TEXT ANALYSIS LEARNING STRATEGIES

Pierre PLANTE
Université du Québec à Montréal

Abstract:

APLEC (l'APrenti-LECteur - or Learning Reader) is an extension of the Déredec software, a programming system devoted to the content analysis and linguistic treatment of texts.

APLEC will associate automatically to any text descriptive grammar (TDG - Grammaire descriptive de texte) a question/answer module where the questions are asked in a given natural language by the user, and where the anwers are tracked down in the textual corpus undergoing examination.

The user's TDGs can inscribe themselves at various levels of analysis (morphological, syntactical, semantical, logical, ...), and it is a singular characteristic of the Déredec that it allows for a polyvalent treatment without interfaces, this being done with only one retentive structure for the information, and only one algorithmic structure for the description and for the retrieval.

Once the user's TDG is applied on the question (it was at first applied on the whole of the text), some exploration models (provided by the user) activate the comparison of the question to the text in order to initiate the tracking down of the answer.

If, due to weaknesses of the grammar, APLEC cannot bring to an end this work, it will track down relevant elements of the problem lifted up, communicate with the user, submit to him the results of its analysis, and wait for him to propose a solution to the problem. If there is such a solution, it will be further on 'learned' by APLEC, that is it will be generalized on the whole text in such a way as APLEC will no more intervene if similar problems arise again.

Hence, the intelligence of the analysis grows gradually as the work of the interactions between the user and his text, via the automaton, goes on; and it shall construct itself by osmosis to the particular semantics of the text undergoing questioning.

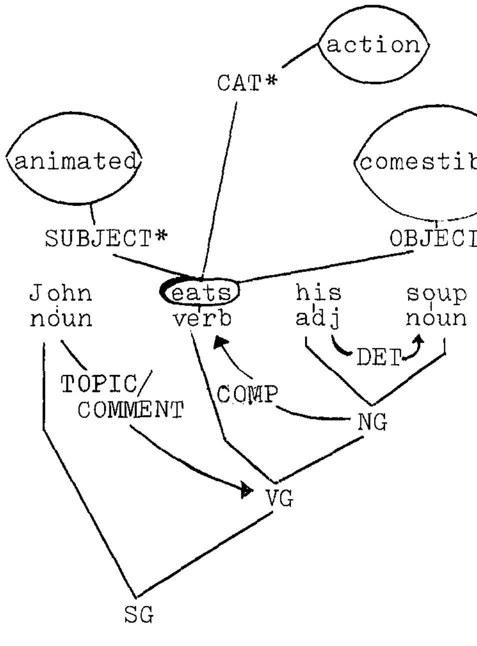
We do not pretend to supply an achieved theory of automatic learning, but rather try to supply different formal strategies functioning whatever the content of the Déredec grammars applied to a given text.

I will first describe the general characteristics of the Déredec programming system, and afterwards will pass on to a few examples of utilization of the question module named APLEC.

The Déredec offers to its users two large classes of functions: (i) functions said to be text descriptive, and (ii) text explorative functions.

The first functions have the form of finished state automata. They are machines that scan the text sequence after sequence (e.g. sentences), and that build tree structures onto the elements of these sequences (e.g. words). The knots of the trees are labelled with descriptive categories, and are also, eventually, linked together by oriented relations. The leaves of the trees (the words of the sequences) can see themselves adjoined with complex semantic networks.

Example:



The user will program his Déredec automata in such a way as they can label descriptive structures to every sequences of his text.

The Déredec remains indifferent to

the nature of these structures which hence can inscribe themselves at syntactical, morphological, semantical, or logical levels.

The Déredec automata scan the sequences in both directions. They are non deterministic, but all degrees of determinism are programmable. These automata are preferentially ascending rather than descending. It is to be noted also that their sensitivity to context can run over the frame of every analysed sequence and spread out on the whole of a corpus. Actually, any decision taken in a Déredec automata (may it be of categorization, composition of a phrase, the labelling of a relation between two knots, the construction or developing of a semantic network...) can be tributary to the result of an investigation carried out in that part of the corpus preceding or following the pointed sequence, or even in any other corpus. This type of enlarged contextual investigation enables a Déredec definition of properly textual recursive grammars in addition to the definition of sentential recursive grammars.

The tree structures (named EXFAD - Expressions de forme admissible, or Expressions of admissible form) will be associated to the corpus sequences in an evolved interactive mode. Here the Déredec software can give assistance to its users in many ways: first, it will automatically track down and give him a diagnosis of programming errors; it also allow him to trace the behavior of an automaton; it further allows interruptions in the automata's work in order to question the state of the description, to modify the grammar, etc.

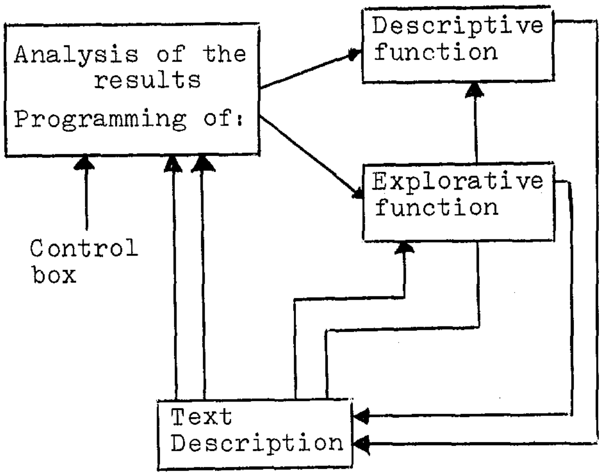
The text descriptions (TD - Description de textes) produced by automata will be analyzed further on by explorative functions whose arguments (called exploration models), given by the user, are pattern-matching structures having a simple writing syntax and a high discernment power. It is by way of the explorative functions that the text descriptive grammars (TDG, i.e. the automata series) will be associated to content analysis objectives.

Thus the programming sessions with Déredec will usually have the aspect of an enchainment whose links are:
(1) an automata construction;
(2) the obtaining of TDs by the application of these automata to the corpus;
(3) the elaboration of exploration models;
(4) the application of explorative functions on the TDs;
(5) the analysis of the results eventually followed by re-explorations or by the reconstruction of new descriptions.

Programming with Déredec essentially signifies to program the production of TDs, then to program their exploration, to analyze the results, and to start over at one or the other stage until the obtaining of satisfying results. The whole of the process is highly facilitated by the fact that the admissible expressions, at the input like those at the output - may they be for descriptive functions or for explorative functions - have exactly the same writing syntax from a computational point of view.

One could think that such a type of experimentation will be the lot of most of the software users. But the highly interested user will surely want to enjoy the Déredec's ACSP procedures, i.e. it's automatic context sensitive programming procedures (procédures de programmation automatique sensibles au contexte: PASC).

These fuctions try in various ways to simulate the behavior of the programmer in the control box of the following diagrams:



What is here aimed at is to take out of the programmer's hands as many as possible of the real operations to be made, and to leave him to take only certain high level decisions as per general planification of the experiments.

As for example, some of the ACSPs deal with the chained reapplication of explorative functions on a TD; the results that are obtained at every exploration will serve to modify the model (or models) of exploration that has (or have) been used, model (or models) which is (or are) given in a primary version by the user as a starting point. The user will control the ACSP by giving certain keys, certain parameters that will guide the whole of the operations.

Other ACSP allow for the progressive enrichment of a TD by constantly modifying a descriptive apparatus whose general structure is given at the start, here again accompanied by general parameters dealing with the iterative process.

It should be noted that the automatic programming procedures have, from a computational point of view, the same form or the same admissibility as the other Déredec functions or operations; it follows that the former are thus compoundable with the latter. This last characteristic is the one that accounts for the 'context sensitive' qualification of the ACSP: the automatic programming procedures are settled in environments that are susceptible to supply the parameters relevant to their execution.

The ACSP machine which is by far the most complex and, in a certain way, the most complete is named APLEC.

APLEC (APrenti-LECteur - the Learning Reader) will associate automatically a question/answer module for all TDG submitted by the user. This Q/A module where the questions are formulated in the natural language of the text, and where the answers are tracked down segments of the latter.

When APLEC cannot track down an answer, it will:
(1) give a diagnosis of the difficulty;
(2) communicate with the user;
(3) ask for a solution;
(4) and - if a solution is proposed by the user - try to generalize the solution onto the overall corpus.
All this is done in order to increase it's forthcoming retrieval power, and to keep interruptions to a minimum.

In addition to the grammar and to a few exploration models that he will consider as being representative of the structures labelled by it, the user will supply APLEC with only very general parameters as per the retrieval conditions and the learning conditions.

Here is a first example of a dialogue with APLEC:

USER: Hello!
APLEC: Hello! Would you like to give an opinion on the hunting of baby seals? I will look through my data for an opposed opinion.
USER: Brigitte Bardot interests are less than honourable!
APLEC: It seems to me that we should praise this defender of nature rather than to run him down by lending to his actions purely selfish motives.
USER: Nevertheless Bardot is a very decent lady!
APLEC: You are contradicting yourself; you were for it and now you are against it. Are you in favor of this hunting, yes or no?
USER: No!
APLEC: Brigitte Bardot has just made a fool of herself by her loony declarations on the hunting of baby seals.
Give another opinion; I will now look for an agreeing argument.
USER: Weber acts to protect the cute little seals!
APLEC: Can you supply an equivalent expression for the expression "seals" among the expressions: "wonder", "earth", "house"?
USER: "Wonder"!
APLEC: Can you consider the expression "Weber" as an equivalent for "celebrity"?
USER: Yes!
APLEC: Other rich celebrities do nothing for the protection of wonders of this earth which is the house of all.
Any other opinion?
USER: Brigitte Bardot comes to defend the nice hunters!
APLEC: I think I discern a contradiction in the evaluative process of your opinion. I cannot give an answer to it.
Any other opinion?
USER: Thank you!
APLEC: Bye, bye!

In that example, the analyzed text is made up of some twenty letters sent to newspapers while was taking place the dispute surrounding the hunting of baby seals in the St-Laurent estuary. APLEC was then fed upon a French surface grammar (PLANTE 1980), and a so-called *discoursive evaluative grammar* (grammaire évaluative du discours) (PANACCIO, 1979). The first grammar tracks down for all french sentences the topic and comment as well as various types of complements and determinatives. It also practices a segmentation of the sentence in its sentential components. The second grammar allows us to know if an agent or a typical act of a discursive formation (e.g. here, the hunting of baby seals) is *praiseworthy* or *condemnable*, this being done after a study of the evaluative transfers occurring on the markers "praiseworthy" or "condemnable" in the sentences. These markers are at first labelled manually on certain words of the corpus (e.g. "ridicule ": "condemnable"). The second grammar juxtaposes itself on the first. (Both grammars are more thoroughly expounded in PLANTE, 1980.)

When APLEC cannot track down an answer directly in the text (either because of a weakness of the grammar or because of some textual insufficiency on the question), APLEC will intervene, i.e. get in touch with the user, explain the difficulty, and wait for a solution. If a solution is proposed, it will probably facilitate the discovery of the precise answer to the question; moreover, it will be, if the user wishes, generalized to the whole of the text undergoing exploration, thus it will help with the future tracking down of answers to other questions.

Hence APLEC's performance, that is its capacity to supply adequate answers to given questions, improves itself gradually with the conversations allowed. Little by little, conceptual networks are constructed, allowing for a more and more refined and acute analysis because of its being more and more relevant to the explored text. APLEC is in a way sold blank to its users who little by little will transform it into a more personal robot which gets better while adapting to particular textual data.

The solutions that are proposed during the conversations and that are later on generalized can take different forms. In our example, the solution form lies in the supplying of "word-to-word" equivalence relations ("Weber"::"celebrity"). Yet APLEC can accept and generalized solution forms which are a lot more complex. Thus, for example, the short following text:

> John and Mary are at the bookshop.
> Mary would like to offer a book to John.
> But she lacks the money.
> Mary decides to slip the book under her coat.
> The bookshop owner will lose a book but John will be happy.

USER: Did Mary steal?
APLEC: Make "offer" more explicit.
USER: Habitually we offer an object of value! [1]
APLEC: Make "slip" more explicit.
USER: It is an action!
APLEC: Make "steal" more explicit.
USER: It is an action on an object of value.
APLEC: Mary decides to slip the book under her coat.
Any other question?

Thus APLEC learns that "book" is an object of value (since it is "offered" in the text), for it learned that to "slip" and "steal" can now be taken one for the other, since that to "slip" has become an "action" on an object of "value"...

In this second learning strategy, the "word-to-word" equivalences are replaced by more or less complete pattern-matching semantic networks defined in the terms of the experimented grammar and labelled to the words of the question and to those of the text. These networks, proposed only during interruptions, first permit a refinement to the pattern-matching procedures to facilitate the tracking down of the good answer, and secondly permit the augmentation or the construction by themselves of new networks. In the last example, after the first explicit reformulation, all the 'offered objects' in the text received the category "value"...We will note that APLEC authorizes the contextualisation of the learning procedures, i.e. the circumstantial generalization of the solutions proposed by the user, this being done with the help of different execution parameters manipulated by the latter.

We believe that APLEC formally allows the distinction of what is relevant in a semiotic enterprise to the semantics of a group of texts, to the semantics of a particular text, or rather to the semantics of a particular use of a given text. It also allows the delimitation of the semantic boundaries of a given TDG, and to thus make easier the amelioration of the latter.

These last considerations lead us to reflect on the problems that are related to the project of the edification of a theory of natural language description, and, in particular, on the problem of the insertion of semantics in that theory.

Natural languages are objects that, in an evident manner, strongly resist all known formalization techniques. We have to get used to the idea of an object whose rules can change according to the portion of it we are observing, and according to the use we are making of it. We can always think of particular semantics, functional for a given world; many robots have shown that it is possible to simulate the functioning of a natural language for a limited semantic world. These experiments are interesting in an illustrative way, but they miss the essential of what characterizes the normal behavior of a speaker: his capacity to pass from one semantics to another while adapting, or even while transforming when needed the set of rules already constructed. What we need is a software that will constantly facilitate the construction of new sets of primitives, of new axioms, and of new rules of inference, these elements being considered as variables rather than constants.

We are still a long way from perfecting a system that would simulate such learning procedures, still far away from a system where the semantics would be in a fairly good part balanced on the side of the use and not on the side of the internal construction principles. Meanwhile, local experiments of description will be valid only if they are accompagnied by their empirical adequation conditions. From such a point of view, the Déredec wants itself to be a formal framework for the comparison of different functionality indices and of empirical adequation indices of the descriptive rules. But moreover, for a given set of rules, it (and here I am thinking more particularly of APLEC) automatically will track down the sequences or the lexical items of the corpus the description of which must be specifically enriched to elevate the empirical adequation index. The rules will not then be automatically transformed or modified, constituting a very close simulation of human speaking behavior, but generalizations will nevertheless be produced automatically in such a way as to augment the scope of the solutions that are at that moment proposed by the user.

[1] The explanations (making an expression more explicit), i.e. the solutions to APLEC's problems must be represented in a formal language - contrarily to the questions that are asked and to the supplied answers which are both given in natural language (French, English...) - in this example, the explicit reformulations are given in English in order to make the presentation easier.

Bibliography:

PANACCIO, Claude "Des phoques et des hommes, autopsie d'un débat idéologique." Philosophiques, VI, 1 (1979), 45-63.

PlANTE, Pierre Le Déredec, logiciel pour le traitement linguistique et l'analyse de contenu des textes, Manuel de l'usager. Montréal, U.Q.A.M., Service de l'informatique, mars 1980.