# Underspecified Semantic Interpretation in an E-Mail Speech Interface*

**Björn Gambäck** and **Maria Cheadle**
{gamback,maria}@sics.se
Swedish Institute of Computer Science
Box 1263, S-164 29 Kista, Sweden
http://www.sics.se/dumas

**Abstract**

The DUMAS project constructs a generic framework for the development of multilingual speech-based dialogues systems. As an initial test of the generic framework we will build a mobile phone-based e-mail interface whose functionality can be adapted to different users, different situations and tasks. The paper describes the semantic processing which we envision needed in that type of system, for the actual e-mail system commands on the one hand, and for extracting knowledge from the e-mails at such on the other.

## Introduction

It is clear that future communication with mobile electronic systems will require dynamic and adaptive capabilities. The users will demand systems that can learn from the interaction and can adapt their functionality to different users, situations and tasks. This is quite in contrast to the current state of affairs where the systems instead require the user's adaptation to the system. A newly-started project supported by the EC, Dynamic Universal Mobility for Adaptive Speech Interfaces (DUMAS) aims to develop a generic and modular framework for such multi-lingual speech-based applications by integrating research on robust text processing, flexible dialogue management, advanced user interface techniques, and dynamic user modelling.

To ensure generality of the implemented methods, the applicability of the generic framework will be tested during the course of the project by the construction of several prototypes, for different domains and different types of adaptive speech applications. Among these we envisage speech-based SMS messaging, text-TV speech interfacing, information retrieval from spoken documents, and cross-language multimodal information retrieval (e.g., an English-speaking journalist searching through a Finnish news database which contains both news in text format and audio sequences of spoken material). However, as a first application area, we will address the issue of an interface to e-mail system operating over a mobile phone communication channel.

Such an 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. On the one hand then, 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 figuring out the user intentions that may be directly or indirectly expressed in the conversation. Different languages and cultures have very different communication strategies and the dialogue management must be able to support this, as well as be able to trace and coordinate different topics of a conversation and to select information that is appropriate to the user in the situation at hand. Complementing the dialogue strategy, a user model which can be updated automatically will reflect both the user's preference on how the system should behave on her account (e.g., which e-mails she finds important) and the user character, i.e., the innate peculiarities of the communicating individual (e.g., the vocabulary and grammar used to construct sentences, the verbosity of her commands or intonation, or possibly her speech properties such as dialect, sex, etc.).

On the other hand, the text-based input comes in the form of e-mails on a wide variety of topics, mails that need to be processed at different levels of granularity depending on the induced user intention. So, if the user wants to hear something like "*a summary of all mails concerning Frank's thesis*," the system first has to execute the information retrieval task of finding all such e-mails and then do information extraction / summarisation on the relevant texts. Since e-mail messages can contain any mixture of different languages in unpredictable combinations, this offers a rich and challenging domain for investigating various multilingual interaction and information processing aspects.

The rest of the paper is laid out as follows, first we discuss the peculiarities of e-mail text processing and of dependency-based analysis methods. We then move on to the heart of the matter, the semantic interpretation process and underspecification. Finally this is narrowed down by the special needs of the DUMAS project and its initial e-mail application.

*This work is part of DUMAS (Dynamic Universal Mobility for Adaptive Speech Interfaces), a project running October 2001 – September 2004 as part of the European Union's Information Society Technologies Programme (contract IST-2000-29452). See the project web page http://www.sics.se/dumas/ for details.

## Text analysis

It is clear that the task of parsing e-mail messages has more to learn from current approaches to parsing speech than it has from traditional approaches to parsing written language. Written language tends to be more "grammatical" than spoken language. However, the language used in e-mails often closer resembles the spoken language in this sense: people more often use incomplete sentences in e-mails than in more traditional types of written language; the sentences and the "turns" (messages) tend to be shorter, more resembling a conversational setting, etc.

To process the type of language found in e-mails and speech we need a robust methodology in which 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. By combining fault-tolerant functional dependency grammar parsing methods with underspecified, template-based semantic annotations the system will produce a set of partial parses, pieces of information spanning parts of the input. Different partial parses then have to be combined, and competing parses eliminated using some evaluation metrics that will decide which piece is the most important.

## Dependency-based parsing

Following Tesnière (1959), 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. Thus the categories in a dependency grammar are lexical projections only, no phrasal categories exist. Constituency does exist, but is derived rather than built into the concept, as in phrase structure grammar. A constituent consists of any head together with its direct and indirect dependents. In a dependency analysis, intuitively 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 sometimes is difficult, especially with coordination structures or when dealing which speech input. Instead, Sunnehall (1996) has suggested a model which creates all possible dependency relations. The resulting structure is a packed dependency structure. The output is optimally one syntax tree, but it might consist of a forest of trees, in the worst case the same number of trees as words. This is done by using a simple $O(n^2)$ algorithm, putting an upper bound on the complexity. Of course, this is too naïve for real-life applications; there is, instead, clear evidence from one of the DUMAS partners, Conexor Oy, that large-scale implementations of dependency parsers can be made to run in $O(n \log n)$ time (Tapanainen 1999).

## Semantic interpretation

The semantic interpretation is the mapping from natural language utterances to some representation of the meaning. This representation might be a predicate logic form matching the statement, or a maybe a database search command. One way of processing natural language is to first build a syntactic representation and then process this in a semantic module. The other way is of course to integrate the processing, and perform both the syntactic and semantic processing at once. Likewise, when aiming at parsing robustness, the "recovery" from incomplete structures in the semantics can be allowed to apply either after or in parallel to the parsing process. A third approach could be to apply some preprocessing to the input. The preprocessing strategy is the least common within the (still very small) field of robust semantic processing; however, it is quite common to use it in other areas, such as machine translation. Of course, what method to use depends on the requirements on the system, the grammar type used and the parser.

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 have a common representation for all of the possible interpretations of an ambiguous expression. Current research (see e.g., Reyle 1993 and Pinkal 1996) uses the term *underspecification* to describe this idea: One does not use representations that encode a single concrete interpretation, but rather *a set* of interpretations.

In the semantic processing, if the meaning of $X$ 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; thus $X$ is then the head and $Y$ is the dependent. With this view, and the basic assumption that semantics is compositional — that the meaning of a sentence in whole is a function of the meaning of its parts — it is straight-forward to build an (underspecified) 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., about complements sought). Assuming a binary-branching dependency structure, we need two such ways, or application rule-types: functor-argument and modifier-argument.

In functor-argument application the bulk of the semantic information is passed between the mother node and the functor (semantic head). In modifier-argument application the argument is the semantic head, so most information is passed up from that. The difference between the two application types pertains to the (semantic) subcategorisation schemes: In functor-argument application, the functor subcategorises for the argument, the argument may optionally subcategorise for the functor, and the mother's subcategorisation list is the functor's, minus the argument. In modifier-argument application, the modifier subcategorises for the argument (only), while the argument does not subcategorise for the modifier. Its subcategorisation list is passed unchanged to the mother node.

A semantic processing like this, where the semantics is built directly from a syntactic representation, either in parallel with it or sequentially after it, would be ideal in the sense that the same semantical rules would be suitable for all tasks, i.e., for everything that a user is trying to do. Likewise, 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. Still, different tasks require different levels of information. The problem with the general approach is that it is very difficult to perform a complete analysis. However, Worm & Rupp (1998) suggest a robust method for speech understanding in which by combination of partial parses to form more complete ones. They use heuristic rules to decide which fragments should be combined.

## Semantics in DUMAS

In DUMAS we will complement the general approach by employing another way of semantic interpretation by trying 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 utterance. 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).

There has so far been very little work along these lines in the dependency grammar tradition; however, the object-level type of description closely resembles that of McCord's (1990) Slot Grammar, while the overall semantic slot-filling strategy is in the style of most approaches to information extraction and closely resembles the ones of Milward (2000) and Sunnehall (1996): 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 and Genthial (1998), who use unification for the task of passing the semantic information.

Thus, in DUMAS the dependency analysis will be enhanced with senses and slot information. All the objects in the analysis will be processed to obtain one or more McCord style frames for every nucleus in the output on the dependency parser. We will add senses to these semantic lexicon frames in two ways, automatically by machine learning methods (such as Sahlgren 2002) and semi-automatically for English by utilising the underspecified approach to semantic tagging used in Buitelaar's (1998) CoreLex. The CoreLex database is based on WordNet (Miller 1990) and covers nearly 40,000 nouns and 126 underspecified semantic types. Words that are not in the CoreLex are classified making use of the linguistic context. (Similar resources for Swedish and Finnish have to be created during the project, however, obviously at a less ambitious level.)

After adding senses to the semantic lexicon frames we will add slot information. In Slot Grammar the slot information is a declaration of syntactic relations, but we will use it to express semantic relations. The slot information of a semantic lexicon frame for a nucleus declares with which other structures its slots may or must be filled in order to be a valid structure. The semantic lexicon frames will contain slot information for at least all verbs and quantifiers and some especially interesting words chosen from the user model vocabulary. The rest of the words will have empty slot information, i.e., be identity mappings in the sense of Milward (2000). The enhanced dependency structures will be parsed by a bottom-up-parser to produce semantic representations and they will be used to extract instructions for the *AthosMail* application.

We will construct general templates along the lines of Milward to match the basic functionality of the *AthosMail* application covering as much of the expected language use as possible. For other tasks, such as finding specific information in e-mail messages or summarizing the content of an e-mail message, domain specific templates will be built.

As pointed out in the introduction we will in DUMAS in fact have two main types of task to address. Firstly, the actual interpretation of commands to the system, that is, 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. Secondly, we have the interpretation of the actual documents in the application, the e-mails in *AthosMail*. For those we need a more schematic interpretation, in order to retrieve some relevant e-mails for a user query, to extract a particular piece of knowledge from some e-mails, or for summarising the e-mails selected by the user.

In the first case, when working with spoken telephone input in the *AthosMail* application, we have to be prepared for receiving just bits and pieces of complete utterances, depending on the quality of the speech recognizer, its vocabulary, and the speech environment (noisiness, etc.). So the main aim must be to merge pieces of incomplete knowledge into sensible e-mail system function calls in the cases where either syntactic or semantic (probably both!) analysis break down. The frame representations for those commands thus represent the type of information needed in order to match one specific system function.

The strategy suggested for DUMAS in the second case can be combined with the possibility of entering information in free form; with the techniques at our disposal, we expect to be able to structure and summarize the key message content on demand.

## Acknowledgements

Thanks to Kristiina Jokinen, Markku Turunen and all the other members of the DUMAS project consortium — which includes participants from UMIST, University of Manchester Institute of Science and Technology, England; ETeX Sprachsynthese AG, Germany; KTH, Royal Institute of Technology, Sweden; and four partners from Finland: University of Tampere, University of Art and Design Helsinki, Conexor Oy, and Timehouse Oy.

Special thanks to Anders Ljusberg, Mind AB and Joel Sunnehall, Axon IT who both did their Master Thesis work at SICS on subjects related to the topic of the present paper.

# References


Buitelaar, P. 1998. *CoreLex: Systematic Polysemy and Underspecification*. Doctor of Philosophy Thesis, Brandeis University, Dept. of Computer Science, Waltham, Massachusetts.

Courtin, J., and Genthial, D. 1998. Parsing with dependency relations and robust parsing. In Kahane, S., and Polguère, A., eds., *Processing of Dependency-based Grammars: Proceedings of the Workshop, COLING-ACL'98*. Montreal, Canada: ACL.

McCord, M. C. 1990. Slot grammar: A system for simpler construction of practical natural language grammars. Research report, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

Miller, G. A. 1990. WordNet: An online lexical database. *International Journal of Lexicography* 3(4). Special issue.

Milward, D. 2000. Distributing representation for robust interpretation of dialogue utterances. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong, China: ACL. Also available as SRI International Technical Report CRC-081, Cambridge, England.

Pinkal, M. 1996. Radical underspecification. In Dekker, P., and Stokhof, M., eds., *Proceedings of the 10th Amsterdam Colloquium*, volume 3, 587–606.

Reyle, U. 1993. Dealing with ambiguities by underspecification: Construction, representation and deduction. *Journal of Semantics* 10:123–179.

Sahlgren, M. 2002. Towards a flexible model of word meaning. In Karlgren, J.; Gambäck, B.; and Kanerva, P., eds., *Notes from AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. Stanford University, California: AAAI.

Sunnehall, J. 1996. Robust parsing using dependency with constraints and preference. Master of Art Thesis, Uppsala University, Uppsala, Sweden.

Tapanainen, P. 1999. *Parsing in Two Frameworks: Finite-State and Functional Dependency Grammar*. Doctor of Philosophy Thesis, University of Helsinki, Dept. of General Linguistics, Helsinki, Finland.

Tesnière, L. 1959. *Éléments de Syntaxe Structurale*. Paris, France: Libraire C. Klincksieck. (in French).

Worm, K. L., and Rupp, C. 1998. Towards robust understanding of speech by combination of partial analyses. In *Proceedings of the 13th European Conference on Artificial Intelligence*, 190–194. Also available as Verbmobil Technical Report 222.