Semantic Decomposition of Sentences in the System Supporting Flight Services

K. Pepelnjak, F. Mihelič, N. Pavešić

Artificial Perception Laboratory, Faculty of Electrical and Computer Engineering, University of Ljubljana, Slovenia

The paper describes the semantic part of the linguistic analysis of a recognized word hypothesis lattice, as implemented in an information dialogue system, capable of recognizing and understanding Slovenian speech. It describes the new Slovenian speech database from a semantic point of view. Semantic analysis is performed in three different steps: first, parsing of complex temporal expressions; second, parsing of simple noun phrases: this method is used for the departure/arrival location determination and in the third step we are looking for keywords defining the intended question. We propose some templates dealing with timetables, departure and arrival times, airline companies, etc. The different templates compete on each utterance. Finally, the template with the best score generates a database query. Some experimental results of the semantic decomposition are presented.

Keywords: semantic analysis, dialogue system, speech database, semantic processing, keyword recognition.

1. Introduction

We describe the semantic analysis in a speaker independent speech understanding and dialogue system. Input to the system is continuous Slovenian speech. The system task is to pass airline timetable information to a client. A vocabulary of about 900 words is used. The different phases of the recognition and understanding task are performed by separate modules. The modules operate sequentially, building a hierarchical bottom-up system. The low-level acoustic processing, based on an extensive HMM network, produces a lattice of scored word hypotheses [1],[2]. A DCG based semantic model extracts task-relevant semantic information from tagged sentence hypotheses. Semantic keywords are passed over to a dialogue module. In Figure 1 the structure of a Dialogue System is presented.

The Slovenian language belongs to the group of

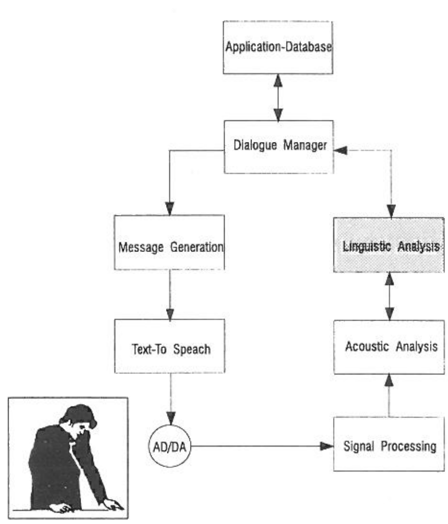

Fig. 1. The Dialogue System

Slavic languages. The problem of recognition and understanding the Slovenian language increases due to a large number of inflected word forms which result in a rather free word order. Furthermore, the Slovenian language involves a rather elaborate number system, including dual besides singular and plural. These language specific problems, which occur during the understanding process, have to be solved by the linguistic analyzer.

2. The Slovenian Speech Database

The Slovenian airline company Adria Airways was kind enough to allow us to record random customer inquiries and interactions between anonymous customers and the airline front-desk personnel, totaling 15 hours of recording. We have transcribed the recordings and extracted typical dialogues [3]. The dialogues were further analyzed with the following results:

- based on the word recordings we estimate that the age span of a typical customer is between 20 and 60 years;
- gender is uniformly distributed between males and females;
- we have encountered syntactic irregularities in nearly all dialogues;
- some speakers are speaking with a strong dialect although it could be noticed that they tried to speak dialect-free Slovenian;
- customers changed their mind in the middle of a spoken sentence and restarted the question, sometimes even changing the subject completly (On Wednesday I would like to fly to ... at three o'clock from London);
- many questions were unclear, forcing the airline personnel to ask for further explanation;

We were able to classify recorded dialogues into four major groups:

1. Airline timetable inquiries ("Are you flying to Vienna tomorrow.").
2. Ticket reservations and cancellations ("I would like to cancel the reservation on tomorrow's flight to Moscow.").
3. Pricing and discount information ("How much does a return ticket to Zuerich cost? Do I have any discount if I stay over the weekend?").
4. Miscellaneous information (group flights, chartered flights ...).

Detailed domain analysis has shown relative high complexity of target system, which would be very hard to implement with limited resources available. The most complex part of the whole system proved to be the semantic analyzer and its interaction to the airline's database. We have therefore decided to implement only a limited subset of the major groups identified, namely airline timetable inquiries.

We have selected nearly 300 typical sentences out of 285 dialogues including:

- around 50 destinations;
- 10 airline companies;
- temporal expressions, including:
    - days in a week;
    - all twelve months;
    - numbers from 0 to 60 used in different date and time expressions;
    - relative clauses (tomorrow, the day after tomorrow, after two weeks, last month, next month ...).
- different startup and finishing sentences (good morning, hello, thank you, good bye ...)
- short positive and negative phrases (Yes, No, What ?, No way, Precisely that ...)

2.1. Semantic decomposition of recorded database sentences

We were able to divide recorded sentences into three main groups:

- context-independent sentences;
- context-dependent sentences;
- non-understandable sentences.

The following tables present examples of different recorded sentences from the first two groups.

CONTEXT-INDEPENDENT SENTENCES		
Sentence type	*Semantic category*	*Example*
INITIAL GREETING	greeting	Good morning. Good evening. Hello, my name is Julia.
INTRODUCTION	introduction	I have a question. I would like to ask you something. Is this Adria Airways?
DIALOG CORE	relation inquiry	Are you flying from Ljubljana to Madrid?
	departure_time	When is the plane for Frankfurt leaving?
	arrival_time	When does the plane from London land?
	timetable	Which days of the week are you flying to Paris?
	flight_duration	How long does the flight to Moscow take?
	airline_company	Which airline is flying between Ljubljana and Rome?
FINAL SENTENCES	conclusion	Thank you. See you later. Good bye.

Tab. 1. Context-independent sentences

CONTEXT-DEPENDENT SENTENCES		
Sentence type	*Semantic category*	*Example*
DIALOG INQUIRIES	repetitions	Could you, please, repeat that information?
	additional_information	What about next week?
	confirmation	Yes, I would like to fly tomorrow.

Tab. 2. Context-dependent sentences

3. Linguistic analysis

The main goal of our system is understanding untrained spontaneous speech (from naive users). As expected, we encountered problems that are well known during design and implementation of such a system [4]:

- unknown or mispronounced words;
- filled pauses (eee, ah, uh,. . .);
- restarts — repeating a word or phrase;
- different dialects;
- interjections — extraneous phrases (I would like to . . . , maybe you can . . .);
- ellipses (On Wednesday I would like to fly to . . . at three o'clock from London);
- non-grammatical constructions.

The problems stated above are mainly due to the nature of spontaneous speech, which is usually grammatically and syntactically incorrect. Word recognition module represents another major source of errors that are introduced during speech recognition. These errors usually manifest themselves in the following manner:

- the noun case is invalid;
- the preposition is missing (I would like to fly_Ljubljana.);
- phonetically similar words with completely different syntactical and semantical meanings (pronounced želimo → recognized genevo; semantic interpretation of "genevo" → arrival_city: Geneva; semantic interpretation of "želimo" → "we would like to").

The linguistic analysis part of the system can be either syntax-driven or semantics-driven. In our case we demand from the system to be as robust as possible. Spoken input is often ungrammatical and speakers use colloquial expressions. While syntax-driven systems have the advantage of domain independence and provide useful information for further analysis, they are unable to handle ungrammatical sentences since they must produce a complete parse of the sentence. We decided to develop a semantics-driven system (Figure 2) as it exhibits the advantage of being able to handle ungrammatical and extragrammatical sentences.

However, semantics-driven systems are known to break down on more complex sentences and they are not easily transferable to new domains. Since the application domain of flight information retrieval is a rather restricted one, problems caused by complex sentences usually do not arise.

4. Semantic processing

The input into the semantics-driven analyzer (also called Template Matcher)[5] is a set of the most probable sentence hypotheses generated by the word recognition module [7],[8].

The semantic analyzer performs three different steps:

• parsing of temporal expressions;

• parsing of simple noun words. Their meaning is often deduced from a simple semantic category which they belong to, and from their position in the sentence. This method is used for departure/arrival city determination;

• the rest of the sentence is parsed using a very simple parser that tries to locate as many keywords as possible. We use these keywords in further processing to determine the sentence type (typical keywords would be: timetable, direct flight _)

The semantic analyzer is implemented in Prolog, with most of the parsing code being written in tagged DCG grammar[6].

4.1. Temporal expression parsing

The Slovenian language has many ways of expressing temporal expressions. Our parser recognizes nearly all of the commonly used expressions and some rarer ones. The following phrases are recognized as valid temporal expressions:

• date and/or time (July fifth at thirteen twenty-five);

•interval between two dates (from tenth to seventeenth of July);

• part of a time unit (early next week, second half of next month); relative expressions (next week, last Tuesday ...);

• part of the day (morning, noon, afternoon, evening, early morning ...);

• days of week (Monday, every Monday, ...);

• other valid phrases (the day before yesterday, the day after tomorrow ...).

The temporal expression parser is the syntactically strictest part of the semantic analyzer due to the above-mentioned richness of temporal expressions. We impose many restrictions on case matching (for example, noun cases and gender shall match with numbers etc.), relaxing the restrictions somewhat if the exact match fails. We have therefore tried to find a fine balance between misunderstood sentences (due to overly relaxed parsing) and rejected sentences (due to too strict parsing).

The temporal expression parser replaces time phrases with a complex expression that uniquely represents semantic meaning of a recognized time phrase. The parser does not try to convert relative time expressions (today, tomorrow ...) into absolute time, that task is left for a higher-level module that will give a semantically recognized sentence its proper meaning.

Sample output from the temporal expression parser is shown in the table below:

Time phrase	Result
noon	[time(rel 0 ++ 12:0)]
Wednesday at 5 PM	[time(wed ++ 17:0)]
early tomorrow morning	[time(rel 1 ++ 3:0 >> 6:0)]
beginning of next week	[time(int(nxt week, 0 >> 0,25) ++ 0:0 >> 24:0)]

4.2. Departure/arrival town determination

Special treatment is given to departure/arrival town determination. The task should be a relatively easy one given the fact that the Slovenian language allows us to uniquely determine whether the user is asking about leaving town or coming to town based on noun case. Nevertheless, we found that the word recognition module fails to recognize proper case in a large percentage of sentences. We were therefore forced to implement additional heuristic procedures that try to extract whether a town is a flight source or destination based on other information present in the sentence and the town's position within the sentence.

The parser tries to locate town names during keyword recognition phase, storing the noun case in the semantic context of the recognized keyword. A post-processing module tries to determine whether a recognized town represents departure or arrival town based on several criteria:

- proper noun case. Departure town should be in second case, arrival town in fourth or fifth case;
- preposition accompanying the noun. Preposition "iz" uniquely determines the town as a departure town even if the noun case is invalid;
- other information presented in the sentence, for example: an unrecognized town in a sentence that contains a departure town is very probably an arrival town;
- vicinity of keywords. Two town names in sequence (like Ljubljana — Vienna) normally represent departure and arrival town;
- position within a sentence. The post-processing module assumes that the departure town precedes the arrival town in most sentences.

4.3. Keyword recognition

The semantic parser tries to recognize as many different keywords as possible. These keywords are defined as belonging to various semantic classes and it is believed that it is possible to determine speaker's intention based on keywords present in the sentence, like:

- words meaning *departure*;
- words meaning *arrival*;
- words meaning *flight connection*;
- words representing airline companies ...

4.4. Template matching

A parsed sentence is post-processed (see departure/arrival town determination) and passed to a template matching module that determines semantic contents of the sentence based to its likeness to predefined templates. We defined many templates that cover most of the sentences encountered in recorded dialogues [5], like: departure information, arrival information, timetable inquiry, airline inquiry etc.

Each template contains a list of semantic categories that are scored. A semantic category in a template can be a keyword (the presence of which would give the template a higher score) or a category that carries additional meaning that should be passed on (like arrival time or departure city) along with increasing template's score. Templates are defined in a special format that allows us to specify the output format of sentence's semantic contents along with scoring information. A good example is the *departure* template:

```
(departure, [carrier^T1/airline(T1)+10,
      departure^T2/city(dep/T2)+10,
      arrival^T3/city(arr/T3)+10,
      departure_time^T4/time(T4)+10,
      keyword(odhod)+30,
      keyword(letalo)+15,
      keyword(leteti)+20,
      keyword(imeti)+20,
      q(kdaj)+15,
      keyword(zveza)+5]).
```

This template anticipates keywords like *departure, airplane, fly, have, connection, when, ...* and semantic categories like *airline, city* and *time*. The semantic meaning of these categories is automatically formatted and passed on to a higher-level module.

The template matching module tries to match every template with the parsed sentence and tries to match as many terms in each template as possible. Each template is scored and the templates are output sorted in decreasing order. The default output of the semantic analyzer is

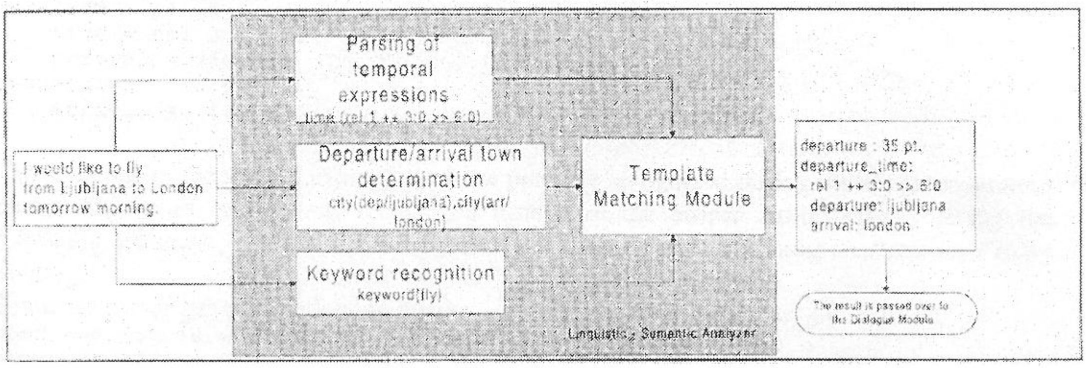

Fig. 2. The Linguistic Analyzer

the formatted output of the highest-score template, although it is possible to output several highest-ranking templates to give the dialogue module different options.

5. Test results

We tested the semantic analyzer described in this paper on over 2250 test sentences that were generated from word recognized spoken sentences[7] from our speech database. The semantic analyzer was able to deduce proper semantic meaning from over 84% of all the sentences. It is important to notice that not all of the properly parsed sentences were syntactically correct or correctly recognized.

We will present several successful and unsuccessful examples of semantic analysis. The first example is a successful semantic decomposition of a simple sentence *Ali letite pojutrišnjem zvečer na Dunaj?* (Are you flying to Vienna tomorrow evening?). The sentence was parsed giving the following result:

```
test ==> [are,you,flying,to,vienna,
tomorrow,evening].
remainder: []
result: [q(are),keyword(fly),time(rel
2 ++ 18 : 0 >> 23 : 0),city(arr /
vienna)]
```

The parsed sentence is passed to the template matching module that scores the sentence likeness to different templates, outputting the templates in decreasing score order. The template that carries the highest score is passed on to the dialogue module. We will show the full output in this example, whereas the following examples will present only the winning template.

```
sentence ->[are,you,flying,to,vienna,
tomorrow,evening].
departure : 35 points (pt.)
   departure_time = rel 2 ++ 18 : 0 >>
   23 : 0
   arrival = vienna
carrier : 29 pt.
   time = rel 2 ++ 18 : 0 >> 23 : 0
   arrival = vienna
arrival : 28 pt.
   arrival_time = rel 2 ++ 18 : 0 >>
   23 : 0
   arrival = vienna
timetable : 18 pt.
   time = rel 2 ++ 18 : 0 >> 23 : 0
   arrival = vienna
duration : 14 pt.
   time = rel 2 ++ 18 : 0 >> 23 : 0
   arrival = vienna
connection : 10 pt.
   arrival = vienna
```

The sentence in the second example was not properly recognized during the word recognition phase. The word 'velja' was recognized instead of the proper word 'želim', giving the following sentence: *V petek zvečer velja biti v Parizu.* (It is worth being in Paris on Friday evening).

Semantic parser gave the following result:

```
test ==> [it,is,worth,being,in,paris,on,
friday,evening].
```

```
remainder: []
result:[time(fri ++ 18:0 >> 23:0), worth,
keyword(arrival),city(arr/paris)].
```

with the winning template:

```
arrival: 50 pt.
   arrival_time = fri ++ 18:0 >> 23:0
   arrival = paris
```

The semantic analyzer successfully extracted the semantic meaning of the sentence. The analyzer is therefore robust enough to give proper meaning to sentences that were not fully recognized. There are limits to its robustness, though: it cannot recognize sentences with missing crucial words or with words that are (wrongly) recognized as an important keyword, as can be seen in the following example:

The word recognition module recognized word želela (I would like to) as Geneva, giving the sentence *Genevo bi leteti iz Skopja v Maribor.* (Genevo fly from Skopje to Maribor.) The semantic analyzer recognized the keyword Genevo as an arrival or departure town name. There were two other town names in the sentence and the post-processing module deduced that the word Genevo represents an arrival town based on additional information present in the sentence. The final semantic information is invalid as it gives us two arrival cities, which is clearly impossible. The winning template selected the first town in the template, giving sentence a completely different meaning.

```
test ==> [genevo,fly,from,skopje,to,
maribor].
remainder : []
result: [city(arr / geneva),keyword(fly),
city(dep / skopje),city(arr / maribor)]
```

The output of the semantic analyzer is not correct:

```
departure : 35 pt.
    arrival = geneva
    departure = skopje
```

We could have devised different solutions to this problem:

• improving word recognition module (which would be the best solution);

• coding every possible exception in the semantic analyzer (hard to do, impossible to maintain);

• rejecting meaningless sentences in dialogue module (giving real-time interaction with the user).

The final system will probably incorporate elements from all three solution sets.

6. Conclusions

The semantics-driven linguistic analyzer is able to process text, whether grammatical or ungrammatical (assuming such a distinction exists), and it is able to do this quickly and efficiently. It simply gives preference to common semantic combinations over unusual ones.

Apart from obvious recognition problems, there are several phrases and syntactically valid sentences that are not yet recognized by our semantic analyzer, for example:

• sentences where the time phrase is divided throughout the sentence. It is perfectly valid in the Slovenian language to have one part of the time phrase at the beginning of the sentence with another part following the noun word at the end of the sentence (example: *Tomorow* I would like to fly from Frankfurt to Ljubljana *at five o'clock in the afternoon.*);

• arrival/departure town determination could fail in those rare examples where there is no difference between first and fourth case of a noun. The solution to this problem lies in experimentally defined rule selection that will give us the highest successful recognition rate;

• we have already encountered subjective time understanding. The sentence *'is there a flight to Frankfurt at three o'clock'* very probably implies 3 PM as the departure time, since there are not many people regularly asking for 3 AM flights to Frankfurt. The question *'is there a flight to Zurich at seven o'clock'* is a harder nut, though, since it is very hard to decide whether the time represents a morning or an evening flight. We still miss experimental data to determine the time range in which an average speaker would normally provide AM/PM information to resolve time ambiguity.

References

[1] F. MIHELIČ, I. IPŠIĆ, S. DOBRIŠEK AND N. PAVEŠIĆ, "Feature representation and classification procedures for Slovenian phoneme recognition", *Pattern Recognition Letters*, no. 13, pp. 879–891, 1992.

[2] I. IPŠIĆ, F. MIHELIČ, E. G. SCHUKAT-TALAMAZZINI AND N. PAVEŠIĆ, "Generating word hypotheses in the Slovene continuous speech recognition system", *Proc. of the SDRV and OAGM Workshop*, Maribor, Slovenia, May 1995.

[3] S. DOBRIŠEK, J. GROS, F. MIHELIČ, K. PEPELNJAK AND I. IPŠIĆ, "GOPOLIS: Slovenian Speech Database of Spoken Flight Information Queries", submited to the *SDRV Workshop — Speech and Image Understanding*, Ljubljana, 1996

[4] W. ECKERT AND H. NIEMAN, "Semantic Analysis in a Robust Spoken Dialog System", *Proc. of the ICSLP*, Yokohama,1994.

[5] R. KUHN, "Classification trees for speech understanding systems", *PhD Thesis*, School of Computer Science Mc Gill University, Montreal, 1993.

[6] F. PEREIRA AND D. H. D. WARREN, "Definite Clause Grammar for language analysis — a survey of the formalism and a comparison with augmented transition networks", *Artificial Intelligence 13*, pp. 231–278, 1980.

[7] I. IPŠIĆ, "Word recognition in continuous speech", *PhD Thesis in preparation*, Faculty for Electrical engineering, Ljubljana, 1996.

[8] K. PEPELNJAK, J. GROS, F. MIHELIČ AND N. PAVEŠIĆ, "Linguistic Analysis in the Slovenian Information Retrieval System for Flight Services", *Proc. of the ESCA Workshop*, Vigso, Denmark, 1995.

Contact address:
K. Pepelnjak, F. Mihelič, N. Pavešić
Artificial Perception Laboratory
University of Ljubljana
Ljubljana, Slovenia
e-mail: karmen.pepelnjak@fer.uni-lj.si

KARMEN PEPELNJAK was born in 1965. She received the B.Sc. degree in electrical engineering in 1989 from the Faculty of Electrical and Computer Engineering at the University of Ljubljana. In 1992 she became a Research Staff Member in the Laboratory of Artificial Perception at the Faculty of Electrical and Computer Engineering in Ljubljana, where she is presently a graduate student and working towards M.Sc. degree in electrical engineering. Her current research interests are around Automatic Speech Recognition and Understanding of Slovenian Language.

FRANCE MIHELIČ studied at the Faculty of Natural Sciences, Faculty of Economics and Faculty of Electrical and Computer Engineering all at the University of Ljubljana. There he received thr B.Sc. degree in Tehnical Mathematics, the M.Sc. degree in Operational Research and the Ph.D. degree in Electrotehnical Sciences in 1976, 1979 and 1991, respectively. Since 1978 he has been a staff member at the Faculty of Electrical Engineering in Ljubljana, wher he is currently Assistant Professor. His research interests include Pattern Recognition, Speech Recognition and Understanding, and Signal Processing.

NIKOLA PAVESIĆ was born in Rijeka, on December 7, 1946. After his final exam at the natural science and mathematics department of the II. Grammar school in Rijeka, he studied at the Faculty of Electrical Engineering, University of Ljubljana. There he received the B.Sc. degree in electronics, the M.Sc. degree in automatics and the Ph.D. degree in electrical engineering in 1970, 1973 and 1976, respectively. He was the recipient of the Mario Osana Award in 1974, the Vratislav Bedjanić Award in 1976, and the Boris Kidrič Fund Award in 1982. Since 1970 he has been a staff member of the Faculty of Electrical and Computer Engineering in Ljubljana, currently occupying the positions of the professor of Systems, Automatics and Cybernetics, Head of the Laboratory for Artificial Perception and that of a Chairman of Department. His research interests include Pattern Recognition and Image Processing, Speech Recognition and Understanding, and Theory of Information. He has authored or co-authored more than 100 papers and 3 books addressing several aspects of the above areas. Dr. Nikola Pavesić is a member of IEEE, the Slovene Association of Electrical Engineering and Technicians (Meritorious Member), The Slovene Pattern Recognition Society (President), and the Slovene Society for Medical and Biological Engineering. He is also a member of editorial boards of several technical journals.