ISIS: Interaction through Speech with Information Systems

Afzal Ballim, Jean-Cédric Chappelier, Martin Rajman, Vincenzo Pallotta

Computer Science Dep. – Swiss Federal Institute of Technology (EPFL) – Lausanne

Abstract. This paper presents the result of an experimental system aimed at performing a robust semantic analysis of analyzed speech input in the area of information system access. The goal of this experiment was to investigate the effectiveness of such a system in a pipelined architecture, where no control is possible over the morpho-syntactic analysis which precedes the semantic analysis and query formation.

1 Introduction

The general applicative framework of the ISIS project[1] [10] was to design an information system NLP interface for automated telephone-based phone-book inquiry. The objective of the project was to define an architecture to improve speech recognition results by integrating higher level linguistic knowledge. The availability of a large collection of annotated telephone calls for querying the Swiss phone-book database (i.e the Swiss French PolyPhone corpus [11]) allowed us to propose and evaluate a very first functional prototype of software architecture for vocal access to database through phone.

The processing of the corpus data is performed at various linguistic levels by pipelined modules, where each module takes as input the output of the preceding module. The main goal of this architecture is to understand how far it is possible go without using any kind of feedback and interactions among different linguistic modules. The proposed architecture for the functional prototype contained 3 modules:

- a speech recognition (SR) system [2], taking speech signals as input and providing N-best sequences in the compact form of a lattice;
- a stochastic syntactic analyzer (i.e. parser) extracting the k-best analysis [8];
- a semantic module in charge of filling the frames required to query the database [5].

The present paper describes first the parsing module and then the semantic module.

[1] ISIS project started on April 1998 and finished on April 1999. It was funded and overseen by SwissCom; the partners were EPFL, ISSCO (Institute Dalle Molle, University of Geneva) and IDIAP (Institute Dalle Molle, Martigny).

2 Syntactic analysis

The forest of syntactic trees produced by this phase have been used to achieve two different goals:

1. the k-best analyses are used to disambiguate speech recognizer hypothesis as, due to the lack of a good language model for the application (lots of proper nouns), the lattices obtained from the SR module were of low quality: almost none of them contained a single syntactically correct sentence;
2. syntactic parses served as supplementary input for the robust semantic analysis which aims at producing query frames for the information system.

Resources. Low-level processing (morphological analysis and tagging) was performed by ISSCO using tools developed in the European Linguistics Engineering project MULTEXT. For syntactic analysis, ISSCO developed a Feature Unification Grammar (FUG) based on a small sample of the Polyphone data.

The FUG was translated into a equivalent Context-Free Grammar (CFG) necessary for parsing the lattices coming out of the SR module. The reasons for translating the FUG in a CFG were: (1) to be able to directly parse the lattices coming out of the speech-recognition module, without listing their whole content explicitly; (2) to have a very fast parsing module and (3) to be able to do robust partial analysis in the (very frequent) case of no complete parsing at the sentence level.

Some may argue that context-free models are not very realistic from a linguistic perspective and that higher-level descriptions[2] should be used instead. Others may argue that context-free models are not realistic for real-time speech recognition since they may lead to important overheads in computational cost for real-world applications.

For real-world applications, only methods with low algorithmic complexity can be considered. Such a constraint imposes severe limitations on the sophistication of the linguistic models that can be used, and most often, only Finite-State models, which allow efficient parsing (linear time), can be considered. However, lower-level language models such as Finite-State Automata do not provide the analyzed sentence with useful syntactic structure that may be necessary for the subsequent processing modules (e.g. semantic).

We argue that CFG have both advantages to be efficient enough for implementation in real-time speech recognition applications and to incorporate at least some linguistic descriptions.

We are nonetheless well aware of the fact that the writing of a grammar (mandatory for applications that require good performances) cannot reasonably be made within a context-free formalism. In our opinion, CF formalism is rather to be considered as a kind of "machine code" for syntactic descriptions, provided that some translation mechanisms from higher level description formalisms are available.

[2] HPSG for instance.

The original FUG grammar had 94 rules. We implemented an optimized CFG translation leading to 28'253 context-free rules with 6'648 non-terminals (whereas the CF backbone of the FUG grammar had 25 main categories). The lexicon had 6'962 words and 174 Part-of-Speech tags.

Experimental Results We experimentally addressed the two questions: (1) Does the pipelined coupling of a CFG parser to the SR module significantly improve the recognition? (2) Is the parsing feasible in a reasonable time?

The first preliminary experiments we made were, for both aspects, promising [9]. Experiments were carried out for several different parameter settings of the speech recognizer[3]. On the average over all the experiments, in 35% of the cases the coupling with a SCFG strictly improved the results and in 67% of the cases it did at least as well as without SCFG. Furthermore, when restricted to the two parameter sets for which the speech-recognizer produced its best results, the former results improved to 50% and 80% respectively.

It is worth emphasizing that, using our computationally efficient parser [8], the overhead in time due to the adjunction of a CFG parser is negligible with respect to the speech recognition time.

3 Robust Semantic Analysis

In theory, a complete dialogue management system requires total semantic understanding of the input. However, it is still very difficult to get any useful semantic representation from free dialogue, even for a limited domain as here considered. A different approach considers that a dialogue management can be achieved by a light parsing of the texts. This method needs neither a full semantic understanding of the language nor a deep investigation in the meaning and senses of the words. It is merely based on the knowledge of certain *cue-phrases* able to describe a shallow semantic structure of the text. These cue-phrases or terms should be relevant enough to give a coherent semantic separation of the different parts[4]. Nonetheless, the set of terms must not be rigid to avoid boolean results, but there must be a set of semantically similar terms with a degree of confidence for each. This would generate hypothetic semantic descriptions. In fact, these terms which correspond to semantic fields, are able to isolate texts parts. In case of a failure in obtaining a full and precise semantic description, a minimum description would indeed be derived. Therefore in all cases only relevant parts would undergo the understanding process. While the semantic analysis will in general reduce the degree of ambiguity found after syntactic analysis, there remains the possibility that it might *increase* some degree of ambiguity due to the presence of coherent senses of words with the same syntactic category (e.g., the word "Geneva" can refer to either the canton or the city). A simple notion of context can be useful to fill by default those slots for which we

[3] Several values for the "acoustic factor" were tested.
[4] A similar approach has been proposed by Grefenstette [12].

have no explicit information. For instance it is possible to fill automatically the *locality* slot using information about the caller's prefix number in case of he does not explicitly specify any locality. We assume that his request actually refers to the same locality denoted by the above prefix number. For doing this type of hierarchical reasoning we exploit the meta-programming capabilities of logic programming and we used a meta-interpreter which allows multiple inheritance among logical theories [7].

Robust Definite Clause Grammars LHIP (Left-corner Head-driven Island Parser) [4, 14] is a system which performs robust analysis of its input, using a grammar defined in an extended form of the Definite Clause Grammar (DCGs) formalism used for implementation of parsers in Prolog. The chief modifications to the standard Prolog 'grammar rule' format are of two types: one or more right-hand side (RHS) items may be marked as 'heads', and one or more RHS items may be marked as 'ignorable'. LHIP employs a different control strategy from that used by Prolog DCGs, in order to allow it to cope with ungrammatical or unforeseen input. A number of tools are provided for producing analyses of input by the grammar with certain constraints. For example, to find the set of analyses that provide maximal coverage over the input, to find the subset of the maximal coverage set that have minimum spans, and to find the analyses that have maximal thresholds.

We introduced weights into LHIP rules with the main goal of inducing a partial order over the generated hypotheses. The following schema illustrates how to build a simple weighted rule in a compositional fashion where the resulting weight is computed from the sub-constituents using the minimum operator. Weights are real numbers in the interval $[0, 1]$.

```
cat(cat(Hyp),Weight) ~~> sub_cat1(H1,W1), ...,sub_catn(Hn,Wn),
   {app_list([H1,...,Hn],Hyp), min_list([W1,...,Wn],Weight)}.
```

This strategy is not the only possible since the LHIP formalism allows a greater flexibility. Without entering into formal details we can observe that if we strictly follow the above schema and we impose a cover threshold of 1 we are dealing with *fuzzy DCG grammars* [13, 3]. We actually extend this class of grammars with a notion of *fuzzy-robustness* where weights are used to compute confidence factors for the membership of islands to categories[5].

In our case study we try to integrate the above principles in order to effectively compute hypotheses for the query generation task.The weighted LHIP rules are designed considering two kind of knowledge: *domain knowledge* is exploited to provide quantitative support (or confidence factor) to our rules. *Linguistic knowledge* is used for determining constraints in order to prune the hypotheses space. We are concerned with lexical knowledge when we need to specify lexical LHIP rules which represent the building blocks of our parsing system.

As pointed out in [6], lexical knowledge plays an important role in Information Extraction since it can contribute in guiding the analysis process at various

[5] Development of this notion is currently under investigation and not yet formalized.

linguistic level. In our case we are concerned with lexical knowledge when we need to specify lexical LHIP rules which represent the building blocks of our parsing system. *Semantic markers* are domain-dependent word patterns and must be defined for a given corpus. They identify *cue-words* serving both as *separators* among logical subparts of the same sentence and as *introducers* of *semantic constituents.* The generation of frame filling hypotheses is performed by rules that specify the orders of chunks interleaved by semantic markers (e.g. separators and introducers), compose weights and filter hypotheses by using domain knowledge. The grammar should also provide a mean to provide an empty constituent when all possible hypothesis rules have failed.

4 Conclusions

From a very superficial observation of the human language understanding process, it appears clear that no deep competence of the underlying structure of the spoken language is required in order to be able to process acceptably distorted utterances. On the other hand, the more experienced is the speaker, the more probable is a successful understanding of that distorted input. How can this kind of fault-tolerant behaviour be reproduced in an artificial system by means of computational techniques? Several answers have been proposed to this question and many systems implemented so far, but no one of them is capable of dealing with robustness as a whole. The use of domain knowledge has turned out to be crucial since our particular goal is to process a queries without any request of clarification from the system. Due to the inaccuracy and ambiguity generated by previous phases of analysis we need to select the best hypotheses and often recover information lost during that selection.

Although robustness can be considered as being applied at either a syntactic or semantic level, we believe it is generally at the semantic level that it is most effective. This robust analysis needs a model of the domain in which the system operates, and a way of linking this model to the lexicon used by the other components. It specifies semantic constraints that apply in the world and which allow us, for instance, to rule out incoherent requests. Taking the assumption that the information system being queried is relatively close in form to a relational database, the goal of the interpretative process is to furnish a query to the information system that can be viewed in the form of a frame with certain fields completed, the function of the querying engine being to fill in the empty fields.

We can conclude that robustness in dialogue is crucial when the artificial system takes part in the interaction since inability or low performance in processing utterances will cause unacceptable degradation of the overall system. As pointed out in [1] it is better to have a dialogue system that tries to guess a specific interpretation in case of ambiguity rather than ask the user for a clarification. If this first commitment results later to be a mistake a robust behaviour will be able to interpret subsequent corrections as repair procedures to be issued in order to get the intended interpretation.

References

1. J.F. Allen, B. Miller, E. Ringger, and T. Sikorski. A robust system for natural spoken dialogue. In *Proc. 34th Meeting of the Assoc. for Computational Linguistics*. Association of Computational Linguistics, June 1996.
2. J.M. Andersen, G. Caloz, and H. Bourlard. Swisscom "advanced vocal interfaces services" project. Technical Report COM-97-06, IDIAP, Martigny, December 1997.
3. Peter. R.J. Asveld. Towards robustness in parsing - fuzzifying context-free language recognition. In J. Dassow, G. Rozemberg, and A. Salomaa, editors, *Developments in Language Theory II - At the Crossroad of Mathematics*, Computer Science and Biology, pages 443–453. World Scientific, Singapore, 1996.
4. A. Ballim and G. Russell. LHIP: Extended DCGs for Configurable Robust Parsing. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 501 – 507, Kyoto, Japan, 1994. ACL.
5. Afzal Ballim and Vincenzo Pallotta. Robust parsing techniques for semantic analysis of natural language queries. In Rodolfo Del Monte, editor, *Proceedings of VEXTAL99 conference*, November 1999. to appear.
6. R. Basili and Pazienza M.T. Lexical acquisitiion and information extraction. In Pazienza M.T., editor, *Information Extraction – A multidiciplinary approach to an ermerging information technology*, volume 1299 of *LNAI*, pages 44–72. Springer Verlag, 1997.
7. A. Brogi and F. Turini. Meta-logic for program composition: Semantic issues. In K.R. Apt and F.Turini, editors, *Meta-Logics and Logic Programming*. The MIT Press, 1995.
8. J.-C. Chappelier and M. Rajman. A generalized cyk algorithm for parsing stochastic cfg. In *1st Workshop on Tabulation in Parsing and Deduction (TAPD98)*, pages 133 – 137, Paris, April 2-3 1998.
9. J.-C. Chappelier, M. Rajman, R. Aragues, and A. Rozenknop. Lattice parsing for speech recognition. In *Proc. of 6ème conférence sur le Traitement Automatique du Langage Naturel (TALN'99)*, pages 95–104, July 1999.
10. J-C. Chappelier, M. Rajman, P. Bouillon, S. Armstrong, V. Pallotta, and A Ballim. ISIS project: final report. Technical report, Computer Science Department - Swiss Federal Institute of Technology, September 1999.
11. G. Chollet, J.-L. Chochard, A. Constantinescu, C. Jaboulet, and Ph. Langlais. Swiss french polyphone and polyvar: Telephone speech database to model inter- and intra-speaker variability. Technical Report RR-96-01, IDIAP, Martigny, April 1996.
12. G. Grefenstette. Light parsing as finite-state filtering. In Kornai A., editor, *Proceedings of the ECAI 96 Workshop on Extended Finite State Models of Language*, pages 20–25, 1996.
13. E.T. Lee and L.A. Zadeh. Note on fuzzy languages. *Information Science*, 1:421–434, 1969.
14. C. Lieske and A. Ballim. Rethinking natural language processing with prolog. In *Proceedings of Practical Applications of Prolog and Practical Applications of Constraint Technology (PAPPACTS98)*, London,UK, 1998. Practical Application Company.