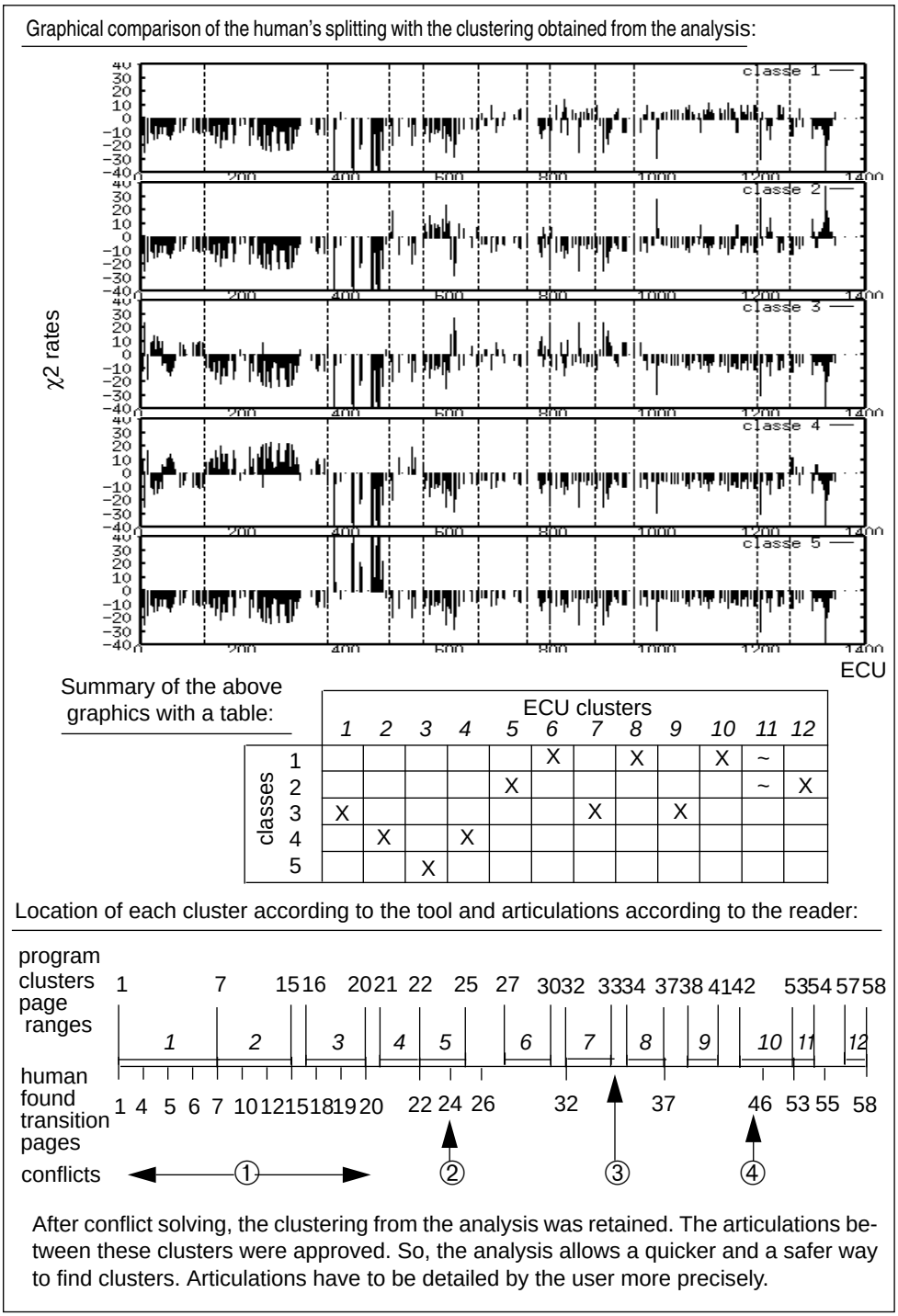


Figure 10: Example of text clustering and comparison between the reader's and the program results.

3.2.2. Exploitation and refinements

In the few experiments we have done, the structure suggested at the end of the analysis was always relevant. The resolution of conflicts concerning text splitting shows us that the reader tends to be too precise or to miss some articulations he was able to retrieve with a careful reading (in figure 10 see conflicts ①,② and ④ due to excessive preciseness, and conflict ③ due to insufficient preciseness). These two phenomena depend on the text, with at least one of them observed in all comparisons between the reader and the program. Another type of conflict occurs when comparing the reader's type attributions and class types for each part. The program is not intended to give the right ones, but the class types it provides are useful to help the human interpretation or to check them. Results for the analysis of the single text corpus are reported in figure 11. For each of the four conflicts noticed, we have read again the related parts with possible types in mind. Three of them are not true discrepancies as the part subject interpretations are ambiguous. The other one is a conflict and the reader has given the right meaning. So, we can say that program class types lack precision. But in most cases, the program gives the right classe type and the conflicts point ambiguous parts of the text. Anyway, the user has to read the text at least one time to make precise the articulations between the parts and the subject of each part. The same work without the help of the statistical analysis is somehow dull, takes a long time and it is difficult and time-consuming to check it.

3.2.3 Conclusion

As regards our experiments and tests, the usefulness of the analysis is at least the guidance of the reader in the research of articulations in the structure of a single text corpus. Another aspect of its usefulness is to check the title that the reader gives to each part. According to the type of classes, we can type the clusters, in fact parts of the text. In our experiments, the reader was sometimes misled in this task. He gave titles that were not in accordance with the ones related by classes. For each conflict, a deeper reading of the incriminated parts of the text revealed that the class types from program were mostly the right ones. In some cases, the reader's types and the program types are different but not conflicting: both are possible and the interpretation of these sorts of parts is ambiguous.

To sum up, the statistical analysis results help the user to find the relevant parts from a simgle text corpus and to locate articulations between them. He

only has to prepare the analysis and to interpret its results, then he specifies the structure with one or two readings of the text. It is a safer and quicker way than the same task conducted without the help of the analysis, which requires at least three or four readings, with no guarantee of the correctness of both splitting and meaning of each part.

3.3 Multiple texts and multiple experts

In this case, the main goal is not to discover the structure of the corpus, but to obtain some useful information to compare several texts. As each text characterizes one or several experts working together, from this sort of analysis we expected some help for establishing expert's knowledge and reasoning differences. As several parameters occur at the same time and many combinations between texts are relevant, we have not investigated all the ways. We have especially studied the case of a single text corpus and a multiple text corpus that includes this text. We present here the kind of results we obtained.

3.3.1 A case study

To test this kind of analysis, we put about 15000 lines of text from experts' interviews within the same corpus, which gives less than 10000 ECU (ECU length of 12 and 14). The organization of that corpus is described in figure 4. The obtained classes presented in figure 13 lead to a corpus structure very close to the existing ICU partition, e.g. the initial texts. Precisely the program points out 11 classes. Each of them is directly relevant for one or two ICU:

(a)- 5 ICU are relevant for a single class (ICU 2, 5, 7, 9 and 10),

(b)- two ICU for two classes (ICU 1 and 4),

(c)- one ICU for three classes (ICU 6),

(d)- and the last two ICU (3 and 8) are not directly characterized by a class, but they are weakly related to two classes.

In the case (a), we have verified that the semantic contexts described by the classes are in accordance with the related texts. Case (b) leads to the same conclusion. The two classes related to ICU 1 and 4 correspond to the same domain with a slight point of view difference. Case (c) is a very interesting one and we investigate it in further detail below. Anyway, classes related to ICU 6 are in accordance with the previous analysis of that text we detailed before (section 3.2). At that time, we have no explanation for the case (d) and

The five class interpretations:

classes	labels from profile	labels from anti-profile	final class types
1	collision,vehicle escapes kinematics sequences impact balancing	infrastructure, crash area driver's behaviors	A
2	crash according to drivers's point of view	kinematics analysis	C
3	marks from the crash on the road	infrastructure, accident context	A
4	infrastructure, driver's approaches crash area	kinematics analysis	I,D
5	impact marks on car-1	driver's interviews	A

Final results with conflict resolution:

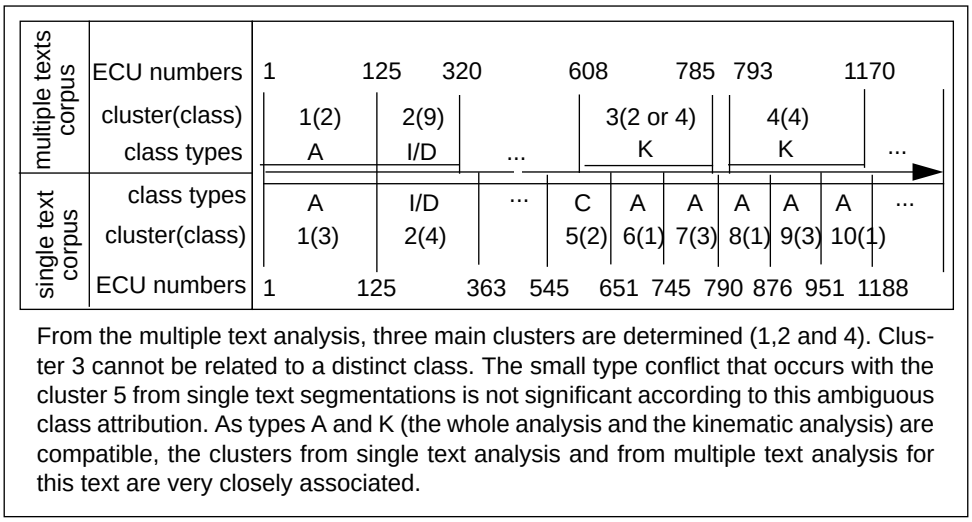
text part	class	type	reader's interpretation, (type)	conflict resolution
1	3	A	crash facts from the analysis of both vehicles (A)	(no conflict)
2	4	I,D	analysis of accident pieces concerning the crash (I)	(no conflict)
3	5	A	the car-2 trajectory (A/C)	the reader was right
4	4	I,D	interviews from drivers (C)	ambiguous, both are right
5	2	C	accident context before the crash (C)	(no conflict)
6	1	A	impacts, kinematics analysis (A)	(no conflict)
7	3	A	analysis of car-2 tyre marks (A)	(no conflict)
8	1	A		(no conflict)
9	3	A	crash kinematics equilibrium (A)	(no conflict)
10	1	A		(no conflict)
11	2	C	reconstitution of the impact (A)	both are combined
12	2	C	conclusions from the reconstitution (A)	both are combined

Figure 11: Final results for a single text structure analysis.

we relate some issue in the next section. So, we already can conclude that this kind of analysis makes precise similitudes and differences between the texts entered as ICU in the corpus. For each of them we obtained a synthetic point of view of their main subject (cases (a) and (b)).

Relations between single text and multiple text analysis results:

Now, let us analyze the case (c) deeply: we make the comparison with its previous analysis (see figure 10). From the multiple text analysis, we reported the structure given by the three related classes (2, 4 and 9) as shown in figure 13 (bold **X**). From the segmentation presented in figure 10 we have specified division page numbers with ECU numbers within the articulations are located. The multiple analysis leads to four clusters. Three of them are very close to the single text clusters: the first two classes obtained from both



From the multiple text analysis, three main clusters are determined (1,2 and 4). Cluster 3 cannot be related to a distinct class. The small type conflict that occurs with the cluster 5 from single text segmentations is not significant according to this ambiguous class attribution. As types A and K (the whole analysis and the kinematic analysis) are compatible, the clusters from single text analysis and from multiple text analysis for this text are very closely associated.

Figure 12: Correspondence between partitions of a given text (*dan-jlo-24) from a single text analysis and a multiple text analysis.

segmentations match and each of the two last clusters from multiple text analysis gathers together three clusters of the single text analysis segmentation. The class types related to clusters are in accordance. Then, from the multiple text analysis, we have a more abstract view of the same text than the coarser level of detail from the single text analysis of the same interview.

3.3.2 Class characterization

The huge number of terms and ECU contained in this sort of multiple text corpus allows a very contrasted class characterization. $\chi 2$ rates are higher than in the case of single text that makes the interpretation of classes easier. For each class, a small set of terms are preponderant. But each of them is very precise and specific, such as the class 8: "the accident factors for the case 003, central lane and indicator ambiguity". This class is relevant for the text 9 and its characterization is exactly the conclusion of a manual analysis we made before.

Many classes strongly characterize expert's specific vocabulary and skills. So, we obtain specific terms for these experts from these classes. The main problem is to obtain the shared terms and to identify the way each expert uses them. We suppose that a link exists between this shared vocabulary and the only unexplained phenomena about the weakly characterized texts 3 and 8.

3.4 Conclusion

From the analysis of both single text and multiple text corpora we conducted, we get two results. Firstly, both inter-text and intra-text comparisons are possible and secondly, for a single text corpus, we obtain its precise structure refined by the knowledge engineer. Above a given number of different kinds of text, e.g. ICU in the corpus, the overall text differences are given back by classes. Texts strongly characterized by specific subjects lead to peculiar classes. Side effect phenomena reveal some specific features of very precise parts of texts. So, according to the way the classification is carried out, different grains and structure levels can be reached. The main remaining problem is to put relevant texts together within a corpus to highlight a level of detail or some desired differences between texts and so, expert's knowledge particularities.

4 Related Work

Statistical approach is far from being a new one in text processing. It has been widely used in text categorization (Jacobs, 1992 ; Register and Kannan, 1992), as well as in language analysis (Hush, Wu and Tan, 1992 ; Charniak, 1993). Some applications have been made to the knowledge acquisition field.

	classes	*yve-int	*pie-int	*man-int	*jlo-int	*fra-int	*dan-jlo-24	*dan-man-002	*dom-fra-24	*man-pie-jlo-003	*man-solo-003
interviews, Yve's detailed accident studies (DAS)	1	X	~	~							
duo, impacts, traces and car deformations	2						**X**	X	~		~
*pie-int (χ2 of 1332 for an average of 32.15)	3		X								
crash, kinematics reconstitution	4				X		**X**				
road infrastructure and its users	5			~		~					
perception of infrastructure by the drivers	6	~									
the grip tyres on the roadways	7					X					
accident factors, central lane and indicator	8									X	X
infrastructure and approach before the crash	9						**X**	~	~		
the whole accident analysis, data and indices	10				X						
the accident situation	11	X									
class interpretations		1	2	3	4	5	6	7	8	9	10

X states a strong correlation between the class and the text (more than half of the ECU that characterized the class belong to these texts).

~ states a weak but existing correlation between the class and the text.

Figure 13: Correlation between texts and classes from a multiple text corpus.

Statistical methods are mostly used to strengthen a conceptual method (Register and Kannan, 1992 ; Fall, Crawford, Souders and Rabin, 1989 ; Jacobs, 1993). Tools such as NLDB (Jacobs, 1993) and SKIS (Register and Kannan, 1992) combine both conceptual and statistical approaches within a hybrid system. Most applications tend to provide a set of tools such as NLDB, which is a "set of statistical method", or to offer an assistant to a human expert, like INLEN (Michalski, Kaufman and Kerschberg, 1991) that "performed a sophisticated data analysis", KITTEN (Shaw and Gaines, 1987) where "the knowledge acquired is being feedback to facilitate the intelli-

gence of people", MOCA (Fall, Crawford, Souders and Rabin, 1989) that "helps analysts to cope with large quantities of intelligence data" and SADC (Moulin and Rousseau, 1992). A few of them pretend to perform automated analysis, such as NLDB and SKIS, but they require the user to build lexicons (NLDB) or well defined sets of criteria (keyword->category correspondences in SKIS) adapted to the user purpose and to the considered domain.

Texts in natural language format serve as entries in KITTEN, NLDB, SADC and SKIS. Corpora are used in various ways, from "a set of entities that are sentences whose features are words they contain" (IMS project, Gaines and Shaw, 1994) to sequences of keywords (SKIS). The ALCESTE segmentation method is close to the one used in KITTEN with the constraint of the ECU fixed length. Statistical treatments mostly use term weighing. Some algorithms, such as TEXAN in KITTEN or the "similarity measuring component" in SKIS implement a distance measurement too. The only algorithm with a pure distance measurement is TEXAN. But it is a "simple distance-in-text measure" that lacks to take into account the whole corpus. Crossed entries of non-noisy words and ECU in the matrix of the presented top-down hierarchical classification allow a measurement based on a $\chi 2$ distance. It meets the requirements proposed in MOCA, as "it does not need to input the number of clusters desired", it allows "overlapping cluster" on non-noisy words and it analyses the corpus as a whole to produce clusters.

The intended results of discovering the corpus structure can be compared to the one of the text logical structure as in SADC method, despite that this latter uses a purely conceptual approach. The work of Hearst (Hearst, 1994a et 1994b) adresses this goal too. We must take into consideration this latter approach as its goal is nearly the same as ours, but with some differences, both in the method used and in the results obtained. Her goal was to "partition expository texts into coherent multi-paragraph discourse units which reflect the subtopic structure of the text". This is the first difference with our method, as expository texts are a bit more structured than texts from expert interviews. She uses a "texttiling" algorithm based on similarity determination. She processes a corpus segmentation into "token-sequences" of a fixed length. A token-sequence includes a predetermined number of consecutive tokens (~20) and it is similar to ECU in our method. Then she uses a window including a fixed number of token-sequence (~6) to perform "comparison of adjacent pairs". This algorithm performs a *local* analysis of the corpus syntax. It only gives similarity indications without any other detail than the

token occurrence frequency. She does not seem to use the pre-existent structure of the corpus.

The algorithm we presented in this paper performs an overall analysis and a correspondence between text units. We think it is more suitable for ill-structured text, such as interviews, than Hearst's texttiling algorithm. With our method, we obtain more than the overall corpus structure. For instance, detailed indications about term distribution and correlation enable us to tag the structure and can be used later for other purposes, such as ontology building.

To conclude, the presented method and algorithm meet previous work. They are candidate to be a "more sophisticated text analysis technique" (Shaw and Gaines, 1987), but till now with a human assistance. The corpus organization (ICU), its segmentation (ECU) and the clustering algorithm (cf. section 2.3) contribute to enhance and refine the set of statistical tools and methods commonly used ; then the overall extension reaches a useful feature to help the knowledge acquisition process. We detail further possible extensions in the next section.

5 Conclusion and Further Research

In this paper we have detailed the basis of the method to enlighten both the structures of corpora and the subjects of their parts. In the last section we have briefly evoked terminological issues. The first remark is about the typing of classes and the parts of the structure. The user can perhaps be helped in the determination of a type grid, but it is not an advantage. As the experts' conversations and interviews are very spontaneous, a too constraining tool can lead to bad results. Anyway, the knowledge engineer must have to read those texts and to understand them. Our method results in a spare of time and facilitates management of a huge amount of expertise texts. As the class typing is only a means to check the class analysis correctness and to guide the knowledge engineer in his reading, no refinement such as the use of a predefined nomenclature seems to be useful to improve this method. With the help of a statistical learning method (Charniak, 1993), we can expect to gain some automated extension to perform the class tagging. No research has been done yet in this way, as far as we know.

To identify specific vocabulary, terminologies of domains and subdomains, our method can be useful. The tool already allows the research of terms

related to a given one, without class consideration. Combined with the multiple text corpus analysis, we hope to find a way to help the knowledge engineer to establish ontologies from a given domain or expert. The use of decision tree algorithms as in CART (Crawford, 1989) can be a way to help exploitation of the big amount of produced data.

So, the use of term correspondence analysis by the means of the top-down hierarchical classification provides useful results to help the knowledge engineer to manage sources of expertise. Implicit structures of single texts and differences inside a set of texts can be discovered. It guides the expert in his work, which becomes quicker and safer than reading them without help. The method to exploit a single text is well established and the multiple text corpus exploitation seems promising. One interesting way we investigate concerns the characterization of domain terminologies, according to a given expert or not. An interesting work in this direction was made by D. Bourigault with the help of the tool LEXTER (Bourigault, 1995), a terminology extraction software. It mainly produces an hypertext version of the discovered domain ontology. As it uses a morpho-syntactical analysis to build a grammatical network of term relevant to the domain, it is a conceptual approach, opposed to the statistic one.

Acknowledgments

All examples come from a research supported by the "Ministère de la Recherche et de l'Espace" under contract n° 92 C 0757 and the "Ministère de l'Equipement, des Transports et du Tourisme" under contract n° 93.0003. Special thanks to Max Reinert and Krystel Amerge who lead me to the text statistical analysis, especially with the help of the ALCESTE program. Thanks to Rose Dieng and Olivier Corby who have encouraged me in this work. Last but not least, thanks to the experts of INRETS, on the expertise of whom our analysis relied.

References

Bourigault, D. (1995). LEXTER, a Terminology Extraction Software for Knowledge Acquisition from Texts. Proceedings of the 9th Knowledge Acquisition for Knowledge Based Systems Workshop, Banff, Canada, 1995.

Benzecri, J.P. (1973). L'analyse des correspondances, in L'Analyse des Données, tome II, Paris: Dunod.

Charniak, E. (1993). Statistical language learning. Bradford books, Cambridge: The MIT Press.

Crawford, S.L. (1989). Extensions to the CART algorithm. International Journal of Man-Machine Studies, Vol. 31, pp.197–217.

Fall, S.K., Crawford, T.C., Souders, S.L. and Rabin, M.J. (1989). Automated knowledge acquisition techniques for intelligence analysts. In M. Mohan (Eds), Applications of Artificial Intelligence VII, Vol. 1095 of SPIE, pp. 66–77.

Gaines, B.R. and Shaw, M.L.G. (1994). Using knowledge acquisition and representation tools to support scientific communities. Proceedings of the Twelve National Conference on Artificial Intelligence, Vol.1, pp. 707–712, AAAI Press/The MIT Press.

Hearst, M.A. (1994a). Context and structure in automated fulltext information access. Report no. UCB/CSD-94/836, Computer Science Division, University of California, Berkeley, California 94720.

Hearst, M.A. (1994b). Multi-paragraph segmentation of expository text, in Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, June 1994, Las Cruces, NM.

Hsu, L.S., Wu, Z.B. and Tan, C.L. (1992). A survey on statistical approaches to natural language processing. Technical Report TRA4/92, Department of Information Systems and Computer Science, National University of Singapore, Kent Ridge, Singapore 0511.

Jacobs, P.S. (1992). Joining statistics with NLP for text categorization. Proceedings of the Third Conference on Applied Natural Language Processing, pp. 178-185, Morristown: Assoc. Comput. Linguistics.

Jacobs, P.S. (1993). Using statistical methods to improve knowledge-based news categorization. IEEE Expert, Vol. 8, No. 2, pp. 13–23.

Kaufman, K.A., Michalski, R.S. and Kerschberg, L. (1991). Knowledge extraction from databases: design principles of the INLEN system. Proceedings of the 6th International Symposium on Methodology for Intelligent Systems, pp. 152-161, Berlin: Springer-Verlag.

Kendall M. and Stuart, A. (1967). Inference and Relationship. The advanced Theory of Statistics, Vol 2, London: Charles Griffin & Co Ltd.

Moulin B. and Rousseau D. (1992). Automated knowledge acquisition from regulatory texts. IEEE Expert, Vol. 7, No 5, pp. 27–35.

Register, M.S. and Kannan, N. (1992). A hybrid architecture for text classification. Proceedings of the Fourth International Conference on Tools with Artificial Intelligence, pp. 286-292, Los Alamitos: IEEE Comput. Soc. Press.

Reinert, M. (1979). Classification descendante hiérarchique pour l'analyse de contenu et traitement statistique de corpus. PhD thesis, Université Pierre et Marie Curie, Paris 6, Paris.

Reinert M. (1992). Notice du logiciel Alceste, Version 2.0.

Shaw M.L.G. and Gaines B.R. (1987). KITTEN: Knowledge Initiation and Transfer Tools for Experts and Novices, in International Journal of Man-Machine Studies, Vol 27, pp. 251–280.

Sowa, J.F (1984). Conceptual Structures, Information Processing in Mind and Machine. Addison-Wesley System Programming Series.

Unité de recherche INRIA Lorraine, technopôle de Nancy-Brabois, 615 rue du jardin botanique, BP 101, 54600 VILLERS-LÈS-NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, domaine de Voluceau, Rocquencourt, BP 105, LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur

Inria, Domaine de Voluceau, Rocquencourt, BP 105 LE CHESNAY Cedex (France)

ISSN 0249-6399