

A DISTRIBUTED MULTI-AGENT ARCHITECTURE FOR NATURAL LANGUAGE PROCESSING

Danilo Fum^o, Giovanni Guida*, Carlo Tasso*

^o Dipartimento dell'Educazione, Università di Trieste, Italy

* Dipartimento di Matematica e Informatica, Università di Udine, Italy

Abstract - The paper presents a distributed multi-agent architecture for natural language processing. This architecture proposes a novel concept of distributed problem solving, which incorporates in a unitary framework the following key-points: large-grained heterogeneous agents, centralized knowledge-based control, and mixed event-driven and goal-driven operation. It provides, moreover, a flexible tool for the design of natural language processing systems, both motivated from the cognitive point of view and computationally effective and robust. The proposed architecture has been implemented in a fully running prototype system, and has been successfully applied in the domain of text understanding.

1. Motivations and objectives

Building artificial systems for natural language processing capable of performing with the same ability of humans, still presents several challenging issues. This complex cognitive task implies, in fact, to model and replicate knowledge, capabilities, and skills that humans exploit in their linguistic activity. The main point is therefore that of identifying these skills and capabilities, discovering the knowledge on which they are grounded, and figuring out how this knowledge can be effectively put into use. Thus, natural language processing turns out to be a knowledge intensive task, based on the cooperation among a variety of different knowledge sources.

Traditionally, natural language has been considered by linguists as organized into a series of levels, whose structure, organization and function are to some extent independent from each other. Linguistic knowledge has been considered, therefore, as comprising a number of different components, each one related to a different functional level and cognitive process (e.g., morphology, syntax, semantics, pragmatics, etc.). One of the first ideas in designing natural language processing systems has been to take advantage of this stratification by organizing the comprehension process in a sequence of stages, each one operating on the output produced by the preceding one and providing input for the following. Such staged organization shows, however, several drawbacks. From a strictly computational point of view, a system organized in a sequential way is generally brittle. If a problem appears in an early stage the whole process fails, if, on the other hand, it arises only in the last stages most of the work done is wasted. Every stage represents a separate bottleneck for those that follow it, and no feedback among the various stages is allowed. Moreover, if we look at the language processing activity in humans, it is clear that a simple

sequential model is inadequate from a cognitive standpoint (Marslen-Wilson, 1975; Tyler and Marslen-Wilson, 1977; Johnson-Laird, 1983). To overcome these difficulties, in the last few years several other kinds of organization have been proposed which are more flexible, robust, and psychologically reliable. It is possible, for example, to maintain the advantages of stratification while increasing the flexibility and power of a natural language processing system by separating the structural levels from the processing sequence. More precisely, it seems appropriate to adopt a distributed organization, where the natural language processing problem is decomposed into a number of distinct subproblems, each one being tackled by a dedicated problem solver. Depending on the criteria used for operating this kind of decomposition, several approaches may result, each one featuring specific characteristics (Cullingford, 1981; Huang and Guthrie, 1986; Small, 1981; Slack, 1986; Waltz and Pollack, 1985).

This paper presents an approach to natural language processing, which proposes a novel distributed problem solving architecture. This is based on the concept of incremental, cooperative problem solving, and features two main distinctive points. First, it includes a set of large-grained, possibly heterogeneous, independent specialists, each one embodying competence in one of the several aspects of language processing (e.g., morphology, syntax, semantics, common sense inference, anaphora resolution, quantification, temporal reasoning, etc.). Second, it is based on a centralized, knowledge-based control concept, which can implement flexible problem solving strategies including both event-driven and goal-driven operation. The design of this architecture relies upon the results of a wide experimental activity (Costantini, Fum, Guida, Montanari, and Tasso, 1987; Fum, Guida, and Tasso, 1987), and has been exploited and tested in the implementation of a prototype system for descriptive text understanding.

2. Requirements

As a first step towards the goal of designing a novel distributed architecture for natural language processing which will be both computationally effective and psychologically reliable, a thorough analysis of functional requirements has been carried out. The main results of this analysis are reported below.

1. The architecture should support functional decomposition of the natural language processing problem at hand into interacting subproblems that correspond to the main cognitive tasks which are faced during natural language processing, such as

morphology, syntax, semantics, common sense inference, anaphora resolution, quantification, reference, temporal reasoning, etc. The expected number of subproblems should be limited and their granularity medium to large.

2. The architecture should host a collection of individual problem solvers, each one devoted to a single subproblem. Furthermore, each problem solver should be independent from the others, possibly exploiting different representation and reasoning techniques. It should be possible to develop problem solvers independently and to run them in parallel.
3. While no problem solver is supposed to have enough competence to solve the entire natural language processing problem at hand, no constraints are put on the number of problem solvers which might be devoted to a single subproblem. The resulting system may thus be redundant so as, whenever appropriate, a single subproblem can be tackled from several different viewpoints, thus extending the overall capabilities of the global system.
4. The architecture should provide a specific dedicated mechanism for controlling the cooperation among the individual problem solvers, in order to direct the global system behavior towards the complete solution of the natural language processing problem at hand. This mechanism should allow full separation between knowledge about the specific problem solvers and knowledge about strategies and methods for the global problem solving task.

3. Architecture overview

The above requirements have motivated the design of an architecture where a set of autonomous agents, called specialists, cooperate together in order to solve a natural language processing assignment. Specialists are large-grained and heterogeneous. None of them is capable to solve the whole problem at hand, and more than one of them can be devoted to the same subproblem.

The architecture is split into two different levels, namely: a cooperation level, where cooperation and interaction among specialists is dealt with, and a problem solving level, where problem solving in the specific subject domain of each individual specialist takes place. Cooperation level activities are centralized and they are performed by a single dedicated specialist called the cooperation manager. This is specifically devoted to identify and implement appropriate solution strategies for the problem at hand, and to coordinate the activity of the other specialists at problem level towards the achievement of a global goal. Specialists at problem level do not have any mutual knowledge or self knowledge: only the cooperation manager knows about specialists and their competence. The architecture is conceived to operate in a multi-processor environment, where all the specialists (the cooperation manager included) can operate in parallel.

4. Basic mode of operation

The basic mode of operation of the proposed architecture at cooperation level is iterative co-routined (Lesser and Corkill, 1981): tentative partial results produced by single specialists are progressively accumulated and iteratively revised under the supervision of the cooperation manager, until the desired final result is produced. At problem level, specialist operation is organized in an "assign-execute-report" fashion: each specialist works at specific and precise tasks assigned to it by the cooperation manager, and whenever it obtains some (positive or negative) result, it reports to the cooperation manager.

Communication among specialists is achieved through a message-passing mechanism, which allows exchange of information between the cooperation manager on one side and the specialists on the other. More specifically, messages are devoted to carry control and coordination information, and to allow the cooperation manager to have

a full visibility of the problem level activities carried on by the specialists. Direct specialist to specialist communication is not permitted, according to the above mentioned choice of centralizing knowledge about specialist capabilities in the cooperation manager.

A message from the cooperation manager to a specialist may concern: (1) the assignment of a new problem to solve, (2) the answer to a help request issued by the specialist. On the other hand, a message from a specialist to the cooperation manager may concern: (1) the solution to an assigned problem, when the specialist has succeeded in its problem solving activity, (2) a fail announcement, when the specialist is unable to solve an assigned problem, (3) a help request, when the specialist has been successful in decomposing and partially solving an assigned problem, but it needs help from other specialists to proceed further in the solution process.

A major effort in defining the above mentioned communication mechanism has been devoted to the design of the interface between specialists and communication manager, which constitutes the only logical link among specialists.

5. The specialists

As already mentioned in the previous sections, specialists can be designed and implemented according to any approach (algorithmic, non-deterministic, knowledge-based, etc.) which might be appropriate for each specific natural language processing activity. They share, however, a common interface towards the architecture: the way they communicate with the cooperation manager and manage internal problem solving tasks is the same for all of them. The general structure of a specialist is divided, therefore, into two parts: (1) the interface, devoted to handle the links with the external world (the rest of the architecture), and (2) the processor, which implements the actual problem solver in the specific subject domain of the specialist. From the architectural point of view, only the interface is of interest here.

The main point to be considered in the design of the interface is that specialists operate concurrently, and the cooperation manager can request to the same specialist the solution of a new subproblem, before the previous assignment has been completed. For example, a syntax specialist could be asked to find the subject of a given sentence, while already engaged in checking a noun-pronoun agreement in another part of the text. Therefore, a specialist has to be able to manage more than one assignment at a time. Also, the processing status of each assignment can be: (1) active: the processor is currently working on it, (2) ready: waiting for the processor to work on it, or (3) suspended: a help request for that assignment has been issued in the past and processing has to be delayed until an answer to the help request will arrive from the cooperation manager. The interface includes therefore appropriate policies to manage these internal scheduling problems.

In the experimental activity performed so far concerning descriptive text understanding, nine specialists have been implemented, devoted to the following competence domains: morphology, dictionary look-up, syntax, access to encyclopedic knowledge, semantics (two specialists), quantification, reference, and time.

6. The cooperation manager

The main capability of this dedicated specialist is to manage the global problem solving strategy and to dynamically assign to the various specialists specific subproblems to solve. Moreover, the cooperation manager is able to appropriately react to events happening at the problem level, i.e. to messages coming from the specialists, namely: solutions to previous assignments, fail announcements, or help requests. To these purposes two kinds of operation modes have been designed: (1) goal driven or top-down, where the cooperation manager

develops autonomously its own problem solving strategies and assigns appropriate tasks to problem level specialists, and (2) event driven or bottom-up, where the cooperation manager acts as an intelligent dispatcher of the messages received from the specialists. These two modes of operation are dynamically combined, in such a way as to assure high levels of flexibility, adaptativity, and cognitive evidence.

The cooperation manager is designed using knowledge-based techniques, and incorporates explicit knowledge devoted to support cooperation level reasoning. Due to the large variety of cognitive tasks, competence and skills which should be covered by the knowledge base of the cooperation manager, both procedural and declarative representation paradigms have been utilized. The first one, implemented through event-graphs, is devoted to encode structured strategic knowledge, which is in a sense precompiled and immediate and requires little or no explicit reasoning. The second, implemented through production-rules, concerns fragmentary unstructured knowledge needed to implement more complex and sophisticated activities, where deep reasoning has a critical role, such as decision making, planning, error recovery, etc. Procedural and declarative knowledge are inter-related, and are processed by an inference engine based on a modified "recognize-acts" cycle, which includes matching, conflict resolution, and execution phases.

7. Experimental activity, results and future directions

The architecture illustrated in the previous sections has been implemented in a prototype system written in Lisp and running on a SUN workstation. This prototype implements a general tool for designing distributed multi-agent systems, and has been used to develop an experimental application in the field of descriptive text understanding. The application system can map short (one page) excerpts extracted from scientific papers and textbooks on operating systems into a formal internal representation expressed in a rich, layered, propositional language developed by the authors in the frame of a previous research project (Fum, Guida, and Tasso, 1984). It has been extensively tested with sample cases and has proved the adequacy of the distributed approach proposed in several interesting cases of difficult parses.

The work reported in the paper has brought two main contributions. From the point of view of distributed problem solving, a novel general architecture has been proposed which can fit a variety of applications in the broad domain of cognitive modeling. A critical comparison with related approaches (Lesser and Corkill, 1981; Davis and Smith, 1983; Ensor and Gabbe, 1985) would be appropriate, but is omitted here due to space constraints. From the more specific perspective of natural language processing, the major advantages obtained are:

- the system is sufficiently sound and reliable from the cognitive point of view;
- it allows integration of different theories of cognitive processing, since it does not commit to a particular point of view (e.g., theories of lexical access, grammar representation, etc.);
- it supports adoption of heterogeneous techniques for implementing individual specialists, so as, for example, a morphology specialist can be implemented using a traditional imperative language, while a reference specialist may be designed using a knowledge-based technique;
- specialists can be developed, debugged, tested, and refined in isolation, each one largely independently from the others;
- redundancy of specialists gives the system a high degree of robustness;
- the global operation and performance of the system can be changed in a quite transparent and effective way through appropriate actions at the cooperation level, so as system tuning, refinement, debugging and experimentation are easy and natural.

The current research activity is mainly devoted to an extended experimentation of the distributed multi-agent approach proposed, in order to better assess its validity and to focus some challenging open problems at the cooperation level (including, among others: relationships between procedural and declarative knowledge, design of skilled global problem solving strategies, and implementation of elementary learning mechanisms based on generalization from past cases). In particular, two experimental application systems are presently in progress: one devoted to understanding and importance evaluation of descriptive texts (Fum, Guida, and Tasso, 1987), and the other concerned with natural language dialogue in the field of intelligent information retrieval (Brajnik, Guida, and Tasso, 1986).

References

- Brajnik, G., G. Guida and C. Tasso. 1986. An expert interface for effective man-machine interaction. In L. Bolc, M. Jarke (Eds.), Cooperative Interfaces to Information Systems, Springer-Verlag, Berlin, FRG, 259-308.
- Costantini, C., D. Fum, G. Guida, A. Montanari and C. Tasso. 1987. Text understanding with multiple knowledge sources: An experiment in distributed parsing. Proc. 3rd Conf. of the European Chapter of the Association for Computational Linguistics, Copenhagen, DK, 75-79.
- Cullingford, R.E. 1981. Integrating knowledge sources for computer "understanding" tasks. IEEE Trans. on Systems, Man, and Cybernetics 11, 52-60.
- Davis, R. and R.G. Smith. 1983. Negotiation as a metaphor for distributed problem solving. Artificial Intelligence 20, 63-109.
- Ensor, J.R. and J.D. Gabbe. 1985. Transactional blackboards. Proc. 8th Int. Joint Conf. on Artificial Intelligence, Karlsruhe, FRG, 340-344.
- Fum, D., G. Guida and C. Tasso. 1984. A propositional language for text representation. In B.G. Bara, G. Guida (Eds.), Computational Models of Natural Language Processing, North-Holland, Amsterdam, NL, 121-150.
- Fum, D., G. Guida and C. Tasso. 1986. Tailoring importance evaluation to reader's goals: A contribution to descriptive text summarization. COLING-86, Proc. 11th Int. Conf. on Computational Linguistics, Bonn, FRG, 256-259.
- Fum, D., G. Guida and C. Tasso. 1987. Variable-depth text understanding: Integrating importance evaluation into the comprehension process. In I. Plander (Ed.), Artificial Intelligence and Information - Control Systems of Robots - 87, North-Holland, Amsterdam, NL, 31-39.
- Huang, X. and L. Guthrie. 1986. Parsing in parallel. COLING-86, Proc. 11th Int. Conf. on Computational Linguistics, Bonn, FRG, 140-145.
- Johnson-Laird, P.H. 1983. Mental Models. Cambridge University Press, Cambridge, UK.
- Lesser, V.C. and D.D. Corkill. 1981. Functionally accurate, cooperative distributed systems. IEEE Trans. on Systems, Man, and Cybernetics 11, 81-96.
- Marslen-Wilson, W.D. 1975. Sentence perception as an interactive parallel process. Science 189, 226-228.
- Slack, J.M. 1986. Distributed memory. A basis for chart parsing. COLING-86, Proc. 11th Int. Conf. on Computational Linguistics, Bonn, FRG, 476-481.
- Small, S.L. 1980. Word Expert Parsing: A theory of distributed word-based natural language understanding. TR 954. Department of Computer Science, University of Maryland.
- Tyler, L.K. and W.D. Marslen-Wilson. 1977. The on-line effect of semantic context on syntactic processing. Journal of Verbal Learning and Verbal Behavior 16, 683-692.
- Waltz, D.L. and J.B. Pollack. 1985. Massively Parallel Parsing: A strongly interactive model of natural language interpretation. Cognitive Science 9(1), 51-74.