

Robust Semantic Analysis for Adaptive Speech Interfaces^{*}

Maria Cheadle and Björn Gambäck

SICS – Swedish Institute of Computer Science AB
Box 1263; S-164 29 Kista; Sweden
{maria.gamback}@sics.se

Abstract

The DUMAS project develops speech-based applications that are adaptable to different users and domains. The paper describes the project's robust semantic analysis strategy, used both in the generic framework for the development of multilingual speech-based dialogue systems which is the main project goal, and in the initial test application, a mobile phone-based e-mail interface.

1 Introduction

The DUMAS (Dynamic Universal Mobility for Adaptive Speech Interfaces) project develops multilingual speech-based applications, and more specifically, will investigate adaptive multilingual interaction techniques to handle both spoken and text input and to provide coordinated linguistic responses to the user. We will construct a generic framework for multilingual speech-based applications, the *Athos* architecture. It supports the development of adaptive speech applications by implementing interaction agents especially designed to handle different interaction situations. The *Athos* architecture supports adaptivity to both the individual user and the particular domain. Based on the *Athos* architecture and various interaction techniques, we are building *AthosMail*, a mobile phone-based e-mail application that will deal with multilingual issues in several forms and environments, and whose functionality can be adapted to different users, situations and tasks.

The communication between the agents in an *Athos* application is handled by the *Information Manager* which controls the *Information Storage* where all application specific data relevant to other agents is stored (Turunen & Hakulinen 2000). This solution makes it easy to add, remove and reuse agents when modifying an *Athos*-based application. The applications will benefit from the advantages of robust and fault-tolerant semantic analysis, which combined with the dialogue management agents will handle user interaction in a very robust manner. The *AthosMail* application will be able to distinguish between different languages, both in e-mail messages and in the user utterances. The adaptive qualities of the *User Modelling Agents* and *Presentation Agents* will ensure that the user is addressed in her preferred languages based on experiences from the current and previous sessions. The *Information Extraction and Retrieval Agents* in collaboration with the text processing agents will continuously analyse the content of the user's inbox in order to enable advanced search functions. The content of the inbox and the interaction history of a

^{*} Work sponsored by the European Union's Information Society Technologies Programme under contract IST-2000-29452, DUMAS. www.sics.se/dumas Thanks to all the project participants from UMIST, UK; ETeX Sprachsynthese AG, Germany; KTH, and SICS AB, Sweden; and U. Tampere, U. Art and Design Helsinki, Connexor Oy, and Timehouse Oy, Finland.

particular user will also be used to generate specialized speech recognition grammars and vocabularies thus aiding the *AthosMail* application in improving speech recognition accuracy.

To achieve the kind of context-aware behaviour described above, fault-tolerant and robust semantic processing is an essential contribution. The agents performing the semantic analysis will naturally rely on the work carried out by the agents for syntactic analysis and sense annotation, described in Section 2. This paper will, however, mainly address the semantic analysis processing designed for *AthosMail* (Section 3). Still, the agents we will build are designed to be general enough to be reused in future applications based on the *Athos* architecture.

2 Text Processing in *Athos*

The overall objective of the different agents making up the text processor in the *Athos* framework is to syntactically analyse the surface word-string, assign correct semantic interpretations to it, and extract the key information content. The *AthosMail* e-mail interfacing system outputs speech only (in the form of spoken responses and read e-mails) to the user, but gets two types of input: user speech and textual e-mails. The spoken user input comes in the form of a dialogue with the system and mainly consists of different commands to be executed. The challenge here lies in interpreting the commands by figuring out the user intentions that may be directly or indirectly expressed in the conversation. In addition, the applications the *Athos* framework is built to handle are inherently multilingual. So will, for example, not only different e-mails in *AthosMail* be written in different languages, but also will each e-mail possibly contain passages in different languages.

2.1 Syntactic Analysis

To process the type of language found in e-mails and speech we need a robust methodology: the system should produce a parse even if the input contains errors or lacks information. Experience shows that intelligent combinations of statistical and machine learning methods with linguistic and lexical tools deliver noticeably better results than approaches that use statistical or linguistic methods alone. The crux is in combining different types of method without sacrificing effectiveness. A method of achieving this is to allow for partial results to be stored. The parse then results in a set of pieces which will have to be combined, and different evaluators must be used to decide what piece is the most important. To this end the DUMAS project will utilize fault-tolerant functional dependency grammar parsing methods. Dependency grammar approaches characterize syntactic structure in terms of dependency relations between node elements, or *nuclei*. These may be words, but are in general the basic semantic units, the minimal units in a lexicographical description. One nucleus is the head of the whole sentence; every other nucleus depends on some head and may itself be the head of any number of dependents. Several parsing algorithms for dependency-type grammars have been suggested; however, many of these rely on the possibility of determining clause boundaries, which is difficult when dealing with speech input. In DUMAS we will use the functional dependency grammar (FDG) parsing scheme described by Tapanainen (1999). The FDG analysers from Connexor Oy provide morphological and dependency syntactic analysis and produce output in a representation language known as *Machinese*.

2.2 Sense Annotation

The *Machinese* Syntax analysis will be enhanced with senses and slot information. All the objects in the analysis will be processed so that for every nucleus in the parser output we will obtain one or more frames in the style of McCord's (1990) Slot Grammar. In Slot Grammar the slot information is a declaration of syntactic relations, but we will use it to express semantic relations.

The FDG parser for English produces output in the extended Machine Semantics format which already adds most of the lexical semantics needed in the *Athos* application domain. For Swedish and Finnish, however, that information needs to be created and added to the Machine Syntax. The sense information for some of the *Sense Annotation Agents* is produced by Vector-Based Lexicon Acquisition Modules which utilizes a machine learning strategy described by Sahlgren (2001). For English, this is combined with the underspecified approach to semantic tagging used in Buitelaar's (1998) CoreLex which covers nearly 40,000 nouns and 126 underspecified semantic types. The main part of the lexical database creation for Swedish and Finnish will rely on the machine learning extraction agent; however, resources similar to CoreLex for Swedish and Finnish will also have to be created during the project, albeit at a less ambitious level.

3 Semantic Analysis

The DUMAS project has two types of semantic analysis tasks to address. First, the interpretation of commands to the e-mail system in *AthosMail*. For those statements we need to detect an as "full" interpretation as possible in order to map the commands into functions in the e-mail reader. Second, we have the interpretation of the actual documents in the application, the e-mail messages. For those we need a more schematic interpretation, in order to retrieve some relevant messages for a user query, to extract a particular piece of knowledge from a set of messages, for summarising the messages selected by the user, or for improving on the quality of the synthesized speech read to the user. The *AthosMail* application will have three different types of agents to handle the semantic analysis tasks: the *Logical Form Builder*, the *Robust Representation Builder for User Utterances* and the *Robust Representation Builder for E-mail Messages*. The Logical Form Builder is derived from the Parasite language understanding system (Ramsay & Seville 2000). The two robust representation builders are described in Section 3.3. First, however, we give a description of the overall strategy for the semantic processing in the *Athos* architecture.

3.1 Underspecified Semantics for Dependency Structures

Semantic interpretation is the mapping from natural language statements to some representation of the meaning. This representation might be a predicate logic form matching the statement, or a maybe a database search command. If the meaning of a piece of information X is incomplete without the meaning Y , but not vice versa, then X is a functor and Y is an argument (of X). Viewing dependency in a semantical way, heads are functors and dependents are arguments. With this view, and the assumption that semantics is compositional, it is straight-forward to build a semantic representation incrementally by including semantic information in the dependency rules and, in particular, in the lexical entries. The grammar rules should allow for addition of already manifest information, e.g., from the lexicon, and ways of passing non-manifest information, e.g., on complements sought. When aiming at parsing robustness, the "recovery" from incomplete structures in the semantics can be allowed to apply either after or in parallel with the parsing process. The problem is that it is very difficult to perform a complete analysis. However, Worm & Rupp (1998) suggest a robust method for speech understanding using heuristic rules to decide which partial parse fragments should be combined to form more complete ones.

A basic problem with semantic interpretation is that language is ambiguous. Amongst others, ambiguity may be due the fact that many words do not have one unique meaning, that more than one syntactic structure may be assigned to an expression, or that scope relations are not clear. A way around this dilemma is to use *underspecification* (Reyle 1993), i.e., to have a common representation for all of the possible interpretations of an ambiguous expression. In the analysis process, ambiguities will sometimes need to be resolved, and sometimes not. Gambäck & Bos (1998)

describe an algorithm for scope resolution in underspecified semantic representations where scope preferences are suggested on the basis of semantic argument structure. The approach maintains an underspecified semantic representation, while suggesting a resolution possibility. In short, the algorithm assumes that the most plausible scopal resolution works in the *opposite* way to the semantic head structure. Thus, if a nonhead daughter is scope-bearing (i.e., is a quantifier, a particle, etc.), it is given scope over the head daughter, as the default case resolution.

3.2 Slot Filling

In DUMAS we will employ two different ways of semantic interpretation, one working directly on the parser output, and another which robustly tries to match the syntactic structure to semantic templates. These can be partially or fully instantiated and will be used to incrementally build the meaning of a statement. The templates may be described using a minimalistic semantic description language based on recursive typed feature structure representations. This covers both “object-level” structures (such as semantic functions) and meta-level structures (such as speech acts). *Slot Filling Agents* will process all objects in the analysis to obtain one or more frames for every nucleus in the output of the Machine parsers. A slot structure can be filled with representations from one or more statements if necessary or applicable. The slot information declares with which other structures its slots may or must be filled in order for it to be a valid structure. All quantifiers and verbs will introduce slot information, as well as some words typical of conversations with an e-mail application. Any other words in a statement will have empty slot information.

The basic functionality of the *AthosMail* application will be expressed in predefined slot-filler structures, *templates*. We will construct templates based on the vocabulary of a set of multilingual dialogue corpora we have collected. The corpora were tagged using a dialogue act taxonomy that among other things covered the commands put to the system. The dialogue acts corresponding to the basic functionality of *AthosMail* and the statements used to express them will be used to construct the templates. So when a word such as “message” is processed it will implicitly activate templates that are able to semantically represent the possible commands in which it is likely to have occurred. Also for tasks such as finding specific information in e-mail messages or summarizing the content of a message we will build domain specific templates. There has so far been very little work along these lines in the dependency grammar tradition; however, the overall semantic slot-filling strategy is in the style of most approaches to information extraction and closely resembles that of Milward & Knight (2001): semantic representations are mapped to predefined template rules that recognise key meanings and extract the desired information. Including semantic information directly in the dependency grammar structure is also suggested by Courtin & Genthial (1998), who use unification for the task of passing the semantic information.

3.3 Robust Representation Builders for User Utterances and for E-Mails

The semantic analysis of user utterances in the DUMAS project will be carried out by the Robust Representation Builder Agents for User Utterances (RBU) and by the Logical Form Builder. The *AthosMail* evaluators assign the task to the agent best suited to perform the particular task. The RBU agents will handle spoken telephone input and must be robust enough to build semantic representations although receiving just bits and pieces of complete utterances. The speech environment may be noisy, the speech recogniser’s language model and grammar may not cover the speech input or the special characteristics of spontaneous speech will not be represented in the linguistic tools that analyse the input before it is made available to the semantic analysis components. Thus the main aim for the RBU agents is to merge pieces of incomplete knowledge into sensible *AthosMail* commands in the cases where either syntactic or semantic (probably

both!) analysis break down. The templates for those commands represent the type of information needed in order to match one specific system function. Should the slot structure of the utterance be incomplete, the module will process its neighbouring utterances to see if any of the information in them will produce valid slot structures when applied to each other. Those of the completed or partially completed slot structures that originated from templates are the ones that will be used to extract the instructions to the *AthosMail* application.

The Robust Representation Builder Agents for E-Mail Messages will construe semantic representations of textual e-mails. When slot information is added by the Slot Filling Agent the structure of the e-mail activates templates suited to describe the contents of the message. For example, the presence of a signature at the bottom of the message would invoke a template that looks for things that would typically be found in signatures, like telephone numbers, addresses, etc. The aim is to provide schematic, rather than full interpretations, suitable for the particular task at hand, be it summarization of some e-mail messages, or information retrieval or knowledge extraction from a set of messages.

4 Conclusions

We have described the robust semantic analysis strategy of the DUMAS project which develops multilingual speech-based dialogue applications adaptable both to different users and to different domains. The approach outlined here is aimed in particular at the project's first target application, a mobile phone-based mail interfacing system where language understanding is needed both for interpreting the commands to the mail system and for retrieving and extracting knowledge from the actual e-mails. Still, the agents we build are designed to be reused in the general framework.

References

- Buitelaar, P. (1998). *CoreLex: Systematic Polysemy and Underspecification*, PhD Thesis, Brandeis University, Waltham, Massachusetts.
- Courtin, J. & Genthial, D. (1998). Parsing with dependency relations and robust parsing, Workshop on *Processing of Dependency-based Grammars*, Montreal, Canada; ACL.
- Gambäck, B. & Bos, J. (1998). Semantic-head based resolution of scopal ambiguities, *Proc. 17th COLING and 36th ACL*, Montreal, Canada, Vol. 1, pp. 433-437; ACL.
- McCord, M.C. (1990). Slot Grammar: A system for simpler construction of practical natural language grammars, Tech. report, IBM T.J. Watson Research Center, Yorktown Heights, NY.
- Milward, D. & Knight S. (2001). Improving on phrase spotting for spoken dialogue processing, Workshop on *Innovation in Speech Processing*, Stratford-upon-Avon, England; IOA.
- Ramsay, A.M. & Seville, H. (2000). Unscrambling English word order, *Proc. 18th COLING*, Saarbrücken, Germany, pp. 656-662; ACL.
- Reyle, U. (1993). Dealing with ambiguities by underspecification, *J. Semantics* 10, 123-179.
- Sahlgren, M. (2001). Vector-based semantic analysis: Representing word meanings based on random labels, in Lenci *et al*, eds, *Acquiring and Representing Semantic Knowledge*; Kluwer.
- Tapanainen, P. (1999). *Parsing in Two Frameworks: Finite-State and Functional Dependency Grammar*, PhD Thesis, University of Helsinki, Helsinki, Finland.
- Turunen, M. & Hakulinen, J. (2000). Jaspis – A framework for multilingual adaptive speech applications, *Proc. 6th ICSLP*, Beijing, China; ISCA.
- Worm, K.L. & Rupp, C.J. (1998). Towards robust understanding of speech by combination of partial analyses, *Proc. 13th ECAI*, Brighton, England, pp. 190-194; John Wiley.