

The Practical Use of Valencies in the Erlangen Speech Dialogue System CONALD

Günther Görz and Bernd Ludwig, Erlangen

11. August 2005

1 Motivation

Within the broad spectrum of applications of computational linguistics the conversational computer is often regarded as the ultimate challenge.

“The conversational computer paradigm provides a way to articulate the properties and challenges of natural language applications and the ways those challenges are being addressed within the field of computational linguistics.”

(from “Applications of Language Technology” by Cole et al. in the *International Encyclopedia of Linguistics*)

2 Applications of Language Understanding

Linguistic computer applications do not just involve language per se, but also *interactions* between linguistic knowledge and other areas of knowledge. These interactions pose the main challenge for any more or less general approach to natural language understanding: Constructing the meaning of a natural language phrase or sentence is guided by different principles than constructing sentences for a formal (artificial) language used for the representation of knowledge in a computing environment. There are two main differences:

2.1 Natural versus Formal Languages

Formal languages do not encode implicitly semantic relations between parts of a sentence. On the contrary, in the natural language example

I want to watch a thriller tonight.

grammatical markers (*subject, object, and attribute/adverbial*) are used to establish relations between the described intention *want to watch*, the agent *I* who has this intention, the TV programme *thriller*, and the indication of a time span (*tonight*) during which the programme should be on air.

In formal languages, such relations are non-ambiguous because each constituent of a sentence serves unambiguously as a functor or argument of another constituent:

want-to-see(*I, thriller, tonight*)

This functor-argument-structure defines precisely and without any doubt that *tonight* indicates the time span when *want to see* should take place. This would probably be the standard interpretation for the natural language sentence as well.

However, depending on the content of the sentence, grammatical markers may be insufficient for avoiding ambiguities, as in the example:

I want to watch the thriller at 8 pm.

In this sentence, there are no grammatical rules preventing the prepositional phrase *at 8 pm* from being attached to the noun phrase *the thriller*. As a consequence, it depends on the evaluation of both phrases which reading is the correct one. If there was a thriller recorded the day before it could be just that one the user wants to see, not necessarily a thriller transmitted when the utterance is made.

2.2 Context-Dependent Evaluation

The need to evaluate phrases in order to understand them hints to the execution of algorithms in order to make the evaluation effective. Such algorithms apply the information contained in phrases for input and compute output that is in turn used to determine all the possibilities to understand an utterance and to react on it. A consequence of this approach is that the capabilities to understand an utterance are determined and limited by the capabilities to process given information with the help of problem-specific algorithms.

As an example, consider an intelligent interface to a TV set that tries to propose programmes to the user that match certain user-defined criteria. A programme is described formally by a filled data structure as the following:

```
<programmes>
  <IsNowNext/>
  <TransportStreamId>1073</TransportStreamId>
  <OriginalNetworkId_ServiceId>93787</OriginalNetworkId_ServiceId>
  <EventId>43932</EventId>
  <StartTime>2005-07-28 12:00:00</StartTime>
  <Duration>00:30:00</Duration>
  <RunningStatus>1</RunningStatus>
  <Pil>238764127</Pil>
  <Title>Eisenbahnromantik</Title>
  <ShortInfo>Im Zug von Bratislava nach Ungarn</ShortInfo>
  <LanguageCode>7693668</LanguageCode>
  <ExtendedInfo>Der "Eisenbahnromantik"-Sonderzug fährt in die schöne
    Slowakei und nach Ungarn. Gefahren wird mit Diesel-, Dampf- und
    elektrischen Zügen. Der erste Teil der Reise führt uns über
    Bratislava, Zvolen, Poprad-Tatry an die ungarische Grenze.
  </ExtendedInfo>
  <ExtendedLanguageCode>7693668</ExtendedLanguageCode>
</programmes>
```

In order to evaluate the user query

Are there any documentaries about travelling now?

three different types of pragmatic evaluation have to be distinguished:

- *documentaries*: Look for programmes of this genre!

- *about travelling*: Which programmes cover this topic?
- *now*: The programme should be transmitted at the time of this saying.

Three different algorithms have to be employed for evaluating each of the above types:

- Data base lookup for finding the right genre,
- Analysis of the `ExtendedInfo` field to find the right content,
- Temporal reasoning for finding the right time interval.

This situation is typical for complex software systems. As a consequence, it is not practicable to rely on a single formal language for covering all types of evaluations. On the other hand, how can one still implement modules for natural language analysis that can be configured for different applications and therefore be used in contexts with totally different and often unpredictable types of evaluations?

Dialogue systems, e.g. for assisting human users in performing practical tasks, are a typical example for the variety of scenarios and applications just addressed:

In cooperation with some technical application — e.g. a database system providing information of some kind — train information, weather, stock market, theatre programmes, etc., a system to order merchandise, a system to control devices — a goal expressed by the user, usually in spoken language, shall be satisfied, if possible. If not, the system should be able to explain why and provide further help.

Aside from dialogue systems, there are of course many other applications of computational linguistics such as

- Information retrieval (“the Google challenge”),
- Automatic machine translation,
- Automatic text summarization,

but also

- to evaluate linguistic hypotheses, as far as they are fully formalized,
- to simulate human language processing,

and many more.

3 Grammar and Valencies in Natural Language Understanding

The solution we propose is that any formal representation of natural language within a natural language understanding system must resort to a formal language that is able to represent explicitly what is entailed explicitly and implicitly in a natural language utterance. This language must be sufficiently expressive to cover valencies, (generalized) quantifiers, modifiers, modalities, and coordination operators that are not available in almost every formal (logical) language due to limitations in computability and decidability.

In a second step of understanding, a statement that represents the semantics of a natural language sentence or utterance has to be translated — by applying evaluation as in the example above, not just syntactical transformations — into an expression of the formal language(s) the underlying computational environment works with.

In this paper, we will focus the discussion on the use of valencies and address only marginally the more broader problem of dialogue analysis and generation. For a better understanding of the role of valencies in our work, we first present the general framework and will then point out how valency plays an essential role in its linguistic analysis component.

3.1 The Erlangen Dialogue System CONALD

Our research on language understanding is embedded in work on dialogue understanding. It aims at systems for rational dialogues based on a pragmatic approach. So, our overall perspective is that of language as action in which speech acts play a central role. As a consequence, the traditional “computational linguistics pipeline” — syntactic derivation, semantic resolution and inference, and, eventually pragmatic analysis — is turned upside down. Pragmatics gets control in a way that processing is controlled by a dialogue management module.

CONALD is a spoken language dialogue system system¹ which enables the interaction between users and a technical system in dynamic environments. The primary goal of its design is to achieve quick configurability for various applications. It combines deep syntactic and semantic analysis, discourse processing, and language generation and features a complex semantics-pragmatics interface in the sense of [BRIETZMANN AND GÖRZ1982]. Semantics is defined in terms of an extended version of discourse representation theory (DRT). Discourse and application pragmatics are considered independent; user utterances affect application pragmatics when their speech acts are executed. They can be specified in a script language interpreted by the dialogue manager.

In dynamic environments, changes can be consequences of user requests as well as of external events in the application. The design of our system allows configuration for a wide range of applications (like household applications, the automotive environment, medical purposes, etc.). The user may give commands in natural language by speaking into a microphone.

The parser transforms the user’s utterance into a semantic representation, from which the dialogue system derives a goal how to change the environment. A plan to reach this goal is computed and executed by a group of agents. Under the assumption of a closed world, the overall system features hierarchical planning, plan execution, and plan observation by several agents with different responsibilities and capabilities. In this way, system knowledge is distributed and organized hierarchically corresponding to the tasks each agent has to execute.

Agents are organized in a hierarchy of layers. On top there is the Assistance System (see Fig. 1) which plays the role of an interface between the application and the dialogue system. The main task of the Assistance System is to satisfy the user’s goals by computing and executing high-level plans.

Negotiation between system and users is handled by a dialogue manager. For user utterances, input from a speech recognizer is processed by a parser resulting in a representation of the meaning of the utterance.

¹Acknowledgment: Our work is supported by the Bavarian Research Association FORSIP

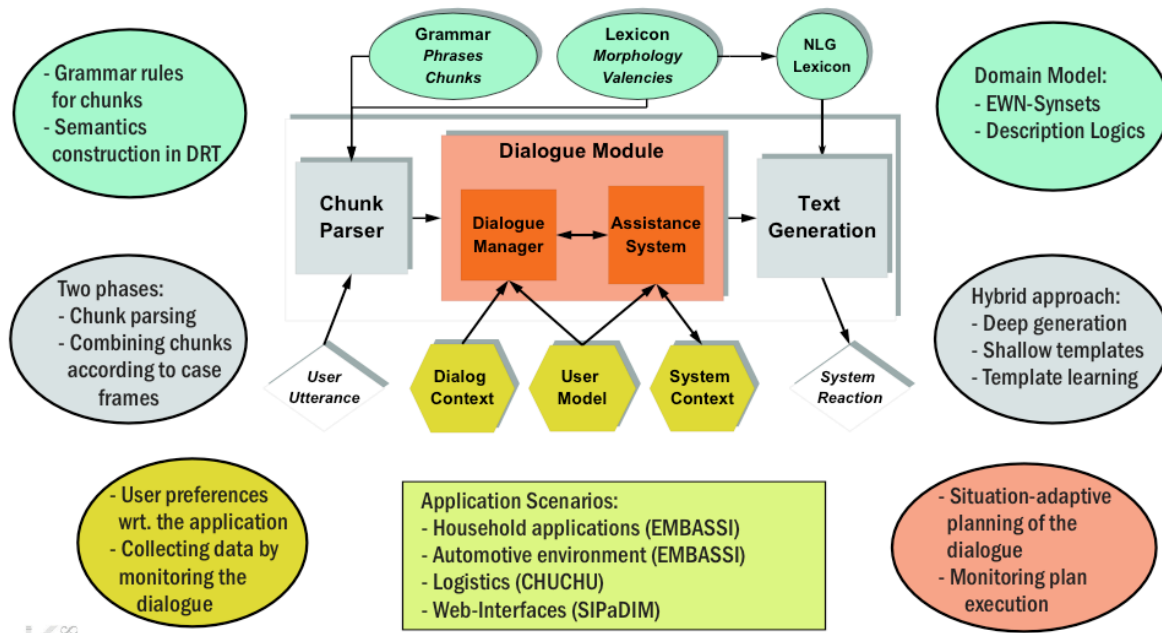


Abbildung 1: The CONALD system architecture

4 Incremental Semantic Composition

If we want human-computer dialogues to be natural, we must allow humans to talk to the computer as they do to humans. Spontaneous speech often is incomplete or incorrect, full of interruptions and self-corrections, leading to ungrammatical input to the parser. Additionally, given the error rates of speech recognizers, even with correct input the speech recognizer may produce an output which is not grammatical. Apart from that, parsing German input is difficult, as German is a language with fairly free word order, also allowing for discontinuous constituents. Therefore, the grammar cannot rely only on linear input sequences as its main concept. We try to overcome these problems by the design of a two-phase parsing process (as presented in [BÜCHER ET AL.2002]).

First, the speech recognizer's output is segmented into chunks [ABNEY1991]. These have to be translated into constraints for a (partial) description of a system state. For that purpose, an approach motivated by dependency theory is applied: Valencies for the syntactic head of each chunk are analyzed if they can serve as the dependent for some other chunk (its regent). Dependent and regent have to meet three classes of criteria in parallel: syntactic constraints, semantic constraints (is the semantic part of the valency satisfied?), pragmatic constraints (can a constraint in the application domain be derived from the triple regent, thematic role, and dependent?). For the example utterance "I want to watch a thriller tonight!" the parser computes the following analysis:

<i>Chunk</i>	<i>Semantics (informal)</i>
I want to watch	assistance task
a thriller	object for task
tonight A	time span for task

4.1 Applying Case Frames to Chunks

The three chunks shown above are connected by semantic relations which have to be identified during the second phase of the parsing process. It relies on a kind of dependency grammar which for each chunk of phase 1 gives a list of possible syntactic functions the chunk may have:

$$C_1 \text{ has } C_2 \rightarrow \langle \text{synfunc} \rangle$$

(constraint equation)

e.g.:

$$VP \text{ has } PP \rightarrow \text{adverbial}$$

$$NP \text{ has } PP \rightarrow \text{attribut}$$

$$VP \text{ has } NP \rightarrow \text{subject}$$

$$NP \text{ agr case} = \text{nom,}$$

$$NP \text{ agr num} = VP \text{ agr num.}$$

The options are constrained by the morphological features of the chunk, e.g. an *NP*-chunk functions as subject only if it has nominative case.

For each chunk there is a case² frame for its semantic head that stores information about the valencies³. The valencies of each chunk are filled by combining it with other chunks, e.g. building a *VP* from a verb and an *NP* that functions as its direct object, or expanding a *VP* by an adverb. The suitability of the combination of two chunks is determined by the semantic constraints of the application ontology. Take the case frame for *sehen*:

infinitive: kommen

syntactic function	thematic role	EWN concept
subject	involved-agent:	Person1
adverbial	involved-timespan:	TimeInterval1

From the case frame we derive hypotheses about possible fillers of a complement position of a chunk using the syntactic functions. Whether a hypothesis is satisfiable is determined by the concepts of the chunks. If they fit, the DRS can be computed.

In our example, the *VP want to see* can be combined with the *NP a thriller* and the adverbial *tonight* since in the case frame of *sehen* there are valencies allowing semantic relations to be established.

5 Building a case frame database

We use our approach to semantics construction in different applications. As a consequence, we gathered a huge amount of semantic definitions (i.e. taxonomic chains) and case frames (i.e. thematic roles) defined by these applications. Some of these data are specific to a given application, whereas others are used by several applications. This made the need for a tool

²The term case is used in the way of Fillmore [FILLMORE1969] meaning thematic roles

³The term *valency* here is used in a broader sense: it includes not only obligatory elements needed to make a phrase syntactically complete; more than that, the case frames list all semantically and pragmatically suitable modifications and their syntactic representations, e.g. attributes for nouns or adverbials for verbs.

that enables efficient storage and easy and fast access, as well as preparing the data required by the parser to be of prime importance.

For this purpose, we have developed a lexicon tool that helps editing semantic data, checks their coherence according to the algorithm presented in sect. 4, and visualizes them as well (see fig. 2).

The tool depends on the following resources as a basis for its data:

- the EuroWordNet (EWN) ontology,
- the SUMO ontology (a generic reference ontology), and
- semantic lexica.

In this respect, it is worth to highlight the differences between our frame data base and FrameNet [BAKER ET AL.1998]. FrameNet is an online lexical resource⁴ for English based on the principles of frame semantics and supported by corpus evidence. It can serve as a dictionary, for it includes definitions and grammatical functions of the entries. And hence entries are linked to the semantic frames in which they participate, FrameNet can serve as a thesaurus as well.

However, the information provided by FrameNet is not sufficiently formalized to be directly applicable within our system; in other words, it is not possible to use FrameNet to parse utterances directed to the system or to construct semantic representations for them. So, from a practical point of view, what we need is a formal specification for the information represented in FrameNet and which, on the one hand, can directly be encoded in Description Logic (which is the logical framework we use), and on the other hand, can be used with an efficient inference mechanism.

Another difference is that the current FrameNet is basically constructed for the English language and hence can be used only in systems based on English. Since our application is multilingual, our representation scheme is based on the ILI-representation of EWN, which makes our tool language independent.

6 Conclusions

To conclude, there are some points in common with other contributions in this volume which are important to our work:

- first of all, the *data-orientation* in general, which by the way is quite common for everybody in the speech recognition community, and in connection with that,
- the importance of *storage* of patterns in the lexicon, (which is, among other properties, a prerequisite for what in Artificial Intelligence is called “case-based reasoning”), and agrees quite perfectly with the lexicalist character of our grammar,
- an emphasis on *pragmatics* and *context*, i.e., in our case, the conceptual domain model, the discourse context maintained in the dialogue manager, and the situation context represented in the application system providing the key to disambiguation.

⁴<http://www.icsi.berkeley.edu/framenet/>



Abbildung 2: A screenshot of the valency editor

Furthermore, the success of any application system depends crucially on the availability of appropriate and comprehensive linguistic resources. Although this sounds trivial, the actual situation we face — at least for German — is not that easy.

It may improve within one or two years the availability of a new generation of resources like GlobalWordNet and German FrameNet. In the long run, we would like to have access to a powerful valency lexicon database — in particular considering that multilinguality is of increasing importance —, for the *technical* aspects of which we can give some advice from the viewpoint of computer science:

- As far as the formal representation of the lexicon is concerned, the *expressivity* of the language is of extreme importance. By that, we don't want to emphasize encoding in some XML language — which is state of the art — but rather its expressivity in logical terms, i.e. whether it can express conjunction, disjunction, negation, quantification, subsumption and inheritance, and whether it provides means to express non-monotonic notions like defaults, a notion which lies beyond standard first-order logic. And we must be aware that in a real, i.e. empirically based lexicon we will be confronted with inconsistency, and we will have to deal with it.
- A minor, but nevertheless important issue is the interface, i.e., what is the expressivity of the query language? And, furthermore: Is access strictly sequential or is there a possibility of parallel access?

Acknowledgement. We would like to thank the current and former members of our research group for their contributions: Kerstin Bücher, Martin Klarner, Yuliya Lierler, Peter Reiss, Bernhard Schiemann, and Iman Thabet.

Literatur

- [ABNEY1991] Steven Abney. 1991. Parsing by chunks. In R. Berwick, S. Abney, and C. Tenny, editors, *Principle-based Parsing*. Kluwer, Dordrecht.
- [BAKER ET AL.1998] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley framenet project. In *Proceedings of the COLING-ACL 1998*, pages 86–90, Montreal.
- [BRIETZMANN AND GÖRZ1982] Astrid Brietzmann and Günther Görz. 1982. Pragmatics in speech understanding – revisited. In J. Horecky, editor, *Proceedings of the Ninth International Conference on Computational Linguistics*, pages 49–54, Amsterdam. North-Holland.
- [BÜCHER ET AL.2002] Kerstin Bücher, Michael Knorr, and Bernd Ludwig. 2002. Anything to clarify? report your parsing ambiguities! In *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 465–469, Lyon.
- [FILLMORE1969] Charles J. Fillmore. 1969. *Universals in Linguistic Theory*. Holt, Rinehart, and Winston, New York.