# Natural Language Generation in Dialog Systems

Owen Rambow Srinivas Bangalore Marilyn Walker
AT&T Labs – Research
Florham Park, NJ, USA
rambow@research.att.com

## ABSTRACT

Recent advances in Automatic Speech Recognition technology have put the goal of naturally sounding dialog systems within reach. However, the improved speech recognition has brought to light a new problem: as dialog systems understand more of what the user tells them, they need to be more sophisticated at responding to the user. The issue of system response to users has been extensively studied by the natural language generation community, though rarely in the context of dialog systems. We show how research in generation can be adapted to dialog systems, and how the high cost of hand-crafting knowledge-based generation systems can be overcome by employing machine learning techniques.

## 1. DIALOG SYSTEMS AND GENERATION

Recent advances in Automatic Speech Recognition (ASR) technology have put the goal of naturally sounding dialog systems within reach.[1] However, the improved ASR has brought to light a new problem: as dialog systems understand more of what the user tells them, they need to be more sophisticated at responding to the user. If ASR is limited in quality, dialog systems typically employ a **system-initiative dialog strategy** in which the dialog system prompts the user for specific information and then presents some information to the user. In this paradigm, the range of user input at any time is limited (thus facilitating ASR), and the range of system output at any time is also limited. However, such interactions are not very natural. In a more natural interaction, the user can supply more and different information at any time in the dialog. The dialog system must then support a **mixed-initiative dialog strategy**. While this strategy places greater requirements on ASR, it also increases the range of system responses and the requirements on their quality in terms of informativeness and of adaptation to the context.

For a long time, the issue of system response to users has been studied by the Natural Language Generation (NLG) community, though rarely in the context of dialog systems. What have emerged from this work are a "consensus architecture" [17] which modularizes the large number of tasks performed during NLG in a particular way, and a range of linguistic representations which can be used in accomplishing these tasks. Many systems have been built using NLG technology, including report generators [8, 7], system description generators [10], and systems that attempt to convince the user of a particular view through argumentation [20, 4].

In this paper, we claim that the work in NLG is relevant to dialog systems as well. We show how the results can be incorporated, and report on some initial work in adapting NLG approaches to dialog systems and their special needs. The dialog system we use is the AT&T Communicator travel planning system.We use machine learning and stochastic approaches where hand-crafting appears to be too complex an option, but we also use insight gained during previous work on NLG in order to develop models of what should be learned. In this respect, the work reported in this paper differs from other recent work on generation in the context of dialog systems [12, 16], which does not modularize the generation process and proposes a single stochastic model for the entire process. We start out by reviewing the generation architecture (Section 2). In Section 3, we discuss the issue of text planning for Communicator. In Section 4, we summarize some initial work in using machine learning for sentence planning [19]. Finally, in Section 5 we summarize work using stochastic tree models in generation [2].

## 2. TEXT GENERATION ARCHITECTURE

.

NLG is conceptualized as a process leading from a high-level communicative goal to a sequence of communicative acts which accomplish this communicative goal. A communicative goal is a goal to affect the user's cognitive state, e.g., his or her beliefs about the world, desires with respect to the world, or intentions about his or her actions in the world. Following (at least) [13], it has been customary to divide the generation process into three phases, the first two of which are planning phases. Reiter [17] calls this architecture a "consensus architecture" in NLG.

- During **text planning**, a high-level communicative goal is broken down into a structured representation of atomic communicative goals, i.e., goals that can be attained with a single communicative act (in language, by uttering a single clause). The atomic communicative goals may be linked by rhetorical relations which show how attaining the atomic goals contributes to attaining the high-level goal.

- During **sentence planning**, abstract linguistic resources are chosen to achieve the atomic communicative goals. This includes choosing meaning-bearing lexemes, and how the meaning-bearing lexemes are connected through abstract grammatical constructions (basically, lexical predicate-argument

[1]The work reported in this paper was partially funded by DARPA contract MDA972-99-3-0003.

.

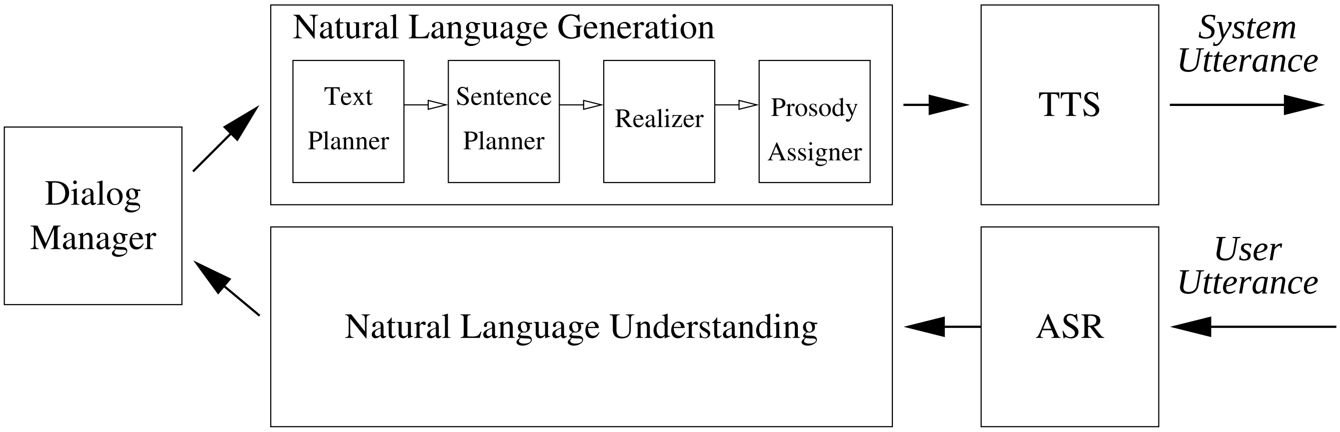


**Figure 1: Architecture of a dialog system with natural language generation**

structure and modification). As a side-effect, sentence planning also determines sentence boundaries: there need not be a one-to-one relation between elementary communicative goals and sentences in the final text.

- During **realization**, the abstract linguistic resources chosen during sentence planning are transformed into a surface linguistic utterance by adding function words (such as auxiliaries and determiners), inflecting words, and determining word order. This phase is not a planning phase in that it only executes decisions made previously, by using grammatical information about the target language. (Prosody assignment can be treated as a separate module which follows realization and which draws on all previous levels of representation. We do not discuss prosody further in this paper.)

Note that sentence planning and realization use resources specific to the target-language, while text planning is language-independent (though presumably it is culture-dependent).

In integrating this approach into a dialog system, we see that the dialog manager (DM) no longer determines surface strings to send to the TTS system, as is often the case in current dialog systems. Instead, the DM determines high-level communicative goals which are sent to the NLG component. Figure 1 shows a complete architecture. An advantage of such an architecture is the possibility for extended plug-and-play: not only can the entire NLG system be replaced, but also modules within the NLG system, thus allowing researchers to optimize the system incrementally.

The main objection to the use of NLG techniques in dialog systems is that they require extensive hand-tuning of existing systems and approaches for new domains. Furthermore, because of the relative sophistication of NLG techniques as compared to simpler techniques such as templates, the hand-tuning requires specialized knowledge of linguistic representations; hand-tuning templates only requires software engineering skills. An approach based on machine learning can provide a solution to this problem: it draws on previous research in NLG and uses the same sophisticated linguistic representations, but it learns the domain-specific rules that use these representation automatically from data. It is the goal of our research to show that for dialog systems, approaches based on machine learning can do as well as or outperform handcrafted approaches (be they NLG- or template-based), while requiring far less time for tuning. In the following sections, we summarize the current state of our research on an NLG system for the Communicator dialog system.

## 3. TEXT PLANNER

Based on observations from the travel domain of the Communicator system, we have categorized system responses into two types. The first type occurs during the initial phase when the system is gathering information from the user. During this phase, the high-level communicative goals that the system is trying to achieve are fairly complex: the goals include getting the hearer to supply information, and to explicitly or implicitly confirm information that the hearer has just supplied. (These latter goals are often motivated by the still not perfect quality of ASR.) The second type occurs when the system has obtained information that matches the user's requirements and the options (flights, hotel, or car rentals) need to be presented to the user. Here, the communicative goal is mainly to make the hearer believe a certain set of facts (perhaps in conjunction with a request for a choice among these options).

In the past, NLG systems typically have generated reports or summaries, for which the high-level communicative goal is of the type "make the hearer/reader believe a given set of facts", as it is in the second type of system response discussed above. We believe that NLG work in text planning can be successfully adapted to better plan these system responses, taking into account not only the information to be conveyed but also the dialog context and knowledge about user preferences. We leave this to ongoing work.

In the first type of system response, the high-level communicative goal typically is an unordered list of high-level goals, all of which need to be achieved with the next turn of the system. An example is shown in Figure 2. NLG work in text planning has not addressed such complex communicative goals in the past. However, we have found that for the Communicator domain, no text planning is needed, and that the sentence planner can act directly on a representation of the type shown in Figure 2, because the number of goals is limited (to five, in our studies). We expect that further work in other dialog domains will require an extension of existing work in text planning to account better for communicative goals other than those that simply aim to affect the user's (hearer's) beliefs.

implicit-confirm(orig-city:NEWARK)
implicit-confirm(dest-city:DALLAS)
implicit-confirm(month:9)
implicit-confirm(day-number:1)
request(depart-time)

**Figure 2: Sample text plan (communicative goals)**

| Realization | Score |
|---|---|
| What time would you like to travel on September the 1st to Dallas from Newark? | 5 |
| Leaving on September the 1st. What time would you like to travel from Newark to Dallas? | 4.5 |
| Leaving in September. Leaving on the 1st. What time would you, traveling from Newark to Dallas, like to leave? | 2 |

**Figure 3: Sample alternate realizations of the set of communicative goals shown in Figure 2 suggested by our sentence planner, with human scores**

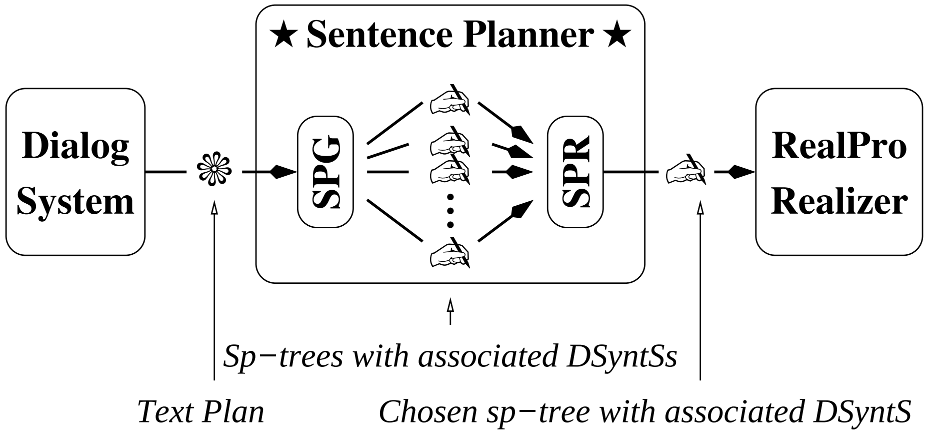


**Figure 4: Architecture of our sentence planner**

## 4. SENTENCE PLANNER

The principal challenge facing sentence planning for dialog systems is that there is no good corpus of naturally occurring interactions of the type that need to occur between a dialog system and human users. This is because of the not-yet perfect ASR and the need for implicitly or explicitly confirming most or all of the information provided by the user. In conversations between two humans, communicative goals such as implicit or explicit confirmations are rare, and thus transcripts of human-human interactions in the same domain cannot be used for the purpose of learning good strategies to attain communicative goals. And of course we do not want to use transcripts of existing systems, as we want to improve on their performance, not mirror it.

We have therefore taken the approach of randomly generating a set of solutions and having human judges score each of the options. Each turn of the system is, as described in Section 3, characterized by a set of high-level goals such as that shown in Figure 2. In the turns we consider, no text planning is needed. To date, we have concentrated on the issue of choosing abstract syntactic constructions (rather than lexical choice), so we map each elementary communicative goal to a canonical lexico-syntactic structure (called a **DSyntS** [11]). We then randomly combine these DSyntSs into larger DSyntSs using a set of clause-combining operations identified previously in the literature [14, 18, 5], such as RELATIVE-CLAUSE, CONJUNCTION, and MERGE.[2] The way in which the elementary DSyntSs are combined is represented in a structure called the **sp-tree**. Each sp-tree is then realized using an off-the-shelf realizer, RealPro [9]. Some sample realizations for the same text plan are shown in Figure 3, along with the average of the scores assigned by two human judges.

[2]MERGE identifies the verbs and arguments of two lexico-syntactic structures which differ only in adjuncts. For example, *you are flying from Newark* and *you are flying on Monday* are merged to *you are flying from Newark on Monday*.

Using the human scores on each of the up to twenty variants per turn, we use RankBoost [6] to learn a scoring function which uses a large set of syntactic and lexical features. The resulting sentence planner consists of two components: the sentence plan generator (SPG) which generates candidate sentence plans and the sentence plan ranker (SPR) which scores each one of them using the rules learned by RankBoost and which then chooses the best sentence plan. This architecture is shown in Figure 4.

We compared the performance of our sentence planner to a random choice of sentence plans, and to the sentence plans chosen as top-ranked by the human judges. The mean score of the turns judged best by the human judges is 4.82 as compared with the mean of 4.56 for the turns generated by our sentence planner, for a mean difference of 0.26 (5%) on a scale of 1 to 5. The mean of the scores of the turns picked randomly is 2.76, for a mean difference of 1.8 (36%). We validated these results in an independent experiment in which 60 subjects evaluated different realizations for a given turn [15]. (Recall that our trainable sentence planner was trained on the scores of only two human judges.) This evaluation revealed that the choices made by our trainable sentence planner were not statistically distinguishable from the choices ranked at the top by the two human judges. More importantly, they were also not distinguishable statistically from the current hand-crafted template-based output of the AT&T Communicator system, which has been developed and fine-tuned over an extended period of time (the trainable sentence planner is based on judgments that took about three person-days to make).

## 5. REALIZER

At the level of the surface language, the difference in communicative intention between human-human travel advisory dialogs and the intended dialogs is not as relevant: we can try and mimic the human-human transcripts as closely as possible. To show this, we have performed some initial experiments using FERGUS (Flexible Empiricist-Rationalist Generation Using Syntax), a stochastic surface realizer which incorporates a tree model and a linear language model [2]. We have developed a metric which can be computed automatically from the syntactic dependency structure of the sentence and the linear order chosen by the realizer, and we have shown that this metric correlates with human judgments of the felicity of the sentence [3]. Using this metric, we have shown that the use of both the tree model and the linear language model improves the quality of the output of FERGUS over the use of only one or the other of these resources.

FERGUS was originally trained on the Penn Tree Bank corpus consisting of Wall Street Journal text (WSJ). The results on an initial set of Communicator sentences were not encouraging, presumably because there are few questions in the WSJ corpus, and furthermore, specific constructions (including *what* as determiner) appear to be completely absent (perhaps due to a newspaper style file). In an initial experiment, we replaced the linear language model (LM) trained on 1 million words of WSJ by an LM trained on 10,000 words of human-human travel planning dialogs collected at CMU. This resulted in a dramatic improvement, with almost all questions being generated correctly. Since the CMU corpus is relatively small for a LM, we intend to experiment with finding the ideal combination of WSJ and CMU corpora. Furthermore, we are currently in the process of syntactically annotating the CMU corpus so that we can derive a tree model as well. We expect further improvements in quality of the output, and we expect to be able to exploit the kind of limited lexical variation allowed by the tree model [1].

## 6. CONCLUSION

We have discussed how work in NLG can be applied in the development of dialog systems, and we have presented two approaches to using stochastic models and machine learning in NLG. Of course, the final justification for using a more sophisticated NLG architecture must come from user trials of an integrated system. However, we suspect that, as in the case of non-dialog NLG systems, the strongest arguments in favor of NLG often come from software engineering issues of maintainability and extensibility, which can be difficult to quantify in research systems.

## 7. REFERENCES


[1] S. Bangalore and O. Rambow. Corpus-based lexical choice in natural language generation. In *38th Meeting of the Association for Computational Linguistics (ACL'00)*, Hong Kong, China, 2000.

[2] S. Bangalore and O. Rambow. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany, 2000.

[3] S. Bangalore, O. Rambow, and S. Whittaker. Evaluation metrics for generation. In *Proceedings of the First International Natural Language Generation Conference (INLG2000)*, Mitzpe Ramon, Israel, 2000.

[4] G. Carenini and J. Moore. A strategy for generating evaluative arguments. In *Proceedings of the First International Natural Language Generation Conference (INLG2000)*, Mitzpe Ramon, Israel, 2000.

[5] L. Danlos. G-TAG: A lexicalized formalism for text generation inspired by tree adjoining grammar. In A. Abeillé and O. Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analysis, and Processing*. CSLI Publications, 2000.

[6] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. In *Machine Learning: Proceedings of the Fifteenth International Conference*, 1998. Extended version available from http://www.research.att.com/ schapire.

[7] E. Goldberg, N. Driedger, and R. Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Expert*, pages 45–53, 1994.

[8] K. Kukich. *Knowledge-Based Report Generation: A Knowledge Engineering Approach to Natural Language Report Generation*. PhD thesis, University of Pittsuburgh, 1983.

[9] B. Lavoie and O. Rambow. RealPro – a fast, portable sentence realizer. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP'97)*, Washington, DC, 1997.

[10] B. Lavoie, O. Rambow, and E. Reiter. Customizable descriptions of object-oriented models. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP'97)*, Washington, DC, 1997.

[11] I. A. Mel'čuk. *Dependency Syntax: Theory and Practice*. State University of New York Press, New York, 1988.

[12] A. H. Oh and A. I. Rudnicky. Stochastic language generation for spoken dialog systems. In *Proceedings of the ANL/NAACL 2000 Workshop on Conversational Systems*, pages 27–32, Seattle, 2000. ACL.

[13] O. Rambow and T. Korelsky. Applied text generation. In *Third Conference on Applied Natural Language Processing*, pages 40–47, Trento, Italy, 1992.

[14] O. Rambow and T. Korelsky. Applied text generation. In *Proceedings of the Third Conference on Applied Natural Language Processing, ANLP92*, pages 40–47, 1992.

[15] O. Rambow, M. Rogati, and M. Walker. A trainable sentence planner for spoken dialogue systems. In *39th Meeting of the Association for Computational Linguistics (ACL'01)*, Toulouse, France, 2001.

[16] A. Ratnaparkhi. Trainable methods for surface natural language generation. In *Proceedings of First North American ACL*, Seattle, USA, May 2000.

[17] E. Reiter. Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the 7th International Workshop on Natural Language Generation*, pages 163–170, Maine, 1994.

[18] J. Shaw. Clause aggregation using linguistic knowledge. In *Proceedings of the 8th International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Ontario, 1998.

[19] M. Walker, O. Rambow, and M. Rogati. A trainable sentence planner for spoken dialogue systems. In *2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'01)*, Pittsburgh, PA, 2001.

[20] I. Zukerman, R. McConachy, and K. Korb. Bayesian reasoning in an abductive mechanism for argument generation and analysis. In *AAAI98 Proceedings – the Fifteenth National Conference on Artificial Intelligence*, pages 833–838, Madison, Wisconsin, 1998.