Text Generation Methods for Dialog Systems

Helmut Horacek

Universität des Saarlandes
FR 6.2 Informatik, P.O. Box 15 11 50
D-66041 Saarbrücken, GERMANY
horacek@cs.uni-sb.de

Abstract

Text generation systems are typically more powerful than generation components of dialog systems. In order to exploit their advanced capabilities for dialog purposes, we discuss the extension potential of NL generation components of dialog systems on the basis of methods embedded in text generation system. We investigate architectural concerns and crucial system features in a comparison, and we formulate conditions for a successful reuse of generation techniques. Finally, we sketch how we envision to substantiate this investigation for promoting the proof explanation system *P.rex* into a generation component of a tutorial dialog system for teaching theorem proving skills.

Motivation

Some limitations of dialog systems are caused by the restricted capabilities of their generation components. For the generation of text rather than dialogs, several systems exist that exhibit considerable presentation capabilities. Hence, it seems reasonable to incorporate these techniques into the less elaborate generation components of dialog system. This investigation, however, it not so simple as it appears to be. Differences in the demands of conducting dialogs as opposed to producing texts and the poorly understood relation in NLG between linguistic functionalities and system architectures suitable for exhibiting these functionalities complicates the transfer of methods.

In order to facilitate the use of generation techniques in dialog environments, we examine the discrepancies between dialog and text generation systems, and we sketch strategies to bridge them for reusing purposes.

NLG for Dialogs and for Texts

The tasks of guiding dialogs and producing texts impose different demands on systems built for either of these purposes. Dialog contributions typically consist of a few and frequently only a single sentence, each carrying out individual speech acts. Generated texts, in contrast, are typically of paragraph-length, with an internal structure based on rhetorical grounds that gives much more freedom in ordering and combining propositions through expressing them by the lexical material available. Moreover, the dialog context, that is, the preceding utterances, has relatively stronger impacts on the appearance of the utterance to be produced than user-specific preferences in text generation systems.

The differences in the utterances typically produced by each of these system classes also manifests itself in the kind of prominent phenomena. Elliptic utterances mostly occur in dialogs, and also "retro-aggregation" (Lemon et al. 2002) is typically found in conversations only. (Lemon et al. 2002) call sequences of sentences with missing repeated constituents "retro-aggregation" (e.g., "I have cancelled flying to the school. And the tower. And landing at the base"), as opposed to the usual "pre-aggregation" (as in "I will fly to the tower and the hospital"). Conversely, systematic and larger-scale aggregation (Shaw1998a, 1998b) is relevant for texts only. However, limited aggregation of this kind also occurs in conversations when sequences (Di Eugenio et al. 2002) or lists of items (Walker et al. 2002) are communicated.

An interesting observation in the comparison between NLG methods for dialog and for texts is the approach towards lexicalization, which we think is a key point for an NLG system if not the most important one. All dialog systems we are aware of apply simple correspondences between concepts and words. In particular, the strong restriction in that correspondence enabled a clean design of the German dialog system HAM-ANS (Hoeppner et al. 1983), which we believe to be a major reason for the success of this pioneer work. For generating texts, more flexibility may be provided, depending on the envisioned system capabilities. Especially for multi-lingual and machine translation purposes, lexicalization is one of the most prominent and elaborate issues.

In order to handle the relevant issues effectively, system architectures may differ significantly between text and dialog approaches. As a tendency, most text generation systems are widely oriented on what is considered a consensus architecture (Reiter 1994). For dialogs, different levels of representation may be chosen in dependency of the depth of analysis required for handling an individual dialog contribution.

	Dialog systems	*Text generation systems*
Typical requirements	Short utterances Context and speechact-driven	Longer text passages Organization through text planning
Typical functionalities	Ellipsis, retro-aggregation Simple concept-lexeme mapping	Systematic aggregation Flexible concept-lexeme mapping
Typical architecture	Simplified representations Flexible processing	Stratified representation levels Uniform processing

Table 1. Comparison between generation components of dialog systems and text generation systems

The considerations about contrastive features of text generation systems and generation components of dialog systems are summarized in Table 1.

Consequences for Building Systems

Given the discrepancies as outlined in the previous section, we believe that extending the functionality of generation components, notably those of dialog systems, requires particular foresight. In order to adopt techniques from text generation systems purposefully, we are convinced that the following fundamental condition must be met: An architecture needs to be chosen which is considered appropriate for handling all aspects of the functionality ultimately envisioned. To a large extent, this architecture may be inspired by an appropriate existing text generation system. In contrast, we believe that incremental extensions are problematic, since there is little agreement on the order of subtasks in the generation process, which is oriented on the specific functionality of implemented systems (Cahill et al. 1999). In particular, this holds for the approach chosen towards lexicalization; incrementally extending its coverage would mean augmenting the repertoire of recasting operations in mapping conceptual specifications onto lexical items – this has strong impacts on all other subprocces. In addition, adopting an architecture of a text generation system for dialog purposes requires some modifications:

(1) Specifications must be processed in partitions, separately for each of the speech acts to be produced,
(2) Preceding dialog contributions must be structurally incorporated in the system's context representation.
(3) Processing certain kinds of speech acts should be handled as architectural short-cuts.

The latter aspect tends to favor the exploitation of preceding dialog contributions in reference and ellipsis generation rather than avoiding repetitions.

An Example – Our Future Project

In the near future, we intend to adopt this strategy for making use of the system PROVERB (Huang and Fiedler 1997) and its successor *P.rex* (Fiedler 2001), in order to build a dialog system that is able to guide tutorial dialogs in teaching mathematical theorem proving. In this task, exchanging short and typical dialog contributions are interleaved with occasional, longer descriptions of proof techniques. This setting seems to be ideal for combining techniques from dialog systems and text generation.

In its current state, *P.rex* is able to produce paragraph-length explanations of machine-found mathematical proofs. Thereby, it varies degrees of detail and explicitness in its presentations, in dependency of user specifications. The system is a typical applied text generation system following the "consensus" architecture in a stratified way. It has a text planner that reorganizes and linearizes a rhetorically inadequate textplan, which is the presentation-oriented perspective on the proof graph, that is the representation of the machine-found proof. Its sentence planner includes a rule-based subsystem for performing aggregation operations, a lexicalization module with paraphrasing capabilities, mostly synonyms and variations in word categories (e.g., "associative", and "associativity") and a reference generation mechanism based on focus spaces in Grosz' and Sidner's style. All these components, while being general in their architecture, are tailored to the domain of mathematics and to the presentation demands of theorem proving to a certain extent.

When promoting *P.rex* into a dialog system, all its present features will be maintained, since the exposition of a full proof may still be required, although this will not frequently be the case. Most of the time, the system will have to produce confirmations, rejections, or corrections, investigate clarification subdialogs, and most importantly, provide hints. For all these kinds of utterances, neither the present representations of domain concepts provide sufficient information nor do the capabilities of the text generation module give support. Therefore, extensions are not only required in terms of domain representations and associated uses (e.g., what a hint is, and the domain concepts, which the system should be able to hint at), but also in terms of proper generation capabilities.

In order to pursue these issues in a systematic manner, the first thing to be settled is the lexicalization mechanism. Our application is ambitious through its high degree of variations (ranging from simple utterances, such as confirmations or denials to elaborate explanations), but it does not require a detailed command of natural language subtleties, as this would be the case in an application related to machine translation. Therefore, we confine ourselves with the existing approach to lexicalization as adopted by *P.rex.*, by which compositionality is guaranteed so that this subprocess easily fits into a standard pipe-line architecture. If it will later turn out that some cases require alternative conceptualizations, we intend to handle this when building the specifications for the utterance or as a restructuring operation within the text planning process, depending on the kind of alternation required.

In addition to this architectural decision, meeting the new demands requires several extensions to *P.rex*, addressing dialog-relevant properties. These concern

(1) A generalization of the focus mechanism of *P.rex*; at present, this mechanism is only able to build a structure for a system's monolog. In the future, it must overarch several utterances of both conversants.
(2) The repertoire of speech acts, which currently consists of domain-specific communicative acts about mathematical concepts, is augmented to cover hints, confirmations, denials, etc. Obviously, this also comprises expansions of the domain models and the vocabulary.
(3) A model for producing elliptic answers, especially in clarification dialogs needs to be developed and integrated into the system in an economic way.

Some potentially useful extensions may not prove relevant to our application, for instance expanding the aggregation operators to handle “retro-aggregation”. If it will turn out that this feature is needed, this means that a new set of operators is to be built and used alternatively to the existing one, depending on whether the specifications cover an elaborate explanation or a sequence or similar speech acts.

Finally, the overall organization of the generation procedure is modified. Instead of processing all the material in one pass, which is beneficial for producing elaborate explanations in an adequate form, a partitioning is imposed on the basis of the sequence of speech acts to be produced. Depending on the kind of speech act, one or several subphases may be skipped, thereby introducing processing *short-cuts*. In particular, we intend to use this techniqe for handling ellipsis.

Conclusion

In this paper, we have discussed the extension potential of NL generation components of dialog systems on the basis of methods embedded in text generation system. We have addressed the problems imposed by this investigation, including architectural concerns and crucial system features, and we have sketched our future plans for the proof explanation system *P.rex*. We believe that this issue will also be relevant for future systems in the typical dialog applications travel and restaurant information, when pure information-seeking is enhanced by advisory capabilities.

References

Cahill, L.; Doran, C.; Evans, R.; Mellish, C.; Paiva, D.; Reape, M.; Scott, D., and Tipper. N. 1999. In Search of a Reference Architecture for NLG Systems. In Proceedings of the 7th European Workshop on Natural Language Generation, 77-85.

Di Eugenio, B.; Glass, M.; and Trolio. M. 2002. The DIAG Experiments: Natural Language Generation for Intelligent Tutoring Systems. In Proc. of the 2nd International National Language Generation Conference, 120-127.

Fiedler. A. 2001. Dialog-Driven Adaptation of Explanations of Proofs. In Proc. of the 17th IJCAI, 1295-1300.

Hoeppner, W.; Christaller, T.; Marburger, H.; Morik, K.; Nebel, B.; and O'Leary, M. 1983. Beyond Domain-Independence: Experience with the Development of a German Language Access System to Highly Diversible Background Systems. In Proc. of the 8th IJCAI-83, 588-594.

Huang, X.; and Fiedler. A. 1997. Proof Verbalization as an Application of NLG. In Proc. of the 15th IJCAI, 965-970.

Reiter, E. 1994. Has a Consensus NLG Architecture Appreared and is it Psychologically Plausible?. In Proc. of 7th International Workshop on Natural Language Generation, 163-170.

Lemon, O.; Gruenstein, A.; Battle, A.; and Peters, S. 2002. Multi-Tasking and Collaborative Activities in Dialogue Systems. In Proceedings of the 3rd SIGDial Workshop on Discourse and Dialogue, 113-124.

Shaw. J. 1998a. Segregatory Coordination and Ellipsis in Text Generation. In Proceedings of the 36th Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, 1220-1226.

Shaw. J. 1998b. Clause Aggregation Using Linguistic Knowledge. In Proc. of the 9th International Workshop on Natural Language Generation, 138-147.

Walker, M.; Whittaker, S.; Stent, A.; Maloor, P.; Moore, J.; Johnston, M.; and Vasireddy. G. 2002. Speech-Plans: Generating Evaluative Responses in Spoken Dialogue. In Proc. of the 2nd International National Language Generation Conference, 73-80.