



HAL
open science

Application of clustering in a system of query reformulation. Presentation of Saros

Nicolas Turenne, Francois Rousselot

► **To cite this version:**

Nicolas Turenne, Francois Rousselot. Application of clustering in a system of query reformulation. Presentation of Saros. Workshop on Knowledge Acquisition, Modeling and Management, Apr 1998, Banff, Canada. hal-03373962

HAL Id: hal-03373962

<https://hal.science/hal-03373962>

Submitted on 11 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Application of clustering in a system of query reformulation
Presentation of Saros**

Nicolas Turenne François Rousselot

ERIC (Equipe de Recherche en Ingénierie des Connaissances)
Laboratoire d'Informatique et d'Intelligence Artificielle (LIIA)
Ensaïs/Univ. Louis-Pasteur
24 Bld de la Victoire 67000 Strasbourg
tel:(33)3-88-14-47-53
e-mail {turenne,rousse}@eric.u-strasbg.fr
WWW: <http://www-ensais.u-strasbg.fr/ERIC>

Abstract: A huge quantity of information is available owing to increasing computerization and network access (internet, intranet). We enjoy to this information necessitates our making some quality extraction out of all this documentary information. One of the problems concerns managing of extraction so that the querying user obtains the most relevant information according to his needs. The system presented in this paper is made up from an internet server of document indexation. It orientates itself towards a solution, based on groups of compound words of a frequency greater than two. This solution is a cluster-based reformulation generating a semantic context more homogeneous than simple words. The compound words are extracted directly and dynamically from the whole base of document and are offered as clusters. The system is tailored to provide access in quasi real-time.

Keywords: clustering; information filtering; repeated segments; query reformulation; web search tool

Introduction

Searching for documents on the web often ends up with an excessive selection of texts answers to a given question. One tends, indeed, to alleviate this drawback by giving the user the possibility of specifying his query through reformulation. This project is designed for public information server for small firms and industries at ADIT (Agency for Technology Information Dissemination). A new technique of reformulation is outlined in this paper. The server consists of surveying known URLs on internet. Nevertheless the reformulation technique is also adaptable to a blind server such as AltaVista.

In this paper we present a new system of query reformulation named Saros (help system to reformulation by successive operations). The processing of the query by an information extractor confronts us with several complex linguistic problems: synonymy, polysemy, variations to which may be added the pragmatic problem of indecision on the part of the user in comparing to the formulation of his information need.

This solution, as optimized for the French language, proposes, firstly, a lexical series of simple and compound words made up carefully such a way as to regroup them into cluster features. Secondly, these suggested words and terms are clickable. They call clusters of associated terms in a manner to cluster meaning. These clustered terms evolve also from the base. Some terms taking in these clusters may be added to the final query according to the user's model. The solution is similar, for instance, to the one implemented by AltaVista, with the difference that list of terms is calculated with the content base and not by consulting a hand-made thesaurus. More over, Saros uses repeated segment and simple words.

We here with, present, the server architecture composed of 9 elements: namely, a crawler or spider used to gather web pages judged interesting, a management program of the base of documents, a module giving information access, an extractor of nominal syntagm and simple word, a box of initial dialogue, a classifier, an interface of dialogue user, an extractor of document (i.e indexation engine) for displaying results. The architecture controls the flow of dynamic information coming from the web, and offers an help reformulation system, also dynamically in accord with the base. This help reformulation indicates a way in which to suggest elements of a theme related to a general initial query. This rough thematisation is realised by collecting single nouns and compound frequent expressions of the base. This thematisation ought to permit initial user query to be refined in towards the final query.

We will present stages of the method. The stages are twofold. The one is a preprocessing which fully indexes documents, extracts terms (i.e repeated segments and discriminant words) and clusters terms. The second is a real-time processing of the initial user query which calls lexical series from which the user accesses to a cluster and the adjacent ones. From these clusters the user chooses some items of the clusters

for adding them to the query which becomes a final query processed by a full indexer to extract documents classified by relevance and by site.

1 The state-of-the-art of query formulation

The problem of reformulation is well known in the realm of electronic document management (EDM). In EDM one emphasizes the importance of the query which will express the user's information need. The extraction of the document is sensitive to the processing of initial query. Consequently the qualitative extraction is as much high as an implemented linguistic processing is sharp. One finds 2 common types of reformulation: the first consisting of using a thesaurus revealing synonymic links between terms. This scheme can be found in most classic document extraction engines (Fulcrum^a, Search97^a, PLS^a, TextFinder^a, Darwin^a,...). Hyperindex system constructs structures of a tree of words detected in the titles of web pages and on which one can browse to refine the query [Bruza & Dennis, 1997].

The second type is the relevance feedback, in other words, from a simple query the user will receive some documents. He will choose those which seem close to his search, these will serve as reformulation for extracting semantically similar documents. This type concerns the Spirit^a system. A finer method consists of creating an index base thanks to selected documents. In this case reformulation learning is developed vis-a-vis the user model. A neural network method using backpropagation has been implemented in the Mercure system [Boughanem & Soulé-Dupuy, 1997].

The certain system of reformulation closest is the proposed one on the web called AltaVista's Livetopics^a. This technique is based on the extraction of meta-keywords coming from collected html pages. The html pages writers are informed of the mean of taking part in correct development of these keywords tags. A routine program collects the keywords to create lists. This technique betrays the following drawbacks: it is not drawn up according to the base, propositions concern only uniterms (i.e simple words) often polysemic and finally headings proposed are relatively irrelevant to the content. We use this system to compare the efficiency of the clustering offered by Saros. The results are presented in the 4th part. On the web one also discovers a search engine, Askjeeves, which bases itself on lists beforehand hand conceived. In this type of conception the calculable risk is to never come across the documents admitting the terms of the query. A server with the PacProspector tool offers an effective processing by local grammar of the query, but this processing is not in real-time and does not explore the extent of the query. At the Institut du Multimedia of Pôle Universitaire Leonard-de-Vinci (Paris), one finds a product not applied on web called Sémiomap^a which suggests a similar approach to Saros though access to information is directed by selecting one or two terms from a cluster of terms. The user selects repeated segments from clusters to gain access to documents of an intranet base of documents. Sémiomap^a does not guarantee real-time processing.

In our application of conceptual clustering with simple words and repeated segments, we endeavour to approach the problem of the query reformulation by thematisation of the document base. Association of words by co-occurrence permits a revelation of interest in the conceptual classification of terms of natural language [Mikheev & Finch, 1992][Smadja, 1990][Hearst, 1994]. Clustering achieves a knowledge structuring. The aim is to suggest clusters as concepts. Conceptual clustering opened to several modules is essentially based on pure mathematical methods. Some persons proposed several interesting solutions but often heavy [Michalski, 1990][Carpineto & Romano, 1996][Fisher & Schlimmer 1997]. A search engine, SQLET, implements a clustering of terms based on series word+adjective or word+adverb [Grefenstette, 1997]. Another engine Inquiry realizes an expansion of the query, giving consideration, on the one hand, to varying forms of selecting words beginning with 3 or 4 first letters, one the other hand, considering cooccurrent words with frequency higher than a threshold [Croft, 1995]. The two systems previously quoted process the information database for creating groups of words (i.e terms classes) but without forming repeated segments. SQLET offers classes to the user but families are too incomplete by their structure while Inquiry takes account of direct possible expansions in the query, creating a considerable potential for the retrieval of documents.

The application presented in the paper offers a dynamic query model to the user, being compared with the content of the document base. The architecture presented in figure.1 shows that several softwares to get a fast operability of the system. They have been used in a resources integration fashion so as to achieve a component architecture. In the following commentary we shall present each module which contributes to the functionalities of the architecture, from the beginning to the end of the chain of the information processing.

2 Preprocessing

2.1 Pages collecting

In English a crawler or spider, is a robot (or agent) which will identify and collect html pages on http protocol. In this case, the parametrizing consists of specifying homepages addresses of the sites to visit and the depth of node to go through per site.

All in all 120 French sites are specified together with the integrality of the page to collect. The whole documents amount to 200,000 that is 402 Megaoctets of textual data. The documents are gathered in a base which is next to be processed. The crawler used is called Ecila^a [Dachary, 1996].

The crawling time is around 5 hours on a Sun Sparc station. The crawling is realized every 7 days, with modification identification of html page edition before collecting them.

Note: The temporary address of the application is: <http://195.101.154.68> (userid: *angers*, pass: *jupiter*), it would become <http://www.cyrano.gouv> in next time

2.2 Extraction of the terms

The next step is extraction of terms. This technique is similar to one in the extractor called Lexter [Bourigault, 1992]. Method used is the "bounds" one: a list of words is developed in a given language, in our case the French language. This list of words builds up to a dictionary of "bounds". This dictionary possesses French stop words such as verbs (to go, to finish...) and conjunctions/pronouns (and, or, when...).

The nominal syntagms are collected in the way that, their locating is carried forward from one known or not "bound" word to another known or not "bound" word (or a punctuation), in agreement that no stop word be a part of the nominal syntagm except the articles. The dictionary of "bound" words has 45,000 words and their derivations (gender, number, verbal declentions). One may parametrize the maximum number of words of a syntagm, in our case we chose 6. For instance in the following sentence "the goal of the exploratory statistics is to put to light the properties of a sample"; a first syntagm begins from "goal" to "statistics" and a second beginning after "to put to light", is "properties of a sample". For greater convenience we use Genet^a to achieve extraction [Constant, 1996].

Another location deals with simple words that deviate the most. One defines deviance (D) as the difference between the word frequency in the document and the frequency in common language. For a given word its deviance higher than zero means it is semantically interesting regards the document; we call deviating words the words having a deviance higher than a threshold. A dictionary of frequencies of the common language for 25,000 words permits this calculus. If a word does not appear in the dictionary its frequency in the common language is considered as to be zero.

It is not sure that this criterion is the best way to define the discrimination nature of simple words. It is clear that a nominal syntagm will be less ambiguous than a simple word, and a sentence will be less ambiguous than a nominal syntagm. That is why one needs to be careful with simple words when being considered a longside statistical criterions, because they are very polysemic. Simple words make many co-occurrences and take up memory. It imposes heavy constraint to include simple words in clusters.

Within the framework of the document extraction it would be interesting to test the relevance of the distribution of frequent (or averagingly frequent) simple words on a low number of documents to weigh their discriminant capacity. In any case, the calculus of deviance needs a priori to know the frequency of the word in the language. It limits action of this technique regarding technical words, which are nevertheless very interesting because they are discriminatory and not very polysemic. In Saros the admission of simple word and repeated segment is not subject to review.

One can parametrize the number of simple words to take into account on each document and the quality of extraction of repeated segment by the considered "bound" word. No manual erasing a posteriori is realized after clustering. Obviously, some noise is created by certain simple words which are too generic and repeated segments which do not carry much meaning. Solely methods improving the extraction of terms [Rousselot & Frath 1996] and the clustering technique can optimize the noise.

2.3 Creation of the clusters

The following step permits us to associate terms with one another. Since [Michalski, 1980] computational linguistics is enriched with an efficient computerized conceptual clustering approach so as to create classes of terms. The term clustering for which terms often appear together is sensible in representing a similar semantic context.

The technique takes terms found previously in input to calculate their co-occurrence pairwise, and to form clusters. Thus a cluster is constituted as a result of repeated segments and simple words mutually associated. In the ideal case considered by Saros, one supposes that the environment of these detected associations reveals semantic proximity, or at least a similar semantic context concerning the members of a cluster. In fact, it would of course need to take into account contextual factors and particularly linguistic ones to form a real semantic correlation between several terms. A morphological linguistic processing will react to variation to avoid redundance; for instance "custom office" and "office of the customer" will be considered identically as they are variants of a same representation. Singular terms will be bring back in their plural equivalent. Some ambiguous terms are manually classified in a dictionary, for example "pari"

(i.e bet) and "Paris" (i.e capital of France), in order to take them into consideration in the clusters without distorting them.

To estimate a co-occurrence one draws up a unique window for calculus, which extends the paragraph being located by break lines. If a break line is not identified one "bounds" the zone to 500 characters. If two terms are detected in a zone one considers that there is co-occurrence. One leads to a matrix of co-occurrence in which line space (i) is the same as column one (j): the simple and compound words (around 300,000). The common term of the matrix (ij) is the co-occurrence value in the base of documents between the i and j terms.

An association parameter from [Michelet, 1988] is used to quantify force of association between two terms, in fact it has the meaning of a correlation normalized: $E_{ij} = C_{ij}^2 / (Freq_i * Freq_j)$

(1)

where C_{ij} is co-occurrence number between i and j, $Freq_i$ is the frequency of term i, and $Freq_j$ is the frequency of term j in the base of documents.

Some parameters set the characteristics of a cluster as at the maximum number of elements of a cluster and the maximum number of external link to a cluster. One fixes the maximum number of elements of a cluster as between 12 and 15. The choice is empirical by seeing with many runs of the system the semantic quality of clusters. One observes that the semantics of the clusters deteriorate by themselves when the number of elements is too small (<5) or too large (>20). The setting of all clusters is organized into a lattice, so that the number of external links, which link other clusters, is fixed at 10 so as not to drown the user in a flow of adjacent clusters. For the clusteiring component we integrated an existing module (Sampler^a) [Jouve, 1997].

The calculus of clusters in the base of documents, drawn up with an automaton as explained below, is done in 1h20 for the entire base on a Sun Sparc station. This time includes the extraction time of the repeated segments and the clustering time. The clustering lasts around 40 minutes. This calculus is done periodically and fully with the crawler every 7 days so as to take into account any base updates, its purpose being to supply conceptual survey of watch documents. Indeed, detected clusters correspond to the most relevant ones.

2.4 Access to the terms

For access to terms the clustering module uses an automaton which permits to effect an alphabetic sorting for compressing base of terms. The speed of access to terms is of the nature of 160,000 words per second on a Sun Sparc station 20. This automaton is a finite state compiled automaton which, owing to its rapidity, permits us to accelerate calculus of clusters 10 times, compared to a classic algorithm [Constant, 1996]. An automaton is an optimized decision tree. It authorizes choices to be made in data series particularly alphanumeric characters. The automata originally used for compiling are nowadays spread out more in the field of natural language processing. We integrated a useful automaton module Genau^a which achieves this fast access. The automaton is also used by an interface of the user dialogue to call up lexical series. This automaton has the advantage of allowing a high-standard processing speed in the search for information, and needs small memory space for data storage and quick compiling. Steps are as follows: effectuate joining of words succeeded by an index (for instance :s for a stop word or:d for a determiner) in a file managed by dynamic automaton. A fast automaton is created from the previous one by minimization. The memory is managed dynamically, segmented and then reentering without reallocations. Use of such an automaton allows us to ultimately require a quasi-real-time system. The access times to expressions and clusters (corresponding to different steps of reformulation) are less than 15 seconds.

3 Processing and user view

The processing is done in five steps. The first step consists in capturing an initial user query written in a dialogue box. The second step proposes lexical series to the user linked with words of initial query. The third step permits to access to the cluster for which an element of the lexical series is member. The fourth step consists in adding to the final query elements of one or several adjacent displayed clusters. The last step is to submit the final query to retrieve relevant documents.

3.1 initial query

The user has to explicit a wish of document search through posing an initial query as for example "artificial intelligence" (fig 2).

The set of words of the base is:

$$B = \{m_i / i \in \mathbb{N}\} \quad (2)$$

The set of words of a document d is:

$$D = \{m_i / i \in I \subset \mathbb{N}\} \quad (3)$$

The set of expressions containing n words is:

$$\Pi_n = \{p_n = (m_{i_n}) / n < 6, (i_n) \in \text{IN} \wedge m_{i_n} \in B\} \quad (4)$$

The query of m expressions is:

$$Q_m = \{(q_m) / i \in \text{IN} \forall m \in \text{IN} q_i \in \Pi_n\} \quad (5)$$

and the initial query:

$$Q_m^{\text{in}} = \{(q_m^{\text{in}}) / \forall i \forall m \in \text{IN}, q_i^{\text{in}} \in \Pi_1\} \quad (6)$$

As the base collects French servers only, the base is composed of documents in French, therefore the query has to be written in French. Sometimes English terms are found in the lexical series under proposal because a server may contain several pages deliberately written in English by the owner of the French web server.

Often there is question of technical reports having been written in French. Therefore, to improve the quality of the clusters containing English terms, English "bound" words have to be adjoined to the automaton.

The formulation of the initial query is expressed intelligently in free natural language. Verbs are not processed; neither in the infinitive nor conjugated. Stop words (articles, prepositions...) are ignored. In the case of the previous example, the system will take into account the items "artificial" and "intelligence" to effect a grouping of features about these items.

3.2 lexical series

By the action of the automaton on the list of clusters, the system scans the lexical series concerning "artificial" and "intelligence" taking into account the minimum of morphological rules (substantives/adjectives inflexions, number, gender). These rules cover 25,000 words of the French, but are not applicable to every new word; even if it be of foreign origin. The technique is based on the detection of suffixes. A dictionary of 260 suffixes is generated by the user having a knowledge of the language. A routine program will identify for each word of the base the number of character to take off at the end of the word, identifying the end of the word with a suffix. For instance, if "tical" is a suffix, you will get "political: 5" (5 being the number of character to retrieve of the word to get the root). The root become "poli". Every word having the same root is grouped into the same family. The word of the query is processed in the same way so that its root selects a family if existing.

The lexical family for the root q_i of q_i is:

$$F(q_i) = \{(f_i) / \forall i \in \text{IN}, q_i \tilde{\subset} f_i\} \quad (7)$$

Taking the word "office" one gets the following lexical family: "office", "officer", "offices" and "official". The system searches for expressions (simple or compound words) finding, first of all, the whole of the initial query words or of expressions which have one word less, etc... This search is made, of course, from the expressions of the clusters. A screen presents the detected lexical series to the user.

3.3 navigation in clusters

The third step consists of navigating closely on the clusters semantically through the initial query expressed and in selecting some of their elements. The succession of selected elements builds the final query. The user constructs his model by comparison with the base content and then becomes the activator of the reformulation .

Each expression of the lexical series is an hyperlink which opens out to the corresponding cluster to which it belongs. The user can open up a required cluster. A multiframe window appears with, on the left, a navigation history and, on the right, a table containing elements of the cluster of associated terms. Below the table external links of this cluster are displayed. A nether frame renders the final query visible by operating its search button. For each expression of the table of a displayed cluster, a check box permits the user to add or take off the expression to the final query. Some external links to a cluster are suggested ensuring a navigation in the lattice, which means navigation in the neighbouring clusters of the displayed cluster. For a displayed cluster, as soon as some required expressions in the query are checked, a validation button will add them to the final query.

The set of elements of a cluster is:

$$G = \{(g_i) / \forall i \forall j \in \text{IN}, E_{ij}(g_i, g_j) > \text{Threshold}\} \quad (8)$$

one define adjacent clusters of G_1 by:

$$\Gamma(G_1) = \{(G_i) / g' = g / \exists g' \in G_2 \exists g \in G_1 g' = g\}, \quad (9)$$

on a $G_1 \in \Gamma$

3.4 final query processing

The system processes the query as following, the final query is a chain weighting of two sub-queries:
 -the first, being of weight at 100 associates, in chaining them, the query items in forcing them to appear in the neighbourhood of a maximum of 50 words each other (function <near> of the indexation engine)
 -the other, weighted at 50 associates, in chaining every item linked by a Boolean-or (function <or> of the indexation engine). The repeated segments are included as a frozen string between inverted commas.

The set of expressions of a reformulated query is: $\forall m \forall i \forall j \in \mathbb{N}$

$$Q_{m+n}^{fi} = \{(q_m^{in}, (g_{in})) / n < 6, q_m^{in} \in Q_m^{in} \wedge f_j^{in} \in F^{in}(q_j^{in}) \wedge (f_j^{in} = g_{jn}) \in G \wedge (g_{in}) \in \Gamma(G)\} \quad (10)$$

In fact, in the case where the first sub-query should lead to a empty result (i.e no documents), the second one ought to lead to only one item present in a document of the result with a low relevance value. In order to get round this problem, it would be of interest to produce combinations between n and two elements in a chain (n being the number of items in the final query). The query would be a weighted multichain made up of the combinations 2,3,4... elements.

Numbers of combinations (between n and 2 elements) rising exponentially to

$2^n - (n+1)$ one would have to arrest at n=8 in the case of 9 items or more in the final query, one divides n by 2 to calculate all combinations of each sub-set. The final query for n>8 would be the sum of all combinations of each sub-set. Within this combinatory consideration one would be sure to have at least 2 items, or even more, from the final query in thus proposed documents after their extraction.

3.5 Extraction of documents

A document extraction module (i.e indexation engine) is used to reach the final query and to give back the relevant documents. The technique employed is full indexing. In accordance with the syntax of the query, should each item of the query appear in a document, it will be displayed first at the head of the results by a ranking method which takes into account statistical weight of item of the query in a document.

The set of relevant documents is:

$$\mathfrak{R} = \{d / D.Q_{m+n}^{fi} \neq 0\} \quad (11)$$

The list of results are displayed sorted and clustered by site. In the most relevant 200 documents answering to the final query, one presents the sites displaying the best ranked document of each relevant site. This formal clustering ensures avoidance of an excessive pollution in the results from all the relevant pages of a particular site. A final query will, on average come up with 5 to 15 sites, each containing the most relevant document presented. The search engine on web, Ifind^a, admits this approach of document clustering by site. The indexation is managed by Search97^a in the architecture.

Finally the user can click on a button to restart the final query on the list of results of the site and consult 1 to 50 documents relevant tot it answering the final query.

4 Limitations and evaluation of the system

4.1 limitations

The simplicity and efficiency of the choosen methods have although some drawbacks. The linguistic processing should be improved. The homogeneity of the semantic clusters depends, of course, on the quality of repeated segments extraction. In the actual method a fitting work can to be considered. For instance stop word such as "cédex" or words of foreign language does not have to appear in a cluster. Actually this fitting is realized by a manual examination of the clusters. A modifiable list of stop and "bound" words managed by the automaton permits us to correct this noise. It takes into account of stop words so as to avoid their appearance in expressions. Expressions are cleant in the following way: a list of expressions making the clusters is accessible to the administrator of the application in an ascii file. It is sufficient to locate stop words included in expressions (like cédex in "Paris cédex") and then add them to the list managed by the automaton. But it will not eliminate expressions making little sense; one will still have noise.

A sharp linguistic processing of the query is not implemented although it should be. For example they take no syntactic consideration in the actual processing: "homme grand" (big man) and "grand homme" (famous man) will call up the same lexical family. Expressions in the final query are not optimally linked each other to ensure a good weight of them in the document. The problem is that relying on a complex linguistic analysis should lead to loss of the real-time faculty of the system.

A more efficient method of statistico-semantic conceptual clustering should be implemented rather than the co-word analysis actually used. It would return more homogeneous and efficient clusters with more relation between conceptual classes and real world data.

A last point concerns ergonomics which may be improved. Effectively there are too many steps from initial query to the final query. The system should increase its user-friendly property in reducing number of steps.

Our team projects to develop a new module of repeated segments extraction and a new conceptual clustering technique working on large corpora.

4.2 evaluation

Evaluation is difficult in our case because the crawler gathers pages automatically and we can not impose a content for retrieving. So it is impossible to calculate precision and recall without knowing the exact number of relevant pages for a given query among the 200,000 pages contained in the base. An absolute evaluation being hard to develop we aim to consider a comparison method more flexible towards a lack of parameter. The comparison is principally qualitative based on the number of proposed documents for a query and the content of the first relevant pages.

One evaluates the system by taking into account on the one hand the relevance of the clusters proposed for reformulation on the other hand the quality of results according to the final reformulated query. Firstly, we compare efficiency the clusters with the AltaVista's Livetopics ones for a same query. Secondly, a classical box of user dialogue is also inclusive of our server. This dialogue box is based on a Boolean formulation without reformulation. It is used to compare results from a query with and without Saros reformulation for a same information need.

First test: "classification conceptuelle" (conceptual classification)

1-without reformulation:

the phrase taken in commas contains all the words ("*classification conceptuelle*")

the query gathers one server (*inria*) and 16 documents

user ask *classification et conceptuelle* without reformulation (conceptual and classification). We get 9 servers (*inrp*, *inria*, *equipement*, *ac-montpellier*, *ac-lyon*, *senat*, *onf*, *inrets* and *inra*) with a total of 450 documents

2-with reformulation:

the phrase is *classification conceptuelle* without commas

in the lexical series proposed the user chooses "*classification automatique*" (automatic classification), the correspondent cluster is:

<i>question ouverte</i>	(open question)
<i>analyse des réponses</i>	(answers analysis)
<i>inconvenients des lignes</i>	(line drawbacks)
<i>classification automatique</i>	(automatic classification)
<i>réponses orales de commerçant blot</i>	(oral answer of blot merchant)
<i>dispositif de gestion performante</i>	(efficient management system)

the user explore 7 others adjacent clusters and among them: "*relation de service*" (service relation)

<i>double contrainte</i>	(double constraint)
<i>relation de service</i>	(service relation)
<i>analyse sémantique</i>	(semantic analysis)
<i>analyse logistique</i>	(logistic analysis)
<i>variables qualitatives</i>	(qualitative variable)
<i>diagrammes batons</i>	(chart diagram)
<i>caractéristiques de classes</i>	(class characterisation)

The user insert: automatic classification, semantic analysis and class characterisation

The user obtain 7 servers (*inrp* with 50 documents, *edf* with 50 documents, *inria* with 50 documents, *inist* with 15 documents, *cnet* with 50 documents, *senat* with 50 documents and *meteo* with 19 documents) gathering a total of 279 documents

3-On AltaVista

with the aid of refining of the initial query: *conceptual and classification* the user obtain the following 20 clusters:

- 1 50% *conceptual, classification, concepts, framework, concept*
- 2 20% *prerequisite, prerequisites, instructor, credits, consent, corequisite, permission, corequisites*
- 3 19% *semantic, retrieval, indexing, thesauri, thesaurus, acm, sigir, salton*
- 4 16% *reasoning, logics, nonmonotonic, monotonic*
- 5 14% *cognitive, ...*
- 6 13% *knowledge, ...*
- 7 12% *theories, ...*

- 8 11% *modelling*, ...
- 9 11% *semantics*, ...
- 10 11% *artificial,intelligence*,...
- 11 11% *inference*,
- 12 11% *spatial*,...
- 13 10% *classifying*,...
- 14 10% *methodological*,...
- 15 10% *perceptual*,...
- 16 10% *modeling*,...
- 17 10% *causal*,...
- 18 10% *neural*,...
- 19 9% *clustering*,...
- 20 9% *organizational*,...

If the user select two clusters:1 + 3, we get 1681 documents!

If user select six clusters: 1+3+5+6+9+19, we get 668 documents among which the 10 first documents are bibliographies writing about conceptual structures but not especially conceptual classification.

In the first test the classical formulation with phrase admits a very low recall. Using words and Boolean operators this formulation has a very low precision. The reformulation technique gets an intermediate position between the two precedent. The Livetopics clusters contain typical themes of the query. They seem interesting for the user because closely linked with concept and classification. The clusters contain simple words which may be polysemic; and the choice of one cluster imposes the validation of all elements of the cluster, so the problem observed is the same that for the simple formulation: low precision (if one or two clusters are selected) or low recall (if some clusters are selected).

Second test: "exonération d'impôt sur les bénéfices" (tax exoneration on profit)

1-without reformulation:

the phrase taken in commas contains all the words ("*exonération d'impôt sur les bénéfices*").

The query gathers two servers (*senat* with 3 documents and *ance* with 1 document) with a total of 4 documents

user asks (*exoneration avec impôt*) et *bénéfice* ((*tax with exoneration*) and *profit*). He gets 4 servers (*senat* with 50 documents, *ance* with 10 documents, *cité* with 5 documents, and *finance* with 50 documents) with a total of 115 documents

2-with reformulation:

the phrase is *exonération d'impôt sur les bénéfices* .

In the lexical series proposed, we choose "*exonération des bénéfices réalisés*" (exoneration of realized profit)

the correspondant cluster is:

<i>entreprise existante</i>	(existing entreprise)
<i>exonération temporaire</i>	(temporary exoneration)
<i>constitution de dossier</i>	(constitution of a file)

We explore 6 others adjacent clusters but none seems interesting.

We insert: existing entreprise, temporary exoneration and constitution of a file.

We obtain 5 servers (*senat* with 50 documents, *fiannces* with 50 documents, *cité* with 50 documents, *ance* with 50 documents and *dree* 50 documents) gathering 250 documents

3-On AltaVista

with the aid of refining of the intial query: *tax exoneration*, we obtain the following 9 clusters:

- 1 4% *psn*,...
- 2 4% *township*,...
- 3 4% *formalities*, ...
- 4 4% *cession*,...
- 5 4% *valorem,tax,profit*,...
- 6 4% *levies*, ...
- 7 2% *incertives*, ...
- 8 2% *enterprise*,...
- 9 2% *charter*,...

If we select three clusters: 3 + 5 + 8, we get 4 servers and 6 documents (*mincom.gov* with 2 documents, *bakerinfo* with 2 documents, *infonetsa* with 1 documents and *aexp.com* 1 document). All documents are relevant.

If we don't use the refining, the query: *+tax +exoneration +profit* , gives 195 documents; *mincom* appears at the 21th position and *bakerinfo* at the 25th one. In this case the 10 first documents seem to be relevant.

In the second test, observations are a little bit different. The simple formulation using phrase has again a low recall. However the reformulated query brings back more documents than the simple Boolean query (250 on to 115); The reformulation generates more noise and leads to a lower precision than classical formulation. We can note there the efficiency of the Boolean-with operator (i.e near operator) whereas in reformulation expressions are linked by a Boolean-or in the general case. AltaVista always proposes interesting clusters while sometimes not very homogeneous when for example tax and profit are together; it is the goal of the query of course but the semantic of tax is not the same than profit all firms know it well. The action of the clusters is heavily weighted and leads to a low recall of documents.

Conclusion

One had presented a new help system of query reformulation (Saros) on a web server. A parametrized crawler will periodically collect pages on predefined sites to create an information database. A reformulation system permits an extractor of document access to the relevant documents of the base.

The originality is twofold. Firstly, repeated segments contained in the base documents are proposed. The repeated segments are offered to the user according to a simple initial query. Secondly, repeated segments are structured into a lattice of clusters which summarizes conceptually the content of the base. A navigation around neighbour clusters of the first cluster called is possible.

Each element of our displayed cluster may be added to the final query. All the steps of reformulation as well as extraction are realized in a quasi real-time. The final query is a logical expression, making a search owing to a Boolean bi-chain (an "and" chain with an "or" chain with the elements of the final query).

Finally, a display of the relevant documents by site is proposed. Each document represents the most relevant document of a site compared to the final query. After that, the final query can be applied exclusively to a chosen site in visualising the list of results. While the clustering possesses limitations [Turenne & Rousselot, 1997], this technique brings very much to the thematic structuring of the knowledge thanks to the repeated segments, being distributed in several clusters.

The action of reformulation, compared to a classical formulation querying a phrase or words linked by Boolean operators, is situated intermediately between a low recall and a low precision position. The Livetopics method permits access to interesting clusters but the simple words generated and grouped for a global selection in a query lead to either a low recall of documents either a lot of noise gathering large documents (as bibliography) containing much words of the query. Our method consisting in extracting expressions from the base dynamically grouped in clusters to propose them to query permits to better correlate precision and recall of relevant documents. Expressions in our query reformulation are unfortunately not overlapped and not processed syntactically differently from a Boolean-or which is a source of noise.

According to consideration of a rapid development and functionality of the system we often used existing modules for the first implementation. An important part of the future work will aim in integrating a new module of repeated segments extraction and a new conceptual clustering module. In the actual clustering module the repeated segments are associated through statistical link and without processing their semantic property. We have to partition the query transmitted to the extractor into a more complex Boolean chain. This chain has to make documents gradually containing less and less query terms in accord with the decreasing ranking display. Finally we will try to imagine a more user-friendly interface to get the results.

Appendices

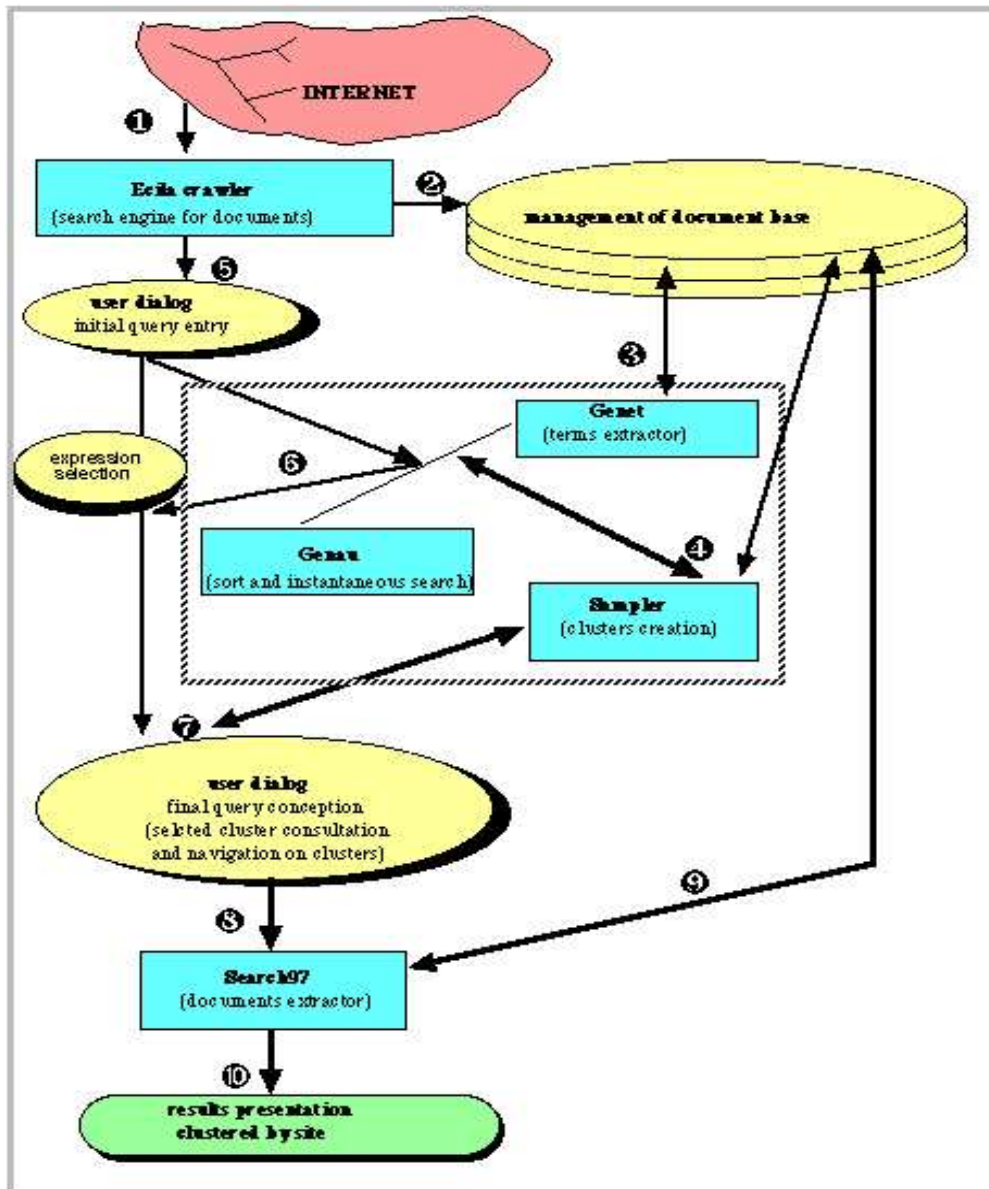


Figure.1

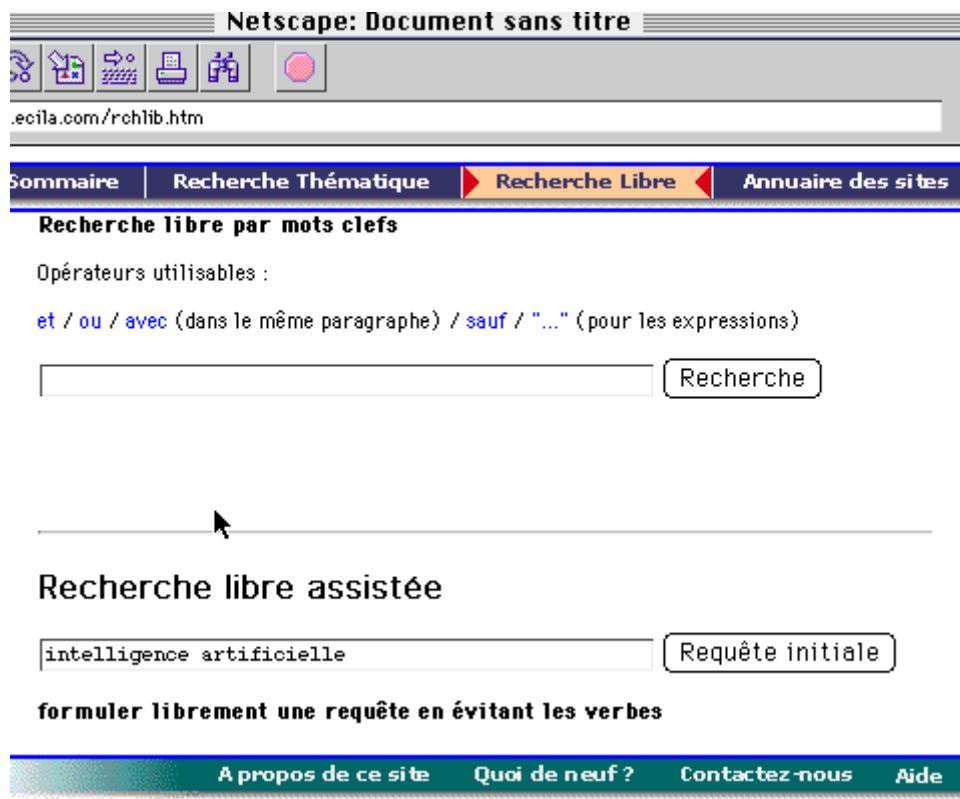


Figure.2

Expressions contenant le terme *intelligence artificielle*

Cliquer sur une expression ci-dessous pour faire apparaître les expressions associées


expressions contenant 2 mots de la question

- [intelligence artificielle distribuée](#)

expressions contenant 1 mot de la question

- [and in artificial intelligence](#)
- [artificial intelligence](#)
- [bonne intelligence](#)
- [distributed artificial intelligence](#)
- [intelligence au niveau](#)
- [intelligence du réseau](#)
- [intelligence économique](#)
- [logics for artificial intelligence](#)
- [machine intelligence](#)
- [pattern analysis and machine intelligence](#)

Figure.3

 [Sommaire](#)
[Recherche Thématique](#)
[Recherche Libre](#)
[Annuaire](#)

[Cliquer ici pour mettre à jour l'historique de l'aide à la formulation](#)

Cliquer sur une expression pour revenir en arrière

- [intelligence artificielle distribuée](#)

Expressions associées à *intelligence artificielle distribuée*

Ce tableau présente un groupe d'expressions fortement liées entre elles dans la base de documents

Expressions	Supprimer de la requête finale	Ajouter à la requête finale
prc-gdr intelligence	<input checked="" type="radio"/>	<input type="radio"/>
representation des connaissances	<input type="radio"/>	<input checked="" type="radio"/>
systemes multi-agents	<input checked="" type="radio"/>	<input type="radio"/>
intelligence artificielle distribuée	<input checked="" type="radio"/>	<input type="radio"/>
american journal of sociology	<input checked="" type="radio"/>	<input type="radio"/>
structuration des échanges	<input checked="" type="radio"/>	<input type="radio"/>
systeme multi-agents	<input checked="" type="radio"/>	<input type="radio"/>

ajouter les termes sélectionnés à la requête

Liens vers des groupes voisins de *intelligence artificielle distribuée*

- [interprétation de signaux](#)
- [anne boyer](#)
- [raisonnement temps réel](#)
- [informations temporelles](#)
- [univers multi-agents](#)

requête finale

intelligence artificielle "representation des conn

chercher les documents

Figure.4

References

[Boughanem & Soulé-Dupuy, 1997]

Boughanem,M & Soulé-Dupuy,C "Query modification based on relevance backpropagation" in RIAO'97 1997

[Bourigault, 1992]

Bourigault,D "Lexter, vers un outil linguistique d'aide à l'acquisition des connaissances" in 3rd journées d'acquisition des connaissances (JAC) France 1992

[Bruza & Dennis, 1997]

Bruza,P & Dennis,S "Query reformulation on the internet: empirical data and the Hyperindex search engine" in RIAO'97 1997

Knowledge and Acquisition Workshop (KAW) '98 Banff Canada

[Carpineto & Romano, 1996]

Carpineto,C & Romano,G "A lattice conceptual clustering system and its application to browsing retrieval" in Machine Learning 1996

[Constant, 1996]

Constant,P "Genet and Genau technical documentation" ed Systal 1996

[Croft, 1995]

Croft,W.B "Effective text retrieval based on combining evidence from the corpus and users" in IEEE Expert 1995

[Dachary, 1996]

Dachary,L "Ecila technical documentation" ed Ecila 1996

[Fisher & Schlimmer 1997]

Fisher,D & Schlimmer,J "Models of incremental concept learning: a coupled research proposal" on web 1997

[Grefenstette, 1997]

Grefenstette,G "SQLET: short query linguistic expansion techniques, palliating one-word queries by providing intermediate structure to text" in RIAO'97 1997

[Hearst, 1994]

Hearst,M "Contextualizing retrieval of full lenght documents" in a technical report by Rank Xerox 1994

[Jouve, 1996]

Jouve,O "Sampler technical documentation" ed Cisi 1996

[Michalski, 1980]

Michalski,R "Knowledge acquisition through conceptual clustering.A theoretical framework and algorithm for partitioning data conjunctive concepts analysis" in International journal of policy and informatics systems 4,219-243 1980

[Michelet, 1988]

Michelet,B "Association des mots" in PhD Univ of Paris VII 1988

[Mikheev & Finch, 1992]

Mikheev,A & Finch,S "Towards a workbench for acquisition of domain knowledge" 1992

[Smadja, 1990]

Smadja,F "Automatically extracting and representing collocations for language generation" 1990

[Turenne & Rousselot, 1997]

Turenne,N & Rousselot,F "Evaluation of four methods of clustering used in text mining" in a technical report by ERIC-LIIA 1997