POMMIER: a classification approach for assisted textual comprehension

Ioannis Kanellos, Xavier Simon, Francois Riviere, Minh-San Nguyen, Emmanuel Mayer, Julie Canonge.

Ecole Nationale Superieure des Telecommunications de Bretagne, Technopole de Brest Iroise, BP. 832, 29285 Brest Cedex, FRANCE

KEYWORDS: interpretative semantics, classification algorithms, text comprehension

AFFILIATION: Department of Artificial Intelligence and Cognitive Sciences, France Telecom University, Brest, FRANCE

E-MAIL: ioannis.kanellos@enst-bretagne.fr
xavier.simon@enst-bretagne.fr
francois.riviere@enst-bretagne.fr
ms.nguyen@enst-bretagne.fr
emmanuel.mayer@enst-bretagne.fr
julie.canonge@enst-bretagne.fr
FAX NUMBER: (+33) 98 00 10 30
PHONE NUMBER: (+33) 98 00 14 35

Abstract

The aim of our work is to set up semi-automatic devices helping in text interpretation by using robust classification methods. In the first part we give some elements of the theoretical background sustaining the overall approach from a linguistic point of view; they essentially come from the theory of interpretative semantics; we try to explain in which sense the problem of the plausible comprehension of a text may usefully meet classification preoccupations. In the second part we give a quick overview of algorithmic ideas close to our aim motivating our technical choices. This part is completed by details of the implementation achieved, as well as a quick description of the interface, its basic use and utility for humanities, as long as they use texts. Finally, we discuss some results and suggest the main directions of future development.

1. Introduction

Our work comes from a rather simple idea; it consists in considering reading as a classification oriented activity."Reading" here has to be understood as a semantic process as a process aiming at organizing a particular textual input for comprehension objectives. Whatever the very cognitive nature of such a process may be, one may plausibly model it as a classification task, in so far as classification is the archetype of organizational activities. Whatever the case, classes are good operational paradigms for dealing with semantic categories.

In reading, one tries of course to catch something of the encoded information as a particular structure of linguistic features; but, also, one may discover structures not intentionally included by the author; such structures highly depend on one's knowledge, sensitivity, state of preparation etc. From the point of view of the linguistic theory, all these structures reflect possible sense categories involved in the text. They are the effect of interpretative strategies operating over several levels of the textual material. Indeed, the notion of interpretation is the key notion for understanding reading as an effective procedure over a given textual input [Ras87], [RCA94]. It reveals that there are many possible senses in a text, related to different interpretations; one seeks to justify a certain sense correlated to specific interpretational schemata rather than to describe a supposed unique sense. Thus, interpretation may be characterized as a classification process. On the basis of a set of selected textual units (lexical, grammatical, semantic, pragmatic, stylistic, narrative...) one tries to define some classes representing semantic categories of the text. Then one tries to refine, extend or even modify this first state of affairs in the light of additional information gleaned through the text. Indeed, trying to understand a given text, one generally has to read it many times, each time confronting previous structures to new data, import additional external information, integrate multi-level organizations... In short, one works on the text in a rather complex manner in order to refine one's comprehension, and this is done, by giving a unified structural scheme to the information processed. From this point of view, reading appears as a never-ending process; but clearly not infinite. Reading is always a converging process, where the converging point is a structural configuration of great stability. Such stability may be formalized as the amount of structural changes needed to establish an original class distribution. It is clearly related to the very algorithmic competence used in the classification.

These ideas form the core of our theoretical approach. Furthermore, they make clear the motivations of our application; it aims to assist the user in organizing his textual material and thus to help him in choose and refine his interpretational strategies. Here is the idea of the basic protocol: first, the user chooses a sample of textual elements (items) and characterizes them by a set of attributes (and associated values); then he classifies some of them defining a group of classes on the basis of a first and rather intuitive comprehension he initially elaborates. Then the algorithm operates on

more extended sets of textual elements and classifies them automatically into one of the defined classes. At any moment and level, the user may reorient classification. This interaction combines intuitions of the user and rigorous classification criteria in a unified scheme. Moreover, it can be stopped at any moment – when the user judges that the overall interpretation scheme is satisfactory. The classes obtained furnish interesting indications for text interpretation, in so far as they operate generic semantic characterizations of large amounts of textual elements.

2. Algorithmic issues

The main objective of our application is to predict which semantic class a specific textual element is in. In so far as the classes defined are assumed disjoint, such an objective becomes equivalent to the searching for a correct classification procedure. In our case the measurement space is completely supplied by the user; thus it represents his own (initial) understanding. On the other hand, the user also specifies a limited training sample: he associates semantic classes on a subset of vector measurements defined by the values given to the selected attributes; no particular restrictions have to be imposed on it at this level – in fact, it may be set according to quite intuitive criteria; but a natural way is to choose textual inputs whose classification corresponds to basic interpretational directions.

Clearly, tree structured classifiers offer a powerful and natural way of solving this kind of problem. What they do is to repeatedly split every measurement sub-space into two disjoint descendant subsets – the initial being specified by the user. Terminal subsets are designated by a class label; the split process is based on the coordinates (values) of the measurement space and the split conditions are extracted from the training base. The fundamental idea is to select splits such that the data in each of the descendant nodes are "purer" than the data in the parent node. At the end of the recursion, each leaf must only contain textual elements of one class. In practice, a subset is considered as terminal if it contains few textual elements or if most of them belong to the same semantic class. We then obtain a maximal tree and it is necessary to "prune" it in order to get the less complex optimal tree which has almost the same accuracy. The CART (Classification And Regression Trees) algorithm that we have used is introduced in [BFOS84] and comes from a similar methodology. Moreover, it possesses features of great value for our purpose: generality, naturality, simplicity and low complexity. Let us take a closer look at how CART works in our application.

2.1. Generation of the maximum-size decision tree

This part of the algorithm operates on a given set of textual elements – initially the training set; it recursively repeats the following procedure which finishes either in (a) regrouping, if possible, the textual elements into a leaf or in (b) creating a node and splitting the set of elements into two subsets.

(a) A leaf is created if one of the following conditions is satisfied:

- All the elements belong to the same class
- The number of elements is less than to N_0, an integer defined by the user. N_0 quantifies the limits of any semantic class likely to be of interest for interpretational purposes.
- The impurity function of the set of elements is below the impurity threshold S_0, also a user-defined number. The lower the value returned by this function is, the more homogeneous is the set of elements (in terms of class distribution).

(b) If (a) is not possible, then the initial set is split into two subsets according to a test on one of the item attributes. Since all the possible tests are not equivalent, we use a particular criterion in order to quantify the discriminating power of each test and choose an optimum one. Thus a node is created, with the chosen test associated with it. And the items satisfying the test are sent to the left descendant of the node, and the others to the right. So, two new (and smaller) sets of items are constructed to each of which procedure (a) is applied.

2.2. Dealing with tree complexity

The tree obtained from the previous step is very good at classifying the items of the training set. However, its size is large, partly because it is overspecialized in the data of the training set. Some of its tests may be very specific to these data, and appear irrelevant or unnecessary when the tree is used to classify new textual material. Indeed, it is possible to find a smaller tree whose classification performances are equivalent to those of the maximum-sized tree with items from the training set, and which are even better with new training sets. Therefore, a "pruning" method is used to build a sequence of trees by cutting one "least relevant" branch of the large tree after another. This part is quite opaque for the user and is used for optimization purposes.

2.3. Quantifying classification errors

For each tree of the final sequence, an error rate is calculated according to the results given by the tree in classifying the items of the testing set. The tree with the smallest error rate is kept as the final decision tree. Thus the user may use the error

indication in order to adjust his interpretational schemata to the performance of the classification.

3. Implementation

Our work is divided into two main parts. The first one is the data processing and the implementation of the CART algorithm; the second one is the building of a user-friendly interface. As these parts are quite different, they may be envisaged in different programming languages.

The CART algorithm was implemented in a Prolog language (ECLiPSe). There are several reasons for this. First, Prolog is a high-level programming language allowing this type of application to be developed with great transparency and without major problems. It provided us with a set of powerful functions to manage data bases involved in the CART algorithm. Secondly, we needed a data structure to manipulate trees. Such a structure involves nodes and leaves, so by using the list structure of Prolog, we did not have to build up our own library for lists. Finally, a Prolog dialect allows easy correction of code and good verification of the algorithmic progress.

The interface was implemented in the TclTk language. TclTk is easy and generic enough for our purpose. Furthermore, a specific Prolog library is implemented allowing Prolog to use TclTk functions. Therefore, the interfacing between the two languages is quite natural and highly efficient. The interface is divided into five parts: four windows give explicit information about training, annex (pruning), test and non-classified bases. The first one is used for the menu bar, containing all the functions the user would desire in interacting with textual material.

Once a file is loaded, the first three windows are automatically filled. All of them will be used by the CART algorithm. Determined or random (with a specified percentage) parts may be used in each base. The user can act directly at all phases and on every formal entity. For instance, he can add, remove or modify textual input, attributes, values and even semantic classes. Furthermore, he can modify the training and annex bases and his action may be reconfigurable (using the mouse or a keyboard...). He can also reconfigure most of the visual entities of the interface.

Finally, the tree can be entirely built up visually; the learning menu contains several options allowing the user to change tree parameters (see previous section). Once the tree is built up, the user can select items to be classified; the results appear in the same window. The classes obtained are the support of generic interpretations covering the whole textual input. Furthermore, in looking for the path leading to a decision, the user can interpret the classification result semantically and thus be able to compare interpretational schemata.

4. Results and discussion

Classification is an essential issue for rationalizing categorization effects. Thus, sense emergence and evolution through interpretational schemata may naturally be thought of as classification processes. As the user always has the possibility of refining classes, the opposition between formal classes and semantic categories is not very sound. Even if the latter are somehow continuous and not necessarily disjoint, one may, by defining new classes, have good approximations of their structural effect. On the other hand, the extent of such an approach is wide and its reutilisability direct.

POMMIER has already been applied to drama monologues giving interesting results. For instance, for Tchekhov's "Les Mefaits du Tabac" two groups of classes have been used to explore different approaches to the subject (apparent and dissimulated) of the play, the aim and the character of the person, of some pragmatic relations involved, some psychological connotations etc. These depend on what idea one initially validates in order to orient one's interpretation. The same final classes are not obtained if one understands the whole play as a product of fear, of an evolving battle between frustration and the desire for freedom, of an invocation of death or even of a rigorous calculation of the person taking advantage of the situation for personal – both intellectual and psychological – satisfaction. POMMIER offers generic rationalizations of all these possibilities.

References

[BFOS84] Breiman, Friedman, Olshen, and Stone. Classification And Regression Trees. Wadsworth and Brooks, 1984.

[Ras87] Francois Rastier. Semantique Interpretative. P.U.F., 1987.

[RCA94] Francois Rastier, Marc Cavazza, and Anne Abeille. Semantique pour l'analyse, de la linguistique a l'informatique. Masson, 1994.

[Tch86] A.P. Tchekhov. Theatre complet. Gallimard, Paris, 1986.