

Generating Natural Language Text in a Dialog System

Mare Koit

Department of Programming
2 Juhan Liivi Street
Tartu State University
202400 Tartu Estonia USSR

Madis Saluveer

Artificial Intelligence Laboratory
78 Tiigi Street
Tartu State University
202400 Tartu Estonia USSR

Abstract

The paper deals with generation of natural language text in a dialog system. The approach is based on principles underlying the dialog system TARLUS under development at Tartu State University. The main problems concerned are the architecture of a dialog system and its knowledge base. Much attention is devoted to problems which arise in answering the user queries - the problems of planning an answer, the non-linguistic and linguistic phases of generating an answer.

1. Introduction

Several problem domains can be named where the task of automatic generation of natural language texts, not sentences, has become the "hot topic", e.g. machine translation, dialog between the user and the computer in a dialog system, etc. Given the two central problems of natural language generation, that of what to say and that of how to say it, we discuss shortly the main components of a dialog system which should enable the system to solve the problem of how to decide what to say, and then proceed to a more detailed treatment of the second task. The present approach draws heavily from the dialog system TARLUS being developed at Tartu State University, Estonia, USSR. Several modules of the system have been working independently for some time already, and experiments are under way in merging these modules into a unified dialog system. Though the surface generation is carried out in the Estonian language the authors hope that a number of ideas which have found their way into this system will be of more general interest.

2. Main Components of the Dialog System

The dialog system consists of the following modules: Linguistic Processor, Turns' Interpreter, Turns' Generator, Planner, Dialog Monitor, Solver. In addition to this the dialog system includes several knowledge bases for long-term knowledge: goals' knowledge base, problem domain knowledge base, linguistic knowledge base, dialog knowledge base, partner's knowledge base and self-knowledge base. To store short-term knowledge the system contains a number of models: of activated goals of the system, of the text of preceding dialog turns, of the communicative situation, of the partner, of the system itself.

Dialog partners always follow certain goals in their interaction. The goals of the

dialog system may be thought of as implicit questions the system is seeking answers to during the dialog. The long-term goals of the system are kept in goals' knowledge base where they have attached to them priority assessments. In the course of interaction these goals (or subgoals) may rise or lower. For every new goal the system must set a priority assessment.

There can be three types of questions in the dialog system: user questions to the system, the questions of the system to itself, and the questions of the system to its user. Every question in its turn may concern either the problem domain or the process of interaction. The central notion of the dialog is a "turn". In natural dialog both interlocutors generate their turns in certain order and thus we may represent a dialog as a sequence of interchanging turns

$$t_1^a, t_1^b, t_2^a, t_2^b, \dots, t_k^{a/b}$$

where t_1^a, t_2^a , etc. are the turns of the first interlocutor, and t_1^b, t_2^b , etc. are the turns of the second interlocutor. Every turn may consist of one or several communicative steps, e.g. the turn refusal may consist of the communicative steps REFUSAL + MOTIVATION, where REFUSAL dominates MOTIVATION.

The dialog system functions as follows. The Linguistic Processor carries out morphological and syntactic analysis of the input turn with the help of linguistic knowledge base. A later task of the Linguistic Processor is the generation of the surface answer from its semantic representation.

The Turns' Interpreter fulfills several tasks:

i) it constructs the semantic representation of the turn with the help of problem domain knowledge base. For the interpretation of the following turns it may be necessary to take into account the preceding turns of the partners. To this end the Turns' Interpreter simultaneously constructs the semantic interpretation of the dialog text already analysed.

ii) the Turns' Interpreter should recognize in a turn the corresponding communicative step(s). For this task it uses the dialog knowledge base. At the same time the Turns' Interpreter forms the model of communicative situation, by supplementing it with typical structures of recognized communicative steps and combining them into bigger units on the basis of recognized turns and turn cycles.

iii) it establishes the activated goals of the dialog system during a dialog proceeding from the turn under interpretation. The questions which the user poses to the system and the questions that the system formulates

on the basis of recognized communicative steps with the help of "interest rules" are carried into the model of goals and supplied with priority assessments.

When the Turns' Interpreter has finished its job the Planner scans the goals in the goals knowledge base and the model of activated goals. It also tries to find answers to the remaining questions by addressing the Solver when the question concerns the problem domain, or the Dialog Monitor when the problem is about communication.

The Turns' Generator selects the type of answer turn on the basis of the set of questions chosen by Planner and the communicative steps which will form the prospective turn. For instance, the question may be conveyed as communicative steps ANSWER, EX-
PRESSION OF DOUBT, etc. The question that has not yet been answered may be conveyed as communicative steps QUESTION, REQUEST, ORDER, etc. Secondly, the Turns' Generator constructs the semantic representation of the future turn and adds it to the model of text of preceding dialog turns. The generation of an answer turn is finished by the Linguistic Processor which transforms semantic representation of a turn into a text.

To organize cooperation between the different modules is a very complicated process. We share the opinion that it cannot proceed linearly but is rather organized as cooperation between experts permanently exchanging information among themselves (Dim et al., 1984).

3. Knowledge Used in Text Generation

3.1. Dialog knowledge base

Dialog knowledge base contains type structures of communicative steps and rules which the dialog system uses in interpreting the replies of its partner and generating its own turns.

The main structural components of a communicative step are:

- SETTING - facts describing the situation where the given communicative step takes place: preconditions which hold about the author of that step or, in the author's mind, about the partner as well as about the objective reality
- PLOT - contents/theme of the given communicative step
- GOAL - communicative goal of the author of the step
- CONSEQ - the outcome of the communicative step, i.e. the changes in the communicative situation which take place as a result of that communicative step.

For the cooperation with the dialog system to be natural the dialog knowledge base should also contain certain rules of communication which the system should follow when carrying out the dialog. Among them the most important are the principle of cooperation (Grice 1975) and the principle of politeness (Leech 1983). In dialog knowledge base these principles are contained as fixed types of

rules.

In the following we exemplify some of these rules together with some examples of using them in live dialogs.

1. The general form of rules of behavior is as follows:

IF <situation> THEN <action>

For instance,

IF interlocutors A and B have a common goal G and A thinks that there exists an obstacle on the way of achieving G

THEN A has the right to demand from B the discussion of that obstacle and discovering of the possible ways of overcoming it

These rules, on the one hand, limit the activity of the author of a turn in constructing his turns and, on the other hand, help the addressee understand these turns by drawing implicatures.

Implicatures are inferences drawn if two conditions are met. First, the turn of the partner violates a principle of communication and, secondly, the communicative situation does not contain any clues that it is done intentionally. Therefore the addressee starts making hypotheses, i.e. drawing inferences which help him construct such an interpretation for the input turn which satisfies the principle of cooperation. If there are no counterarguments to this hypothesis the addressee supposes this to be the intended meaning of the input turn.

Drawing of implicatures in the dialog system proceeds according to special procedures based on rules of behavior.

2. A special case of rules of behavior are pragmatic inference rules:

IF <type of communicative step>
THEN <default GOAL'>

Where GOAL' is a goal inferred by default from the GOAL of the author of the turn. For instance, when A asks B how to reach a goal (e.g. How can I get to the railway station?) then his goal may be to achieve that result (i.e. to be at the station).

3. Rules of interest have the form

<interest source> → <question/problem>

They determine from the type of a communicative step the questions, or "interests", which the interlocutor must find answers to. They typically concern such problems, as

- what does the author suppose about the addressee when asking a question or making a proposal
- does the claim of the author hold
- are there any obstacles to the plan put forward in a communicative step, etc.

As interest sources may function various communicative situations with wide difference in their complexity. The questions they trigger are typically related to the structural parts SETTING and GOAL of the communicative step.

Here are some examples of the rules (A= author, B= addressee):

A: REFUSAL → B: Why A refused (what is the MOTIVE of REFUSAL)?

A: CLAIM P → B: Does P hold?

4. Rules of logical inference use data within one communicative step only. We have treated that type of rules in detail in (Öim, Saluveer, 1985) and will therefore not discuss them here.

5. Rules of turn compilation are used in constructing and interpreting a turn which consists of more than one communicative steps. For example, a turn expressing a refusal may consist of only one communicative step REFUSAL, but more common are such combinations as REFUSAL plus MOTIVE, only MOTIVE, REFUSAL plus ALTERNATIVES, etc. The rules of turn compilation fix the possible combinations of communicative steps and their possible sequence in a turn (the sequence of turns is important because the steps are not simply linearly ordered but there exist fixed subordination relations between the communicative steps within a turn). These rules have the general form

type of turn → C₁, C₂, ..., C_k
(k ≥ 1), where C₁, ..., C_k are types of communicative steps.

6. To rules of dialog coherence belong first and foremost rules which determine from the components SETTING and CONSEQ of a partner's turn the contents of the component SETTING in the other partner's turn.

There are several subgroups within this general group:

(i) default rules are used in such situations in a dialog where the turn of a partner is "blank", i.e. when the partner does not answer to a remark. E.g., when somebody asks "Don't you believe me?" then a silence from his partner is equal to a negative turn (the partner does not believe the author).

(ii) rules determining cycles of turns in coherent dialog. It has been pointed out that as a minimal unit in interaction functions not a pair of turns but a triplet, e.g.

A: INITIATION
B: REACTION
A: ACCEPTANCE OF REACTION

A: Would you pass the sugar, please.
B: Here you are.
A: Thanks.

Here are some other examples of such triplets in dialogs:

A: QUESTION - B: ANSWER - A: THANKING
A: QUESTION - B: SPECIFYING QUESTION - A:
ANSWER TO SPECIFYING QUESTION - B: ANSWER
A: THANKING - B: ACCEPTANCE OF THANKING

A: Can I have a bottle of brandy?
B: Are you twenty-one?
A: No.
B: No.

The rules of these two groups may be best represented in the form of augmented transition networks where types of communicative steps correspond to nodes and the steps which can follow one another in a dialog are connected with arcs (Metzing, 1980).

3.2. Linguistic knowledge base

This base includes knowledge about morphology, syntax and to a certain degree of semantics of the language.

The lexicon stores declarative knowledge of the language in the form of following entries:

<primary form> <stem> <type of stem>
<semantic characteristics of word>

Morphological rules should guarantee the morphological analysis and synthesis of the words used, i.e. a transition from the word form to its morphological representation (number, case, tense, person) in analysis and the reversed transition in generation.

The output of syntactic analysis (and input to syntactic generation) is a tree of dependencies.

In order to reduce the number of possible resulting dependency trees we may use instead of purely syntactic rules syntactico-semantic rules which combine syntactic and semantic features of a word:

	word 1	word 2
IF	morphological information 1	morphological information 2
		+
	semantic characteristics 1	semantic characteristics 2
THEN	word 1	relation → word 2

Linguistic knowledge base is used mainly by the Linguistic Processor. During parsing the input to the Linguistic Processor is the user's utterance in natural language, the output is the syntactic representation of the turn in the form of dependency trees. In surface generation the input to the Linguistic Processor is the dependency tree(s) and the output is an answer turn in natural language.

3.3. Problem domain knowledge base

To this base belong definitions of all the objects and relations between them in that problem domain and also the methods of solving the problems the system deals with. The definitions of objects and their relations may be represented in the form of frames, the algorithms of solving problems as procedures with parameters. Some procedures may be fillers of frame slots. This knowledge base is used by both the Turns Generator and Interpreter, as well as by the Planner (when solving problems which have cropped up during the dialog).

4. Answering the User: Text Generation

4.1. Planning the answer

In planning its answer the dialog system proceeds from its current activated goals. The Planner chooses questions from the model of goals which then underlie the output turn. The choice is made according to the priorities of the questions, which may concern either the problem domain or interaction. Planning the answer turn is carried out

simultaneously with interpreting the user's input turn. In case of questions which are connected with the problem domain the Planner makes use of the Solver. The Solver tries to answer the questions put to the system by the user and/or by the system itself and marks in the model of goals these questions which it has succeeded in finding an answer to.

In order to answer questions about interaction the Planner turns to the Dialog Monitor. Most questions about interaction belong to the domain "system questions to itself". Rare exception are questions of the type "How dare you speak to me like this?" The dialog system usually does not direct the questions about interaction to its partner except in cases when the partner's turn somehow concerns the dialog system as a "personality" (in man-machine dialog this is yet an unimportant aspect of interaction). To find out such questions the dialog system uses its knowledge about dialog, as well as interest rules and dialog coherence rules and its own and the partner's models. In interpreting a turn the dialog system must also check whether the partner has stuck to all rules of communication. In the opposite case a question appears in the column "system questions to the user" which the dialog system may ask about the violation of communicative rules.

4.2. Non-linguistic synthesis of the answer

As a result of the work of the Planner in the model of goals of the dialog system there are a number of questions from the system to its user from which the Turns' Generator must construct the semantic representation of the future turn. With the highest priority are questions about the problem domain. The Turns' Generator determines

- i) the possible types of answer turns - answer, refusal, request etc.
- ii) the choice among the possible alternatives with the help of rules of behavior
- iii) the use of rules of turn compilation by deciding which types of communicative steps the turn of the dialog system must consist of, and filling in concrete information of the chosen typical structures of communicative steps.

As a result of all these actions the semantic representation of the future turn is formed.

4.3. Linguistic synthesis of the answer

The generation of surface text from its semantic representation takes place in the Linguistic Processor and can be divided into three stages: transformation of semantic representation into syntactic (semantic synthesis), transformation of syntactic representation into morphological (syntactic synthesis) and transformation of morphological representation into the surface text (morphological synthesis).

4.3.1. Semantic synthesis

In the process of semantic synthesis it is necessary to "slice" the semantic representa-

tion of the future text (text frame in case of TARLUS) into sentence representations, i.e. sentence frames. To achieve this the frames which belong to the semantic category of ACTION must be separated from one another according to their sequence in time. Every action tree frame is transformed into a dependency tree of a (simple) sentence. A number of slots in the action frame containing irrelevant information from the point of view of the user are disposed of (e.g. slot SUP referring to the generic notion of a category, procedural slots in frames, etc.). The remaining slot fillers serve as labels of the nodes of the dependency tree, while slot names serve as labels for the arcs on the tree. The preliminary order of the nodes will be determined by the corresponding verb patterns for that action. A verb pattern will determine the order of verb and its attributes in an isolated sentence but not in actual text. Verb patterns depend upon target language.

The text frame is composed of either terminal or conceptual frames. The names of the former are words of the target (i.e. Estonian) language. The names of the latter are names of semantic categories, e.g. ACTION, ANIMAL, TRANSFER, etc. If the dependency tree node is labelled by a semantic category, a word of the target language must be substituted for it depending on the context. E.g. instead of the conceptual frame TRANSFER the system has to choose one of the words from the list of verbs such as buy, borrow, rob, make a present, etc. The procedure of choosing a correct lexical item for a semantic category node is a category-oriented pass through a binary tree each node of which presents a discrimination procedure. The tree gradually limits the set of possible candidates until finally there will be only one word left. The choice of a word among near synonyms is a means for achieving a greater coherence of the text (cf. lexemes like steal, pilfer, nab, purloin, etc. for the semantic notion of stealing). When choosing among these near synonyms the Linguistic Processor should also take into account the model of the partner: the output text should not contain words which are unknown to him.

4.3.2. Syntactic synthesis

This stage can in its turn be divided into two steps: first, transformations on dependency trees with the aim of achieving greater coherence of the text and, secondly, supplying the lexemes with morphological information.

To achieve a greater smoothness of the output text it is necessary to perform some modifications on the dependency trees during this phase of generation:

- i) reordering of nodes

The primary order of nodes in a dependency tree is determined by the verb pattern which does not take into account the place of the sentence in the text. Therefore, it will be necessary sometimes to change the order of nodes: the nodes expressing the theme of the sentence will be placed higher

and those representing the rheme will be placed lower. To accomplish this reordering of nodes, a mechanism of three stacks is used. The first two stacks contain the labels of the two immediately preceding dependency trees. In the third stack those labels which have occurred in the previous two stacks are also placed lower. Experiments have shown that it is sufficient to take into account the word order of only two immediately preceding sentences. Even more - if the system "remembers" too much from the preceding information, the smoothness of the text may get lost. The use of this method allows us to get text (2) instead of text (1):

(1) John took a book from John's briefcase.
John gave Mary the book.
John left John's briefcase on the table.

(2) John took a book from John's briefcase.
The book he gave to Mary.
John's briefcase John left on the table.

ii) Use of pronouns

A pronoun may be substituted for a lexeme corresponding to a node of a dependency tree according to special rules. The application of these rules gives us text (3) instead of text (2):

(3) John took a book from his briefcase.
The book he gave to Mary.
His briefcase John left on the table.

iii) Deletion of repeated phrases

If there exist similar subtrees in two dependency trees then in the second tree the stem may be substituted for the subtree. The result is text (4):

(4) John took a book from his briefcase.
The book he gave to Mary.
The briefcase he left on the table.

iv) Integration of two or more dependency trees into a coherent graph.

One of the rules in this domain states:

IF in several immediately following dependency trees one and the same lexeme fulfills the role of agent/patient
THEN all these trees may be integrated into one coherent graph by removing from the second tree downward the nodes with identical label, and connecting arcs to the corresponding node of the first tree

This rule helps us get text (5) instead of text (1):

(5) John took a book from John's briefcase,
gave Mary the book and left on
the table John's briefcase.

The use of these rules in different order results in different output texts.

To ascribe morphological information to lexemes syntactic rules are used which determine from the syntactico-semantic relations between two words the morphological characteristics of the words, and as a result we get the morphological representation of the text.

4.3.3. Morphological synthesis

On the basis of morphological representation with the help of primary forms of words and their morphological characteristics concrete word forms are built. If it is possible to construct several parallel forms as, for example, are short and long forms of the plural nouns in Estonian, then the choice of one of them is an additional means for achieving fluency of the text.

From the above mentioned facts it may be concluded that coherent text generation differs in many respects from single sentence generation, and the regularities governing this process must be taken into account from the very start of the generation process.

5. Conclusion

Several modules are involved in generation of natural language turns in a natural language dialog system. First, the Planner chooses among the currently activated goals of the dialog system those which the system would carry out in its reply to the user. Secondly, Turns'Generator constructs the semantic representation of the answer turn, i.e. chooses the necessary communicative steps for carrying out the goals laid down by the Planner. It also fills in these steps with concrete data by using knowledge about the problem domain and/or about the communicative process. The Linguistic Processor gives the finishing touches to the surface text.

But before a turn is communicated to the user there should be a "check-up" interpretation of it - the dialog system carries out the possible interpretations of this turn in its "mind", using additional knowledge from its model of the partner and its knowledge base. The result is compared with the intended meaning of the turn and in case of discrepancies the system should return to the module where the interpretations were still similar in order to try another possible way of generating the answer turn.

Of the modules listed above there is a complete version of the Linguistic Processor both for parsing and generation running on a Ryad 1060 computer in the language PL/1. The Turns'Generator is being tested at the present time.

References

- Grice H.P. 1975. Logic and Conversation. - Syntax and Semantics, vol. 3. Speech Acts. New York: Academic Press.
- Leech G. 1983. Principles of pragmatics. Cambridge: Cambridge University Press.
- Metzger D. 1980. ATNs used as a procedural dialog model. Proc. COLING-80, Tokyo, 487-491.
- Õim H., Koit M., Litvak S., Roosmaa T., Saluveer M. 1984. Reasoning and discourse: experts as a link between high and low level inferences. - Papers on Artificial Intelligence, vol. VII, Tartu, 176-190.
- Õim H., Saluveer M. 1985. Frames in linguistic descriptions. - Quaderni di Semantica, vol. VI, N 2, 282-292.