A Software for the Analysis of Scripted Dialogs Based on Surface Markers

Sylvain Delisle & Mathieu Dugré
Université du Québec à Trois Rivières
Département de mathématiques et d'informatique
Trois-Rivières, Québec, Canada, G9A 5H7
www.uqtr.ca/~delisle

and

Bernard Moulin
Université Laval
Département d'informatique, Pavillon Pouliot
Sainte-Foy, Québec, Canada, G1K 7P4
www.ift.ulaval.ca/~moulin

ABSTRACT

Most information systems that deal with natural language texts do not tolerate much deviation from their idealized and simplified model of language. Spoken dialog is notoriously ungrammatical however. Because the MAREDI project focuses in particular on the automatic analysis of scripted dialogs, we needed to develop a robust capacity to analyze transcribed spoken language. This paper presents the main elements of our approach, which is based on exploiting surface markers as the best route to the semantics of the conversation modelled. We highlight the foundations of our particular conversational model and give an overview of the MAREDI system. The latter consists of three key modules, which are 1) a connectionist network to recognise speech acts, 2) a robust syntactic parser, and 3) a semantic analyzer. These three modules are fully implemented in Prolog and C++ and have been packaged into an integrated software.

1. THE *MAREDI* PROJECT

Research in computational linguistics is influenced by the increasing importance of a number of application domains including text translation, the analysis of texts for knowledge acquisition or information retrieval, text summarization and the analysis of conversations. However, most systems manipulating natural language inputs cannot deal with entries that do not conform to their "ideal model of language", which limits their use in practical applications. All these systems, as traditional natural language processing systems, are based on the assumption that the texts to be analyzed follow the ideal grammar that is implemented in the system. Oral inputs are notoriously difficult to analyze because of their imperfections relative to conventional natural language grammars: misspellings, ungrammatical constructions, word omissions, repetitions, misuse of words, etc. Hence, finding flexible and robust approaches to the analysis of oral discourse is a major challenge for the years to come.

In the MAREDI Project (MArkers and REpresentation of DIscourse), we worked on written transcriptions of task-oriented conversations and explored how surface markers detected in the text sentences can help determine the text's semantic content in order to build a conversation model. We propose a new approach to the automatic analysis of conversations which combines the exploitation of surface markers and the use of a robust syntactic parser to deal with a variety of ungrammatical constructs (repetitions, missing or unnecessary terms, ellipses, etc.).

A detailed account of this work will appear in the *Journal of Natural Language Engineering* (Cambridge University Press): "Surface-Marker-Based Dialog Modelling: A Progress Report on the MAREDI Project", by Sylvain Delisle, Bernard Moulin, and Terry Copeck.

2. AN EXAMPLE DIALOG

This project is not one of speech processing: the reader interested in such matters can refer to the abundant literature on speech processing—see [16] and [11]. In fact, we worked on a corpus of transcriptions of dialogs in which an instructor gives directives to a manipulator in order to draw a geometric figure. The instructor and the manipulator were located in different rooms, each being in front of a terminal. They interacted using a microphone system through which the dialogs were recorded. The instructor gave directives to the manipulator who drew a picture on her terminal and simultaneously, the instructor could see it on her own terminal. Here is an example of an actual dialog in which we can observe various anomalies—'Inst' stands for instructor, 'Man' for manipulator, and i_x and m_y are the utterance's identifiers.

Inst (i_1)	*uh now you take a small a small square*
Inst (i_2)	*and put it in the bottom left corner*
Inst (i_3)	*in order to draw a house to uh*
Man (m_4)	< drawing gesture >
Inst (i_5)	*now you take another one*
Man (m_6)	< taking gesture > + < dragging gesture >
Man (m_7)	*with a roof, right?*
Inst (i_8)	*no no*

In this example we find a good sample of the main characteristics of these dialogs. Sentences are usually ill-formed (word repetitions, utterance interruptions, uncompleted statements) and noisy (i.e. *uh*). They contain various surface markers (i.e. *now*, *right*, *in order to*) which can be used to determine the kind of speech act that the locutor is performing. These conversations are multi-modal dialogs in which speech and gestures are combined in order to build the dialogue meaning (for instance the gesture that the manipulator performs in m_6 indicates implicitly that she accepts the goals set by the instructor in utterances i_1, i_2 and i_3). The initiative is mixed and can be taken opportunistically by either locutor.

3. THE CONVERSATIONAL MODEL

A discourse results from the linguistic productions that the involved agents (persons, software) create cooperatively. Agents choose the form and content of their utterances in a conversation, taking into account the context of the interaction, their objectives and knowledge, as well as their social relations with other participants. In technical terms we can say that an agent's mental states (beliefs, intentions, preferences, wishes etc) provide the conceptual background for the accomplishment of her linguistic actions or speech acts. Furthermore, they choose the content of their utterances and perform speech acts so as to influence the mental states of the other agents participating in the conversation.

In previous work ([15], [12]) we developed a multi-agent model to simulate conversations between software agents which can communicate together, reason about their mental states, and plan and perform non-linguistic actions as well as speech acts. The speech acts which such agents perform contain conversational objects (COs). A conversational object is a mental attitude (belief, goal, wish, etc.) along with a positioning that the speaker transfers to the addressee during a conversation ([13]). The speaker takes a position with respect to a mental attitude by performing actions like proposing, accepting, rejecting: this is called the speaker's positioning relative to that mental attitude. COs are aggregated in the model of the conversation contained in a persistent memory managed by a special agent called the conversation agent. Agents participating in the conversation can reason about the content of their own mental models as well as the model of the conversation.

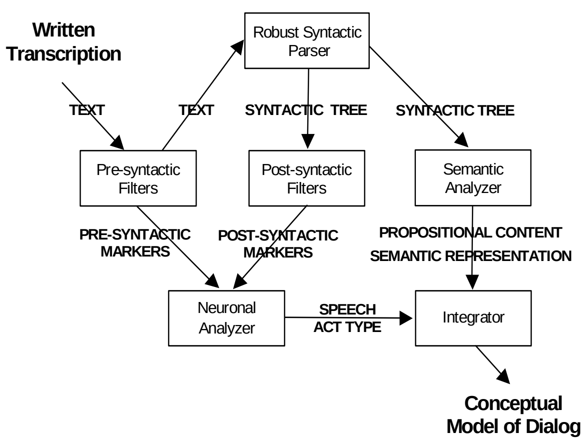
A conversation is modeled as a network of COs and the locutors' positionings with respect to them. Presume the instructor says to the manipulator, *Uh now you take a small a small square*. In doing so she proposes—this is the positioning—to the manipulator a CO, the mental state whose goal is that the manipulator take the small square. If the manipulator answers *OK*, she positions herself by accepting the proposed CO and integrates the corresponding mental state into her mental model. The manipulator could also refuse or request an explanation from the instructor. These would correspond to other types of positionings: refusal and inquiry. The manipulator could even propose alternative COs. In this way the conversation unfolds as a network of COs created by the locutors' speech acts. Such a model of the conversation provides conceptual structures to represent an unfolding conversation and the various exchanges of mental states performed by its agents: it illustrates the negotiation process which takes place during a conversation. These structures can be used by agents to perform different kinds of reasoning ([12]).

4. THE *MAREDI* SYSTEM

The objective of the MAREDI project is to develop a system for analyzing transcriptions of oral dialogs in order to generate a conceptual representation of the conversation as a network of conversational objects (COs). We assume that the various markers found in a discourse can be used to orient analysis ([4]), and more specifically, to identify the categories of locutors' speech acts ([6]). The main steps of our automated analysis are:

1. Divide the original text into utterances;
2. Remove noisy terms such as *uh*;
3. Identify markers such as *now*, *right*, *so* in utterances and extract them;
4. Proceed with the syntactic analysis of utterances: well-formed, ill-formed, or fragmentary;
5. For each utterance, identify the category of speech act performed by the locutor. This pragmatic analysis takes into account the numerous indirect speech acts found in oral dialogs;
6. Based on the category of speech act, identify the locutor's positioning and the type of mental state it applies to, using a correspondence table between categories of speech acts and (positioning, mental state) couples;
7. Proceed with a semantic analysis of the speech act's propositional content based on the results of the syntactic analysis;
8. Create the conversation's conceptual model.

The MAREDI System is an analyser of transcriptions of oral dialogs. In these written dialogs, we found elements which can be composed of one or several sentences (which are not always well-formed) or even only the indication of a gesture. These elements correspond to the verbal and non-verbal actions performed by the locutors and are marked by the identity of the corresponding locutor. Written transcriptions of oral dialogs are processed by a Pre-syntactic Filter which detects lexical markers (such as *Uh*, *right*, *OK*). The filtered text (which may be fragmentary) is processed by the Robust-Syntactic Parser that generates a syntactic tree and sends it to a Post-syntactic Filter. This filter identifies syntactic markers (such as imperative or interrogative constructions) and sends them to the Neural Analyzer which processes them along with the lexical markers provided by the Pre-Syntactic Filter. The Neural Analyzer identifies the category of speech act that was performed in this utterance and sends it to the Integrator. The syntactic tree is also sent to the Semantic Analyzer which determines the semantic interpretation of the speech act's propositional content. The Integrator integrates the information provided by the Neural Analyzer and the Semantic Analyzer in order to generate a conceptual representation of the utterance which is integrated in the Conceptual Model of the Dialog. The diagram below shows MAREDI system's architecture.



5. THE THREE CENTRAL MODULES

The **Neural Analyzer** ([4], [6]) is a localist neural network, i.e. one without learning capabilities, which takes the text markers as input and proposes corresponding speech act categories. The core of the Neural Analyzer is a sub-network (or subnet) that matches combinations of markers with categories of speech acts. In many cases more than one speech act category corresponds to a given combination of markers. For instance, *like that* may denote either a question or a confirmation or even an assertion about the manner in which to do something.

The Neural Analyzer contains two subnets that take into account the roles played by the speakers (i.e. manipulator and instructor) and the context of the utterance (i.e. the particulars of the preceding utterance). These two subnets allow the system to limit the number of possible continuations for one utterance in terms of the markers found in the following utterance. Colineau ([6]) showed experimentally that the Neural Analyzer is able to analyze almost 85% of the utterances of the drawing task corpus and 88% of the utterances of a second task-oriented corpus. She also showed how to improve these results.

The **Syntactic Parser** ([1], [2]) is composed of four modules: 1) the Supervisor coordinates the interactions of the three other modules; 2) the Kernel-Analyzer uses a standard grammar of written French in order to process well-formed utterances; 3) the Recovery Module detects anomalies (repetitions, interruptions, noise and ellipses) in ill-formed utterances and corrects them, transforming the syntactic tree into the canonical form required by the Semantic Analyzer; and 4) the Lexical Ambiguity Resolver validates the syntactic category of each word in the utterance. The system uses a specification of the French grammar based on Chomsky's theory of Government and Binding ([3]). According to this theory, syntactic rules take into account the argument structure of verbs, nouns and adjectives; this greatly facilitates the resolution of lexical and syntactic ambiguities. Chomsky's theory also provides transformation mechanisms which are used by the Recovery Module. Anomalies are detected using heuristics which we developed after analyzing our corpora. The proper transformations are then applied to restore an ill-formed utterance to a syntactically correct structure. The Syntactic Parser is thus able to analyze not only well-formed French sentences but also a number of ill-formed structures, providing at least fragments of syntactic trees.

We tested the current version of the parser on two corpora of 100 utterances apiece. Results for undistorted utterances, i.e. well-formed statements, are worse than for distortions. The main reason proved to be gaps in the extent of linguistic phenomena covered by the core grammar. Notwithstanding this, we are satisfied with the performance of the distortion repair module. The experiments verified the two important points: 1) a variety of oral distortions were repaired successfully and without loss of information; and 2) analysis applied the core and peripheral processes in correct order. The heuristic recovering from noise was particularly effective, successfully handling 87.5% of instances on average. Repetitive inputs were repaired with only slightly less success; 86% of instances were corrected without losing information. These experiments show clearly that it has been productive to focus our efforts on handling distortions in the input. They show equally that it is time to shift our attention to increasing the coverage of the core grammar.

The **Semantic Analyzer** ([9], [10]) is based on case-based analysis ([8], [7]) and on a template-based and inference-driven mapping approach ([14]). This approach takes advantage of the correspondence between syntactic patterns and semantic cases. An utterance is composed of a modal part (tense, mode, modality) and a clause. The clause is composed of a main predicate (the verb) and a collection of nominal components (nouns and adjectives) linked to the verb by various grammatical relations. The objective of the analysis is to identify the concepts, or predicates, corresponding to the verb, nouns, and adjectives in an utterance and the case relations among them. Our analytic strategy is centered on a semantic characterization of verbs found in utterances. We use a dictionary of concept schemata ([17]) for verbs used in our corpus. Each verb has one or more schemata (or templates) in which its meaning is linked to acceptable concept types by conceptual relations or semantic cases. Cases may be mandatory or optional, enabling the system to analyze similar sentences which omit certain cases. The Semantic Analyzer uses a concept type lattice ([17]) to check that words found in utterances meet the semantic requirements of verb cases. For each word found in our corpus, it lists the possible concept types. This lattice is also very useful for anaphoric resolution. The Semantic Analyzer also uses a set of syntactic patterns of two kinds derived from analysis of our corpora. Modal patterns identify elements of an utterance's modal part. Clausal patterns help the system locate those parts of the syntactic tree that express the verb concept's semantic cases. Each syntactic element is thus associated with a semantic role and validated against the acceptable concept types for a given schema.

6. THE INTEGRATED SOFTWARE

The three central modules described in Section 5 were originally developed separately by three different people, over a period of a couple of years, using different programming languages. Communication between these modules was either simulated or tested via semi-automated means. Despite completion of the independent central modules, we were left with a challenging task: the actual integration of all components into a coherent software package. This is what we managed to accomplish recently: this paper is the first to report on it.

The integrated software that we have developed runs on a PC, under Microsoft Windows, and is fully implemented in LPA Prolog (http://www.lpa.co.uk/) and C++. LPA Prolog was used to create the graphical user interface for the integrated software, as well as a compiled executable version. Predicates used for handling windows and graphical elements, along with an access to core functions of the operating system, allowed the integration of all the components of the MAREDI software. Overall, we were quite satisfied with LPA Prolog.

The Neural Analyzer was initially implemented with Visual C++, version 4, as a normal application with a graphical user interface. We first ported it to Visual C++ version 5. Then, in order to use this module with the integrated software, it had to be recompiled as a Windows Dynamic Link Library (DLL). The C++ methods used to create the neural network and collect the results were exported in this DLL. LPA Prolog could then load and execute the library with the *winapi* predicate. This predicate has access to all the kernel functions of Windows, such as `LoadLibraryA` which loads a DLL inside memory in an area shared by all applications. In the case of the Neural Analyzer, the *winapi* predicate calls the `start_session` method of the DLL with a list of markers. The method then returns the output of the Neural Analyzer, i.e. the most probable speech act.

The Syntactic Parser and the Semantic Analyzer were originally developed with Sicstus Prolog—actually, some early elements were even developed with Quintus under Unix. Generally speaking, the integration of these two central modules within LPA Prolog was more natural. However, some tinkering was necessary to make them work correctly with LPA Prolog. For example, the "nth" predicate returns the N^{th} element of a list. This predicate is present both in LPA Prolog and in Sicstus Prolog, but their definition are different: this type of error was difficult to detect. For this reason, a small library was developed to simulate the behaviour of Sicstus Prolog in LPA Prolog.

Another example of the use of DLLs is the dictionary used by the parser. The Syntactic Parser needs a large dictionary in order to recognize and analyze all the words in a sentence. This dictionary is much too large to be stored entirely in main memory. Our experiments showed seriously degraded performance when the entire dictionary was loaded in main memory. For this reason, a DLL was created to locate the words present in the current input sentence to be analyzed, and all other words which may be of relevance for this analysis. Only this small set of words was fetched from the dictionary and then loaded in main memory.

The next page presents two figures showing screens from the current version of the integrated MAREDI software. **Figure 1** shows the initial user interface (screen) when the MAREDI application is launched. It allows the user to select the file containing the text to be analyzed, via the *Parcourir* (browse) button, and then begin the analysis with the *Lancer une session* (begin analysis) button, which triggers the screen of **Figure 2**.

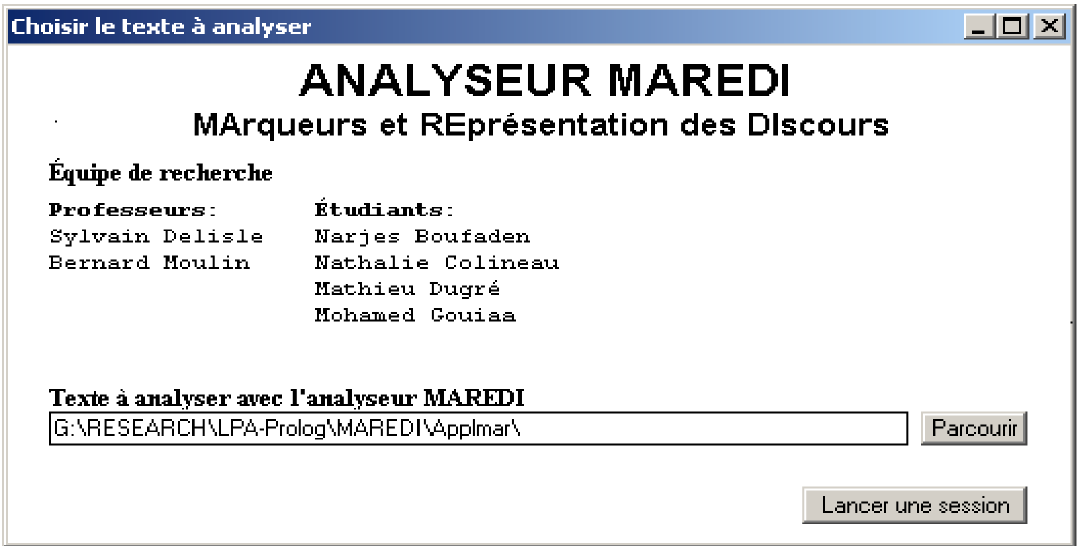


Figure 1: Initial user interface (screen) when the MAREDI application is launched.
(*Parcourir* = browse ; *Lancer une session* = begin analysis, which triggers the screen of Figure 2.)

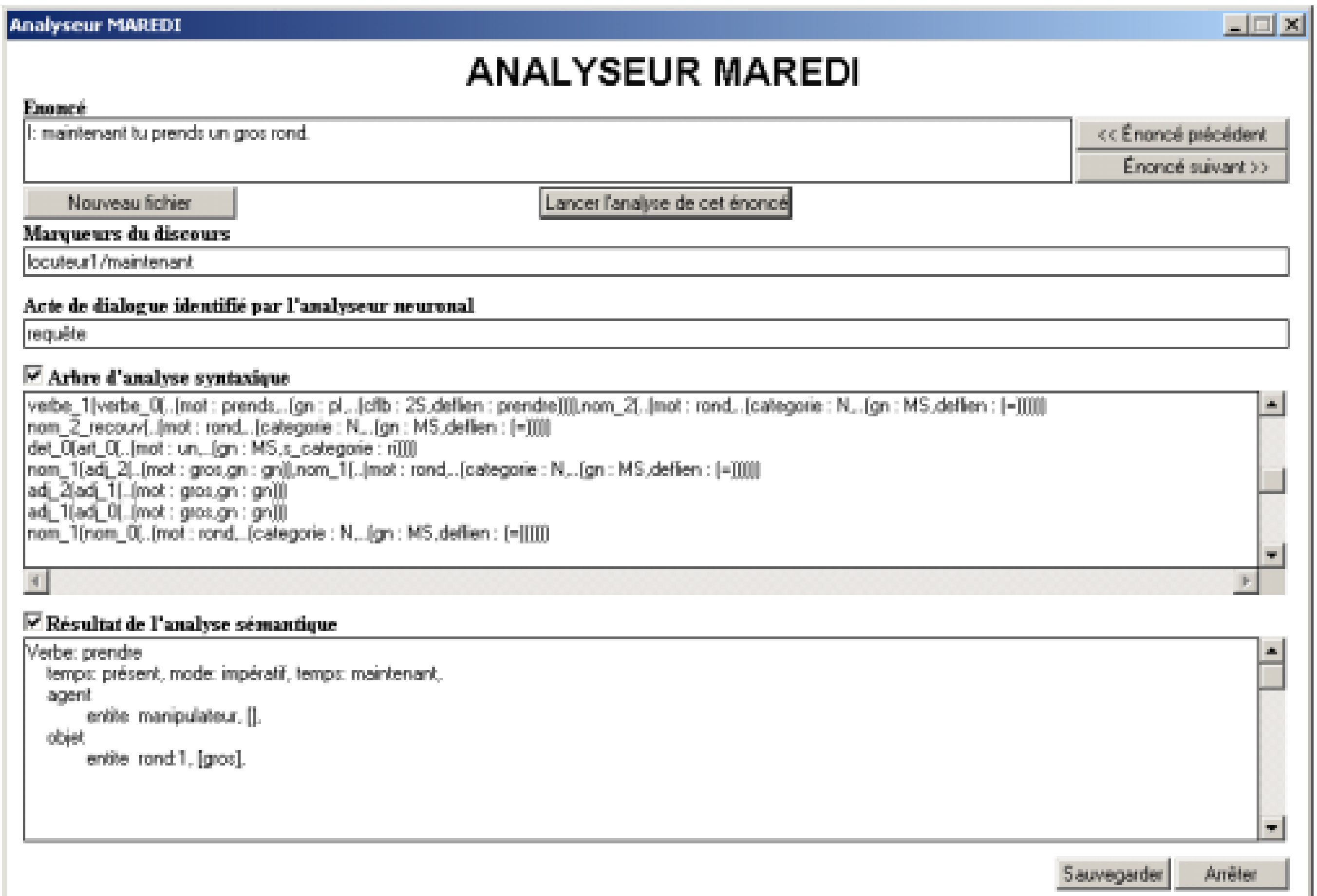


Figure 2: The main screen of the MAREDI integrated software showing the analysis of the input *maintenant tu prends un gros rond* (now you take a large circle).
(*Enoncé* = input ; *Enoncé précédent* = previous input *; Enoncé suivant* = next input ; *Nouveau fichier* = new input file ; *Lancer l'analyse de cet énoncé* = launch the analysis of the current input ; *Marqueurs du discours* = discourse markers ; *Acte de dialogue identifié par l'analyseur neuronal* = speech act identified by the Neural Analyzer ; *Arbre d'analyse syntaxique* = syntactic analysis produced by the Syntactic Parser ; *Résultats de l'analyse sémantique* = semantic analysis produced by the Semantic Analyzer ; *Sauvegarder* = save ; *Arrêter* = stop)

7. CONCLUSION

The MAREDI project involved several people over a period of five years. Our goal in this brief paper was to present its salient features and report on the recent development of the integrated software: obviously, lots of details had to be discarded. We invite the interested reader to consult our publications (see Section 9, or visit our Web pages) for further details. We believe the work related to the Neural Analyzer and the Syntactic Parser is particularly interesting due to its originality, especially in the context of the analysis of scripted dialogs. Also of interest is the integrated approach we took to the problem.

The development of an integrated version of the MAREDI software was also the occasion to discover weaknesses. For instance, since each of the three central modules were developed more or less independently by different people, different assumptions had to be made at certain moments in time to allow them to progress in their own project. When these modules were finally integrated, discrepancies between what one module produced for another module and what the latter expected from the former became apparent. Another source of problem was the fact that the three central modules were individually tested on various subsets of our corpus of scripted dialogs. Thus, when came the time to perform end-to-end testing with the integrated software, results were fluctuating drastically, depending on which set of inputs was used. This is why we decided not to present overall evaluation results here since these were meaningless.

Future work would be required in order to complete and improve our first implementation of the integrated version of the MAREDI software as described here. Extra work would also be necessary to develop to its full extend the implementation of our conversational model (Section 3). What the current version produces is the elements of this model: they would have to be integrated and dsiplayed in an appropriate graphical representation.

8. ACKNOWLEDGEMENTS

We would like to thank the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche de Québec (FCAR) and the Natural Sciences and Engineering Research Council of Canada (NSERC) for granting financial assistance to our research efforts. We also thank the following people for significant contributions to the MAREDI project: Nathalie Colineau for her work on the Neural Analyzer, Narjès Boufaden for her work on the Syntactic Parser, and Mohamed Gouiaa for his work on the Semantic Analyser. Mathieu Dugré integrated the various MAREDI software components, as reported in this paper. Finally, we wish to thank Terry Copeck for his careful editing work on portions of this paper.

9. REFERENCES

[1] Boufaden N. (1998). Analyse Syntaxique Robuste des Textes de Dialogues Oraux. Masters Thesis, Department of Informatics, Université Laval.

[2] Boufaden N., S. Delisle & B. Moulin. (1998). Analyse syntaxique robuste de dialogues transcrits: peut-on vraiment traiter l'oral à partir de l'écrit? *Proceedings of the Fifth Annual Conference on Automatic Natural Language Processing (TALN-98)*, 112-121.

[3] Chomsky N. (1981). *Lectures on Government and Binding. Studies in Generative Grammar*, Dordrecht: Foris.

[4] Colineau N. & B. Moulin (1996a). Proposition d'un modèle connexionniste pour la détermination des actes de dialogue - Application à un corpus de dialogues multi-modaux finalisés, rapport de recherche n°9601, Département d'informatique, Université Laval, Canada, juin 1996, 95 pages.

[5] Colineau N. & B. Moulin (1996b). “Un modèle connexionniste pour la reconnaissance d'actes de dialogue”, *Actes de la conférence Informatique et Langue Naturelle ILN'96*, 157-174.

[6] Colineau N. (1997). Etude des marqueurs discursifs dans le dialogue finalisé. Thèse de 3ème cycle de l'Université Joseph Fourier, Spécialité Sciences Cognitives, décembre 1997, 283p.

[7] Delisle S., K. Barker, T. Copeck and S. Szpakowicz. (1996). Interactive Semantic Analysis of Technical Texts, *Computational Intelligence* 12(2): 273-306.

[8] Fillmore C. (1968). The Case for Case. In E. Bach and R.T. Harms (eds.) *Universals in Linguistic Theory*. Holt, Rinehart and Winston.

[9] Gouiaa M. (1998). Modélisation conceptuelle de conversations à partir de l'analyse de textes de dialogues transcrits : analyseur sémantique et générateur de la structure conversationnelle. Masters Thesis, Department of Informatics, Université Laval.

[10] Gouiaa M., S. Delisle & B. Moulin (1999). “L'analyse sémantique de dialogues oraux transcrits”, *Actes de la Sixième Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN-99)*, Cargèse (Corse), 12-17 juillet 1999, 339-344.

[11] Jurafsky D. & J.H. Martin. (2000). *Speech and Language Processing (An Introduction to Natural Language Processing, Computational Linguistics, and Speech recognition)*. Prentice Hall.

[12] Moulin B. & D. Rousseau. (1994). An Approach for Modelling and Simulating Conversations. In D. Vanderveken et S. Kubo (eds.), *Essays in Speech Act Theory*, John Benjamins.

[13] Moulin B., D. Rousseau & G. Lapalme (1994). A Multi-Agent Approach for Modelling Conversations, *Proc. of the Intl. Conf. on AI and Natural Language*, Paris, 35-50.

[14] Palmer M. S. (1990). *Semantic Processing for Finite Domains*, Cambridge University Press.

[15] Rousseau D., B. Moulin & G. Lapalme (1994). Une approche multi-agents pour modéliser les conversations, *Actes des Deuxièmes journées francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, 3-14.

[16] Smith R.W. & D.R. Hipp (1994). *Spoken Natural Language Dialog Systems: A Practical Approach*, Oxford University Press.

[17] Sowa J. (1984). *Conceptual Structures: Information Processing in Mind and Machine*, Addison Wesley.