Dept. for Speech, Music and Hearing
Quarterly Progress and Status Report

The natural language component - STINA

Carlson, R. and Hunnicutt, S.

journal: STL-QPSR
volume: 36
number: 1
year: 1995
pages: 029-048

KTH Computer Science and Communication

http://www.speech.kth.se/qpsr

The natural language component - STINA

Rolf Carlson and Sheri Hunnicutt

Abstract

In this paper we will give a short overview of a dialogue system and describe the natural language and dialogue component in detail. Our work in this project is focused on a sub-language grammar, a grammar limited to a particular subject domain - that of requesting information from a transportation database. Our parser, STINA, is knowledge based and is designed as a probabilistic language model. It contains a context-free grammar which is compiled into an augmented transition network (ATN). Probabilities are assigned to each arc after training. Characteristics of STINA are a stack-decoding search strategy and a feature-passing mechanism to implement unification.

Dialogue management based on grammar rules and lexical semantic features is implemented in STINA. The notation to describe the syntactic rules has been expanded to cover some of our special needs to model the dialogue. The STINA parser is running with two different time scales corresponding to the words in each utterance and to the turns in the dialogue. Topic selection is accomplished based on probabilities calculated from user initiatives.

Introduction

In this paper we will give a short overview of a dialogue system and describe the natural language and dialogue component STINA in detail. Our research group at KTH[1] is currently building a generic system in which speech synthesis and speech recognition can be studied in a man-machine dialogue framework. In addition, the system is designed to facilitate the collection of speech and text data that are required for development. The system has been presented on several occasions, for example, the Eurospeech '93 conference (Blomberg et al., 1993) and the ARPA meeting '94 (Carlson, 1994).

The demonstrator application

The demonstrator application, which we call WAXHOLM, gives information on boat traffic in the Stockholm archipelago. It references time tables for a fleet of some twenty boats from the Waxholm company which connects about two hundred ports.

Besides the speech recognition and synthesis components, the system contains modules that handle graphic information such as pictures, maps, charts, and time-tables. This information can be presented to the user at his/her request. The application has great similarities to the ATIS domain within the ARPA community, the Voyager system from MIT (Glass et al., 1994) and similar tasks in Europe, for example

[1] The Waxholm group consists of staff and students at the Department of Speech Communication and Music Acoustics, KTH. Most of the efforts are done part time. The members of the group in alphabetic order are: Johan Bertenstam, Jonas Beskow, Mats Blomberg, Rolf Carlson, Kjell Elenius, Björn Granström, Joakim Gustafson, Sheri Hunnicutt, Jesper Högberg, Roger Lindell, Lennart Neovius, Lennart Nord, Antonio de Serpa-Leitao and Nikko Ström.

SUNDIAL (Peckham, 1993), the system for train timetables information developed by Philips (Aust et al., 1994, Oerder & Aust, 1994) and the Danish Dialog Project (Dalsgaard & Baekgaard, 1994).

The possibility of expanding the task in many directions is an advantage for our future research on interactive dialogue systems. In addition to boat time-tables, the database also contains information about port locations, hotels, camping places, and restaurants in the Stockholm archipelago. This information is accessed by SQL, the standardised query language (Gustafson, 1992). An initial version of the system based on text input has been running since September 1992.

Speech recognition and lexical search

The speech recognition component, which has been integrated in the system, handles continuous speech with a vocabulary of about 1000 words. The work on recognition has been carried out along two main lines: artificial neural networks (Elenius & Takács, 1990; Elenius & Tråvén, 1993; Elenius & Blomberg, 1992) and a speech production oriented approach (Blomberg, 1991). Since neural nets are general classification tools, it is quite feasible to combine the two approaches.

The frame-based outputs from the neural network form the input to the lexical search. There is one output for each of the 40 Swedish phonemes used in our lexicon. Each word in the lexicon is described on the phonetic level and may include alternate pronunciations of each word. The outputs are seen as the aposteriori probabilities of the respective phonemes in each frame. An A* N-best search has been implemented using a simple bigram language model (Ström, 1994).

Synthesis

For the speech-output component we have chosen our multi-lingual text-to-speech system (Carlson et al., 1991). The system is modified for this application. The application vocabulary has been checked for correctness, especially considering the general problem of name pronunciation.

Since the recognition and synthesis modules have the same need of semantic, syntactic and pragmatic information, the lexical information will, to a great extent, be shared. In dialogue applications such as the WAXHOLM we have a better base for prosodic modelling compared to ordinary text-to-speech, since, in such an environment, we will have access to much more information than if we used an unknown text as input to the speech synthesiser.

The speech synthesis has recently been complemented with a face-synthesis module. Both the visual and the speech synthesis are controlled by the same synthesis software (Beskow, 1995).

The Waxholm database

We have been collecting speech and text data using the Waxholm system. Initially, a "Wizard of Oz" (a human who simulates part of a system) has been replacing the speech recognition module. A scenario was presented both as text and as synthetic speech to the user. During the data collection, utterance-sized speech files were stored

together with the transcribed text entered by the wizard. The collected corpus has been used for grammar development, for training of probabilities in the language model in STINA, and also for generation of an application-dependent bigram model to be used by the recogniser.

To date, 68 subjects have been recorded and analysed. About 1900 utterances (9200 words) in this database have been used for the experiments reported in this paper. The most frequent 200 words out of the total of 720 words cover 92 percent of the collected transcribed data. About 700 utterances are simple answers to system questions while the rest, 1200, can be regarded as user initiatives. The Waxholm database will be presented in detail in a separate paper in this volume (Bertenstam et al., 1995).

The natural language component - STINA

Our initial work on a natural language component is focused on a sublanguage grammar, a grammar limited to a particular subject domain - that of requesting information from a transportation database.

Some of our fundamental concepts in our natural language component are inspired by TINA, a parser developed at MIT, (Seneff, 1992). Our parser, STINA, i.e., Swedish TINA, is knowledge based and is designed as a probabilistic language model (Carlson & Hunnicutt, 1992; Carlson & Hunnicutt, 1994; Carlson, 1994). STINA contains a context-free grammar which is compiled into an augmented transition network (ATN). Probabilities are assigned to each arc after training. Characteristics of STINA are a stack-decoding search strategy and a feature-passing mechanism to implement unification.

STINA can be used to generate text constrained by the grammar and the trained probabilities. Up to now, this feature has not been exploited to any greater degree in our work. However, the generation facility has aided in discovering weaknesses in our grammar's semantic constraints.

STINA can also be used as part of a text-to-speech system. However, in this application, the grammar is quite different from the subgrammer discussed in this paper. In the case of text-to-speech, the robust analysis is a particularly important feature of the parser, the goal being primarily to find phrases and some relations between them without an exhaustive analysis of the input text.

Rule notation and implementation

Originally the rules in STINA were formulated in text form according to a specific notation. A simple rule system could look like the following:

```
(TOP_LEVEL (NP VP))
(NP (n))
(NP ((art) (adj) n))
```

These rules would be converted to the transition networks in Figure 1.

	NP	VP	END
START	1	--	--
NP	--	1	--
VP	--	--	1

	art	adj	n	END
START	.25	.25	.5	--
art	--	.5	.5	--
adj	--	--	1	--
n	--	--	--	1

Fig. 1. Transition matrixes specified by the example rule system.

The numbers in the matrix correspond to the probability of going from one state to another. The original probabilities are taken from the frequency with which a certain transition is mentioned in the rule system. In this way, some of the paths can be forced to have a higher initial value by simply repeating a rule. It should be noted that the rule notation is meant to describe the transitions in the matrix. This means that a node name can only appear once in the matrix. Since a node can have a transition to itself, some unwanted loops may be specified. However, such overgeneration has not proved to be a disadvantage in our work.

Grammatical features

The basic grammatical features can be positive, negative or unspecified. In our implementation, we have followed our tradition from text-to-speech modelling in which unspecified features match both positive and negative features. This convention has many advantages, such as allowing nouns to have both a singular and plural interpretation. The basic grammatical features including word classes are defined by rule:

[BASIC_FEATURES A ADV ART AUX KONJ N NEU DEF PL .....]

In order to simplify the rule writing, a group of features can be given a specific name, such as the N_UP feature defined by the following rule:

[MAPPED_FEATURE N_UP [PL NEU NOM DEF]]

In our example rule system above, the "n" preterminal node can be specified with features such as the one in the following rule:

(n [N] ^[=N_UP])

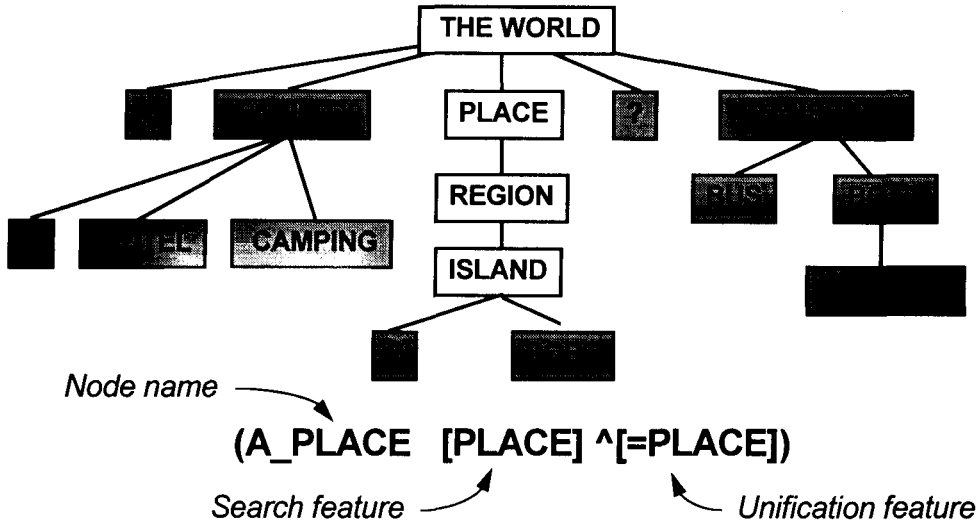
Since there is no transition matrix associated to the "n" node, it is a preterminal node which accepts a noun as a terminal. The last part of the rule tells the parser that the features according to the N_UP mapped feature should be transferred from the lexicon "up" to the node and furthermore that the "=" sign demands agreement with features already specified in the hypothesis. Similarly, the noun phrase node (NP) can have the specification ^[=N_UP] to move the features up the syntactic tree. All features are, by default, moved from the top down the branches. By this convention, the verb phrase (VP) in our example will have features such as the PL of the NP in its leaves. It is also possible to include more detailed feature manipulations in the rule system.

Semantic features

The semantic features, in opposition to the syntactic features, are only specified or unspecified. An unspecified feature is regarded to have a negative property. Semantic features can be divided into two different classes, basic features and function features. Basic features such as BOAT and PORT give a simple description of the semantic property of a word. These features are hierarchically structured.

In Figure 2, we give an example of a basic semantic feature tree. The FACILITY feature is a subdomain of THE_WORLD. Furthermore FACILITY is divided into two subgroups HOTEL, CAMPING. The underlying rule system has the following structure:

[MEANING THE_WORLD [.... FACILITY PLACE ... TRANSPORT]]
[MEANING FACILITY [.... HOTEL CAMPING]]
[MEANING TRANSPORT [BUS BOAT]]
[MEANING BOAT [STEAMER]]
.....



Example: Grinda /GR"INDA/ N SG NOM ISLAND CAMPING

Fig. 2. Example of a semantic feature tree.

The second type of semantic features are the function features. These features are not hierarchical. Typically these features are associated with an action. A typical feature is TO_PLACE indicating the destination in an utterance regarding travel. The function features are also node names in the parser. A verb can have function features set allowing or disallowing a certain type of modifier to be part of a clause. For example, the node DEPARTURE_TIME is disallowed in connection with verbs that imply an arrival time.

This method is also a powerful method to control the analysis of responses to questions from the dialogue module. The question "Where do you want to go?" conditions the parser to accept a simple port name or a prepositional phrase including a port name as a possible response from the user. This property of STINA gives the parser some of the advantages of a functional grammar parser.

Terminal features

A special type of feature is the terminal feature. These features are not represented by "bits" but by node names. Because it seems uneconomical in some cases to reserve a general feature for a very special purpose, the pre-terminal node name is simply used as a direct description of the acceptable word or group of words having the same name as the feature specification. The infinitive mark or the word "o'clock" are typical examples where this description is profitable. Another example is the word "after" as used in time expressions. It should be noted that "after" also has another lexical entry corresponding to the general preposition.

Lexicon

The lexicon entries are generated by processing each word in a lexical analyser according to a Two-Level Morphology analysis (Koskenniemi, 1983; Karlsson, 1990). Each entry is then revised by removing all unknown homographs. New grammatical and semantic features which are used by our algorithm and special application are then added. A phonetic transcription is added to each entry in the lexicon. This transcription is used by the synthesis module. The recognition module, however, has a far more elaborate pronunciation network.

When the STINA parser is used as a module in a text-to-speech system, the manual editing is bypassed and the input text is simply processed by the lexical analyser if the word is not already in the parser lexicon. This makes it possible to run unlimited text though the parser.

Terminal evaluation and feature passing

Terminal evaluation is primarily carried out on the grammatical (terminal) features. If the constraint evaluation passes, the semantic features are also evaluated. The grammatical features that are asked to be unified by the pre-terminal rule are brought in from the lexical entry and compared to the current hypothesis. The constraint evaluation fails if any of the tests give a negative response.

The hierarchical structure has importance for the rule writing. During the unification process in STINA, all semantic features which belong to the same semantic branch in the feature tree are considered.

The rule that Figure 2 depicts uses the feature structure to accept all places, regions, islands and ports. Thus, a unification of the feature PLACE engages all semantic "non-shaded" features in the figure. The whole tree of the lexical entry is moved into the hypothesis including the leaves on the feature tree. A port name will keep its PORT feature even if only the PLACE is noted in the rule. This has several advantages. The rules do not have to be more specific than necessary and the domain knowledge can, to some extent, be part of the lexicon rather than the rules. This mechanism is extensively used in the sublanguage grammar for our application.

Hypothesis progression and stacks

In this section we will give a general description of how the parser is processing an utterance (Fig. 3.) We call an unfinished hypothesis *h,* and the whole ensemble of

unfinished hypotheses *H*. Each *h* consists of a sequence of links. A link contains information about the hypothesis such as features, probabilities and pointers to lexical entries and node specifications. Each *h* has a hypothesis score associated to it, which is used to rank order *H*. The score is updated each time a new link is added to the *h*. The score will be discussed in more detail below. A hypothesis can split into several new individual hypotheses.

Initially a link corresponding to the TOP_LEVEL is placed on an *h-stack*. A process is started with the goal of clearing the *h-stack*. Each *h* is extended and evaluated according to the rules. If the *h* is not accepted, the links in *h* are deleted up to a link that is shared by another *h*. If the *h* is accepted, the *h* is taken from the *h-stack* and placed in the *history stack*. If the new link is a preterminal link, it is placed on the *preterm-stack*. Otherwise it is placed on the *h-stack*. If the *h* has reached an "end of utterance" state and no more input words remain, the *h* is moved to the *n-best-stack*. At some point, the *h-stack* is empty and *H* is placed in the *preterm-stack*, the *history-stack* and the *n-best-stack*.

The next step is to evaluate the *preterm-stack* according to the input. If this evaluation fails, the *h* is deleted and all links describing the *h* are deleted up to the point where a link is shared between/among more than one *h*. If the evaluation passes, the *h* is placed on a *term-stack* ordered according to the hypothesis score with an appropriate pointer to the lexical entry in question. When all pre-terminals have been evaluated, a number of the top links in the *term-stack* are moved back to the *h-stack*. It is possible to specify how many *n-best* solutions should be considered before terminating this iterative process. This criteria is only examined when the *h-stack* is empty. With this method, only two stacks -- the *term-stack* and the *n-best-stack* - need to be ordered according to the hypothesis score.

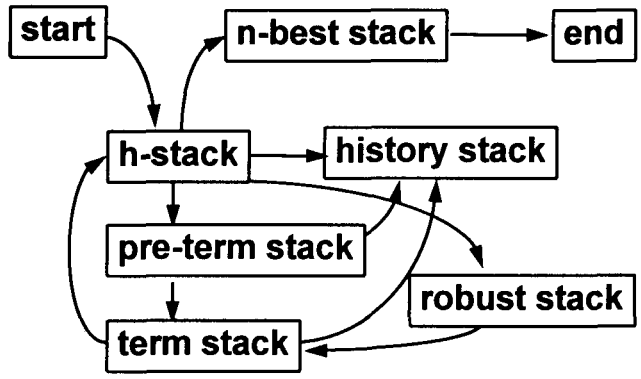


Fig. 3. Block diagram of hypothesis processing.

Probability and score calculation in STINA

Each time a link is added to a hypothesis, the hypothesis score is updated according to the transition probability and stored in this link. When a terminal is added to the hypothesis, the lexical probability is also included in the calculation. If the true probability was used as a score it would get lower and lower values the farther a hypothesis gets. If no compensation were introduced, the hypothesis that had come the

farthest would be in the bottom of the *term-stack*, forcing the parser, in principle, to process all hypotheses, *H*, in parallel. The clear goal in the project is to maintain a reasonable parsing speed and low memory consumption. Thus, a few mechanisms have been included to improve the performance.

The first mechanism introduced in STINA was to give an adjustment factor to each lexical entry. This would increase the score each time a terminal is included in a hypothesis. We have in Figure 4 plotted the mean score according to word number and utterance length. The next step was to train this factor in order to achieve a constant level in mean relative word number.

The next mechanism, inspired by the work at IBM (Bahl, 1993), was to introduce a top-level envelope corresponding to the maximal probability at each word position. When a terminal node is reached by a hypothesis, this envelope is adjusted if necessary. The *term-stack* is ordered according to the probabilities relative to this envelope.

These two mechanisms drastically improved the performance of the parser, regarding both speed and accuracy.

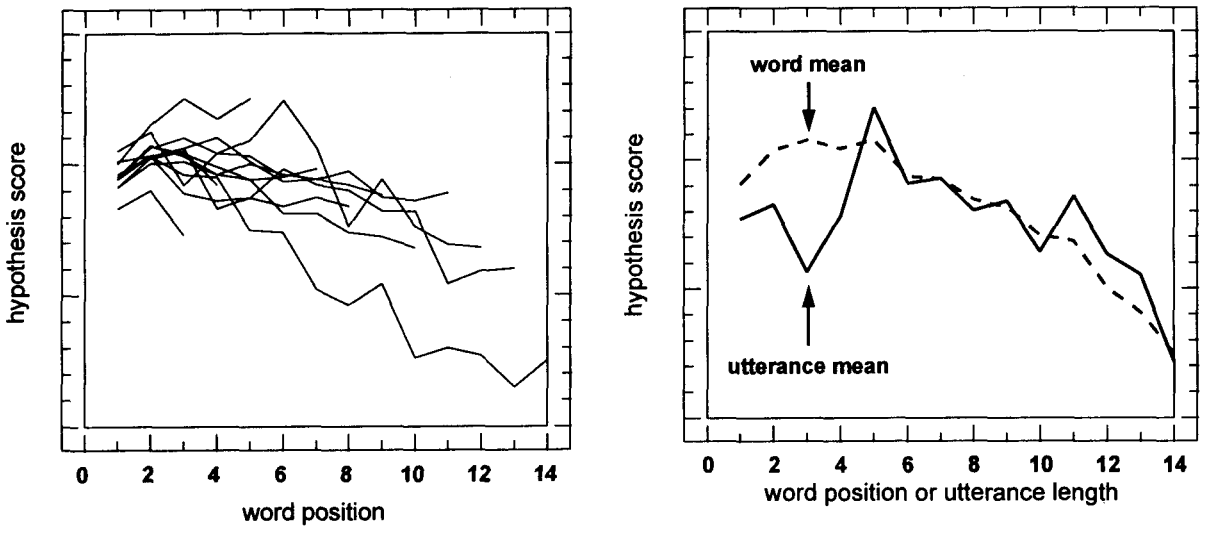


Fig. 4. a) Mean hypothesis score according to word position plotted for each utterance length. b) Mean final hypothesis score plotted according to utterance length and mean hypothesis score averaged independent of utterance length.

Robust analysis

So far, we have discussed only complete parsing but we have also introduced a simple method to do robust parsing. The notation has been expanded to accept certain nodes to be seeds for robust parsing. When such a node is passed, the hypothesis is split into two. One is put on a *robust stack* while the other is pushed forward as before. When the *h-stack* and the *preterm-stack* are empty, the *robust stack* is added to the *term-stack*. Thus, robust analysis is competing with the complete analysis during parsing. The probability of the robust hypothesis is set in relation to the probability envelope discussed above. This method gives a rather reasonable result, bypassing some of the problems such as "restarts" and the occurrence of unknown words.

Parsing results

The parser has been evaluated in several different ways. Using about 1700 sentences in the Waxholm database as test material, 62 percent give a complete parse, whereas if we restrict the test data to utterances containing user initiatives (about 1200), the result is reduced to 48 percent. This can be explained by the fact that a large number of responses to system questions typically have a very simple syntax. If we exclude extralinguistic sounds, such as lip smack, sigh and laughing, in the test material based on dialogue initiatives by the user, we increase the result to 60 percent complete parses. Sentences with incomplete parses are handled by the robust parsing component and frequently effect the desired system response. It should be noted that the test material is not unseen by the system developers.

We have only a few examples of restart (3%) on a word level in the data, preventing a complete parse. However, in most cases, the robust analysis gives a reasonable solution to the problem.

The parser is relatively fast on our HP 735. It takes about 17 msec to process an utterance. It can be changed to run faster if some of the analysis facilities are taken out; a slightly different approach on constraint evaluation would also make it faster. At the moment, the processing speed of the parser is not an important issue.

The perplexity on the Waxholm material is about 150 using an untrained grammar. If the grammar is trained and tested on the same material, we get a reduction to about 30. In the perplexity analysis below (Table 1). We have divided the Waxholm database into four parts. Each speaker only appears in one part. The training has been done on three parts and testing on one part. This procedure has been rotated four times and the reported results in the table are the mean of these four runs.

Table 1. Perplexity for various subsets of material from the data bas. (N=total number of words.)

Test material	Perplexity	N
woz input (untrained grammar)	150.9	6888
woz input (trained grammar)	33.9	6993
no extralinguistic sounds	30.4	7091
only complete parse	23.4	3880

Man-STINA interaction

In the implementation of the parser and the dialogue management, we have stressed an interactive development environment. This makes it easier to have control over the system's progress as more components are added. It is possible to study the parsing and the dialogue flow step by step when a graphic tree is built. It is even possible to use log files, collected during Wizard of Oz-experiments, as scripts to repeat a specific dialogue including all graphic displays and acoustic outputs.

The rule system was originally defined according to the notation described above. We have recently added a graphical interface to the system which presents each network graphically. Both the syntax and the dialogue networks can be modelled and

edited graphically with this tool. Each node's function can be changed and new nodes can be added. It is also possible to store the network and to import it into the parser. This new module has increased the speed of debugging the rule set and it has also made it possible to make quick changes to the system. Earlier work on dialogue modelling such as the Generic Dialogue System Platform in the Danish dialogue project (Larsen & Baekgaard, 1994) has been an inspiration for this expansion.

Semantic analysis

The semantic analysis is a straight-forward process in which the syntactic tree is reduced to a semantic tree, deleting all nodes and branches that contain no semantic information. A semantic node is characterised by having a semantic function feature associated to it or, in the case of a preterminal node, having a lexical entry with associated semantic features which are transferred to the hypothesis. After the tree has been reduced, a special process creates a semantic frame with slots corresponding to attribute-value information taken from the tree (Fig. 5). The semantic frame has a

```
TEXT:
jag vill åka till Waxholm på fredag . (I want to go to Waxholm on Friday.)

PARSE:
(TOP_LEVEL
      (STATEMENT
            (SUBJECT "jag"/PRON )
            (VERBAL "vill"/aux åka"/v_inf )
            (MODIFIERS
                  (MOD (TO_PLACE "till"/TO "Waxholm"/A_PLACE ))
                  (MOD (AT_DAY "på"/PREP_ON "fredag"/A_DAY ))
            )
      )
)

SEMANTIC PARSE:
(TOP_LEVEL
      (STATEMENT
            (VERBAL     "åka"/MOVE/ )
            (TO_PLACE   "Waxholm"/PORT/ )
            (AT_DAY     "fredag"/DAY/ )
      )
)

SEMANTIC FRAME:
Semantic features:  /AT_DAY TO_PLACE VERBAL MOVE PORT DAY/
(VERBAL             "åka"/MOVE/ )
 (TO_PLACE          "Waxholm"/PORT/ )
 (AT_DAY            "fredag"/DAY/ )
```

Fig. 5. Semantic analysis of the utterance "jag vill åka till Waxholm på fredag ."

feature specification describing which features are used in the frame and which information might have been added to the frame from the dialogue history. The latter aspect will be discussed more below. Special code has been added to handle time expressions, which is important for our application. This is currently the only application specific code in STINA.

Dialogue management

Spoken dialogue management has attracted considerable interest during the last years. Special workshops and symposia, for example the special workshop at Waseda University, Japan 1993 (Shirai & Furui, 1994) and the ESCA workshop on Spoken Dialogue Systems inVigsø, 1995, are arranged to forward research in this field. We will not attempt to review this growing field in this paper. We will, however, describe in some detail the current effort to model the dialogue in the Waxholm system. Our objective is to develop a dialogue management module which can handle the type of interaction that can occur in our chosen domain. The complexity of the task sets the needed number of dialogue elements, as discussed by Bernsen et al. (1994). The Waxholm system should allow user initiatives, without any specific instructions to the user, complemented by system questions to achieve the user's goal. Based on this aim, two major ideas have been guiding the work. First, the dialogue should be described by a grammar. We have chosen to use the same notation and the same software (STINA) to implement the dialogue grammar. In our system, dialogue building blocks are described by nodes. Each node has specifications concerning, for example, constraint evaluation and system response. Second, the dialogue should be probabilistic. Topic selection is accomplished based on probabilities calculated from user initiatives (Carlson, 1994; Carlson & Hunnicutt, 1994).

The topic selection based on probabilities in our system has similarities with the effort at AT&T (Gorin, 1994; Gorin et al., 1994). A different approach, also based on training, has been presented by Kuhn & De Mori (1994) in their classification approach. The dialogue system developed in Denmark is based on a special tool (Larsen and Baekgaard, 1993) in which the dialogue flow can be described by a network of building blocks. These blocks can be edited graphically. The OASIS developed by the GTE Laboratories Incorporated (Zeigler & Mazor, 1994) is based on dialogue prototypes which include building blocks for the acquisition of factual information, for the verification of acquired information, and for reacquisition following a disconfirmation.

Topic selection

In the following description, we have used the term "topic" to describe what type of information a user is requesting or, in some cases, a special response from the system. In Fig. 6, some of the major topics are listed. The decision about which path to follow in the dialogue is based on several factors such as the dialogue history and the content of the specific utterance. The utterance is coded in the form of a "semantic frame" with

slots corresponding to both the grammatical analysis and the specific application (Fig. 5). The structure of the semantic frame is automatically created based on the rule system.

TIME_TABLE
Goal: to get a time-table presented with departure and arrival times specified between two specific locations.
Example: När går båten? (When does the boat leave?)

SHOW_MAP
Goal: to get a chart or a map displayed with the place of interest shown.
Example: Var ligger Waxholm? (Where is Waxholm located?)

EXIST
Goal: to display the availability of lodging and dining possibilities.
Example: Var finns det vandrarhem? (Where are there hostels?)

OUT_OF_DOMAIN
Goal: to inform the user that the subject is out of the domain for the system.
Example: Kan jag boka rum. (Can I book a room?)

Fig. 6. Some of the main topics used in the dialogue.

Each semantic feature found in the syntactic and semantic analysis is considered in the form of a conditional probability to decide on the topic. The probability for each topic is expressed as: p(topic|F), where F is a feature vector including all semantic features used in the utterance (Table 2). Thus, the BOAT feature can be a strong indication for the TIME_TABLE topic but this can be contradicted by a HOTEL feature. The probability has been trained using a labelled set of utterances taken from the Waxholm database. Only utterances indicating a topic (about 1200) have been included in this set. The probability is calculated according to: $p = (n+1)/(N+2)$, where N = number of times a feature can be a terminal node in the feature tree, and n = number of times a feature actually is a terminal node. In the topic prediction we exclude the features which are set by a feature on a lower level in the feature tree. In Figure 7, the features BOAT and PORT are present in the semantic frame corresponding to the utterance. Thus, the features that are shaded in the figure (the world, transport, place, region and island) are not included in the calculation of the topic probabilities.

Evaluation of topic selection

We have performed a sequence of tests to evaluate the topic selection method. The evaluation has been performed using one quarter of the material, about 300 utterances, as test material, and the rest as training material, about 900 utterances. This procedure has been repeated for all quarters and the reported results are the mean values from these four runs. The first result, 12.9% errors in Table 3, is based on the unprocessed labelled input transcription. The eight possible topics have a rather uneven distribution in the material as can be seen in Table 4. One of the topics, labelled "no understanding," is trained on a set of constructed utterances that are not possible to

Table 2. Probabilities that a certain semantic feature is present in a topic-indicating utterance.

FEATURES ↓ TOPICS →

	TIME TABLE	SHOW MAP	FACILITY	NO UNDER-STANDING	OUT OF DOMAIN	END
OBJECT	.062	.312	.073	.091	.067	.091
QUEST_WHEN	.188	.031	.024	.091	.067	.091
QUEST_WHERE	.062	.688	.390	.091	.067	.091
FROM_PLACE	.250	.031	.024	.091	.067	.091
AT_PLACE	.062	.219	.293	.091	.067	.091
TIME	.312	.031	.024	.091	.067	.091
PLACE	.091	.200	.500	.091	.067	.091
OOD	.062	.031	.122	.091	.933	.091
END	.062	.031	.024	.091	.067	.909
HOTEL	.062	.031	.488	.091	.067	.091
HOSTEL	.062	.031	.122	.091	.067	.091
ISLAND	.333	.556	.062	.091	.067	.091
PORT	.125	.750	.244	.091	.067	.091
MOVE	.875	.031	.098	.091	.067	.091

$$\underset{i}{\text{argmax}} \{ p(t_i \mid \vec{F}) \}$$

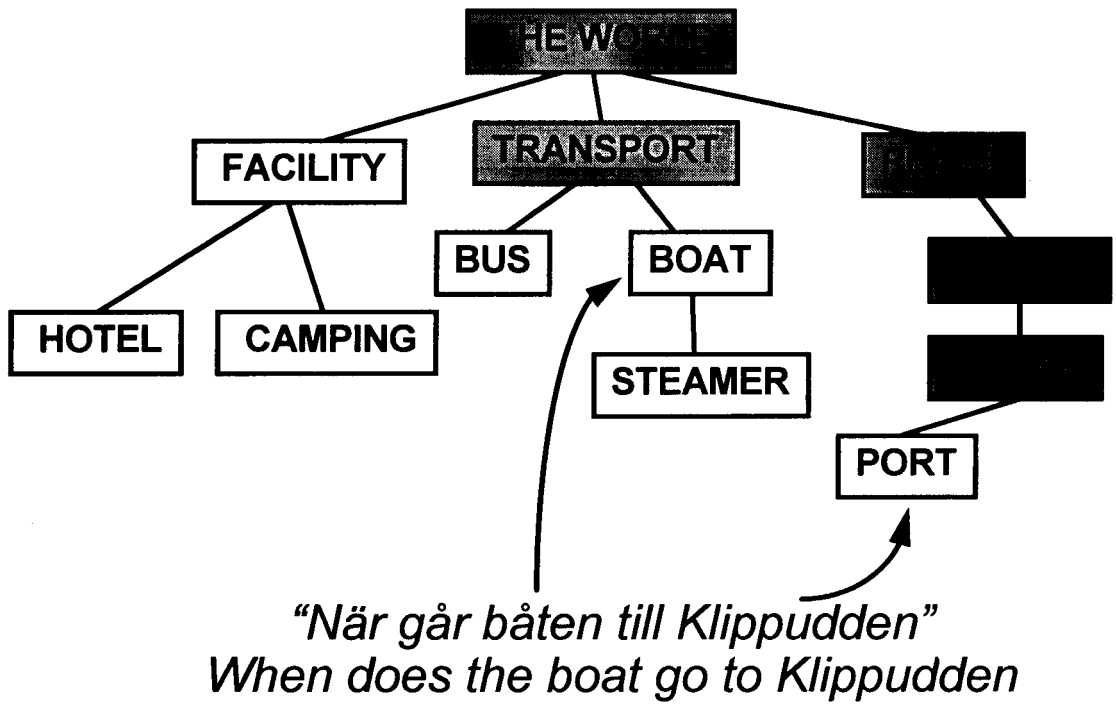


Fig. 7. Features excluded (shaded) in the topic probability calculation.

understand, even for a human. This topic is then used as a model for the system to give an appropriate "no understanding" system response. It should be noted that, in principle, this is not a question of utterances that do not get a reasonable parse. However, the topic prediction is certainly influenced by this fact. It seemed reasonable to exclude the "no understanding" prediction from the result since the system at least does not make an erroneous decision. The accuracy model in word recognition evaluation has the same underlying principle. By excluding 55 utterances, about 5% of the test corpus, predicted to be part of the "no understanding" topic, we reduce the error by about 4%.

Table 3. Results from the topic prediction experiments.

Test material	All material		Excluding no understanding	
	% Error	N	% Error	N
woz input	12.9	1209	8.8	1154
no extralinguistic sounds	12.7	1214	8.5	1159
only complete parses	3.1	581	2.9	580

In the next experiment, we excluded all extralinguistic sounds, about 700, in the input text. This will increase the number of complete parses with about 10% as discussed earlier. The prediction result was about the same compared to the first experiment.

The final experiment included only those utterances that gave a complete parse in the analysis. The errors were drastically reduced. This means that the utterances with a syntax covered by our grammar also were semantically easier to interpret. On the other hand, we do not yet know if an increased grammatical coverage also will reduce the topic prediction errors.

Table 4 shows the confusion matrix for the initial experiment. We can clearly see

Table 4. Confusion matrix for topic prediction. The matrix is calculated as the sum of four experiments.

	N	time table	get pos	exist	trip map	end scen	re-peat	out of dom	no und.	error %
time table	545	502	4	2	7	.	.	19	11	7.9
show map	79	.	65	8	.	.	.	.	6	17.7
exist	293	1	2	263	1	5	.	13	8	10.2
trip map	32	6	.	1	22	.	.	3	.	31.2
end scenario	156	.	.	.	.	140	.	.	16	10.3
repeat	5	.	.	.	.	.	.	.	5	100.0
out of domain	83	8	.	12	1	2	.	56	4	32.5
no understanding	16	2	.	2	.	5	2	.	5	68.8
total	1209	519	71	288	31	152	2	91	55	12.9

the uneven distribution of topics in the database. The "trip map" topic is meant to give the user information on how it is possible to travel in the archipelago. Thus, the topic is relatively close to the "time table" topic and the confusions between the two topics are understandable. The "out of domain" topic, telling the user that the system can not give the requested information, such as ordering tickets or booking rooms, is difficult to handle. A user initiative in a single utterance can easily contain a request for information, some of which is available to the system and some of which is not. In this case, it is unclear if a response to the part of the request for which information is available is the correct response to give. In most cases, the existence of a partial request which indicates "out of domain" has been preferred in the labelling.

Dialogue rules

Dialogue management based on grammar rules and lexical semantic features is implemented in STINA. The notation to describe the syntactic rules has been expanded to cover some of our special needs to model the dialogue. The STINA parser is running with two different time scales concurrently corresponding to the words in each utterance and to the turns in the dialogue. Syntactic nodes and dialogue states are processed according to transition networks with probabilities on each arc.

Each predicted dialogue topic is explored according to the rules. These rules define which constraints have to be fulfilled and what action should be taken depending on the dialogue history. Each dialogue node is specified according to node type, node activity, and constraint evaluation. (Fig. 8).

Dialogue Node Specifications

Node types
- branching or preterminal

Constraint evaluation on
- dialogue flow features
- semantic frame slots and features
- .....

If more information needed
- synthesise question to user
- control parser to accept incomplete sentences
- .....

Node activity:
- record utterance
- synthesise message
- test constraints
- data base search using SQL
- graphic display table
- graphic display map
- graphic display picture
- .....

Fig. 8. Dialogue node specification.

The constraint evaluation is described in terms of features and in terms of the content in the semantic frame. If the frame needs to be expanded with additional information, a system question is synthesised. During recognition of a response to such a question, the grammar is controlled with semantic features in order to allow incomplete sentences. If the response from the subject does not clarify the question, the robust parsing is temporarily disconnected so that specific information can be given to the user about syntactic problems or about unknown word problems. At the same time, a

complete sentence is requested giving the dialogue manager the possibility of evaluating whether the chosen topic is incorrect.

A positive response from the constraint evaluation opens the way for the selected action to take place. The node action list in the figure gives examples of such actions. In Figure 9, we continue our simple dialogue example. The request "I want to go to Waxholm on Friday" predicts the time table topic which is confirmed to the user. However, the system needs information about the place of departure. When asked, the user responds with a single word. It is worth mentioning that the dialogue module in this state primes the syntax analysis to also accept a response other than a complete sentence, and that this response has to be of the FROM_PLACE type. This whole process is handled by the feature passing between the dialogue part and the grammar part of STINA. In most cases, the user *does* answer system questions, as will be discussed in the data-base contribution in this volume (Bertenstam et al., 1995).

TEXT:
Jag vill åka till Waxholm på fredag. (I want to go to Waxholm on Friday.)
PROPOSED TOPIC:
TIME_TABLE
SYNTHESIS:
Jag söker båtar som går till Waxholm på en fredag. Varifrån vill du åka? (I'm looking for boats that go to Waxholm on Fridays. Where do you want to depart from?)
TEXT:
Stockholm .
SEMANTIC FRAME:
Semantic features: /FROM_PLACE /ISLAND/
(FROM_PLACE "Stockholm"/ISLAND/)
SYNTHESIS:
Detta är en tabell över de båtar som går från Stockholm till Waxholm på en fredag. (Here is a table of boats that go from Stockholm to Waxholm on Fridays.)

Fig. 9. Example of a user - system interaction.

Text generation

As yet, the text generation part of the system is unsophisticated. In principle, all messages are generated from system utterance skeletons in which available information is included. These skeletons are part of the dialogue node structure and are defined in the rule system. A simplified example of the text generation is shown in Figure 10. The PRESENT node is used to synthesise information to the user that a time table is shown on the screen with the requested information. The BOAT_LIST slot in the PRESENT text is filled by the information found in the BOAT_LIST node and so forth. The FROM node in our example has a dual function. One is the mentioned text

generation function, but the most important function is to evaluate whether the FROM_PLACE attribute is filled in the semantic frame. If not, the system requests more information with the question "From where do you want to go?". If the constraint is fulfilled or becomes fulfilled after a subdialogue, the content in the semantic frame is pushed forward in the dialogue according to which features are set in the mapped T_HIST feature. In this section we have only attempted to give some simple examples of the text-generating part of the system.

Introduction of a new topic

The rule-based, and to some extent probabilistic, approach we are exploring makes the addition of new topics relatively easy. However, we do not know at this stage where the limits are for this approach. In this section we will give a simple example of how a new topic can be introduced.

```
(PRESENT "Detta är en tabell över de båtar [BOAT_LIST] .")
        "This is a table of the boats [BOAT_LIST]."
(BOAT_LIST "som går [FROM] [T_TO] [T_DAY][AFTER][BEFORE]")
        "that go [FROM] [T_TO] [T_DAY][AFTER][BEFORE]"
(AFTER "efter klockan [FROM_TIME.AFTER_TIME]")
        "after [FROM_TIME.AFTER_TIME]"
(BEFORE "och före klockan [FROM_TIME.BEFORE_TIME]")
        "and before [FROM_TIME.BEFORE_TIME]
(FROM [FROM_PLACE] "från [/ISLAND/FROM_PLACE]"
                    ":Varifrån vill du åka." ^[=T_HIST]])
                    "from [/ISLAND/FROM_PLACE]"
                    "Where do you want to go from?"
  .....
```

Fig. 10. Rule example to generate information about a time table displayed on the screen.

Suppose we want to create a topic called "out of domain." Below is a list of the tasks that are involved in implementing a new topic not contained in the database.

1. Introduce a new dialogue grammar parent node
2. Expand the semantic feature set if needed
3. Specify dialogue children nodes and their function and add to lexicon
4. Construct and label training sentences
5. Train topic probabilities

First a topic node is introduced in the rule system. Some new words probably need to be included in the lexicon and labelled with a semantic feature showing that the system does not know how to deal with the subjects these words relate to. Then a synthesis node might be added with a text informing the user about the situation. Example sentences must be created that illustrate the problem and the dialogue parser must be trained with these sentences labeled with the "out of domain" topic. Since the

topic selection is done by a probabilistic approach that needs application-specific training, data collection is of great importance for the progress of the project.

Final remarks

In our presentation, we have described the natural language and dialogue control components in the Waxholm project. No module is yet considered complete. The dialogue will be naturally restricted by application-specific capabilities and the limited grammar. So far, we also assume that the human subjects will be co-operative in pursuing the task. Recovery in case of human-machine "misunderstandings" will be aided by informative error messages generated upon the occurrence of lexical, parsing or retrieval errors. This technique has been shown to be useful in helping subjects to recover from an error through rephrasing of their last input (Hunnicutt et al., 1992).

The STINA parser handles both the regular grammar analysis and the dialogue control. We have found this approach to be very profitable since the same notation, semantic feature system and developing tools can be shared. The rule-based and probabilistic approach has made it reasonably easy to implement an experimental dialogue management module. It remains to test it in a more realistic environment.

Acknowledgement

This work has been supported by grants from The Swedish National Language Technology Program.

References

Aust H, Oerder M, Seide F & Steinbiss V (1994). Experience with the Philips Automatic Train Timetable Information System. In: *Proc IEEE workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA94),* 67-72.

Bahl LR et al. (1993). Search issues in large vocabulary speech recognition. *IEEE ASR Workshop*, Snowbird (unpublished).

Bernsen NO, Dybkjaer L & Dybkjaer H (1994). A dedicated task-oriented dialog theory in support of spoken language dialog systems design. In: *Proc ICSLP International Conference on Spoken Language Processing,* Yokohama, 875-878.

Bertenstam J, Blomberg M, Carlson R, Elenius K, Granström B, Gustafson J, Hunnicutt S, Högberg J, Lindell R, Neovius L, Nord L, Serpa-Leitao A de & Ström N (1995). Spoken dialogue data collected in the Waxholm project. *STH-QPSR, KTH,* 1: 49-74.

Beskow J (1995). Regelstyrd visuell talsyntes. *Senior thesis*, Dept of Speech Communication, KTH (only available in Swedish).

Blomberg M (1991). Adaptation to a speaker's voice in a speech recognition system based on synthetic phoneme references. *Speech Communication* 10: 453-462.

Blomberg M, Carlson R, Elenius K, Granström B, Gustafson J, Hunnicutt S, Lindell R & Neovius L (1993). An experimental dialogue system: WAXHOLM. In: *Proc Eurospeech '93, 3rd European Conference on Speech Communication and Technology,* Berlin, 1867-1870

Carlson R (1994). Recent developments in the experimental "Waxholm" dialog system. In: *ARPA Human Language Technology Workshop*, Princetown, New Jersey, 207-212.

Carlson R & Hunnicutt S (1992). STINA: A probabilistic parser for speech recognition. In: *FONETIK'92, Sixth Swedish Phonetics Conference, Chalmers Technical Report No 10*, Gothenburg, 23-26.

Carlson R & Hunnicutt S (1994). Dialog management in the Waxholm system. In: *Eighth Swedish Phonetics Conference, Working Papers 43*, Lund, 46-49.

Carlson R, Granström B & Hunnicutt S (1991). Multilingual text-to-speech development and applications. In: (Ainsworth AW, ed.), *Advances in speech, hearing and language processing*, London: JAI Press, UK.

Dalsgaard P & Baekgaard A (1994). Spoken Language Dialogue Systems. In: *Proc in Artificial Intelligence*, Infix. Presented at the CRIM/FORWISS workshop on 'Progress and Prospects of Speech Research and Technology, Munich.

Elenius K & Takács G (1990). Acoustic-phonetic recognition of continuous speech by artificial neural networks. *STL-QPSR, KTH,* 2-3: 1-44.

Elenius K & Tråvén H (1993). Multi-layer perceptrons and probabilistic neural networks for phoneme recognition. In: *Proc Eurospeech '93, 3rd European Conference on Speech Communication and Technology,* Berlin, 1237-1240.

Elenius K & Blomberg M (1992). Experiments with artificial neural networks for phoneme and word recognition. In: *Proc of ICSLP 92,* Banff, 2: 1279-1282.

Glass J, Flammia G, Goodine D, Phillips M, Polifroni J, Sakai S, Seneff S & Zue V. Multilingual spoken-language understanding in the Mit Voyager System. (To be published in *Speech Communication.*)

Gorin A (1994). Semantic associations, acoustic metrics and adaptive language aquisition. In: *Proc ICSLP, International Conference on Spoken Language Processing, Yokohama,* 79-82.

Gorin AL, Hanek H, Rose R, & Miller L (1994). Automatic call routing in a telecommunications network. In: *Proc IEEE workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA94),* 137-140.

Gustafson J (1992). Databashantering som del av ett talförståelsesystem. *Senior thesis*, Dept. of Speech Communication, KTH. (Only available in Swedish.)

Hunnicutt S, Hirschman L, Polifroni J, & Seneff S (1992). Analysis of the effectiveness of system error messages in a human-machine travel planning task. In: *Proc ICSLP International Conference on Spoken Language Processing,* Alberta, Canada, 1: 197-200.

Karlsson F (1990). A comprehensive morphological analyzer for Swedish. *University of Helsinki, Dept of General Linguistics. (Manuscript).*

Koskenniemi K (1983). Two-level morphology: A general computational model for word-form recognition and production. *University of Helsinki, Dept of General Linguistics, Publications No. 11.*

Kuhn R & De Mori R (1994). Recent results in automatic learning rules for semantic interpretation. In: *Proc ICSLP, International Conference on Spoken Language Processing, Yokohama*, 75-78.

Larsen LB & Baekgaard A (1994). Rapid prototyping of a dialog system using a generic dialog development platform. In: *Proc ICSLP International Conference on Spoken Language Processing, Yokohama,* 919-922.

Oerder M & Aust H (1994). A realtime prototype of an automatic inquiry system. In: *Proc ICSLP International Conference on Spoken Language Processing, Yokohama,* 703-706.

Peckham J (1993). A new generation of spoken dialog systems: results and lessons from the SUNDIAL project. In: *Proc Eurospeech '93, 3rd European Conference on Speech Communication and Technology,* Berlin, 33-40.

Seneff S (1992). TINA: A natural language system for spoken language applications. *Computational Linguistics,* 18(1): 61-66.

Shirai K & Furui S (1995). Spoken Dialogue. In: (Shirai & Furui, eds.) *Special issue of Speech Communication,* 15(3-4).

Ström N (1994). Optimising the lexical representation to improve A* lexical search. *STL-QPSR, KTH,* 2-3: 113-124.

Zeigler B & Mazor B (1994). Dialog design for a speech interactiv automation system. In: *Proc IEEE workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA94),* 113-116.