Towards Context-adaptive Natural Language Processing Systems

Robert Porzel and Michael Strube

European Media Laboratory GmbH
Schloß-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
{Robert.Porzel|Michael.Strube}@eml.villa-bosch.de

Abstract

In this paper we will discuss several issues and requirements for enabling natural language processing systems to become context-adaptive. Given the fact that emerging systems feature speaker independent continuous speech recognition restricted to individual domains and are equipped with syntactic and semantic parsers for understanding input from these domains, we should begin to envision open multi-domain natural language processing systems. We will address the following issues arising in such open systems: How to divide natural language processing modules in context-variant and context-invariant components and how to enable the ensuing systems to detect contextual changes for processing the context-variant phenomena.

1 Introduction

The complete understanding of naturally occurring discourse is still an unsolved task in computational linguistics. Early approaches attempting to solve this task produced only toy systems. Their aim was to cope with special linguistic problems and/or to model particular cognitive capacities of natural language users. Broad coverage of syntactical constructions, lexical information sources and semantic/pragmatic behavior was not the primary concern and also far outside the scope and capabilities of these systems.

Today's linguistic development environments, representations and methodologies have shown that an approximately complete coverage may be achieved in the areas of morphology and computational grammar. Large lexical resources have been made available for strictly linguistic applications in the area of parsing and also for natural language interfaces to application areas with restricted domains.

Even though a broad coverage of lexical semantics as well as formal methods for dealing with pragmatic factors are still missing, systems are in development that can offer suitable natural language interfaces both on the reception and the production side. These systems can be employed in domain-specific real world applications. In these applications they are commonly linked to other non-linguistic applications, such as databases (Gallwitz et al., 1998), geographic information systems (Zipf & Malaka, 1999), web-based information systems (Hemphill, 2000) or task planning systems (TRIPS, Allen et al. (2000)), customer service systems (HMIHY, Gorin et al. (1997)) or standard applications, such as text-processing software (e.g. Microsoft Word) or web browsers (e.g. Netscape Navigator).

It has, therefore, been shown that, if restricted to singular domains, natural language processing (NLP) can work in real world applied systems. However, natural language processing systems are moving from stationary desktop applications to mobile systems and, at the same time, are coupled to more complex applications (Malaka & Zipf, 2000). Therefore they are likely to be exposed to different contexts and domains. As a consequence they have to get ready to be applied in more than one domain. The central question to be addressed in our paper can be phrased as follows:

> *Knowing that an NLP system A can work in domain X and an NLP system B can work in domain Y, how can we build a system C that works in X and Y?*

In order to address problems arising from this situation we have to determine context-variant and -invariant factors and we have to find ways, methods and technologies of enabling NLP systems to adapt to the context-variant factors.

As a practical implementation of some of the previously mentioned points we present an applied NLP system that enables users to gain intuitive access to various underlying information technology (IT) services without requiring prior knowledge of these services' internal modalities. The fundamental idea is to enable users to employ complex and heterogeneous systems and information sources through a single natural language interface and processing system. After discussing related systems and their solutions (Section 2), we will employ the Deep Map system (Section 3) as an example for such a multi-domain and, hence, context-adaptive system.

2 Applied NLP Systems

2.1 Overview

Despite their enormous contribution to the field of Computational Linguistics we will exclude such systems from our considerations that focus on aiding human-human communication. Research systems such as VERBMOBIL and the C-STAR translators or commercial systems such as the *Personal Translator*© do not feature the kind of interplay that arises in the field of human-computer interaction. These systems do not detect the intention behind the user's input[1] and do not have to *answer* to this input.

There are several academic and commerical tools available which include information extraction systems, information retrieval systems, knowledge acquistion systems, spell-checker, auto summarizer or dictation systems. Usually these tools are seen as components of NLP systems and not as systems on their own. Therefore, we exclude these tools from our considerations as well.

Previous systems focusing on human-computer interaction are by and large either focused on sophisticating their natural language input (understanding) side or their output (production) side. An additional common characteristic of existing systems is that they are bound to single, specific domains and their employment of (a priori) defined scripts for dialog management. However, several fully blown-up NLP systems exist which we will describe shortly here.

- The TRAINS system (Allen et al., 1996) and its successor TRIPS (Ferguson & Allen, 1998) are spoken dialogue systems which attempt to help users to solve tasks. Though this attempt involves a considerable amount of work in developing corpora and NLP components, the main emphasis lies on the planning part of the system. Also, both systems deal with tiny domains.
- Existing multi-modal dialogue systems like the QuickSet system (Cohen et al., 1997) or the Command Talk spoken language system (Stent et al., 1999) are quite narrow in focus and coverage of speech input. The vocabulary of these systems covers only a few hundred entries and the domain knowledge contains only a few dozen concepts. These systems allow interaction only in a very controlled fashion.

[1] We are aware that translation needs context-sensitive understanding of an utterance's meaning, however, that is not always the same as understanding the underlying intention.

- The AT&T telephone-based system *May I help you?* (Gorin et al., 1997) is – like the majority of spoken dialogue systems coming out of AT&T – restricted to a single domain with not much more than a dozen conversational topics. The same is also true for EVAR (Gallwitz et al., 1998) and the Philips train timetable system (Aust et al., 1995).

2.2 Requirements for Multi-domain NLP Systems

Based on the overview in the next section we propose to proceed in the following way in order to address questions concerning the application of NLP systems in multiple domains. Firstly, we have to determine context-variant and -invariant factors in all modules of an NLP system and secondly, we have to enable NLP systems to detect their current context and adjust their context-variant factors to that.

On a rather superficial level we can organize particular (computational) linguistic knowledge sources into context-variant and -invariant ones (see Table 1).

	Context-variant	Context-invariant
Speech Recognition	vocabulary language model	
Syntax & Parsing	open class lexicon parsing	closed class lexicon grammar
Semantics	disambiguation domain knowledge	lexical semantics á la DRT common-sense knowledge
Pragmatics	intention recognition	dialogue acts

Table 1: Context-variant and -invariant levels or modules

However, these decisions are not that simple because each level (or linguistic phenomenon) may be described by several factors. Strube & Wolters (2000) show that the problem of pronominalization can be described by a dozen factors some of which are genre-independent. With the method described in Strube & Wolters (2000) it should be comparatively simple to figure out which factors are context-variant and -invariant. However, after determining a set of relevant factors, one needs a corpus containing data from different contexts or domains. These data have to be annotated with the relevant factors. For discovering differences among contexts standard statistical techniques can be applied for each factor and to each context. One possibility to do that is to train statistical models on all domains except one and then to test extracted hypotheses on this domain. This procedure should be applied to all domains so that differences among domains show up. So, the problem boils down to the availability of annotated dialogue corpora in different domains.

The second problem concerns the adaption of NLP systems to different environments and applications. For deciding which domain should be considered to be currently in focus, an NLP system needs to be context-aware. This can be achieved by employing sensors beyond its NLP capabilities. Examples for such sensoric inputs are global position systems (GPS) for location awareness, computer vision for location and situation awareness or audio sensors for situation awareness. Within the Deep Map research framework introduced below we employ GPS for location awareness and audio for noise and background monitoring.

3 Solving the multiple domain problem in applied NLP systems

Employing the Deep Map system (Malaka & Zipf, 2000) as an example[2] we will exemplify individual approaches for solving the problems shown above. A commonality between all solutions is the fact that additional, partly non-linguistic, knowledge stores and sensors have to be linked to the linguistic parts of the system in order to enable them to adapt to contextual variances and to supply them with the contextually relevant information.

Within Deep Map we differentiate between the NLP system, called *Talking Map* and the underlying IT system. Deep Map is a long term research project that aims at building an intelligent mobile tourist guide. It features a layered architecture consisting of (see Figure 1)

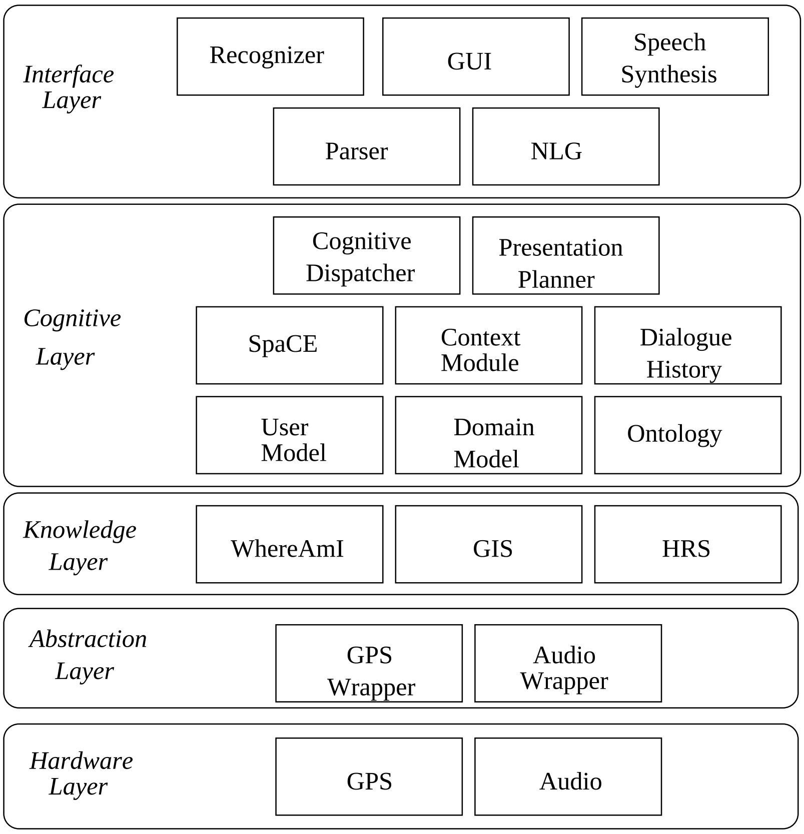

Figure 1: Agents in Deep Map are located in several layers: interface, cognitive, knowledge, abstraction and hardware layer.

- the interface layer (components that directly interact with the user such as the graphical user interface),
- the cognitive layer (components that try to understand what the user meant and to react accordingly, e.g. the presentation planner),
- the knowledge layer (components that provide basic services such as a databases, geographical information systems and hotel reservation systems),

[2]Several of the technologies and approaches discussed here are part of the SmartKom and the EMBASSI research framework and is undertaken in collaboration with our research partners from these consortia (for further information see www.embassi.de and www.smartkom.com).

- the abstraction layer (components that capsule the individual hardware systems and provide interfaces for the service modules),
- the hardware layer (actual devices and sensors).

Unlike many other NLP systems (with the exception of recent systems such as TRIPS (Allen et al., 2000) or the DARPA Communicator), Talking Map is based on an agent paradigm: All components are connected to a shared bus that is used to send messages back and forth. This greatly increases the amount of interaction between them. The underlying communication infrastructure is based on standards (such as CORBA (www.corba.org) and FIPA (www.fipa.org)) allowing platform independent exchange of messages and the seamless addition of new agents. The platform supports asynchronous and synchronous communication, which can take place in a peer-to-peer or a broadcast scenario. Therefore, information either can be directed to a single receiver, or be made available to all agents. Each agent defines a filter for the messages that are relevant to it. This facilitates the addition of new agents - in many cases without the need to modify the other agents. Their additional functionality, then, directly benefits the entire system, as they can obtain messages they have an interest in and distribute their results to the other agents. In order to make that possible the Talking Map system communicates using a shared ontology consisting of roughly two hundred classes and concepts.

Currently, Talking Map has three subsystems that interact to react to utterances of the user:

- A subsystem (NL IN) allowing for natural language input, i.e., spontaneous speech recognizer, syntactic and semantic parsers that recognize what the user said, transform the utterance to an internal representation and augment it with linguistic information.
- A subsystem that takes the output of NL IN and further enriches it with extra-linguistic information. Within this subsystem an appropriate reply to the user's input is computed, and language-independent verbalization plans are produced. Additionally modules, such as the discourse history and context model reside in this subsystem.
- A subsystem (NL OUT) that analyzes the language-independent verbalization plans, generates linguistic surface structures from them and supervises the synthesis of these surface structures into speech.

In the following sections we will look at the individual components and their context-variant and -invariant sides.

3.1 Speaker-Independent Spontaneous Speech Recognition

Within Talking Map we integrated a speaker independent continuous speech recognizer that is capable of processing spontaneous speech input.[3] Next to the employment of crossword context-dependent phoneme models and additional noise models to cover spontaneous speech effects like hesitations, breathing and the like, we employ a statistical language model, that is highly domain dependent.[4] In order to be area-independent, all names for points of interest such as street names, sights, restaurants are mapped to different class symbols. For the target area, i.e. Heidelberg, the relevant names are added to the appropriate class and also to the phonetic dictionary. These names introduce the difficult and yet

[3]The recognition engine we use is the Janus Recognition Toolkit (JRTk3.2) (Finke et al., 1997) that was already successfully employed for a variety of languages.

[4]Research efforts are also underway to include two additional contextual variations in regards to acoustic and phonetic models. One concerns the recognition of the user's language itself, e.g., (Schultz & Waibel, 1999), and the other the adaptation to individual variants such as the user's dialects (Fuegen & Iviza, 2000) and the acoustic situation, e.g., microphones, environments such as cars etc. (Westphal & Waibel, 1999).

unsolved problem of foreign word pronunciation. As a first result of a study where American speakers were asked to read German street and place names aloud, we derived a set of rules how those speakers segmented the words in syllables and how uncommon strings are pronounced. Since the speakers are not consistent in some cases (like for words ending with '*e*'), up to ten pronunciation variants were created for each word.

Nevertheless both statistical models and lexica have to be domain dependent and limited respectively in order to achieve speaker independence. The obvious solution pursued not only within the Deep Map framework, but also within SmartKom (Rapp et al., 2000), is to solve the multi-domain problem by enabling the recognition engine to adapt weights within the lexicon and shift the acoustic model during runtime as well as enabling them to discard grammars, statistical models and lexica completely and load new ones in response to specific contextual changes. It is important to note that first of all the ability to make such (online) adaptions is the first prerequisite for solving the multiple domain problem in speech recognition. The second prerequisite concerns the information needed to trigger these adaptations.

The following examples will show how this information can be gained and supplied in the case of adapting lexica on the fly, changing lexica and changing language models.[5] The stochastic language model's n-gram probabilities for n-tuples of words have the additional effects that the recognition rate for individual words also varies with respect to the language model. A system, for example, using a language model based on corpora where names of places where found as often as references to dates and times may weight individual word occurrences such as the location name *Donnersberg* and day of the week *Donnerstag* equally. The overall recognition rate could potentially be increased if the recognition system was to be notified by the natural language generation system that the user has been asked a *Where* or *When* question. Empirically validated measures of likelihood could then be re-distributed in response to the dialogical context.[6]

A different form of adaption, concerns dynamic lexica. These concern both speech recognition as well as speech synthesis alike. In open systems such as the Deep Map systems where constantly changing and novel information can enter the system during run-time and be presented to the user, e.g. lists of hotel names, restaurants or films running in cinemas, it is very likely that the user will refer to this information, e.g., *book me two tickets for being john malkovich*. It is, therefore, necessary to find ways of processing this information phonetically and structuring it according to the appropriate lexical classifications. These will then be used to create new recognition and synthesis lexica dynamically, which, in turn, enable the system to recognize and synthesize these entities appropriately or in most cases at all. This can be done technically as shown in (Rapp et al., 2000) and is instrumented via a coordination of presentation management and speech recognition and synthesis, whereby the presentation manager informs the speech modules about the given propositional context.[7]

The biggest problem, however, arises in actual cross-domain changes and can, therefore, be called *domain-detection problem*. The two essential prerequisites – before possible solutions to the domain-detection problem can have an effect – are obtaining the individual language models and enabling the recognizer to adapt to a new domain during run-time. Assuming we have enough data and corpora for the targeted domains we can construct individual models for the specific domains and identify the domain variant and in-variant factors, employing the method shown in Strube & Wolters (2000), as we have done for the tourism domain in Deep Map. Given these techniques and models, the remaining problem lies in triggering these switches. In the Deep Map framework a central focus lies on constructing a mobile

[5]For acoustic model and phoneme adaptation techniques see Westphal & Waibel (1999) and Fuegen & Iviza (2000).

[6]However, it is not clear, whether such a strategy actually enhances the performance (though Litman et al. (1999) were using high-level discourse information for improving low-level speech recognition).

[7]Technically, we have implemented this procedure in such a way that the presentation manager informs the context model of changes in the propositional context and the relevant speech modules subscribe to these changes, and are, therefore notified of them.

tourist information system, which therefore needs both geographic and touristic information as well as location awareness (Malaka & Porzel, 2000). A simple example for such a change in situational context, that could trigger a language model switch is a user touring various sights and interrupting that tour by entering a restaurant or a museum. A system such as Deep Map that combines both a geographic information system and a global positioning system has the potential to alert the NLP modules about the fact that the user has entered a building which is an Italian restaurant, a museum of modern art or else. More subtle changes in the situational context concern not only the speech recognition system, but also the parsing and understanding modules, which are described in the following section.

3.2 Parsing and Intention Recognition

In Deep Map the spoken language input is analyzed by the MISO parser (Gavaldá & Waibel, 1998) using semantic grammars for robust speech analysis. Although the generated parse tree contains the semantic information of the sentence, its form is not independent of the syntactic structure of the sentence. For this reason, a semantic construction algorithm converts the parse tree into a normalized partial representation, a set of typed feature structures (Carpenter, 1992). Each feature structure represents either an object in the domain or a relation between objects. A declarative knowledge base called type hierarchy holds information on the domain.

Analogous to the dynamic language model discussed above we can employ multiple type hierarchies or semantic grammars for the possible domains. The problem however is twofold. On the one hand the semantic information of an utterance has to be extracted. On the other hand that information has to be combined with the dialogical, propositional and situational context in order to get at the intention behind the utterance. Considering a simple utterance such as *On Thursday*, a purely linguistic and semantic analysis does not suffice to enable the underlying IT system to do something meaningful with it. Assuming the system really does constitute a multi-domain system with various capabilities, such as integrated hotel reservations systems, public transport schedules or online ticket reservation systems, then each of those sub-systems could potentially be recipient of such an utterance. It is, therefore, vital that each parsed utterance is put in its greater dialogic context.

A more difficult example concerns the extraction of intentions by recourse to world and domain knowledge. This issue has been noted frequently in research on the semantics and pragmatics of questions and answers. Knowing that virtually every question can be answered truthfully and completely uselessly at the same time, by violating the Gricean maxims, we can add an additional constraint that arises in applied systems, i.e., for each task and domain there is task- and domain-specific knowledge which is needed to create an interface between the NLP and IT systems. It does not make much sense in most contexts, for example, to let a hotel reservation agent look for vacancies in hotels without specifying at least arrival and departure dates. Looking at one of the more complex examples such as a user asking *Is there a bakery around here*, we find that it is certainly not part of the NLP system to find out what *here* really is or where bakeries are actually located, however we would like to extract the user's intention in such a way that the system can decide to direct the user to the nearest bakery, if they are open, or suggest alternative means of obtaining food[8]. An ontological model, therefore, is needed that supplies the necessary information that bakeries are places for buying snacks and that they can be open or closed. As for linking this augmented semantic representation to real worlds state of affairs and processing systems we need mapping agents, as are described in the following section.

[8] Again, situational contexts are imaginable where the user's interests actually concern only the existence of specific objects and answers such as *Yes* or *No* would be appropriate.

3.3 Query and Answer Translator

Note that after going through the NL-IN pipeline we are still faced with a representation, that has no relation to the real world. This entails that no knowledge about existing objects or which objects in the real world correspond to a linguistic expression – such as *i need a hotel near the train station* – is available at that moment. In order to answer to any query it has to be translated. This task involves two fundamental capabilities:

- segmenting complex representations into appropriate chunks,
- distributing these chunks to the corresponding recipients.

In our experience there is neither a one-to-one correspondence between representational components and chunks nor a context-invariant mapping between segments and recipients in multi-domain systems. The first recipient of such a complex representation, therefore, has to disassemble the message and find out what kind of actions could result from this. Furthermore it needs to check whether embedded concepts could be considered sub-actions. In our case a module called CoDI[9] would note that in this dialogical context the quantification *near* could be resolved by producing a SpaCEAction[10]

The SPACE agent (Kray & Blocher, 1999; Kray & Porzel, 2000) is responsible for the deeper semantic and pragmatic understanding and interpretation of spatial concepts that the user employed when talking to the system. References to real world objects have to be resolved, spatial relations need to be established, and the parser output must be analyzed further in order to ensure that the intention of the user is fully understood. Then, an internal representation in terms of the system's model of the real world is generated that other agents can work with.

Once the user's goal has been identified, SPACE tries to generate an appropriate reply to any query that is related to space (localization, route directions, object identification, etc.). In this task, it relies on other agents such as a geographic information system or the global positioning system (GPS) to provide basic information (such as the user's location or the names and positions of landmarks) that is needed to reason about an educated response to the user's input. The spatial content of the utterance will be resolved by recourse to the context model, the user model and the GIS. Resulting form the user's current position (WhereAmI Agent), mode of transportation and known user preferences, a polygon describing and area around the specified object, i.e. railway station, can be returned.[11]

3.4 Generating Answers and Supplying Information

Having discussed the problems arising in natural language recognition and understanding[12], we can see that contextual issues in terms of natural language production differ in several respects. First of all, an understanding component has to be able to handle unknown input. In contrast, a generation component always operates with complete knowledge about the linguistic and extra-linguistic information it requires.[13]

[9]CoDI stands for *Cognitive Dispatcher*.

[10]SpaCE is an agent conglomerate that resolves different kinds of tasks related to spatial cognition, hence, the acronym is resolved as: Spatial Cognition Engine.

[11]The motivation for choosing an agent system as underlying infrastructure and computational paradigm was, in part, motivated by the significant complexity of the proactive communication involved. The alternative approach, i.e. to construct a general manager, would have involved fitting every bit of domain-specific knowledge into this module.

[12]One could regard the ability to know how an input can be translated and answered to be part of the understanding of that input. Within the neural theory of natural language processing, for example, the understanding processes result in simulation specifications, that represent the knowledge that is needed to compute a corresponding action, belief change or simulation (Narayanan, 1997).

[13]It is important to keep in mind that we have not discussed problems arising in multi-modal systems allowing for input via speech, gestures, keyboard, mouse, etc. and producing multi-modal output, employing natural language, graphics etc. It is

The central task of natural language generation (NLG) system is take what the underlying IT system has computed as input and to transform this preverbal input or information into natural language. It is clear that this input eliminates the domain-detection problem with respect to NLG as the input serves to specify the current domain. Other contexts within the NLG, such as user modeling and dialogical context constitute well-studied phenomena and already have been applied in various systems (Paris, 1993). Within Deep Map user modelling entails an additional aspect. The produced information needs to be presented fitting to the overall situational context. Knowing that the user is currently driving a car, walking down a street or visiting a church, we can identify additional constraints on the output to be produced. These constraints concern the level of complexity and detail of the surface structures to be produced and depend on the user's available mental resources, or constitute *behavioral* constraints dependent on the given environment.

We see that the context problem in NLG differs as restrictions on lexica and grammars are not the same as in natural language understanding, furthermore many things such as pronominalization, referential status of discourse referents etc. are by and large domain independent. The fundamental task of an NLG system, i.e. to enable the user to create an appropriate mental model of the state of affairs to be conveyed, however, can be improved by taking propositional contexts, dialogical contexts and situational contexts into account.

As we have already noted in section 3.1, dynamic lexica *cum* phonetic modeling are also needed for speech synthesis in open multi-domain systems. Additional factors concern the prosodic structuring of the synthesized speech as well as contexts arising from multi-modality issues, e.g. the synchronization with lip-synchronous avatars. Work on the former, e.g., on the conceptual planning and linguistic realization of information structuring (Endriss & Klabunde, 2000) have not yet progressed to a level where they can be implemented in NLP systems, however, we can already see that these questions relate to the dialogical context and are by and large domain-independent.

4 Conclusions

Human-computer interaction by means of natural language will hardly ever become as context sensitive as human-human interaction. Considering the enormous amount of factors that induce a given context, we find:

- listener-related factors, e.g., back-channeling (Ygnve, 1970), common ground (Krauss, 1987; Horton & Keysar, 1996) or social factors (Buhl, 1996),

- state of affairs-related factors encompassing both our world-knowledge and issues concerning our representations thereof (Hobbs, 1985; Habel, 1996),

- language-related factors, e.g., structural properties (Carroll, 1993).

The two fundamental deficiencies of NLP systems, therefore, can be identified as lack of propositional and tacit knowledge and lack of sensoric input and processing capabilities. The ongoing emergence of complexer dialog systems and their migration into real world applications points to the fact that solving the multi-domain problem will become a focal research issue in computational linguistics.

important to keep in mind, however, that the coordination and synchronization necessary for multi-modal systems creates an additional context problem.

References

Allen, James F., Donna Byron, Myroslava Dzikovka, George Ferguson & Amanda Stent (2000). An architecture for a generic dialogue shell. *Natural Language Engineering*. To appear.

Allen, James F., Bradford Miller, Eric Ringger & Teresa Sikorski (1996). A robust system for natural spoken dialogue. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, Cal., 24–27 June 1996, pp. 62–70.

Aust, Harald, Martin Oerder, Frank Seide & Volker Steinbiss (1995). The Philips automatic train timetable information system. *Speech Communication*, 17:249–262.

Buhl, Heike M. (1996). *Wissenserwerb und Raumreferenz: Ein sprachpsychologischer Zugang zur mentalen Repräsentation*. Linguistische Arbieten 357. Tübingen: Niemeyer.

Carpenter, Bob (1992). *The Logic of Typed Feature Structures*. Cambridge, U.K.: Cambridge University Press.

Carroll, Mary (1993). Deictic and intrinsic orientation in spatial descriptions: A comparison between English and German. In J. Altarriba (Ed.), *Cognition and Culture: A Cross-cultural Approach to Psychology*, pp. 23–44. Amsterdam, The Netherlands: North-Holland.

Cohen, Philip R., Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen & Josh Clow (1997). Quickset: Multimodal interaction for distributed applications. In *Proceedings of the Fifth Annual International ACM Multimedia Conference*, Seattle, Association for Computing Machinery. http://www.cse.ogi.edu/CHCC/Papers/philPaper/acm.html.

Endriss, C. & Ralf Klabunde (2000). Planning word-order dependent focus assignments. In *Proceedings of the 1st International Conference on Natural Language Generation (INLG 2000)*, Mitzpe Ramon, Israel, 12-16 June, 2000.

Ferguson, George & James F. Allen (1998). TRIPS: An intelligent integrated problem-solving assistant. In *Proceedings of the 15th National Conference on Artificial Intelligence & 10th Conference on Innovative Applications of Artificial Intelligence*, Madison, Wisc., 26–30 July 1998, pp. 567–573.

Finke, Michael, Petra Geutner, Herrmann Hild, Thomas Kemp, Klaus Ries & Martin Westphal (1997). The Karlsruhe-Verbmobil speech recognition engine. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP '97)*, Munich, Germany.

Fuegen, Christian & Rogina Iviza (2000). Integrating dynamic speech modalities into context decision trees. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2000)*, Istanbul, Turkey, 5-9 June, 2000, Vol. 3, pp. 1277–1280.

Gallwitz, Florian, Maria Aretoulaki, Manuela Boros, Jürgen Haas, Stefan Harbeck, R. Huber, Heinrich Niemann & Elmar Nöth (1998). The Erlangen spoken dialogue system EVAR: A state-of-the-art information retrieval system. In *Proceedings of 1998 International Symposium on Spoken Dialogue (ISSD 98)*, Sydney, Australia, 30. Nov., 1998.

Gavaldá, Marsal & Alex Waibel (1998). Growing semantic grammars. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, Montréal, Québec, Canada, 10–14 August 1998, pp. 451–456.

Gorin, Allen L., Guiseppe Riccardi & Jerry H. Wright (1997). How may I help you? *Speech Communication*, 23:113–127.

Habel, Christopher (1996). Räumliche Repräsentationsformate für die Sprachproduktion. In R. Meyer-Klabunde & C. von Stutterheim (Eds.), *Proceedings of a Workshop on Conceptual and Semantic Knowledge in Language Production, Arbeiten aus dem SFB 245 "Sprache und Situation", Report No. 92*, pp. 57–86. Universities of Heidelberg and Mannheim.

Hemphill, Charles A. (2000). A voice browser and tool for hands-free computing. In *Proceedings of the International Conference on Wearable Computing (ICWC 2000)*, McLean, Va., 16-17 May, 2000.

Hobbs, Jerry R. (1985). Granularity. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, Los Angeles, Cal., 18–23 August 1985, Vol. 1, pp. 432–435.

Horton, William S. & Boaz Keysar (1996). When do speakers take into account common ground. *Cognition*, 59:91–117.

Krauss, Robert (1987). The role of the listener: Addressee influences on message formulation. *Journal of Language and Social Psychology*, 6:91–98.

Kray, Christian & Anselm Blocher (1999). Modeling the basic meaning of path relations. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1-6 August, 1999, pp. 384–389.

Kray, Christian & Robert Porzel (2000). Spatial cognition and natural language interfaces in mobile personal assistants. In *Proceedings of the ECAI 2000 Workshop on Artificial Intelligence in Mobile Systems*. To appear.

Litman, Diane J., Marilyn A. Walker & Michael S. Kearns (1999). Automatic detection of poor speech recognition at the dialogue level. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, Md., 20–26 June 1999, pp. 309–316.

Malaka, Rainer & Robert Porzel (2000). Integration of smart components for building your personal mobile guide. In *Proceedings of the ECAI 2000 Workshop on Artificial Intelligence in Mobile Systems*. To appear.

Malaka, Rainer & Alexander Zipf (2000). Deep Map: Challenging IT research in the framework of a tourist information system. In *Proceedings of the International Conference on Information and Communication Technologies in Tourism (ENTER 2000)*, Barcelona, Spain, 26-28 April, 2000. Berlin, Germany: Springer.

Narayanan, Srinivas (1997). *KARMA: Knowledge-Based Active Representations for Metaphor and Aspect*, (Ph.D. thesis). Computer Science Division, University of California, Berkeley, Cal.

Paris, Cécile L. (1993). *User Modeling in Text Generation*. London: Pinter.

Rapp, Stefan, Sunna Torge, Silke Goronzy & Ralf Kompe (2000). Dynamic speech interfaces. In *Proceedings of the ECAI 2000 Workshop on Artificial Intelligence in Mobile Systems*, p. to appear.

Schultz, Tanja & Alex Waibel (1999). Language adaptive LVCSR through polyphone decision tree specialization. In *Workshop on Multi-lingual Interoperability in Speech Technology (MIST '99)*. Leusden, The Netherlands, September 1999.

Stent, Amanda, John Dowding, Jean Mark Gawron, Elizabeth Owen Bratt & Robert Moore (1999). The Command Talk spoken dialogue system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics,* College Park, Md., 20–26 June 1999, pp. 183–190.

Strube, Michael & Maria Wolters (2000). A probabilistic genre-independent model of pronominalization. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, Seattle, Wash., 29 April – 3 May 2000, pp. 18–25.

Westphal, Martin & Alex Waibel (1999). Towards spontaneous speech recognition for on-board car navigation and information systems. In *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH '99)*, Budapest, Hungary, 5-9 September 1999.

Ygnve, Victor (1970). On getting a word in edgewise. In *Papers from the 6th Regional Meeting of the Chicago Linguistics Society*, pp. 567–577. Chicago: CLS.

Zipf, Alexander & Rainer Malaka (1999). Web-basierte Planung und animierte Visualisierung von 3D Besichtigungstouren im Touristeninformationssystem Deep Map. In *Symposium für angewandte graphische Informationsverarbeitung (AGIT '99),* Salzburg, Austria,1999.