PROGEN : a Genetic-Based Semi-automatic Hypertext Construction Tool - first steps and experiment

Georges Nault

Laboratoire d'Analyse Cognitive de l'Information (LANCI). University of Québec in Montréal, CP 8888, succ A, Montréal (Qc) H3C 3P8.
E.mail: nault@pluton.lanci.uqam.ca

Vincent Rialle

Laboratoire TIMC-IMAG UMR CNRS 5525, Faculté de Médecine de Grenoble, 38700 La Tronche FRANCE
E.mail : Vincent.Rialle@imag.fr

Jean-Guy Meunier

Laboratoire d'Analyse Cognitive de l'Information (LANCI). University of Québec in Montréal, CP 8888, succ A, Montréal (Qc) H3C 3P8.
E.mail:jean-guy.meunier@uqam..ca

Abstract

We present a method and an interactive software to assist in the creation of hypertext links in a text, whatever the topic of the text and the application domain may be. The method follows three steps : 1) preprocessing of the text for its segmentation and its "vectorization", 2) classification of segments of texts using ART, an unsupervised learning neural net, 3) research by genetic algorithm of the best associations between segments of text by using the classification provided in step 2 and the subjective choices expressed interactively by the user. Steps 1 and 2, carried out thanks to CONTERM software, are presented briefly. Step 3, implemented in PROGEN software, is developed and the results of a first experiment are presented. A discussion on the interest and the limits of the method are given in conclusion.

1 INTRODUCTION

The study presented here is at the crossroads of two great research fields : genetic algorithms (GA), and Computer-Assisted Reading and Analysis of Text (CARAT). CARAT is the computer technology that offers readers assistance in attaining some aspects of the informational or semiotic content of a text (e.g. discursive, lexical, hypertextual, thematic, stylistic, etc.) (Rialle*, et al.*, 1997). Thus CARAT definitely relates to interpretative actions. The processing implied aims at seeking, among the various segments of a text, common units of information, and takes part in various steps of a more global process of reading and analysing a text on behalf of a user. The software which offers such services aims at prolonging the cognitive capacities of a user for very complex tasks, and falls within the perspective of cognitive technologies (Gorayska, 1995).

Much research has been undertaken in the field of textual information processing (Delany and Landow, 1993) some of which uses the GA approach (Gordon, 1988; Gordon, 1991; Chen and Kim, 1993; Petry*, et al.*, 1993; Yang*, et al.*, 1993; Chen, 1995; Rialle*, et al.*, 1997; Rialle*, et al.*, 1998; Vrajitoru, 1998). The one we present here is a methodology along with an interactive computer tool which makes it possible for a user to work out a basic hypertextualisation of a mass of pre-segmented texts in collaboration with the system. We illustrate how a genetic based assisting tool can help in the conception, the modeling, and the experimentation of a semiotic behavior such as CARAT, and how this model calls upon the Genetic Algorithm theory to reach its goal.

2 ISSUES IN THE TREATMENT OF LARGE FULL TEXT

Textual information becomes increasingly abundant thanks to networks, CD-Roms and the strong diffusion of microcomputers. In addition to the capacities of extraction and fast information retrieval in the full texts, today the capacity to quickly move inside a text or from one text to another via hypertext links is also necessary. Therefore the need for transforming a normal text into hypertext is increasingly indispensable in administrative, commercial and academic environnements.

These hyperlinks allow restructuring texts into networks for fast and selective reading. The challenge which arises then to tackle the task of hypertextualising a text becomes multiple:

- Firstly, segmentation of the text into independent entities which can be reached individually is performed.

- Secondly, we classify these segments according to a criterion of resemblance which expresses the best possible similarity of subject covered by these various segments.

- Thirdly, finding which links between these classified segments are most relevant completes hypertextualisation.

This challenge is within anyone's reach for small texts (1 to 10 pages). It becomes increasingly difficult with large full text of several dozens of pages. Moreover, this challenge becomes complicated when the segments of texts do not come from the same original text. This is the case for example when a user gradually constitutes a textual data base of direct extractions from the WEB; hence the need to impose upon the text a new structure of access, *i.e.* new links which supplement those already present without supplanting, nor eliminating them. It is in the spirit of solving these problems that we conceived the following method and system.

3 OVERALL PRESENTATION OF THE METHODOLOGY

3.1 STEPS OF TEXTUAL PROCESSING

The method follows three main steps:

- Firstly, preprocessing of the text in order to segment it and obtain a vectorial representation which makes algebraic computing possible is undertaken (neural net and genetic search).

- Secondly, we classify the segments of text based on a similarity measure and unsupervised neural net.

- Thirdly, a genetic search of best associations between segments for hypertext links is performed.

The steps 1 and 2, summarized below, were carried out using the CONTERM platform (Meunier*, et al.*, 1996). Step 3, exposed in point VI, was implemented in PROGEN software.

3.2 STEP 1 : PREPROCESSING

At this step, the text is broken up into unifs (units of information), and structured into segments. Unifs can be words, compound words, n-grams, etc., and are determined using appropriate procedures commonly used in text processing (Salton, 1988).

Table 1 : matrix (segments × unifs)

	unif 1	unif 2	...	unif m
seg. 1	$A_{1,1}$	$A_{1,2}$	...	$A_{1,m}$
seg. 2	$A_{2,1}$	$A_{2,2}$	...	$A_{2,m}$
...	...	...	...	...
seg. NS	$A_{NS,1}$	$A_{NS,2}$	...	$A_{NS,m}$

Segments can be pages, paragraphs, or fixed length word strings, and are realized using a segmenting procedure. One then constructs the matrix (segments × unifs) giving the presence (bit 1) or absence (bit 0) of each unif in each one of the NS segments (table 1).

3.3 STEP 2 : CLASSIFICATION USING NEURAL NET AND DISTANCE TABLE

The second step aims at marshalling segments into classes using a neural net (NN) along with the matrix (segments × unifs). We have chosen the ART neural net (Carpenter and Grossberg, 1987) for its capacities of incremental adaptive learning from evolutionary text base (Meunier and Nault, 1995). The produced classes of segments provides the first proposal of links between segments. The settings of step 2 are the following:

1- set **T** of the NS segments of the text. Let **T** = $\{S_1, \dots, S_{NS}\}$

2- the *cluster* function realized by the NN : for each S_j belonging to **T**, there exists a class C_i such that $C_i=cluster(S_j)$. Formally speaking then, step 2 is a function, namely *cluster*, which gives for all and every segment of text S_j a class C_i.

3- set **K** of classes. Let **K** = $\{C_1, \dots, C_{NC}\}$, where NC represents the total number of classes.

4- class C_i belonging to K is composed of ni segments. Let $C_i = \{ S_{i_1}, \dots, S_{i_{ni}} \}$. A class is a set of segments which are sufficiently close according to the function of similarity implemented in the NN. The classes are disjoined (*i.e.* a segment can be found in only one class) and complete (each segment of text belongs to a class). Different classes have generally different numbers of segments.

5- the table of classes TC giving the segments composing each class.

6- the table TD of Khi-square distance $D\chi^2$ (S_a, S_b) between each pair (S_a, S_b) of segments. $D\chi^2$ is in the continuous interval [0 … 1]. The Khi-square measure is interesting for full text treatment since it takes into account how frequently unifs occur in segments, hence giving greater importance to rare unifs and weakening the influence of frequent ones (Rajman and Lebart, 1998).

4 STEP 3 : HYPERTEXTUALISATION USING GENETIC SEARCH

4.1 DEFINING THE PROBLEM

Step 3 aims at suggesting to the user the best segment to link to each segment of the text. Here, *link* means *suggested segment to link with the segment under study.* Formally speaking, the problem tackled at this point is the following :

given *P-link* a function that provides to each segment S_j belonging to T a segment S_k belonging to the class C_i such that $S_j \in C_i$ (*i.e.*, *P-link*$(S_j) = S_k$), try to construct *P-*

*link** (for "PROGEN-link"), the *P-link* function that best meets the three following criteria A, B, and C :

- criterion A : criterion of shorter khi-square distance between segments.
- criterion B : user's subjective assessments - represented by the function *User* - of the proposed links to each segment using the previous criterion (A). For a segment S_j and another segment represented by *P-link*(S_j), the values of *User*(S_j, *P-link*(S_j)) belong to the set {-3,…,+3}, the semantics of which goes from very bad (-3) to very good (+3).
- criterion C : an additional criterion for preferring long loops of suggested links : $S_j \rightarrow S_{a1} \rightarrow \ldots \rightarrow S_{an} \rightarrow S_j$. Though optional, this criterion is interesting for "pushing" the process towards a rich running through the text, rather than rapid returns to the starting segment.

Let *Loop*(S_i) the number of links (equal to n+1 in the above example) divided by NS to stay in the interval [0 … 1]. Therefore this stage is both "genetic" and interactive, as shown on fig. 1.

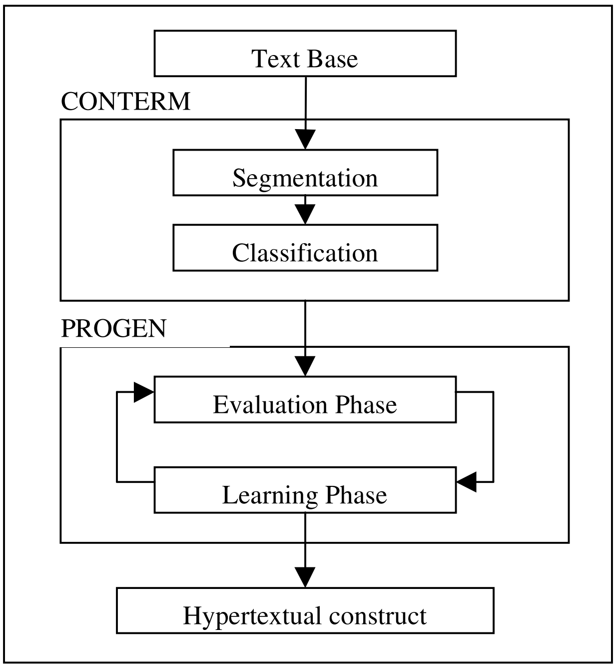

Fig. 1 : Overall architecture and flow of information

The system does not provide complete HTML hyperlinks. It simply suggests associations between segments of text from which the user can possibly construct real HTML links.

4.2 INTERACTIVE SEARCH CYCLE

At the outset, a link for each S_j is randomly chosen in C_i. Then, the following two steps alternate until the user-system couple reaches the best solution *P-link**, that is to say the best suggested hypertextual navigation in the text :

i) user assessment step (criterion B): the user is prompted to give an assessment of each and every suggested links according to an assessment range R (cf. *infra*).

ii) genetic learning step: computation of the current approximation of *P-link**.

The alternation between the user's step in assessment and the learning step is entirely under the control of the user: in practical terms, the user navigates through the proposed links by means of a user friendly interface, and continues to evaluate these links until he decides to trigger a new learning step. The GA invoked at the new learning step will re-compute links using these later elements of the user's assessment.

4.3 FITNESS FUNCTION

The fitness function F that permits the convergence upon an optimal solution has been defined as a linear combination of quantities representing the criteria A, B and C. So, for an individual potential solution x, F(x) is defined as:

$$F(x) = \sum_{i=1}^{NS} [\, \alpha \,.\, D\chi^2 (S_i, \textit{P-link}(S_i)) + \beta \,.\, \textit{User}(S_i, \textit{P-link}(S_i)) + \delta \,.\, \textit{Loop}(S_i) \,]$$

In the experiments the following values have been retained after several trials: $\alpha = 0.8$; $\beta = 0.3$; $\delta = 1.0$

This choice of value weighting is justified by the concern of attaching an importance roughly equal to the three criteria. It was not possible during the experiment to determine whether it was beneficial to significantly modify these values.

4.4 GENETIC SEARCH FOR OPTIMAL LINKS

Within PROGEN, learning is carried out through a fairly normal genetic process. A population of chromosoms is allowed to evolve from one generation to the next using the three standard operators: reproduction, crossover and mutation.

Each of the chromosoms represents a complete hypertext hypothesis for the available segments of text. It is constituted (Fig. 2) by an array of integers the size of which is determined by the number NS of segments. The value k, at position j in the array, is the number the segment S_j is linked to (*i.e.*, S_k).

Fig. 2 : structure of a chromosom in which *P-link*(S_j) = S_k

The size of the population is determined by the user prior to the first learning phase. This size is then fixed for the duration of the working session and does not change form one generation to the next. In the present version of the system, the size can be chosen in a scale of values between 25 and 500. Empirically, we found that a population of 150-200 chromosoms evolving on 200 generations is sufficient to attain a near ideal level of convergeance.

Even though these numbers may seem low in the context of genetic learning, they are comprehensible considering the search space of the problem. For a given segment of text, the number of possible alternative links within the same class of segments, and the evaluations made by the user greatly limit the extent of the search required to find a good link proposal. Therefore it doesn't take long for the system to find a very good solution which is then difficult to improve any further.

This pattern of learning, quite common in genetic algorithms, is attained very rapidly in PROGEN. Any improvements can only come after a fairly extended sequence of evaluation to enrich the possibilities of the search. This is the reason for the iterative approach, alternating learning phases and assessment phases.

After a learning phase the best chromosom from the last population becomes the new hypertext the user is proposed with. The user can then carry out a new evaluation of the hypertext and once again enter into a new learning phase and so on and so forth until he obtains a hypertextualisation to his satisfaction that can be saved in a file. We obtain a hypertextualisation which doesn't mean a hypertext but a set of links. The initial textual data base remains unchanged. Only its access structure is improved.

5 THE EXPERIMENT

An exploratory experiment is currently in progress. Its main objective is to better determine the factors which come into play when using the system. The essential ideas are : 1) to define the best selection criteria for the hyperlinks in order to enrich the adaptation function which controls the genetic selection process, and 2) to better envision the process involved in the interaction between the user and the system. Note also that our goals, at the current exploratory phase, do not aim at establishing performance comparison between the two softwares used, or performing comparative quality measures for the two hypertexts generated. The result of this experiment will be used for parameterizing forthcoming experiments where these aspects will be considered.

The experimentation consists in requiring users acquainted with the task of hypertextualisation to consecutively use two softwares – called "NEUTRAL" and PROGEN, and to verbally give an appreciation of the link proposals carried out by the softwares (and of the workings of the softwares *per se*). These appreciations are then compiled, analysed and interpreted. A standard but rigorous qualitative research protocol is followed to ensure the validity of the results (Patton, 1990; Miles*, et al.*, 1994).

- "NEUTRAL" software presents the user successively with all the segments of the text and, for each segment S_j, it displays the other segments belonging to C_i. The software then prompts the user to subjectively determine among these segments which one seems to be best appropriate for a link with S_j. Thus a complete hypertext is obtained and saved on the disk at the end of the experiment along with the time used by the user to achieve this task.

- PROGEN software asks the user to evaluate the quality of the various links suggested by the GA, and proposes improved links thanks to cycles of genetic search. This alternation between user's assessment and GA search is repeated until the whole set of segments has been gone through at least once. Here too a complete hypertext is obtained and saved on the disk with the experiment duration.

Although the analytical results of the experiments are not yet available, a fast survey of the results of the pre-test were enough to outline several interesting features. Therefore, a quick look at these results has allowed us to confirm the many potential advantages of PROGEN. For example, in the case of "NEUTRAL", the generated hypertext becomes available only after a long phase of work by the user (more than 3 hours for a text of approximately 60 pages segmented in about a hundred segments). In the second case, the generated hypertext is immediately available, and its quality never ceases to increase during the entire work session.

The user's first comments lead us to believe that 3 or 4 learning phases alternating with evaluation phases by the user suffice to produce a hypertext considered as responding to the majority of the user's expectations toward the text base. The size of this article does not allow a presentation of an actual generated hypertext. Furthermore, generating such a hypertext is not the goal of this research.

In PROGEN the user can follow the evolution of the quality of the produced hypertext, and thus the progress carried out by the genetic based learning. The user interface (not shown in this paper) shows two progress curves displayed to the user during one of the experiments. In the X-coordinate we have the evolution in time; the Y-coordinate concerns the grade obtained. The lower curve gives the average obtained by the population to a given generation, the higher curve gives the grade of the best chromosom.

The experiment revealed an interesting phenomenon : in the case of "NEUTRAL", one observes in nearly all classes what one could call an attractor (*i.e.*, a segment of text or a cycle of a few segments) towards which the other links usually converge. Moreover, the user seems to lean towards maximizing the length of the way (i.e., the number of intermediate links) leading to this attractor.

PROGEN produces exactly what one expected of it: thanks to the three criteria A, B, and C, the navigation through the classes (thus in theory the hypertext produced) is significantly better with PROGEN than with "NEUTRAL" software. Indeed, the traversal of the classes produced by PROGEN is either optimal or nearly optimal in respect to the adaptation function used.

6 DISCUSSION AND CONCLUSION

It must be stressed that for the two types of software, the evaluation of the user and the limits of training are constrained by the classification provided at step 2. The relevance of the suggested links is thus related to the quality of this classification.

Although the results are encouraging, the question of finding out to what extent our running through the segments of artifical classes can be generalized and used by a wide range of plain text workers still comes up.

A recurring remark from the users is that it is necessary to better balance the positive grades granted to the "good" links to thwart the strong tendency towards negative evaluation. This tendency is due exclusively to the often poor quality of the step 2 classes. This classification seems to be the weak point of the whole system: the classes always tend to contain parasitic elements which undermine system coherence.

According to the results of this first experiment, it is possible that a total run through of a whole_class is not always perceived as an appreciable quality in hypertext. One must then revalue the effect of the quality of classification, and the parasitic elements in classes.

In case the mechanisms of automatic classification cannot be improved, it would then be necessary to plan not to ensure a complete run through of a class, and accept the presence of "holes" in the hypertext, *i.e.*, of elements that can be attained only by sequential running of the text.

Among the modifications performed on the genetic algorithm in order to increase the speed of its convergence, it is worth noting the protection of very good links against possible mutations. This type of modification is perfectly central to genetic algorithmic thinking and does not constitute a violation of the paradigm. The malleability of genetic algorithms is one of the most appreciated charateristics that had strongly influenced our chosing them.

Acknowledgements

Special thanks to C. T. Schmidt for his preparation of this text. The present work is part of a Franco-Québécois cooperation under the tutelage: for France, of the Ministère de l'Enseignement Supérieur et de la Recherche (MENERT, DISTB) and the Ministère des Affaires Etrangères; for Quebec, of the Gouvernement du Québec - Ministère des Relations Internationales.

References

Carpenter, G. and S. Grossberg (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*(37): 54-115.

Chen, H. (1995). Machine Learning for Information retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms. *Jour. Am. Soc. for Information Science* 46(3): 194-216.

Chen, H. and J. Kim (1993). GANNET: Information retrieval using genetic algorithms and neural networks. *(working paper, CMI-WPS).*

Delany, P. and G. Landow, Eds. (1993). *The Digital Word: Text Based Computing in the Humanities*. Cambridge, MA, MIT Press.

Gorayska, B. (1995). Cognitive Technology. *Information Technology and the Culture of Post-Industrial Society: Human Machine Symbiosis*. K. S. Gill, (ed.). Berlin, Springer Verlag.

Gordon, M. D. (1988). Probabilistic and genetic algorithms for document retrieval. *Communications of the ACM* 31(10): 1208-1218.

Gordon, M. D. (1991). User-Based Document Clustering by Redescribing Subject Description with a Genetic Algorithm. *Jour. Am. Soc. for Information Science* 42(5): 311-322.

Meunier, J.-G., I. Biskri and G. Nault (1996). *Exploration des modèles connexionnistes à l'analyse terminologique*. Projet CONTERM, AUPELF-UREF, Rapport intérimaire.

Meunier, J.-G. and G. Nault (1995). Modèles connexionnistes et traitement de l'information textuelle : Le modèle ART de Grossberg. *Cahier de recherche du LANCI* 95.9(Université du Québec à Montréal).

Miles, M. and Huberman. M. (1994). *Qualitative Data Analysis*. Thousand Oaks, CA, Sage Publ.

Patton, M. Q. (1990). *Qualitative Evaluation and Research Methods*. Newbury Park, CA, Sage Publ.

Petry, F., B. Buckless, D. Prabhu and D. Kraft (1993). Fuzzy information retrieval using genetic algorithms and relevance feedback. *Proc. ASIS Ann. Meet.* Medford, NJ: 122-125.

Rajman, M. and L. Lebart (1998). Similarités pour données textuelles. *JADT -4e Jour. Int. d'Analyse*

Statistique des Données Textuelles. S. Mellet, (ed.). Nice 19-21/02/98, Univ. Nice-Sophia Antipolis: 545-555.

Rialle, V., J.-G. Meunier, S. Oussedik, I. Biskri and G. Nault (1998). Application de l'Algorithmique Génétique à l'Analyse Terminologique. *JADT -4e Jour. Int. d'Analyse Statistique des Données Textuelles*. S. Mellet, (ed.). Nice 19-21/02/98, Univ. Nice-Sophia Antipolis: 571-581.

Rialle, V., J.-G. Meunier, S. Oussedik and G. Nault (1997). Semiotics and Modeling Computer Classification of Text with Genetic Algorithm : Analysis and first Results. *ISAS'97, Proc. 1997 Int. Conf. Intelligent Systems and Semiotics : A Learning Perspective,* A. M. Meystel, (ed.). Gaithersburg, Maryland, National Institute for Standards and Technology : 325-330.

Salton, G. (1988). *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Reading, Massachusettts, Addison-Wesley.

Vrajitoru, D. (1998). Crossover improvement for the genetic algorithm in information retrieval. *Information processing & management* 34(4): 405 - 415.

Yang, J., R. R. Korfhage and E. Rasmussen (1993). Query improvement in information retrieval using genetic algorithms. *Proc. Text Retrieval Conf. (TREC-1)*. Washington, DC, NIST: 31-58.