

Computer-aided Language Processing : Using Interpretation to Redefine Man-machine Relations

Ludovic TANGUY

Laboratoire d'Intelligence Artificielle et Sciences Cognitives
Ecole Nationale Supérieure des Télécommunications de Bretagne
BP 832 29285 BREST CEDEX - FRANCE

AND

Faculté des Lettres et Sciences Sociales
Université de Bretagne Occidentale
BP 814 29285 BREST CEDEX -FRANCE
E-mail : Ludovic.Tanguy@enst-bretagne.fr

Abstract

Focusing on the human-computer interaction for natural language processing, we present a formal notion of interpretation, as a frame for the semantic analysis of texts. Moving away from classical automated processing, we define the role of the computer as a guide towards manual identification of semantic structures. We propose an interactive protocol in order to “extract” and explore a user’s intuitions of the thematic structures of a given text. This protocol, along with the data structure which stores semantic information from the user, is managed by a software system, named PASTEL, which is described herein. Using an example, we show how this method can be useful in deeply exploring textual content.

1 Introduction

We would like to present here a kind of re-thinking of the relation between man and machines as regards computer-aided text analysis. Following the repeated failure of automated language comprehension by computers, several methods have been proposed in which human interaction is required. However, these intrusions of human knowledge are somewhat peripheral: systems are still machine-centered.

Most common approaches concern the statistical analysis of texts [3, 6], in which, beside fixing parameters, human action is only requested *afterwards*. Once clustering of words, for example, is performed, results are presented to the user, who then has full freedom to interpret them.

We would like to discuss here an approach whose goal is to more widely plan these interactions, thus

proposing a complete dialogue between man and his machine. Our theoretical bases will mainly consist of the linguistic theory of interpretative semantics [4, 5], which describes the conditions of constructing meaning, when dealing with texts. This is a structuralist approach, whose formal description, although simple, is flexible enough to permit human “intrusion” in formal representation. Its main advantage is to propose a rationalizing method of representing the impact of global semantic structures on local description. This theory is also motivating for rethinking the place of standard processing, for it cannot accept a classical bottom-up approach in calculating meaning. However, its structuralist influences are advantages toward a simple formalization, enabling us to focus on means of cooperation.

We will present an application of these principles, the PASTEL software, which can be described as a computer-assisted interpretation tool. It has mainly been used on short texts (poetry and newspaper articles), and consists of an interfaced protocol whose goal is to lead and motivate the user throughout his semantic analysis of a given text.

Its computerized part is reduced to the management of a formal structure describing the interpretation, ruled by a number of formal constraints. Thus, when managing the constraints, the computer will continually ask the user for further explanation of his interpretation, possibly leading him in to discover new semantic aspects of the text analyzed.

Schemata of new-generation man-machine cooperative systems are proposed, describing generic

knowledge-base systems [9]. We agree with critics about the danger of high complexity in the logical aspects of knowledge systems [8]. Non-standard logics, for example, increase the gap between machine representation and human cognition. It has been argued that only first-order logics is flexible enough to ensure the user's comprehension of the system's inner mechanics. On the other hand, the area fit for computation has to be restricted: it has been seen that human cognition cannot be reduced to calculus. When it comes to understanding natural language, this is even more important. Most work on NLP automation has concerned syntax, and proved the importance of complex processing. But this is not the case for semantics, partially because a sentence's meaning is not a function of the meaning of its parts. Sense seems to be propagated toward local units, and to come from a non-rationalizable global source.

So, the alternative to more and more complex semantic formalisms and systems is to offer some kind of cooperation between the computer's processing and representation power, and man's creativity and global understanding. This requires a flexible means of semantic knowledge representation, and a tight cooperation between the management of this representation structure and human knowledge.

2 Our goals, with an example

The main text we will be using to exemplify our approach is taken from a French satirical newspaper¹, which mainly deals with French political life. The text mainly consists of a classic criticism of President Chirac's political behavior, but the author uses many religious terms in order to do so². The event at the origin of this newspaper article is the threat our president received from an Islamic fundamentalist terrorist group, when they demanded his personal conversion to Islam. The text, its structure and its difficulties will be discussed here. The main interest, linguistically speaking, lies in the tight mixing of two separate lexical fields (religion and politics) through a number of stylistic figures.

When dealing with such kinds of texts, "classical" computer analyses are very limited. The syntactic structures are very complex, and many puns are present, as well as neologisms, that do not enable use of an automated lexical parsing. As an example, we

¹Le Canard Enchaîné, 1995, paper *Chirac envisage de se convertir, mais à quoi ?* by F. Pagès

²The choice of such a text as an example should not be *interpreted* as any kind of political or religious opinion on behalf of the author. Our only concern here is the stylistic nature of the text used, not its socio-political contents.

can see in this text the neologism *extrême-ponction*, which can be analyzed as a deformation of the expression *extrême-onction* (extreme-unction) through the use of *ponction* (tapping, here in the context of raising taxes). Furthermore, many interesting aspects of the text's thematics are socio-culturally based, and as such require a very fine level of description in the lexicon.

For this last remark, we can try a more cognitive analysis. The hypothesis is that, in many places in such texts, the stylistic features can only be discovered if the reader is aware of the general semantic structures of the whole text. As an example, one can find in the text a declared opposition between *La Mecque* (Mecca) and *Colombey-Les-Deux-Eglises* (a small French town), as possible destinations of president Chirac's pilgrimage. The first place is well known as the main site of Moslem pilgrimages, whereas the second one is more narrowly known as the birthplace of General de Gaulle, the founder of Chirac's political movement (known as *Gaullisme*). The other semantic aspects related to *Colombey-Les-Deux-Eglises* are faded in a classical vision of this text, because the most relevant feature is its political symbolism. And there, the information related to these two lexemes is at a different level of language. So, the semantic description is completely dependent on general structures : the sole semantic relation between these two lexemes, saying that they are both geographical places is not sufficient to describe the local opposition.

On the other hand, the kind of "surface" relations between semantic aspects, such as the relative positions of words belonging to the two main themes (religion and politics) are simple to describe and compute.

So, the cooperative approach we propose for this kind of text will follow the following rules :

- We will let the user lead and propose all of the semantic descriptions. Assuming the fact that not all semantic aspects can be discovered, we will deal with a notion of interpretation, rather than utopian full understanding. The semantic descriptions will be known to be incomplete, but guaranteed to fulfill the reader's vision of a given text.
- The interpreter will lead the analysis by first expressing his *global* intuition. More precisely, he will declare what approximative thematics he has found in the text, and this initial declaration will influence the whole analysis. The need for such a rule from global to local has been discussed above. The local descriptions (words, and expressions) will only come after such general data

has been declared. We therefore remain some distance from classical compositional analysis, such as performed by fully automated NLP, where local descriptions (context-free) are arranged together in order to obtain the semantic structure related to a whole text.

- Throughout the exploration of such general features, we will use the computer to discover new thematics and to refine the original ones. This will mostly be done through the implementation of a data structure, and a projection of general structures upon local units. All the information is extracted from the user's own language knowledge, excluding generic semantic description as can be found in established lexicons. Doing otherwise would violate the above principle that pertinent local descriptions are mostly influenced by general textual structures. This is where the computer can be of much help, precisely because of its being unable to understand language as we do. Relying solely upon formal manipulations of abstract units, the machine will be able to propose and investigate new semantic information, which could have been overlooked by the interpreter, to whom language is more than a complex set of formal units.
- Finally, the computer will also be used, in a more classical way, to calculate and establish relations between semantic structures, in order to characterize both these structures and the general performance of an interpreter. The latter aspect is only comparative, as one cannot define *the* correct interpretation of a given text: comparison can be performed between individuals and/or texts. Of course, computer tools can be used, through the technology of a man-machine interface, to synthetically present such semantic structures.

We will now explain more precisely what kind of semantic aspects our approach can take into account, through a presentation of the linguistic theory we used to develop it.

3 Overview of Interpretative Semantics

As said earlier, our approach was inspired by the linguistic theory developed by F. Rastier [4] : Interpretative Semantics. We will here only develop its main aspects which have had an influence upon our particular approach.

3.1 Principles and basic units

Generally speaking, Interpretative Semantics proposes a description of an interpretation, both as an act and as a result. Beginning with the latter, interpretation can be seen as the act of describing semantic units, not out of context, but located both in a text and a situation. The notion of situation here consists of an interpreter, along with his knowledge of the text, his goals, and his relation to the particular interpreted text (mostly social). Thus, the obtained meaning of a word, for example, is not a definition of this word, as could be found in a dictionary, but rather an explanation of this word's role in the text. Once again, the text and the interpreter are the ultimate decision makers, and they can widely modify the "classical" meaning of a given word.

As for the means of description themselves, this theory uses semantic features, or *semes*, which are tags attributed to local semantic units [1]. These local units, or signified, are called *sememes* and correspond to signifying units which can vary from simple words ('tax'), to compounds ('extreme-unction') and even phrases ('European Economic Community'). The expression of the semes themselves is done through the use of natural language, with no particular constraint : semes can again vary from simple words (/taxation/) to phrases (/economic politics/). Thus, interpreting a given text with such tools consists of a meta-linguistic description of the text's meaning, like some kind of commentary of its contents.

One of the particularities of this theory is to consider semes as tags upon *relations* between sememes, not upon the sememes themselves. It can thus be considered as a structuralist means of description, as it considers the semantic layer as a whole system, structured along simple relations between semantic units. The two main types of relations between sememes that are considered herein are similarity and opposition. As an example, the seme /taxation/ is not abruptly attributed to a sememe like 'income tax', but rather to express and qualify the similarity of 'income tax' to another sememe like 'excise'. In a more formal way, such a seme is attributed to a set of sememes, also called *taxeme*. The given seme can also be seen as attributed to every sememe that belongs to a taxeme, but taking into account the intermediate position of the class, we get a typed seme attribution : in this case, it is a generic seme.

The other kind of relation is opposition, which takes place between two sememes belonging to the same taxeme. As an example, once we have declared 'income tax' and 'excise' as similar, we can distinguish one

the text : the mere knowledge that one can have about a newspaper's style and topics already pre-activate a number of isotopies, even before reading a single word. In our case, as the newspaper we extracted the article from is specialized in political criticism, we can already activate at least a /politics/ isotopy. We will later propose a more formal description of such phenomena, when we have a more rigorous definition of the semantic units and an overview of their use in a cooperative interpretation.

As we take into account these general principles of Interpretative Semantics, we have chosen to develop our formal and computer models upon this theory because, in addition to the expressive power and ease of use the semes give to it, we can also rely on means to rationalize their attributions. This is an advantage over other semantic features theories such as [2]. As we saw that the general notion of interpretation considers semantic description as a partial and subjective one, the minimal concepts we presented above give us a way to explore an interpreter's point of view and establish its cohesion. As we will see, the computerized quest for this cohesion can lead to discoveries on the part of the user.

4 From theory to application : formal and computer models

Before presenting the different functions of our cooperative software, we will first describe the semantic data structure which it manipulates. This formal structure integrates, amongst others, the conceptual units we selected from the linguistic theory. The general framework of our model was inspired by set theory, in a simple and direct way. The reasons for this choice amongst a number of possibilities (logical models for instance) is mainly our wish to preserve transparency between the user's point of view and the processing of data. As the main point of our software system lies in representation and not in deduction, we preferred the overall simplicity of set theory operations. More about this type of modeling and its relation to logical models can be found in [10], and the use of this formal model for interpretative semantics is detailed in [7].

The structure described herein has one main purpose: that is to represent the elements of a user's interpretation of a text. As such, it does not have any connection with stabilized, dictionary-like data bases : all information stored in it depends on its single user.

4.1 Overview of the semantic data structure

We can present the structure as in the figure 1.

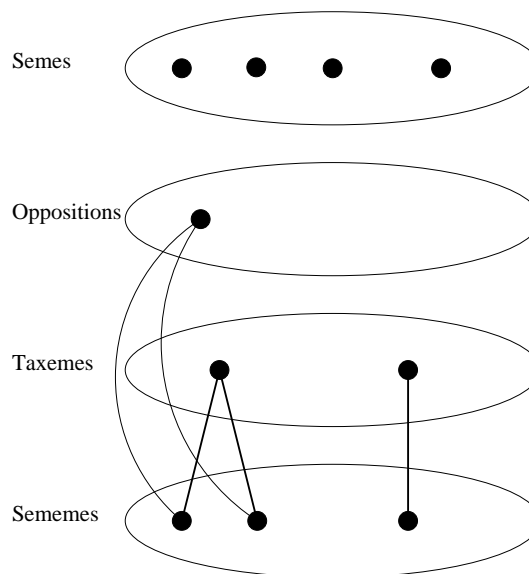


Figure 1: Generic semantic data structure

In the lower parts of the schema, sememes are represented in an unordered set S , while at the other end are the semes SE . The two middle layers correspond to the medial units of classes and binary oppositions that represent the two kinds of relations between sememes. Figure 2 shows an exemplified structure.

This schema can be read as :

- (1) The two sememes 'income tax' and 'excise' belong to the same taxeme
- (2) This taxeme is described by the seme /taxation/. Therefore, both 'income tax' and 'excise' have this seme as a generic one.
- (3) There exists an oriented opposition from 'income tax' to 'excise'.
- (4) This opposition is described by the seme /direct/.
- Thus, 'income tax' has a specific seme /direct/ to distinguish it from 'excise'

Note that, in a real-sized structure, nothing prevents a given seme from being used with different sememes and/or for different purposes (generic for some sememes while specific for others).

The relations between semes and other units are expressed by means of a function I , or *isotopy function*.

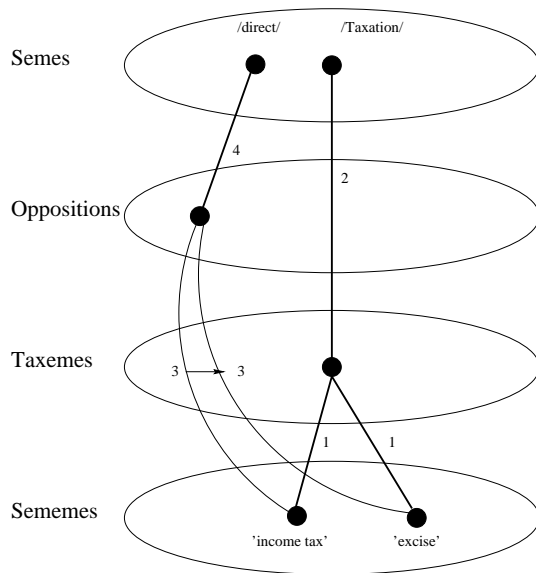


Figure 2: Sample semantic data structure

By using such a function, a given seme can be associated with a number of classes, oppositions and/or direct sememes (see the note below about this third kind of relation). These relations can, through a simple projection upon S , be reduced to simple seme-sememe relations. But the intermediary links (classes and oppositions) allows a classification of these seme attributions (as specific or generic). Thus, our notion of isotopy can cover the different possibilities presented above. In fact, a “classical” notion of isotopy can only be found through this projection. In our small example, the isotopy associated with the seme /taxation/ contains two sememes : 'income tax' and 'excise'.

Note : In our formal model, there exists a third link between seme and sememe, which is a direct connection. Its main use is to cover semantic drifts such as connotation, which could not be expressed because of the separation of taxemes. For example, the now well-known 'extrême-ponction' can so be declared as a full member of a /economics/ taxeme, while it can still be connected to /religion/ by this kind of direct link. Note also that such a double meaning can be expressed by the means of specific semes. This particular sememe can be distinguished from others in the same taxeme by using /religion/ as a specific seme.

The extended notion of isotopy plays the role of a general organizer for the whole data structure. As we

present the different steps of a structure's instantiation, we will see another aspect of isotopies.

In order to give such a structure a minimal formal cohesion, a number of formal constraints have been elaborated. These constraints can be expressed as “do and don't” regarding the different relations between semantic units. Some sample constraints are :

- A sememe belongs to exactly one taxeme
- A taxeme must have at least two elements
- A seme cannot be attributed to both a binary opposition that takes place in a class, and to the class itself
- A sememe inside a class must support at least one opposition qualified by a seme.
- etc...

All these constraints are self-explanatory and guarantee the desired nature of all semantic units. They also define a unique semantic identity for each sememe in the structure. A given structure which satisfies the whole set of constraints is said to be stable. This stability corresponds to a temporarily successful interpretation. Of course, interpretation can still continue, through the identification of new sememes or relations, but reaching this state is a big step from the user's initial vision. In order to obtain such stability, however, the interpreter needs to declare a relatively large number of semes to organize his/her interpretation. During this process, the constraints are not only seen as limits to what he/she can express, but mostly as requests to pursue his/her investigation of a text's meaning.

We will now present this data structure from the computer's point of view.

4.2 Towards the software system

The PASTEL system is used to manipulate and store a data structure such as the one presented above. PASTEL is a French acronym that stand for (approximately) “Computer-aid for the semantic analysis of texts, even literary ones”. It has been developed in Caml, and currently runs on Unix platforms ; a full source code can be obtained from the author on request. Its software architecture can be presented as in figure 3. Its main parts are the database that stores the semantic data structure, and a constraint detection engine that explores this database.

The upper part of the figure represents the classical means of interaction between the user and the computer : through representation, the computer presents

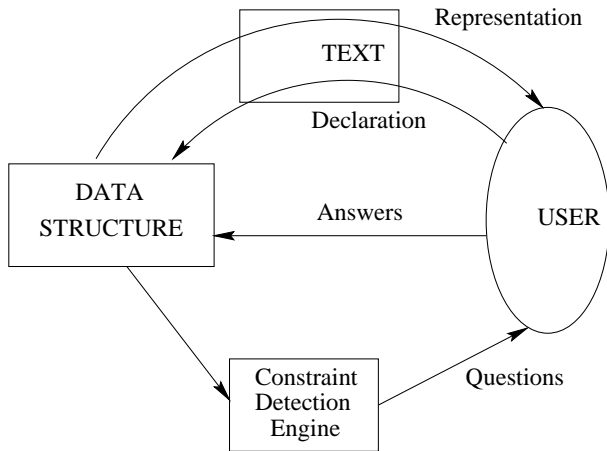


Figure 3: Software system architecture

the current state of the database to the user, both through extensive display of its contents or through the text that is being interpreted. On the other hand, the user can declare his interpretation through a selection of terms from the text and identification of semes. By doing this, he modifies and complete the database.

The lower part deals with the particular interaction method we have developed. The constraint detection engine browses the data structure and finds unsatisfied formal constraints, if any. For example, it finds that the class /Islam/ only contains one sememe : 'Mecca'. It then presents this fact to the user, by asking a simple question : Can you find another expression that deals with /Islam/ ? If yes, the user is requested to identify a new sememe, for example 'pilgrimage', and to place it in the taxeme. If no, then it has to move the sole sememe to another taxeme, etc. The answers to such questions are similar to "free" interaction in the upper part, but most of the information of these actions comes from the constraint that triggered it. In fact, these relations between man and machine are very motivating (and sometimes frustrating !) for the user.

Due to the relative size and complexity of both the semantic units and the relations between them, we preferred to develop these kinds of interactions, rather than let the user freely develop his/her interpretation. This can be justified as both a pedagogical objective of such interactive systems, and as a means of increasing suggestions from the machine towards the user, as we will see below. So we developed a dialogue protocol in order to establish a framework for the cooperative work.

5 Interaction protocol

This protocol is used to guide the user until he reaches the relative stability of the semantic data structure as defined by the satisfaction of a set of formal constraints. As the dialogue proceeds between man and machine, the number of constraints that are used and satisfied increases. As we will see, the dialogue starts with full freedom of speech from the user and ends with a restrictive list of precise questions in order to complete the database.

The developed protocol can be separated into the following steps :

- **Initial data declaration.** The user is absolutely free to attribute any number of semes to a number of extracts from the analyzed text, the sememes. Not all parts of the text does not have to be tagged, only those the user considers as significant enough.
Example : From our example text, the user can declare two main themes, /Religion/ and /Politics/, and for each of these, point out in the text the corresponding signifying units ('income tax,. 'excise', 'tax payer', 'President Chirac', 'election campaign', etc for the second one.)
The user can also split such isotopies-to-be into smaller ones, again with no constraints. For example, out of /Politics/ one can extract an /Economics/ theme, keeping only the first three units.
- **Class creation and first problem solving.** According to the initial data, the software will create semantic classes. It will possibly run into a non-partition case, when a given sememe belongs to several classes. If this occurs, it will ask the user to select a main class from the possible ones, inside which the oppositions which deal with this particular sememe will be settled. The other classes will become secondary ones, not relevant for the opposition phase (they will be expressed in the data structure as direct links).
- **Class structuring - first step.** For each such class, the user will first be asked to identify a number of oppositions, and to attribute one or several semes to them. The user is forbidden to use the seme corresponding to the class. This is done by selecting any pair of sememes from the class and proposing a seme to qualify the oriented relation between these two.
- **Class structuring - second step.** After that, the software will verify some constraints, for each class, requesting additional oppositions to

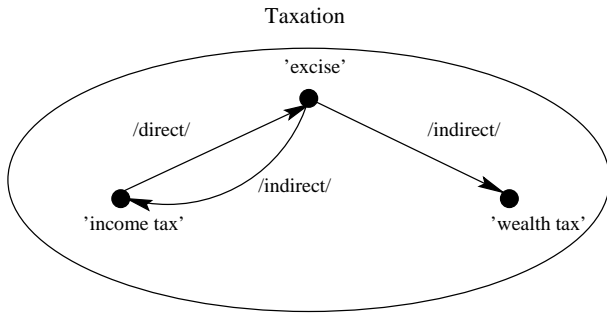


Figure 4: Sample class structuring graph

be qualified if needed. As the tagged oppositions between sememes of a given class form an oriented graph, as showed in figure 4, we require this graph to be complete (without taking orientation into account), and that for at least one edge to come from each node. It is also forbidden to qualify with the same seme two symmetrical oppositions : they are by definition asymmetrical.

- **Acquired stability.** At this stage, the semantic structure has all of its inner constraints verified. The user can visualize the isotopies he has created through this process, and locally modify the structure (add or remove a sememe, change classes, etc.), thus activating local constraint verification.

Let us focus now on the third and fourth steps, that concern the oppositions between sememes. These two steps are the most important, both theoretically and in time consumption. As sememe classification is an easy and natural task, their structuring by opposing similar sememes is a large step towards conceptual exploration. Some oppositions are also natural, like /direct/ to oppose 'income tax' to 'excise', but other ones need more reflection. It is during these steps that the suggestion-giving part of our approach is the most visible. In fact, such a need for a number of oppositions generally leads to interesting analogies across classes. As an example, inside a /Religious/ class, one has to characterize 'Lent', for an example through the use of a /deprivation/ specific seme (opposing it to 'mass'). Later, when having to characterize 'tax increase', the user has the possibility to re-use the /deprivation/ seme. This is done, in the interface, through presenting the user with all semes stored so far in the data base. The program has, of course, no means of "intelligently" proposing a particular seme, but still reminds the user of the results of previous interpretation steps.

Through a number of uses of the PASTEL program, this has generally led to such analogical discoveries. Some of them are too general to be of much interest, like /animated/ or /concrete/, but more precise ones still appear. These results are the reward for extensively investigating conceptual classes, through the use of the question-answer protocol. The user is "forced" to define more precisely a particular term he would have otherwise overlooked, once he has classified it in a semantic set.

6 Feedback to the user

Once the interpretation process can be considered as (temporarily) finished, the user has, of course, the possibility to examine a synthetic overview of its results. We discussed above the possible relations between established isotopies that can characterize the text's semantic structure. Other characteristics can be calculated that give some kind of judgment to the results. This is done through a number of tools, detailed as follows:

- **Richness of the general structure.** This can be expressed through the rough number of semantic classes and density of these classes' descriptions in terms of qualified oppositions.
- **Structure evolution.** This is the difference between the initial intuitions declared by the user at the beginning of the interpretation process and the final structure. There is no guarantee that the user discovers many new thematics through the use of PASTEL, but the simple number of new isotopies so discovered is a good clue to what the user found in the text.
- **Specific isotopies and inter-class relations.** As we saw above, the discoveries of analogies across apparently well-separated semantic classes occur during the two main steps of the interpretation process. Such easily-detectable isotopies are guaranteed to appear only during the intensive dialogue, and can, in turn, be themselves quantitatively characterized. Amongst such criteria, one can cite :

Number of classes. This is simply the number of different classes a specific isotopy has sememes in. If only one class is involved, there is no trans-class analogy. From two onwards, interesting semantic features are present.

Sememe / class ratio. For each class that such an isotopy involves, the number of sememes can be calculated, and the average number of sememes from the isotopy that belongs to the same

class is another aspect of this transversal phenomenon.

Classical quantitative features. Of course, the previously enumerated numerical qualifiers of isotopies still apply to specific ones.

- **Structure transformation.** In the case of a wide specific isotopy that crosses several different classes, it can be interesting to modify the whole structure so as to make a class out of this isotopy. In our example text, the /conversion/ specific isotopy, that takes place in both religious and political themes can become the /conversion/ taxeme. The main advantage of such a transformation is to enable the user to re-focus his interpretation on a newly discovered theme. Such reorganization will lead to the violation of constraints, and to another short dialogue between man and machine. The principle behind this transformation is the similarity between isotopies and semantic class, whose distinction is only due to specific semes.

Such final considerations can be suited for the comparison of different interpretations, be it from one interpreter to another, or from one text to another. Rather than judging an interpretation in the absolute, we propose a way to characterize its result from a cohesive point of view.

7 Conclusion and perspectives

As a conclusion to this presentation, we want to insist on the new kind of relation we propose between man and machine towards textual analysis. The theoretical program that lies behind our particular approach is wider than just this little interactive software. PASTEL has to be seen as a minimal example of such an interactive process, whose main disadvantage is its utter autonomy. The principle of computer aid for language processing cannot be extended too far without the use of huge amount of data, mainly to increase the suggestion-giving part of computer use. Here we have focused on structuring and representing subjective visions of a text's meaning. A further improvement to our software would in fact lie in the integration of a semantic dictionary, but still as a suggestion-only device, not as a decision-maker. Some parts of the protocol could thus be made lighter if the computer could suggest a possible answer that would well suit a user. But there are many obstacles, among which the flexibility of signifying units. By limiting description to simple words for example, in order to match dictionary entries, some interesting semantic phenomena could not be treated, and one would

have to describe much more units than the one the user selected as "interesting" ones. Our approach is different from full-scale parsing in that a user does not have to justify the origin of a semantic feature. For example, describing 'increasing taxes' through the use of a /negative/ seme does not need the use of rules describing the pejorative aspect of taxes, especially when they rise, from a supposed classical citizen point of view. The semantic aspects we deal with in this kind of approach are high-leveled, and only reachable with difficulty through a bottom-up compositional method.

The main uses of such software, be it in the state we have presented it or in a more automated fashion, can easily be thought of as education-oriented. The game-play part of computer for such academic work as literary studies, along with the well guided path it proposes in order to analyze a text can motivate students. The kind of texts that can be analyzed with this method are to be complex enough. Technical texts, for example, would not be as rewarding as literary or stylistically rich ones, but could still be tagged by such means. Perspectives exist toward such assisted tagging for individual-sensitive text classification in corpora. Through such aspects, one can easily connect our software with classical statistical tools, and propose a step forward in "soft" natural language processing.

Acknowledgments

I would like to thank Mrs Ormrod, from the Ecole Nationale Supérieure des Telecommunications de Bretagne for having corrected the many errors in this paper.

References

- [1] E. Coseriu and H. Geckeler *Trends in structural semantics*, Narr, 1981
- [2] J.J. Katz and J.A. Fodor *The structure of a semantic theory*, Language, 39, 1963
- [3] M. Olsen *Signs, symbols and discourses : a new direction for computer-aided literature studies*. Computers and the humanities, 27, 1993
- [4] F. Rastier *Sémantique Interprétative*, Presses Universitaires de France, Paris, 1987
- [5] F. Rastier, M. Cavazza and A. Abeillé *Sémantique pour l'analyse*, Masson, Paris, 1994
- [6] T. Snelgrove *A method for the analysis of the structure of narrative texts*, Literary and linguistic computing, 5.3:221-225, 1990

- [7] L. Tanguy *Traitement automatique de la langue naturelle et interprétation : contribution à l'élaboration d'un modèle informatique de la sémantique interprétative* PhD Thesis, Université de Rennes 1, 1997
- [8] Y.-M. Visetti *Des systèmes experts aux systèmes à base de connaissances : à la recherche d'un nouveau schéma régulateur*, Intellectica, 12, 1991
- [9] Y. Waern *Cognitive aspects of computer supported tasks*, Wiley, 1989
- [10] V.-H. Zaldivar-Carillo *Contributions à la formalisation de la notion de contexte : le concept de "théorie" dans la représentation des connaissances* PhD Thesis, Rennes 1, France, 1995.

