The Text Understanding System LEU/2

Jan Wilms

The concepts of a text understanding system as described in the previous chapters have been implemented in LEU/2, the second 'LILOG Experimentier Umgebung' (LILOG Experimental Environment). This Experimental Environment is used for development work as well as for experiments and to demonstrate the capibilities of the system. The version of LEU/2 described here is 2.4, the version that was shown to the public at Hanover CeBIT trade fair 1991. Therefore not all features of the final version as of end 1991 are discussed here in detail.

For development the engineering environments for linguists (see the paper by J.Dörre in Sect. 1.1 of this book) and for knowledge engineers (see the paper by M.Börkel in Sect. 2.2) have been implemented. For a description of LILOG-KE, the LILOG Knowledge Engineering Environment, also see [Seiffert, Wilms]. These development environments may be used stand alone or together with the complete system. If run with the complete system, they may also be used to inspect intermediate results.

For experiments the behavior of LEU/2 can be modified in a number of ways. The mode of processing can be determined by changing user-defined options like the parsing strategy, the search strategy and calculi used by the inference engine, search depth for forward or backward chaining, and so on. A configurator allows one to specify which modules to use. This is true for program modules as well as for parts of knowledge bases. In this way, it is possible, for example, to experiment with different variants of the parser, the morphology module, or the database management module, or one can use an alternate lexicon or another set of background knowledge rules. By the time when this was written, we only were able to perform part of the experiments that we plan to do. So there is more work under way.

The rest of this paper will describe how we demonstrate that LEU/2 is actually able to comprehend the meaning of texts. Due to the large number of features in LEU/2, it is not possible to cover all of them. So we will focus on some of them to give an idea of how LEU/2 understands texts. As LEU/2 deals with German texts, all examples here are shown in German, but we do provide English translations. The command language of the system, e.g. for all messages issued by the processes, is English. The current version of the system reads texts from travel guides about the city of Düsseldorf (see also the paper by K.v.Luck and T.Pirlein in this chapter) like the following:

> Im Palais Nesselrode ist das Hetjensmuseum, das 1909 eröffnet wurde, untergebracht. Es befindet sich an der Ecke Schulstraße und Hafenstraße. Die Keramiksammlung umfaßt zehntausend Objekte. Der Eintritt der Ausstellung, die von 10 Uhr bis 17 Uhr geöffnet ist, beträgt 2 DM. [1]

When this text is read in, it is processed and a representation of its content is stored in the system's text knowledge base. So let us for the rest of this chapter assume that this text, which

[1] Nesselrode Palace houses the Hetjens Museum, which was opened in 1909. It is located at the corner of Schulstrasse and Hafenstrasse. The ceramics collection of the museum contains ten thousand items. Admission to the exhibit, which is open from 10 a.m. until 5 p.m., costs 2 DM.

is only one of many sample texts, has been read by the system. When using LEU/2, we do not of course have to use this specific formulation of the text. Much more than this, we can express the same thing in many different ways. If, for example, we read in an alternate version of the text like:

> Die Keramiksammlung des Hetjensmuseums, das sich an der Ecke Schulstraße und Hafenstraße befindet, umfaßt zehntausend Objekte. Es ist im Palais Nesselrode untergebracht. Die Ausstellung hat von 10 Uhr bis 17 Uhr geöffnet. Der Eintrittspreis des Museums beträgt 2 DM.[2]

the system will react by saying:

Ja, das war mir bereits bekannt. (Yes, I knew this already.)

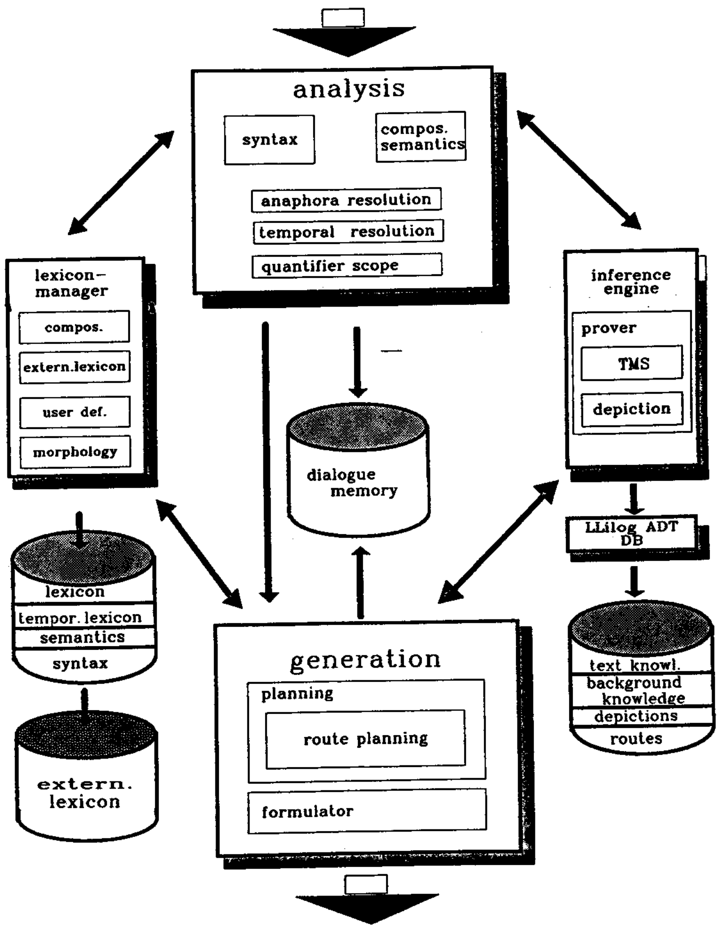

Figure 1: Architecture of LEU/2

[2]The ceramics collection of the Hetjens Museum, which is located on the corner of Schulstrasse and Hafenstrasse, contains ten thousand items. It is housed in Nesselrode Palace. The exhibit is open from 10 a.m. until 5 p.m. Admission to the museum is 2 DM.

So this is also a first indication that the system actually did understand both texts, which enabled it to conclude that the alternate text did not contain any new information. In the next step, one would let the system answer questions that humans would normally be able to answer, given that they have read the text. If, for example, we ask the system:

Wann hat das Hetjensmuseum geöffnet? (When is the Hetjens Museum opened?)

it will answer: Von 10 Uhr bis 17 Uhr. (From 10 a.m. to 5 p.m.)

The answer is evident to anyone who has read the text, but explicitly the text only gives the opening times of the exhibition. So the system must figure out that the exhibition is the exhibition of the ceramics collection, that the ceramics collection is part of the museum and that, whenever the exhibition is opened, presumably the museum will be open too - a fairly complex task of resolving anaphorically definite noun phrases, which relies on fairly sophisticated knowledge representation and reasoning.

In a live demonstration of the system, one can watch the system going through the four major steps which are performed to handle a question: Syntactic Analysis, Semantic Analysis, Knowledge Processing, and Language Generation. To give the reader a rough idea, we give a diagram of the system's architecture here (see Figure 1). Basically these four steps correspond to the first four chapters of this book. So no further explanation should be necessary here. In this way many questions can be answered by the system. Some examples of questions with their answers are shown in Figure 2.

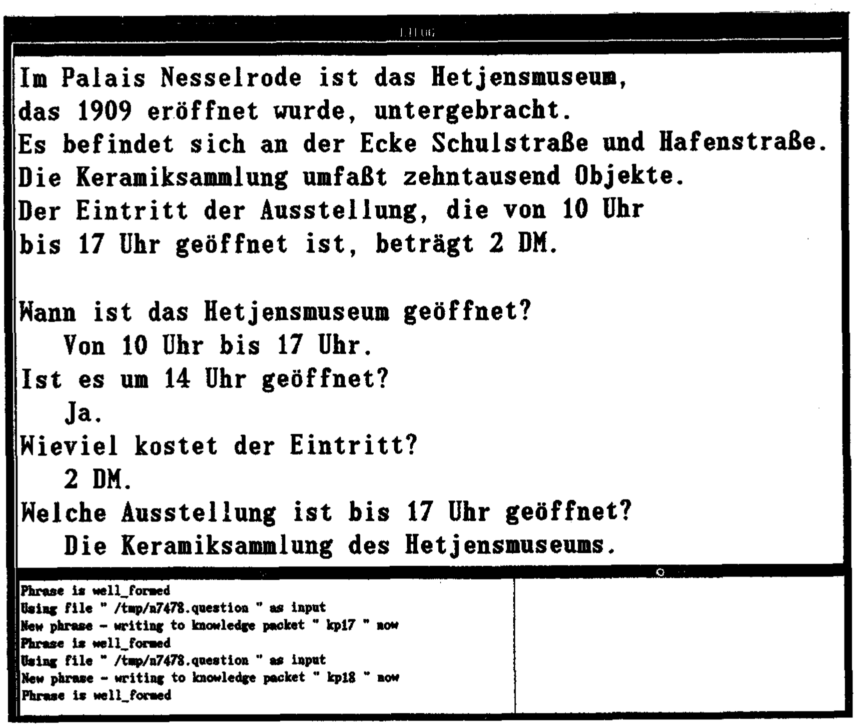

Figure 2: Sample screen of a LEU/2 session

Some of the questions can only be understood appropriately if the discourse is considered. The second question [3], for example, only makes sense following a previous mention of the Hetjens Museum (like the first one), so the 'es' ('it') can be related to the Hetjens Museum. And the third[4] requires a discourse referent, because only then can 'the admission' be interpreted in the intended fashion as 'admission to the Hetjens Museum'[5].

In the next step we demonstrate, how the system can generate text. As an application we choose the description of ways from one point to another, which in our city domain can be used to tell the ways to sights. However in real applications it might be used for explanation components (of expert systems etc.). Finding a path is done based on knowledge about part of the city map of Düsseldorf, but this is not the point of interest here. If we put the question 'Wie komme ich von der Müller Schlösser Gasse zum Eiskellerberg?'[6], ('Müller Schlösser Gasse' and 'Eiskellerberg' are the names of two streets in Düsseldorf) the system would reply by giving the following description:

Gehen Sie zum Stiftsplatz! Gehen Sie über den Stiftsplatz schräg nach rechts! Biegen Sie dann links in die Lambertusstraße ein! Biegen Sie an der nächsten Ecke rechts in die Altestadt ein! Gehen Sie die Altestadt und die Ratinger Straße entlang! Biegen Sie an der dritten Ecke scharf links in die Mühlengasse ein! Gehen Sie durch die Mühlengasse bis zur nächsten Ecke! Da ist dann der Eiskellerberg. [7]

One may observe, that the formulation the system chooses is quite adequate. The phrases 'schräg nach rechts' or 'scharf links' are used rather than simply 'rechts' or 'links', if this helps to distinguish the way. Moreover 'durch die Mühlengasse' may only be used because it is a 'Gasse' (lane).

At this point we can also show how knowledge from new text can be used in answering questions. For example we could ask 'Wie komme ich vom Marktplatz zur Opernstube?'[8] where 'Opernstube' is the name of a restaurant whose location is not known to the system. The answer would therefore be 'Der Anfangspunkt ist unbekannt.'[9]. However if we provide new information by reading in the sentence 'Die Opernstube befindet sich an der Rheinstraße.'[10] and ask the question, then the system is able to find out in a first step where the Opernstube is located, and in a second step construct and describe a path from Marktplatz to that location:

Gehen Sie in das Schloßufer! Gehen Sie das Schloßufer entlang, bis zum Burgplatz! Gehen Sie über den Burgplatz links! Biegen Sie dann schräg nach rechts in die Marktstraße ein! Gehen Sie immer die Marktstraße entlang! An der Rheinstraße ist dann die Opernstube. [11]

---

[3]Is it open at 2 p.m.?

[4]How much is the admission?

[5]Which exhibit is open until 5 p.m.?

[6]How do I get from the Müller Schlösser Gasse to the Eiskellerberg?

[7]Go to Stiftsplatz. Go across the Stiftsplatz diagonally to the right. Turn left into Lambertusstrasse. Turn right at the next corner into Altestadt. Go along Altestadt and Ratinger Strasse. Take a sharp left at the third corner into Mühlengasse (Mühlen Lane). Go through Mühlengasse to the next corner. There is the Eiskellerberg.

[8]How to I get from the Marktplatz to the Opernstube?

[9]The starting point is unknown.

[10]The Opernstube at the Rheinstrasse.

[11]Go into the Schlossufer street. Go along the Schlossufer street as far as the Burgplatz. Go across the Burgplatz to the left. Bear right into Marktstrasse. Keep going along Marktstrasse. At the Rheinstrasse is the Opernstube.

LEU/2 is also able to deal with words, which cannot be found in the system's lexicon (see the paper by W.Emde in Sect. 1.3 of this book). For Example, let a text be read in containing the sentence 'Das Software-Museum wurde 1986 eröffnet.'[12]. The compound 'Software-Museum' of course cannot be found in the system's lexicon. So the lexicon manager trys to analyze it with one of four steps, the third of which is successful here. The trace of these steps for 'Software-Museum' is shown below.

Unknown word: Software-Museum

Searching for entry in temporary lexicon ..
No entry in temporary lexicon!

Searching for a synonym which is known ..
Can not build lexical entry using a synonym

Trying to decompose word ...
Decomposed as Software - Museum
New concept: #Software-Museum inserted as sub-concept of: Museum !
Decomposition successful

continuing calling process ...

At the end of this process lexical information derived from the second part of the composite noun ('Museum') together with information on the meaning (concept) of the word (here: #Software-Museum) is inserted in the temporary lexicon and provided to the calling process, i.e. the parser. On the next occurence of the word, it can (in the first step) be looked up successfully in the temporary lexicon, thus avoiding repeated analysis.

```
'Software-Museum' :=
   'MOR_NOUN',
   'NOUN_TEMP'(norm,'ART'),
   'GENDER'(neut),
   'NOUN_SX'(@('#Software-Museum',sorts))
```

For this composite noun the concept was created dynamically as a subconcept of the sort 'Museum', i.e. the new sort was inserted into the ontology below 'Museum'. Figure 3 on the next page shows an excerpt of the LILOG ontology (as displayed by the browser of the KE tools) showing the dynamically acquired concept '#Software-Museum' and its inherited roles and features.

[12]The software museum was opened in 1986.

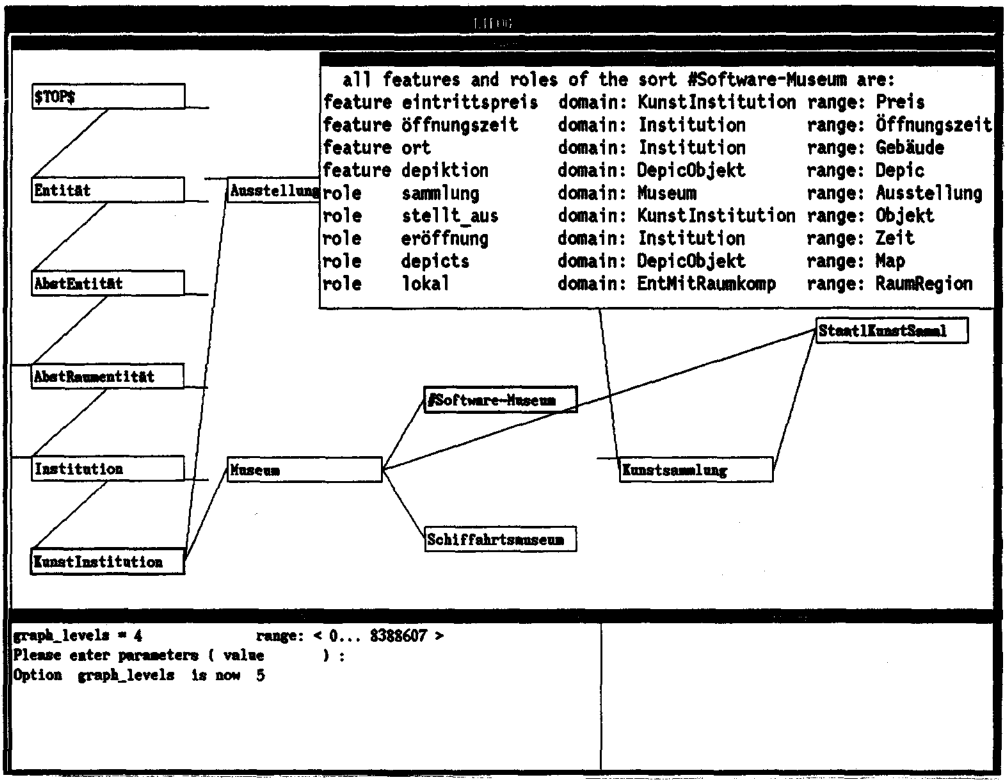

Figure 3: Excerpt of the LILOG ontology showing a dynamically acquired concept

In a LEU/2 working session some intermediate results can be displayed. Just for illustration we will give some types of result here. Others, like feature structures, the ontology, or displays of depictions (imagination and inspection) are not fit to be printed in a book but should rather be inspected dynamically on the screen. So we renounce showing those here and encourage the reader to try to view the LEU/2 video [Video] or a live demonstration of the LEU/2 system. Intermediate results given on the following two pages below are for the first sentence of the 'Hetjens' text quoted above: 'Im Palais Nesselrode ist das Hetjensmuseum, das 1909 eröffnet wurde, untergebracht.'

Intermediate results of the syntactic analysis, which can be inspected by the user cover amongst other things the chart, the parse tree, and the feature structure (see the previously mentioned paper by J.Dörre in Sect. 1.1).

The intermediate result of the semantic module is the EDRS of the sentence, as shown on the next page.

```
% EDRS :

REFLIST:  drfnt52  drfnt53  drfnt54  drfnt55  drfnt56  drfnt57  drfnt58

REFO: drfnt52   SORTS : Unterbringen
   Attributes:   lokal: drfnt53
                 patiens: drfnt55
   Lfeatures:    temp: present
                 det: state
REFO: drfnt53   SORTS : RaumRegion
   Attributes:   inlokspec: drfnt54
   Lfeatures:    det: anon
REFO: drfnt54   SORTS : Palais
   Attributes:   name: Palais_Nesselrode
   Lfeatures:    det: def
REFO: drfnt55   SORTS : Museum
   Attributes:   name: Hetjensmuseum
   Lfeatures:    number: sg
                 gender: neut
                 det: def
REFO: drfnt56   SORTS : Eroffnung
   Attributes:   patiens: drfnt55
                 temporal: drfnt57
   Lfeatures:    temp: past
                 det: event
REFO: drfnt57   SORTS : Zeit
   Attributes:   intempspec: drfnt58
   Lfeatures:    det: anon
REFO: drfnt58   SORTS : Jahr
   Attributes:   einheit: jahr
                 zahlenwert: n1909
                 jhr: n1909
   Lfeatures:    det: def

DEPENDENCIES:
   drfnt54 < drfnt53            drfnt53 < drfnt52
   drfnt56 < drfnt55            drfnt55 < drfnt52
   drfnt58 < drfnt57            drfnt57 < drfnt56
```

The drfntnn used in the EDRS are dynamically created IDs of discourse referents, some of which are pretty printed below. They are inserted in the knowledge base as Reference Objects, e.g.:

```
Showing attribute values of refo drfnt58 of sort Jahr
  in kbs: know_base(demotextkb_1,[kp12])
     feature jhr: 1909
     feature zahlenwert: 1909
     feature einheit: jahr
```

The pretty-print of the L_{LILOG} fact constructed from the EDRS (i.e., the sentence) as inserted in the knowledge base is:

```
axiom ax_inf_74 in demotextkb_1-kp12
        jhr(drfnt58)=1909
    and zahlenwert(drfnt58)=1909
    and einheit(drfnt58)=jahr
    and intempspec(drfnt57,drfnt58)
    and temporal(drfnt56)=drfnt57
    and patiens(drfnt56)=drfnt55
    and name(drfnt55)=Hetjensmuseum
    and name(drfnt54)=Palais_Nesselrode
    and inlokspec(drfnt53,drfnt54)
    and patiens(drfnt52)=drfnt55
    and lokal(drfnt52,drfnt53) .
```

Additionally some facts inferred by forward chaining during the step of knowledge processing are inserted into the knowledge base. One example for the sentence above is:

```
axiom ax_inf_86 in demotextkb_1-kp12
forall      K:Kontextvektor
            T:EntMitZeitkomp
    Lok(def:innenregion(ro:hohlraum(ro:drfnt54)),
                gebiet:refo_inf_10,
                kontext:K,
                le:drfnt55,
                ro:drfnt54,
                temp:T) .

Showing attribute values of refo refo_inf_10 of sort Gebiet
  in kbs: know_base(demotextkb_1,[kp12])
     feature definition: innenregion(ro:hohlraum(ro:drfnt54))
     feature refobjekt: drfnt54
     role le_objekt: drfnt55
```

LEU/2 was completely implemented in Quintus Prolog (2.4 and 3.1) on IBM workstations running AIX (PS/2-70, PS/2-80, RT6150, RS6000). The database interface is implemented in C, the user interface is based on XWindows and OSF/Motif (AIXWindows). Demonstrations of LEU/2 can be given on any of the machines mentioned above, preferably on RS6000s of course.

References

[Seiffert, Wilms] R. Seiffert, J. Wilms. LIKEET, the LILOG Knowledge Engineering Environment and Tools. In J. H. Griesser, ed., *IBM ITL Conference on Expert Systems*, IBM, 219-223, 1990.

[Video] Video-Tape. LILOG - Linguistics and Logic in Text Understanding. IKBS, IBM Germany GmbH, 15 min., 1990.