LANGUAGE ANALYSIS IN SCHISMA

Danny Lie, Joris Hulstijn,
Hugo ter Doest, Anton Nijholt*

1 Introduction

SCHISMA is a research project that is concerned with the development of a natural language accessible theatre information and booking system. In this project two research approaches can be distinguished. One approach is devoted to theoretical research in the areas syntax, semantics and pragmatics.Research on syntax has been conducted on a unifying parsing approach [7], left and head corner grammars [6], on stochastic context-free grammars [1] and on unification grammar parsing [2][4]. Research on semantics and pragmatics has been conducted on dialogue act classification [3] and from a logical point of view [5].

The other approach is more practically oriented in that it is more focused on the realisation of a fully functional prototype of the SCHISMA system. To this purpose, in the past, the interface development system Natural Language™ has been used for implementing a SCHISMA prototype, a Wizard of Oz environment has been developed (with which a corpus of dialogues has been collected), an attempt has been made to develop a SCHISMA prototype for the WWW, and a prototype has been developed for education purposes[1]. Our most recent achievement is a fully functional SCHISMA system (implemented in Java) based on a context-sensitive string rewrite mechanism. Although the system is primitive in nature and not necessarily built on linguistic principles (rather on intuitions), we believe its performance is such that the majority of users will be satisfied. Users will adapt to the system rather than reject it because of poor performance (when compared to human-human dialogues). In the development of a spoken telephone service for the Dutch Public Transport Service a similar position is taken.

In this short paper we will give an overview of the two approaches distinguished here as far as parsing is concerned. Section 2 presents work on classical unification-based parsing, section 3 discusses the rewrite and understand approach in some detail. Section 4 considers future developments and concludes the paper.

2 Unification-based Approach

In [2] a head corner parser for typed unification grammars is designed and implemented in C++; for description of the lexicon and the grammer a specialised specification language TFS is developed. It is argued that typed unification grammars and especially the newly developed specification language are convenient formalisms for describing natural language use in dialogue systems. The system comprises a compiler that compiles TFS specifications into C++ and some specification-independent libraries (input/output, parsing and unification algorithms). After a TFS specification is developed and compiled into C++, the libraries and the specification-dependent parts of the system can be linked together.

Another, more sophisticated system for unification-based parsing is currently under development.[2] Type hierarchies, disjunctive feature structures and productions all are represented by means of DAGs. Other advantages of the system include the optional weighting of disjunctions with probabilities, flexible behaviour in case of conflicts and an incremental type hierarchy that may grow during parsing depending on type conflicts leading to new types.

*Parlevink Language Engineering, Dept. of Computer Science, University of Twente, PO Box 217 7500 AE, Enschede, The Netherlands, (lie | joris | terdoest | anijholt)@cs.utwente.nl

[1] By Joris Hulstijn; it can be downloaded from wwwseti.cs.utwente.nl/~joris/

[2] By Hugo ter Doest; a prototype can be downloaded from wwwseti.cs.utwente.nl/~terdoest/tfs/tfs.tar.gz

3 The *Rewrite and Understand* Approach

As mentioned in the introduction, there have been several attempts to design a SCHISMA dialogue system with the aim to end up with a system that is certainly not perfect but that can nevertheless be accessed by users willing to adapt themselves to the system. Our most recent achievement is a system that consist of two subsequent processes: a *rewrite* process in which natural language utterances are mapped via a sequence of context-sensitive string-to-string transformations onto some semantical normal form, and an *understanding* process, in which the interpretation of the semantic form is made dependent on the current dialogue state. We concentrate on the rewrite process here.

Two basic operations underly the rewrite process: deletion of semantically irrelevant words, and substitution of words by standard synonymous words or expressions. If we abstract from syntactic sugar of the specification language, deletions and substitutions are defined by context-sensitive rewrite rules of the form $x_1 \ldots x_n \rightarrow y_0 x_{k_1} y_1 \ldots y_{m-1} x_{k_m} y_m$ where $m \in 1 \ldots n$ and $k_i \in 1 \ldots n$ for all $i \in 1 \ldots m$ where $x_1 \ldots x_n$ are regular expresions[3] and $y_1 \ldots y_m$ are arbitrary strings. So rewrite rules are potentially context-sensitive. A rewrite rule is applicable to a sentence if its left hand side matches a substring of the sentence; application of the rewrite rule means that the matched part of the sentence is rewritten, the surrounding parts are left as they are. A rewrite specification consists of a sequence of rewrite rules. Given a rewrite specification and a sentence s, a complete rewrite of s means that the sequence of rewrite rules is run through in the order specified applying each rule applicable in the sense described: $s \xrightarrow{r_{i_1}} s_1 \xrightarrow{r_{i_2}} \ldots \xrightarrow{r_{i_n}} s_n$.

The complete rewrite process consists of two phases: a *global* phase and a *local* phase, respectively. The global phase is the same for all sentences; after the global rewrite a keyword filter decides on what local grammar(s) to apply to the sentence. If more than one local grammar applies, all of them are executed; a score accompaning the rewritten strings helps the understanding process decide on what interpretation to continue with. We developed one global grammar and 20 local grammars by closely examining the aforementioned Wizard of Oz corpus. Each of the local grammars cover a part of the semantic functions sentences may have in the SCHISMA domain. Each local grammar finally rewrites its input into a list of feature value pairs accompanied by the name of the action the system should take. The understanding process then interpretes the list in the context of the dialogue.

4 Future Work and Conclusion

In the near feature a combination of the two approaches is foreseen: the rewrite and understand approach will be adapted such that it can be applied as a preprocessor that rearranges input such that difficult problems like discontinuity and embedded sentences no longer burden the unification-based parsing process. Also the rewrite parser will be applied as a backup mechanism in case conventional parsing fails.

References

[1] Rieks op den Akker and Hugo ter Doest. Weakly restricted stochastic grammars. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 927–934, 1994.

[2] Rieks op den Akker, Hugo ter Doest, Mark Moll, and Anton Nijholt. Parsing in dialogue systems using typed feature structures. In *Proceedings of the Fourth International Workshop on Parsing Technologies*, Prague/Karlovy Vary, Czech Republic, 1995.

[3] Toine Andernach. A machine learning approach to the classification and prediction of dialogue utterances. In *Proceedings of the Second International Conference on New Methods in Language Processing*, pages 98–109, 1996.

[4] Hugo ter Doest. *Robustness and Efficiency in Unification-based Parsing Methods.* PhD thesis, University of Twente, Enschede, The Netherlands. To appear.

[5] Joris Hulstijn. Structured information states - raising and resolving issues. In *Proceedings Munich Workshop on Formal Semantics and Pragmatics of Dialogue*, Munich, Germany, 1997. University of Munich.

[6] Klaas Sikkel and Rieks op den Akker. Predictive head-corner chart parsing. In *Proceedings of the Third International Workshop on Parsing Technologies*, pages 267–275, Tilburg (The Netherlands), Durbuy (Belgium), 1993.

[7] Klaas Sikkel and Anton Nijholt. Parsing of context-free languages. In *Handbook of formal languages*, volume II Linear Modeling: Background and Application, pages 61–100. Springer, Berlin; Heidelberg; New York, 1997.

[3] Regular expressions x_i may be optional, one of a set of alternatives, required at the beginning or the end of the sentence, a *joker* * matching any word, or a *wildcard* ** matching any sequence of words.