NATURAL LANGUAGE INTERFACES: INTRODUCTION

Franciska de Jong & Anton Nijholt
University of Twente, P.O. Box 217,
7500 AE Enschede, The Netherlands
fdejong@cs.utwente.nl, anijholt@cs.utwente.nl

1. INTRODUCTION[1]

Research aiming at the design of a natural language interface (NLI) is a very convenient vehicle for doing computational linguistics. If computational linguistics is viewed as the branch of research concerned with the modelling of linguistic knowledge needed for the building of systems that count as the product of language technology, then NLIs are interesting for at least the following reasons:

- More or less sophisticated NLIs require non-trivial parsing. At least some semantic aspects must be incorporated in the analysis of the input language.
- The application domain is restricted. NLIs are typically designed for usage within a specific context. If domain knowledge is accessible for the NLI, early disambiguation is possible. Consequently an NLI is one of the applications of natural language processing for which ambiguity is a problem that can be tackled.
- NLIs are most plausibly integrated with non-linguistic modules such as an SQL query module.
- NLIs do not preclude the integration with interfaces using other communicative modalities.
- Practical use. More and more information is stored electronically, so user friendly retrieval techniques and communication with the storage unit are very welcome.

This is not to say that working on NLP products that require large coverage grammars is uninteresting, or that products that can afford to disregard semantic matters are trivial. But often research capacity is limited. Then research directed towards the design and/or the implementation of an NLI is likely to yield results that are interesting either from a scientific or from an engineering point of view, or both. And indeed a lot of research projects work on the modelling of linguistic and non-linguistic research with an NLI as the intended application.

An interface that can be talked to without any restrictions can be built, but it will not provide human-like reactions. It would need all common sense knowledge that is possessed by a human being and it should not only be able to handle literal meanings of utterances but also tropes like irony, understatement, overstatement, simile, metaphor and metonymy. For these reasons we have to talk about domain-dependent interfaces and domain-dependent language usage. A sublanguage is the language used by a community of language users in discussing a restricted domain. Before building a natural language interface it is necessary to agree about the sublanguage that will be covered. A representative sample corpus is needed. The corpus can be analyzed by hand and a grammar and lexicon can be constructed that covers the corpus. In practice this turns out to be an extremely difficult and time-consuming task. Semantic information should be extracted from the corpus in order to reduce the number of semantically anomalous parses. It is also possible to gather semantic information through interaction with the user. Feedback by the user makes it possible that the system, by using frequency statistics, itself rejects certain parses in the future or performs a kind of preference-based parsing. One approach to the problem of constructing a grammar and a parser is to use a broad coverage natural language grammar and its parser. However, it is not only unlikely that the grammar indeed covers the corpus, but is also probably the case that the parser is too inefficient to be used in a natural language interface. It will be too slow and it will produce many parses that are irrelevant for this particular domain. One solution is to trim the broad coverage grammar. Use its

[1]Part of this paper appeared in the Proceedings of TWLT2, *Linguistic Engineering: Tools and Products*. W. ter Stal, A. Nijholt & H.J. op den Akker (eds.), April 1992.

parser on the corpus, eliminate the irrelevant parses and then collect the grammar rules that have been used in the remaining parses. They constitute a grammar for the sublanguage. It may be necessary to add some extensions in order to deal with idiosyncratic syntactic, semantic or discourse properties of the domain.

Natural language interfaces should be robust. A spelling and grammar corrector should make it possible to handle ill-formed questions or answers. However, grammars describe sentences, not dialogues. If the natural language interface is meant to support a dialogue between human user and system then it should allow input that can be considered as ungrammatical at the sentence level, but that is allowed in the context of a dialogue. Fragmentary or telegraphic style input, ellipses and anaphora are dialogue phenomena that have to be dealt with. If necessary, the interface has to interact with the user to confirm assumptions and to resolve remaining ambiguities in the input. In that case presenting a natural language paraphrase of the query by the system is useful. Users, on the one hand, sometimes give unsolicited comments or over-answer questions, or on the other hand, sometimes give very elliptical answers. A system is sometimes expected to provide more information than literally requested. A system is expected to be co-operative, to infer intentions from utterances and to detect and rectify misconceptions of the user about the capabilities of the system with which he is communicating. In order to live up to a user's expectations a system can maintain a model of a user or a group of users, that is, a knowledge source of assumptions on the user's behaviour in the dialogue. In addition one may ask that systems tune themselves to novice or expert users.

In many cases a natural language interface takes the form of a language understander that translates queries into database commands for a relational database system and a dialogue facility to interact with the user to resolve ambiguities in the queries. In this case we can make use of the structure of the database to define semantics. The main goal of the system is to supply information. This is different from an interface to an expert system, where the user statements have to be mapped into the facts of an expert system and the system is meant to solve a problem. Whatever the back-end of the interface is, for each domain a new interface has to be build. As with other grammar-based natural language processing products development of natural language interfaces can be speeded up by grammar development workbenches.

It is possible to distinguish between domain-independent tools for building and composing the natural language interface and domain-dependent tools or parts of the interface. This separation should be explored to its limit in order to reduce the effort to construct natural language interfaces and to achieve portability of the interface from one domain to another. Portability aims at reducing the efforts to create domain-dependent modules.

2. HISTORICAL OBSERVATIONS

The first natural language interfaces were built in the context of Artificial Intelligence research. Winograd's SHRDLU system can be considered as an example where Artificial Intelligence and Computational Linguistics (CL) meet. However, not many CL researchers took interest in systems that performed tasks in particular domains without modelling language usage in a linguistically acceptable way. Computational models of aspects of language description and processing without reference to a particular domain received most of the research efforts. The emerge of relational databases in the second half of the 1970s and the early 1980s, together with the development of formal query languages, however made a change. In Artificial Intelligence it was the emerge of expert systems that changed the attention for research on general artificial intelligence to attention for research on artificial intelligence for restricted domains of application. In Natural Language Processing (NLP) we see the first natural language interfaces to databases appear in the late 1970s and in the early 1980s. Examples are LIFER, ROBOT, LUNAR, PLANES, LADDER, PHLIQA1, TELI, LDC and TEAM. Some of these systems developed into commercial products. In many of them portability was strived after. In order to get a feeling for these systems we consider few of them in more detail.

In LIFER the domain-independent part is a parser and an interactive language specification facility. LIFER's grammar is a semantic grammar. Its rules describe phrases of sentences in terms of semantic categories. The specification facility allows the user to define (semantic) grammar rules, semantic rules associated with the grammar rules and words. The semantic rules associated with the grammar rules convert user's questions into database queries. TELI (Transportable English-Language Interface) maintains case frame information and has modules for the acquisition of syntactic and semantic knowledge Customization by end users can be done at any time. TEAM (Transportable English Access Data Manager) interacts with a database expert in an acquisition dialogue to obtain the information about the fields of the database. It can interact with end-users to provide answers to queries.

Crucial in the history of the research on NLIs and Question-Answering systems seems the introduction of the notion of multi-level semantics. This notion was implemented in the earlier-mentioned PHLIQUA1 system (see Bronnenberg et al[1979]). The distinction of several phases in the analysis of natural language input is not specific for the design of NLIs. The derivation of a semantic representation via successive phases of syntactic analyses is more or less standard in NLP. But in contrast to e.g. systems for machine translation, systems that process questions about information stored in a database require that the input is mapped onto fully explicit semantic representations or logical forms. A logical form (LF) serves as the basis for the evaluation of the propositional content of a question with respect to the database. Multi-staged semantic analysis functions serves at least the following two goals:

- Using multi-level semantics is a way to derive the kind of LF that is explicit enough to be transformed into the proper SQL-statements.
- Multi-level semantics is a way to organise the process by which the NLI is feeded with domain knowledge in order to disambiguate the input as early in analysis as possible.

The requirement that fully explicit logical forms be derived is perhaps more inspired by theoretical concerns than by the needs of realistic NLIs. However as shown in the paper by SIJTSMA & ZWEEKHORST in this proceedings it is hard to see what kind of reliability could be attributed to an NLI that performs only a limited amount of semantics during analysis.

3. TOPICS OF RESEARCH

3.1 Introduction

The NLP problem is tough, even if we restrict the problem to a certain domain of discourse. If our aim is to build practical systems that help users to obtain the information they want, then there is no reason to restrict ourselves to investigating natural language. Not only in this introduction, but also in other parts of this proceedings it is argued that research on NLIs is an excellent vehicle for research on computational linguistics. However, it is also an excellent vehicle for research on the integration of NLP with non-linguistic techniques for communicating information. This may lead to systems that exploit more heavily the potentialities of current and future computer systems than is done by pure NLIs.

3.2 (Spoken) Language Understanding

One obvious integration is that with speech recognition, synthesis and generation. Rather than writing or typing speech is the most natural mode of human communication. In addition, there are lots of applications in which the user will prefer to use speech or in which the user is not able to use his hands. Speech recognition and synthesis deals with sounds and signal processing. Natural language understanding deals with symbols. While the approach to speech is often statistically based, the approach to language understanding is mostly knowledge based, where knowledge is drawn from linguistic and cognitive theories, but also from sources that are called 'common sense', 'domain' or 'background' knowledge and that provide knowledge in the form of rules. Integration of speech and understanding is receiving more and more attention and is supposed to lead to spoken language understanding systems. Presently there exist several prototype spoken language understanding systems for limited interaction in limited domains (such as airplane travel planning), with very limited vocabularies and with very limited success.

In Cole & Hirschman[1992] areas of natural language research are identified that are critical to progress in the development of spoken language systems. We will summarize these areas here. Obviously, many of the areas are also critical to progress in the development of NLIs that use typed natural language input.

1. **Robust speech recognition:** Needed is minimal, graceful degradation in performance with changes in input conditions. For speech this concerns additive noise and changes in the acoustic signals, e.g., due to different speakers. Automated speech recognition of continuous speech by different speakers is not yet very successful. Improvements can be obtained not

only by using better statistical modelling techniques, but also by using higher-level sources of knowledge than acoustic sources, usually modelled by Hidden Markov Models. Higher-level sources include sources with knowledge about environment, speaker, prosody and the usual syntactic/-semantic/pragmatic knowledge sources.

2. **Automatic Training and Adaptation:** Full-scale spoken language understanding is not to be expected in the near future. Therefore it is useful to conduct research on portability, that is, adapting systems to new domains and new languages. It requires modelling of task-independent and task-specific data and training on these data. Among the challenges are the detection of new words and their syntactic and semantic classes. This means that integration of the usual stochastic models in speech recognition have to be integrated with the knowledge based paradigms of natural language understanding. Just as in NLI research for written or typed language input, the domain model should be linked to the application (e.g., SQL queries for a database). Metrics of portability should be developed to measure progress.

3. **Spontaneous Speech:** Spontaneous speech differs from read sentences. People use ill-formed syntactic constructions, hesitate, restart, correct themselves, and do not bother about improper language usage as long as their conversational partner(s) stay with them. Moreover their conversation can be interrupted by ("uh") pauses, inhalation noises, coughs, sobs and laughter (to name a few). The usual acoustic and language models do not handle these phenomena. On the other hand, spontaneous speech provides prosodic cues about coherence of a dialogue and conveys syntactic and semantic information. Research on spontaneous speech includes the study of conversational dynamics: how are certain kinds of conversational interaction signalled, e.g., turn-taking in a conversation. For spoken natural language interfaces this research can lead to more natural dialogue interaction.

4. **Dialogue Models:** In Cole & Hirschman[1992] dialogue processing is called the enabling technology for speech recognition systems. It may also be called enabling technology for NLIs that go beyond simple question-answering systems. User utterances have to be understood in the context of a dialogue structure. The dialogue and its subdialogues have to be modelled and the model provides expectations of what will be said. Goals and subgoals in the dialogue have to be identified (plan recognition). The interaction between speech recognition and recognition of dialogue structure may increase system robustness. Unfortunately, due to the complexity of dialogue structure and the problems that it causes, e.g., referent resolution, elliptical constructions, etc., there are hardly any research results which can find their way to practical systems. Apart from modelling meaning, plans and discourse structure it is also necessary to model 'initiative' in the dialogue. It is also useful to about the user or group of users. What are their characteristics and can these help in designing a more effective system. The system can also incorporate a model of the individual user and keep (changing) records of vocabulary and abilities.

5. **Natural Language Response Generation:** A NLI is expected to operate in a co-operative way. It should be helpful and answer intelligibly. It should be able to handle errors by users, to paraphrase questions, conduct a clarifying dialogue and, if necessary, correct wrong presuppositions of a user. The response of a dialogue system becomes part of the dialogue and should fit there, i.e., be sensible in the context of that dialogue. Responses can be expressed in many different ways. However, humans do not select randomly but they use expressions that suit the situation, including their dialogue partner. Again, research in this direction allows for more natural human-computer interaction. The knowledge sources for language interpretation should preferably shared with the response generation component. Obviously, a response may consist of a combination of different media, e.g., (spoken) language and graphics.

6. **Speech Synthesis and Generation:** For some NLP systems, e.g. those with telephone access or where there already is too much visual information, voice response can be the only medium that can be considered. Also for other situations voice response or voice response in combination with visual information on a screen can be useful. In text-to-speech generation syntactic, semantic and pragmatic information is not necessarily used. The result is unnatural speech. Spoken response

generation in the context of a dialogue system can make use of the knowledge of syntax, semantics and pragmatics that is available in the system. The availability of an integrated model of syntax, semantics and prosody would make it possible to provide spoken sentences with prosodic patterns and to introduce intonation and duration in order to improve speech quality.

7. **Multi-Lingual Systems:** Multi-lingual spoken language interfaces allow users to address interfaces in various languages. If the user is not required to specify her language then the system needs to identify the language that is spoken or written. A multi-lingual interface needs to respond to each user in her own language and also in other output this language should be used. Most of the research in speech and natural language aims at communication in English. More than other research these systems (including machine translation systems) ask for models for language-independent properties across languages and for language-independent representations of meaning. Obviously, these issues are also of interest if we want to increase the portability of systems.

8. **Interactive Multimodal Systems:** Integration of auditory and visual information for input and output leads to interfaces that are, compared with unimodal interfaces, more accessible, more effective in providing the user with the information she wants and they allow the user to profit from advanced information technology. Research issues are the co-ordination of input and output modalities, resolving ambiguity in one modality by using information from another, and the identification of characteristics of individual communication modalities and interactions among them. In the next subsection more attention will be paid to interactive multimodal systems.

3.3 Domains and Sublanguages

In order to realise a working natural language processing system it is not possible to take into account the whole spectrum of a natural language. Hence, we should start with a predefined domain of application that introduces restrictions on concepts and language constructs to be used. Specifying a so-called sub-language for a certain domain of application involves the stock-taking of the lexicon, the syntax, the semantics and the pragmatics of utterances in the context of the user environment. The aim of this stock-taking is to design a system that allows the user a usage of language that is as close a possible to the natural domain language usage. It also requires the identification of the relevant concepts in the application domain, the objects that are referred to, their properties and their behaviour. The language level should be related to the conceptual level.

4. Interactive Multimodal Systems

In the paper of Stock & Strapparava in this proceedings a useful account is given of the motivations that underlie the integration of different communicative modalities:

- Often humans communicate by non-verbal means. For example, they point to indicate direction and they make gestures. They communicate by doing things. Body movements, facial expressions, nodding, eye contact and eye movements can all be part of delivering a message to another.
- Integration may help in solving NLP problems. For example, menus integrated with NLP can avoid problems with ambiguity. The same holds, as mentioned above, when intonation helps in finding the syntactic structure, when a not yet completed syntactic structure helps to find the next word in a speech recognition system or when touching a screen at a certain position disambiguates a word like 'this' in a sentence.
- Computers give the opportunity to allow communication that goes beyond the usual human-human interaction. NL can be integrated with hypermedia so that a NL user has the possibility to manipulate images, sounds and texts and to explore the information in the system in whatever form that is available.

In the paper these motivations are illustrated with the descriptions of two systems: AlFresco and Maia. The focus in AlFresco is on integrating NL with pointing to a touch screen on which images of frescoes are displayed. The overall Maia project is, among others, concerned with playing the role of a concierge of the authors' institute. Research reported in the paper is on the interaction with a user *about* the dialogue that is conducted with the system. A prototype has been constructed that builds a visible dialogue structure. Interesting is the attempt to let deictic context play a role in establishing dialogue structure.

Different users may have different interaction styles. HULS & BOS argue that existing multimodal interfaces only offer limited choices of interaction style and, moreover, that each function of the system has its own style. They would like to present users the opportunity to combine styles in one single multimodal expression. An attempt in this direction is presented in their EDWARD system. A combination of mouse arrow pointing to objects on the screen, selecting from menus and using command and natural language allows them to illustrate their ideas in a rather restricted domain. An informal user study was conducted in order to observe user behaviour with the system and to obtain indications about the usability.

5. COMMERCIALLY VIABLE NLI TECHNOLOGY?

Many of papers in this proceedings report on the research in projects in which the authors participate and on the insights gained thus far. However, the interest of researchers is always somehow biased. Not always are their design decisions determined by the requirements imposed by the use of NLIs. Often the linguistic theory they adhere to, or the results of former research activities determine the kind of NLI they intend to deliver. Therefore we thought it important to add at least a few papers that would address not the design, but rather the user aspects of NLIs and in particular the commercially available natural language interfaces.

Early examples of commercially marketed natural language interfaces to databases are INTELLECT (1981, Artificial Intelligence Corp.) and THEMIS (1983, Frey Associates). INTELLECT had to be used on IBM mainframes (an IBM pc had only 64 Kbytes internal memory) and costed $70.000. THEMIS was available for $24.000. Presently, there exist commercial building tools for developing natural language interfaces, there exist commercial integrated natural language interfaces (in a DataBase Management System) and there exist commercial optional integrated natural language interfaces. Among the building tools are systems with prices ranging from $900 (*Language Workbench*, Brodie Assoc.) to $80.000 (*Easytalk*, Intelligent Business Systems Inc.). An early example of a commercial development environment is *LanguageCraft* designed by Jaime Carbonell. Apart from a grammar writers workbench it has a caseframe parser and support to connect it to different applications. A well known example of a commercial integrated natural language interface is Symantec's Q&A (*Question and Answer*). Q&A offers an integrated environment which contains a text-editor, a database management system and a natural language interface, the *Intelligent Assistant* (*IA*). IA makes it possible to access a user-constructed database in natural language, rather than in a formal query language. It is not interested in the grammatical correctness of questions that are asked. Its aim is to convert questions into intended queries. The interpretation that is determined by the Intelligent Assistant is shown on the pc's screen for confirmation. New words can be introduced by the Q&A answer. Users sometimes express doubts whether the efforts needed to learn to manage a not always intelligent IA outweigh the efforts to learn a formal query language like SQL. A similar example is *Hal* developed by Lotus Development for its spread-sheet program Lotus 1-2-3. *Natural Language*, a language product developed by Natural Language Inc. (Berkeley, Ca.), allows users to access relational databases (Oracle, Sybase Ingres, etc.) in a natural language. Questions are parsed and translated into SQL statements. A front-end with similar properties will be offered by IBM. It is called *LanguageAccess* and it parses questions and commands into SQL statements. The system is based on syntactic and semantic modules for the analysis and the generation of natural language in- and output. It can handle referents in the same or previous questions and in case of ambiguity it prompts users to indicate correct interpretations.

Any buyer of a natural language interface should consider the syntactic and semantic (or conceptual) coverage of the system, its ability to deal with elliptical utterances, anaphora and ill-formed input. Moreover, for database query it is also possible to learn a formal query language or to use a menu-based system. There exist systems that constrain the user in selecting phrases from menus to compose sentences. An example of a recent system that allows users to construct SQL-queries without having to learn SQL is *Quest* from Gupta Technologies. SESAME is Bull's system for natural language access to relational databases. Also in this case, rather than being an interface it is an environment to build a natural language menu system. Through menus the words and phrases that make up a query are proposed. The query is first translated into a logical form and then into a SQL query. NaturalLink of Texas Instruments is a similar interface.

This proceedings contains a presentation of one of the commercial NLIs that are now on the market. Natural Language from Natural Language Inc. is presented in the contribution by SIMONS. It allows users access to the major relational database management systems through natural loanguage (English). The roots of the design and the development of the system can be found in the first half of the 1980s when portability of natural language interfaces to databases became a research issue (see also section 2).

Beyond the usual interest of researchers there is not only the perspective of the party that is interested in selling. In particular the perspective of users of electronically stored data may help us to understand how remote the ideal system is in view of the state of the art in research. The contribution of HORSMAN to these proceedings offers such a perspective. His paper is a description of the needs felt at the Dutch National Archives for an automated finding aids system and the requirements that should be fulfilled by the user interface. Presently prototypes of the system are being developed in which linguistic knowledge is not yet present. However, when the user interface will be developed the use of linguistic knowledge in order to achieve 'intelligent' information retrieval will be among the design considerations.

In NERBONNE'S paper in these proceedings several issues concerning NLIs are brought up. In particular the complexity of the task, the number of problems and the necessity of human-like intelligence being available in these systems are emphasized. It is argued that currently NL technology is not sufficient to realize possible advantages of NL. One of the conclusions in this paper is that " *in spite of substantial effort and genuine progress, a general and commercially viable NLI technology remains distant*".

Particularly important is the, probably provocative, observation that simple Graphical User Interfaces (GUIs) can be built in a few person-months, while when NLIs are involved, we have to think of millions of research dollars that have been involved. On the other hand, criteria can be given that show that for some applications NLIs are more appropriate than GUIs. Obviously, this is certainly the case when we have to deal with the processing of speech.

A rather detailed and useful review of commercial NLIs, with the emphasis on the natural language fragment implemented in the systems, i.e., the linguistic competence, can be found in the paper of SIJTSMA & ZWEEKHORST in these proceedings. Since the linguistic models underlying these systems were not accessible, the competence had to be inferred by testing on syntactic and semantic phenomena. In this paper the reviews for three NLIs are presented: Intellect/KBMS, Natural Language and Q&A. The hope is expressed that this research can lead to guidelines for NLI engineering. Their research is illustrative for the growing interest in evaluating natural language processing systems and interest in users, user studies and, more generally, the human factors of NLP systems (see also the paper by BOS & HULS in this proceedings).

6. SOFTWARE ENGINEERING, INTERFACE TECHNOLOGY AND NATURAL LANGUAGE

The interest in human factors which we mentioned above is in fact a break with the traditional research in computational linguistics, since it emphasizes functionality of a system (from a users point of view), rather than cognitive or linguistic language modelling.

In this way NLI research has become part of software engineering research and interface technology. This does not only follow from the interest in users, in evaluation and in modalities other than language and speech alone, but also from the interest in current topics of advanced information technology: specification, design methodology, project management, tools and shells, object-oriented approaches to design and programming, protocol design, parallel processes and architectures, neural networks, etc. For example, both in the papers of BEUN and of Neumann in this proceedings we find object-oriented approaches to the design of the systems they discuss and in the paper by MENZEL the design of software architectures built from linked homogeneous modules for speech and language processing is discussed.

The architecture of the DISCO (DIalogue System for Co-operating agents) system, described in NEUMANN's paper has been realised with a development shell and modern design and programming methodologies. DISCO's linguistic knowledge is specified in typed feature structures.

Its grammar is besd on the Head Driven Phrase Structure Grammar (HPSG) formalism. It has a logical form module for translation into various types of back-end systems.

The basic tool for the integration of natural language components in the DISCO system is its Development Shell. In designing the architecture an object-oriented approach was chosen to support the design of modular, robust and flexible systems that are easy to extend and use. The Development Shell consists of a class hierarchy and the specification of class-specific methods. All modules of the system (parser module, morphological module, etc.) have an associated class and inherit from other classes. Methods defined for a class 'controller' are the protocols that specify the flow of control between modules. The object-oriented architectural model allows the exchange of modules and the integration of new modules in the system.

Emphasis on the design of architectures is also the focus of MENZEL's contribution to this proceedings. Strict modularization of architectures for speech and language is advocated. Not only technological, but also arguments on communicative performance and the neuronal and cognitive structure of the human analogon are given. The monolithic model of speech processing is rejected in favour of a heterarchical design on the software level, where an architecture is proposed that is built from homogeneous modules that are linked by well-defined communication channels without predefined direction of information dissemination. As in NEUMANN's contribution, it is argued that the highly modularized systems offer better possibilities for redefining an existing architecture.

7. LITERATURE

Bronnenberg, W.J.H.J., H.C. Bunt, S.P.J. Landsbergen, R.J.H. Scha, W.J. Schoemakers & E.P.C. van Utteren [1979]. The question-answering system PHLIQA1. In: *Natural Language Question Answering Systems* (Natural Communication with Computers, Vol. II). L. Bolc (ed.), Carl Hanser Verlag, München & Wien; Macmillan, London.

Cole, R.A. & L. Hirschman et al [1992]. Workshop on Spoken Language Understanding. Oregon Graduate Institute Technical Report No. CS/E 92-014, September 1992.

Lavelli, A., B. Magnani and C. Strapparava [1992]. An Approach to Multilevel Semantics for Applied Systems. In: *Proceedings of the Third ANLP Conference*, Trento.

Nijholt, A. [1992]. Linguistic Engineering: A survey. In: Proc. TWLT2 (Twente Workshop on Language Technology 2): *Linguistic Engineering: Tools and Products*. W. ter Stal, A. Nijholt & H.J. op den Akker (eds.), Memoranda Informatica 92-29.