The Verbmobil Semantic Database*

Karsten L. Worm
Univ. des Saarlandes
Computerlinguistik
Postfach 15 11 50
D–66041 Saarbrücken
Germany
worm@coli.uni-sb.de

Johannes Heinecke
Humboldt–Univ. zu Berlin
Computerlinguistik
Jägerstraße 10/11
D–10099 Berlin
Germany
heinecke@compling.hu-berlin.de

Abstract

This paper describes the development and use of a lexical semantic database for the Verbmobil speech–to–speech machine translation project. The motivation is to provide a common information source for the distributed development of the semantics, transfer and semantic evaluation modules and to store lexical semantic information application–independently.

Dieser Beitrag beschreibt die Entwicklung und Anwendung einer lexikalisch–semantischen Datenbank für das Projekt Verbmobil zur maschinellen Übersetzung gesprochener Sprache. Die Zielsetzung ist, eine gemeinsame Informationsquelle für die verteilte Entwicklung der Module Semantik, Transfer und Semantische Auswertung bereitzustellen und lexikalisch–semantische Information anwendungsunabhängig zu verwalten.

1 Introduction

The distributed development of the modules of a large natural language processing system at different sites makes interface definitions a vital issue. It becomes even more urgent when several modules with the same intended functionality are developed in parallel and should be compatible with respect to their input–output–behaviour.

*The research reported in this paper was supported by the German Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie under contracts 01 IV 101 R and 01 IV 101 G6. We wish to thank our colleagues in the lexicon, syntax/semantics and transfer groups in the project.

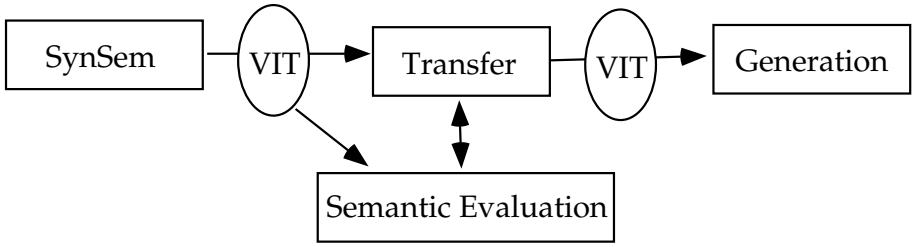

Figure 1: The Verbmobil architecture (simplified)

Another important issue is the acquisition and maintenance of lexical information which should be stored independently of an application in order to make it (re–)usable for different purposes.

This paper describes the design and use of the Verbmobil Semantic Database which we developed in order to deal with these issues in the area of lexical semantics in Verbmobil.

2 The Verbmobil Project

The Verbmobil project [Wah93, BGL+96] aims at the development of a speech–to–speech machine translation system for face–to–face appointment scheduling dialogues. It employs a semantic transfer approach to translation [DE96], i. e., an input utterance is syntactically analyzed, a semantic representation of the content is built up, and this source language semantic representation is mapped to a target language semantic representation by the transfer module. This representation is the input for the target language generation. Additionally, a semantic evaluation module answers disambiguation queries (cf. figure 1).

3 Motivation for the Semantic Database

The architecture of Verbmobil makes it necessary for the semantics, transfer, semantic evaluation and generation modules to agree on the format and contents of the semantic representations they exchange. E. g., the developers of the transfer module need to know how the semantics of the different lemmata in the vocabulary is represented in the structures produced by the syntax–semantics module (synsem for short), i. e., which predicates and structures they have to map to the target language. On the other hand, semantics need to know which readings have to be distinguished by transfer in order to arrive at correct translations.

This need becomes even more urgent when, like in Verbmobil, there are several synsem modules (two for German, one for Japanese), which have to produce

compatible output, and the modules are developed in parallel by partners at different sites.[1]

As a frame for the exchange of semantic representations, a common format, the *Verbmobil Interface Term*, `VIT` for short, has been defined [BES96]. The `VIT` is the central data structure used at the interfaces between the language modules of Verbmobil. A `VIT` is a ten–place term with slots for a list of labeled semantic predicates, sortal and anaphoric information, scope relations, prosodic features, etc.

What is needed then in addition to the `VIT` data structure definition is a definition of the `VIT`'s contents, for each lemma in the vocabulary of the system a definition of the semantic predicates and other types of information, e g., sortal restrictions, it introduces in the `VIT`. E. g., for a verb like *kommen*, we need to specify that it introduces a predicate `kommen(L1,I1)` together with an argument role `arg1(L1,I1,I2)` in the semantics slot and `sort(I1,space_time)` in the sorts slot.

If a source providing this kind of information to the developers of the separate modules is available, the modules delivering (the two synsem modules) or processing (especially the transfer module) `VIT`s conforming to this definition can be developed in parallel. It would also be desirable to use this information source directly in the construction of the linguistic knowledge bases of the synsem modules to guarantee consistency between their output and the specifications.

To meet these goals, we have developed the *Verbmobil Semantic Database*, which we will describe in the remainder of this paper.

4 Design and Implementation of the Database

The database is organized around a set of abstract semantic classes [BES96], which are used to classify the lemmata in the vocabulary. It is implemented using the lexicon formalism LEX [GH95].

4.1 Semantic Classes

The semantic classes in use are originally based on a morpho–syntactic classification of the words in the vocabulary of the system which has been refined to account for semantic properties.

For each semantic class a representation scheme, called the *predscheme*, has been defined, which specifies the predicates together with their arity and arguments appearing in a `VIT` for instances of the class.

As an example consider the class intransitive_verb. A intransitive verb is rep-

[1] In the following, we concentrate on the Semantic Database for German. The database we developed for the Japanese synsem module [Mor96] follows the same principles.

Class	PredScheme	Example
transitive_verb	`R(L,I), argX(L,I,I1), argY(L,I,I2)`	treffen
common_noun	`R(L,I)`	Termin
det_quant	`R(L,I,H)`	jeder
demonstrative	`demonstrative(L,I,L1)`	dieser
wh_question	`whq(L,I,H), tloc(L2,I2,I1), time(L1,I1)`	wann

Table 1: A few examples of semantic classes

resented as `R(L,I), argX(L,I,I1)`.[2] I. e., it introduces some relation `R` and one thematic roles (`I` is the event variable, `L` a label used to refer to the verb's semantic contribution, and `I1` is the instance filling the role). The verb's relation and the thematic roles it assigns have to be defined for each verb in the database. Cf. table 1 for further examples of semantic classes together with their predschemes.

4.2 The Lexicon Formalism *LeXI*

The semantic database makes use of the lexicon formalism *LeXI* developed in the course of the Verbmobil project [GH95].

The **Lex**icon **For**malism *LeXI* has been used since summer 1994 within Verbmobil's lexicon group. It is based on feature-structures (permitting disjunction and negation) embedded in an inheritance hierarchy of classes.

In *LeXI* the task of constructing a lexicon is split up into four parts: Modelling the lexicon (i.e., its linguistic classes), data-acquisition (can be done at the same time by different contributors), definition of the application-interface (data can be compiled into every format needed after being processed by the *LeXI*-machine) and efficient storage.

Modelling a lexicon involves defining classes, their appropriate features and inheritance relations between classes. Examples for defining classes will be given below in section 4.3; appropriateness of features is dealt with in the remainder of this section.

Database entries, called *bases*, are instances of a class. Consequently, they assign values to the features they inherit from their class which are not yet fully specified by the class definition.

4.3 Semantic Classes and their Representation in *LeXI*

The abstract semantic classes of section 4.1 have been modelled in the lexicon formalism *LeXI* along the following lines.

[2] `X` stands for one of the values $\{1, 2, 3\}$, since `arg1, arg2, arg3` are the thematic roles used in Verbmobil.

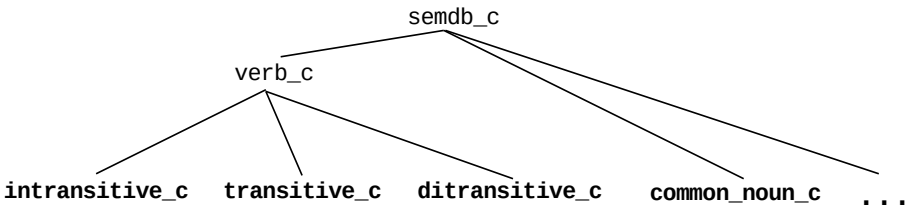

Figure 2: Part of the class hierarchy

Firstly a general superclass `semdb_c` is defined from which all classes inherit features for the lemma, the main predicate's name, the part of speech, etc. The individual subclasses corresponding to the abstract semantic classes additionally introduce a specific predscheme for each predicate associated with words of this class and features for sortal information, thematic roles, etc.

```
class semdb_c :< top >:     %  - Main class from which
                            %    all classes inherit
   predname: top &          %  - Name of the semantic predicate
   lemma: top &             %  - Lemma of the entry
   pos: top .               %  - Part of Speech
```

While the abstract semantic classes are not hierarchically organized, their modelling in $\mathcal{LX}$ makes use of a hierarchy to capture generalizations. E. g., we abstract over the properties all verb classes have in common and place them in an abstract verb class `verb_c` from which all verb classes, e. g., `intransitive_c`, inherit, cf. figure 2 (classes corresponding to semantic classes are shown in boldface) and below.

```
class verb_c :< semdb_c >:  %  - All verbal classes inherit this.
   sort_of_inst: top .      %  - Sort of eventuality.
```

```
class intransitive_c :< verb_c >:  %  - Intransitive verbs
   semclass: intransitive_verb &   %  - Semantic class
   predscheme: 'L,I' &             %  - PredScheme for PredName
   predscheme_a1: 'L,I,I1' &       %  - PredScheme for the argument
   role_a1: (arg1 \ arg2 \ arg3) . %  - Thematic roles of arguments
```

4.4 Representation of Lemmata

A base for a lemma consists of its classification together with its idiosyncratic properties in terms of feature values; it inherits the feature values which are specified in the definition of the class. Among the idiosyncratic information

we have predicate names, sortal restrictions, etc. Thus an entry inherits the predscheme from the class, while the concrete predicate name in the predscheme is defined in the entry itself.

```
base 'kommen' :<< intransitive_c >>: %  - The entry inherits
                                     %    from 'intransitive_c '.
   pos: 'VVFIN;VVINF' &              %  - Further specifications.
   lemma: 'kommen' &
   predname: 'kommen' &
   sort_of_inst: space_time &
   role_a1: 'arg1' .
```

5 Application of the Semantic Database

The Semantic Database is currently being used for creating the semantic lexica of the syntactic–semantic modules of Verbmobil, for producing a table of lemmata with the predicates and other types of information they introduce in a `VIT` and for checking the correctness of the generated interface terms automatically.

To guarantee consistency between the output of the synsem module and the database content, the semantic lexicon of SynSemS3[3] is generated out of the semantic database, e. g., the following entry for *kommen*.

```
sem_lex(Cat, kommen) short_for
     intrans_verb_sem(Cat, kommen, (space_time), [arg1]) .
```

The verbs in the syntactic lexicon contain calls to the macro `sem_lex/2` which are expanded in the semantic lexicon as shown above.[4] The macro `intrans_verb_sem` defines the semantic properties of intransitive verbs [BGL+96].

Additionally, we generate a table of lemmata which is used by the transfer developers and as an information source for the automatic correctness check on `VIT` representations. In the table the example appears as this:

```
kommen VVINF intransitive_verb kommen(L,I),arg1(L,I,I1) I1/space_time
```

[3] SynSemS3 is the syntactic–semantic module developed by **S**iemens AG (syntax), University of the **S**aarland and University of **S**tuttgart (semantics). The other synsem module developed by IBM Germany makes use of the table output of the database to create a semantic lexicon.

[4] The first argument of `sem_lex/2` ranges over entry nodes of the feature structures of the lexical entry used by the grammar formalism.

6 Conclusion

The use of the semantic database has proven to be successful in dealing with about 2000 German and 300 Japanese lemmata for version 1.0 of the Research Prototype. It allows the partners responsible for the syntactic/semantic, transfer and semantic evaluation modules to develop their modules in parallel, relying on the interface specification and the content of the database.

References

[BES96] Johan Bos, Markus Egg, and Michael Schiehlen. Abstract Semantic Classes and Concrete VIT Representations. Verbmobil–Memo 101, Universität des Saarlandes, Computerlinguistik, Saarbrücken, 1996.

[BGL+96] Johan Bos, Björn Gambäck, Christian Lieske, Yoshiki Mori, Manfred Pinkal, and Karsten Worm. Compositional semantics in Verbmobil. In *Proc. of the 15th COLING*, Copenhagen, Denmark, 1996.

[DE96] Michael Dorna and Martin C. Emele. Semantic–based transfer. In *Proc. of the 15th COLING*, Copenhagen, Denmark, 1996.

[GH95] Gunter Gebhardi and Johannes Heinecke. Lexikonformalismus LeX4. Verbmobil Technisches Dokument 19, Humboldt–Universität zu Berlin, Computerlinguistik, Berlin, 1995.

[Mor96] Yoshiki Mori. Multiple discourse relations on the sentential level in Japanese. In *Proc. of the 15th COLING*, Copenhagen, Denmark, 1996.

[Wah93] Wolfgang Wahlster. Verbmobil: Translation of face-to-face dialogues. In *Proceedings of the 3rd European Conference on Speech Communication and Technology*, pages 29–38, Berlin, Germany, 1993.