# Trainable Sentence Planning for Complex Information Presentation in Spoken Dialog Systems

**Amanda Stent**
Stony Brook University
Stony Brook, NY 11794
U.S.A.
stent@cs.sunysb.edu

**Rashmi Prasad**
University of Pennsylvania
Philadelphia, PA 19104
U.S.A.
rjprasad@linc.cis.upenn.edu

**Marilyn Walker**
University of Sheffield
Sheffield S1 4DP
U.K.
M.A.Walker@sheffield.ac.uk

## Abstract

A challenging problem for spoken dialog systems is the design of utterance generation modules that are fast, flexible and general, yet produce high quality output in particular domains. A promising approach is trainable generation, which uses general-purpose linguistic knowledge automatically adapted to the application domain. This paper presents a trainable sentence planner for the MATCH dialog system. We show that trainable sentence planning can produce output comparable to that of MATCH's template-based generator even for quite complex information presentations.

## 1 Introduction

One very challenging problem for spoken dialog systems is the design of the utterance generation module. This challenge arises partly from the need for the generator to adapt to many features of the dialog domain, user population, and dialog context.

There are three possible approaches to generating system utterances. The first is *template-based generation*, used in most dialog systems today. Template-based generation enables a programmer without linguistic training to program a generator that can efficiently produce high quality output specific to different dialog situations. Its drawbacks include the need to (1) create templates anew by hand for each application; (2) design and maintain a set of templates that work well together in many dialog contexts; and (3) repeatedly encode linguistic constraints such as subject-verb agreement.

The second approach is *natural language generation* (NLG), which divides generation into: (1) text (or content) planning, (2) sentence planning, and (3) surface realization. NLG promises portability across domains and dialog contexts by using general rules for each generation module. However, the quality of the output for a particular domain, or a particular dialog context, may be inferior to that of a template-based system unless domain-specific rules are developed or general rules are tuned for the particular domain. Furthermore, full NLG may be too slow for use in dialog systems.

A third, more recent, approach is *trainable generation*: techniques for automatically training NLG modules, or hybrid techniques that adapt NLG modules to particular domains or user groups, e.g. (Langkilde, 2000; Mellish, 1998; Walker, Rambow and Rogati, 2002). Open questions about the trainable approach include (1) whether the output quality is high enough, and (2) whether the techniques work well across domains. For example, the training method used in SPoT (Sentence Planner Trainable), as described in (Walker, Rambow and Rogati, 2002), was only shown to work in the travel domain, for the information gathering phase of the dialog, and with simple content plans involving no rhetorical relations.

This paper describes trainable sentence planning for information presentation in the MATCH (Multimodal Access To City Help) dialog system (Johnston et al., 2002). We provide evidence that the trainable approach is feasible by showing (1) that the training technique used for SPoT can be extended to a new domain (restaurant information); (2) that this technique, previously used for information-gathering utterances, can be used for information presentations, namely recommendations and comparisons; and (3) that the quality of the output is comparable to that of a template-based generator previously developed and experimentally evaluated with MATCH users (Walker et al., 2002; Stent et al., 2002).

Section 2 describes SPaRKy (Sentence Planning with Rhetorical Knowledge), an extension of SPoT that uses rhetorical relations. SPaRKy consists of a randomized sentence plan generator (SPG) and a trainable sentence plan ranker (SPR); these are described in Sections 3

| | |
|---|---|
| strategy: | recommend |
| items: | Chanpen Thai |
| relations: | justify(nuc:1;sat:2); justify(nuc:1;sat:3); justify(nuc:1;sat:4) |
| content: | 1. assert(best(Chanpen Thai))<br>2. assert(has-att(Chanpen Thai, decor(decent)))<br>3. assert(has-att(Chanpen Thai, service(good))<br>4. assert(has-att(Chanpen Thai, cuisine(Thai))) |

Figure 1: A content plan for a recommendation for a restaurant in midtown Manhattan

| | |
|---|---|
| strategy: | compare3 |
| items: | Above, Carmine's |
| relations: | elaboration(1;2); elaboration(1;3); elaboration(1,4); elaboration(1,5); elaboration(1,6); elaboration(1,7); contrast(2;3); contrast(4;5); contrast(6;7) |
| content: | 1. assert(exceptional(Above, Carmine's))<br>2. assert(has-att(Above, decor(good)))<br>3. assert(has-att(Carmine's, decor(decent)))<br>4. assert(has-att(Above, service(good)))<br>5. assert(has-att(Carmine's, service(good)))<br>6. assert(has-att(Above, cuisine(New American)))<br>7. assert(has-att(Carmine's, cuisine(italian))) |

Figure 2: A content plan for a comparison between restaurants in midtown Manhattan

and 4. Section 5 presents the results of two experiments. The first experiment shows that given a content plan such as that in Figure 1, SPaRKy can select sentence plans that communicate the desired rhetorical relations, are significantly better than a randomly selected sentence plan, and are on average less than 10% worse than a sentence plan ranked highest by human judges. The second experiment shows that the quality of SPaRKy's output is comparable to that of MATCH's template-based generator. We sum up in Section 6.

## 2 SPaRKy Architecture

Information presentation in the MATCH system focuses on user-tailored *recommendations* and *comparisons* of restaurants (Walker et al., 2002). Following the bottom-up approach to text-planning described in (Marcu, 1997; Mellish, 1998), each presentation consists of a set of *assertions* about a set of restaurants and a specification of the *rhetorical relations* that hold between them. Example content plans are shown in Figures 1 and 2. The job of the sentence planner is to choose linguistic resources to realize a content plan and then rank the resulting alternative realizations. Figures 3 and 4 show alternative realizations for the content plans in Figures 1 and 2.

| Alt | Realization | H | SPR |
|---|---|---|---|
| 2 | Chanpen Thai, which is a Thai restaurant, has decent decor. It has good service. It has the best overall quality among the selected restaurants. | 3 | .28 |
| 5 | Since Chanpen Thai is a Thai restaurant, with good service, and it has decent decor, it has the best overall quality among the selected restaurants. | 2.5 | .14 |
| 6 | Chanpen Thai, which is a Thai restaurant, with decent decor and good service, has the best overall quality among the selected restaurants. | 4 | .70 |

Figure 3: Some alternative sentence plan realizations for the recommendation in Figure 1. H = Humans' score. SPR = SPR's score.

| Alt | Realization | H | SPR |
|---|---|---|---|
| 11 | Above and Carmine's offer exceptional value among the selected restaurants. Above, which is a New American restaurant, with good decor, has good service. Carmine's, which is an Italian restaurant, with good service, has decent decor. | 2 | .73 |
| 12 | Above and Carmine's offer exceptional value among the selected restaurants. Above has good decor, and Carmine's has decent decor. Above and Carmine's have good service. Above is a New American restaurant. On the other hand, Carmine's is an Italian restaurant. | 2.5 | .50 |
| 13 | Above and Carmine's offer exceptional value among the selected restaurants. Above is a New American restaurant. It has good decor. It has good service. Carmine's, which is an Italian restaurant, has decent decor and good service. | 3 | .67 |
| 20 | Above and Carmine's offer exceptional value among the selected restaurants. Carmine's has decent decor but Above has good decor, and Carmine's and Above have good service. Carmine's is an Italian restaurant. Above, however, is a New American restaurant. | 2.5 | .49 |
| 25 | Above and Carmine's offer exceptional value among the selected restaurants. Above has good decor. Carmine's is an Italian restaurant. Above has good service. Carmine's has decent decor. Above is a New American restaurant. Carmine's has good service. | NR | NR |

Figure 4: Some of the alternative sentence plan realizations for the comparison in Figure 2. H = Humans' score. SPR = SPR's score. NR = Not generated or ranked

The architecture of the spoken language generation module in MATCH is shown in Figure 5. The dialog manager sends a high-level communicative goal to the SPUR text planner, which selects the content to be communicated using a user model and brevity constraints (see (Walker

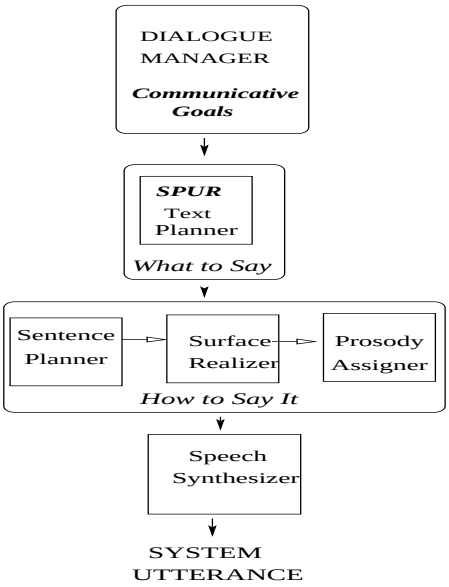


Figure 5: A dialog system with a spoken language generator

et al., 2002)). The output is a content plan for a recommendation or comparison such as those in Figures 1 and 2.

SPaRKy, the sentence planner, gets the content plan, and then a sentence plan generator (SPG) generates one or more sentence plans (Figure 7) and a sentence plan ranker (SPR) ranks the generated plans. In order for the SPG to avoid generating sentence plans that are clearly bad, a content-structuring module first finds one or more ways to linearly order the input content plan using principles of entity-based coherence based on rhetorical relations (Knott et al., 2001). It outputs a set of **text plan trees** (**tp-trees**), consisting of a set of speech acts to be communicated and the rhetorical relations that hold between them. For example, the two tp-trees in Figure 6 are generated for the content plan in Figure 2. Sentence plans such as alternative 25 in Figure 4 are avoided; it is clearly worse than alternatives 12, 13 and 20 since it neither combines information based on a restaurant entity (e.g Babbo) nor on an attribute (e.g. decor).

The top ranked sentence plan output by the SPR is input to the RealPro surface realizer which produces a surface linguistic utterance (Lavoie and Rambow, 1997). A prosody assignment module uses the prior levels of linguistic representation to determine the appropriate prosody for the utterance, and passes a marked-up string to the text-to-speech module.

## 3 Sentence Plan Generation

As in SPoT, the basis of the SPG is a set of clause-combining operations that operate on tp-trees and incrementally transform the elementary predicate-argument lexico-structural representations (called **DSyntS** (Melcuk, 1988)) associated with the speech-acts on the leaves of the tree. The operations are applied in a bottom-up left-to-right fashion and the resulting representation may contain one or more sentences. The application of the operations yields two parallel structures: (1) a **sentence plan tree** (**sp-tree**), a binary tree with leaves labeled by the assertions from the input tp-tree, and interior nodes labeled with clause-combining operations; and (2) one or more **DSyntS** trees (**d-trees**) which reflect the parallel operations on the predicate-argument representations.

We generate a random sample of possible sentence plans for each tp-tree, up to a prespecified number of sentence plans, by randomly selecting among the operations according to a probability distribution that favors preferred operations[1]. The choice of operation is further constrained by the rhetorical relation that relates the assertions to be combined, as in other work e.g. (Scott and de Souza, 1990). In the current work, three RST rhetorical relations (Mann and Thompson, 1987) are used in the content planning phase to express the relations between assertions: the JUSTIFY relation for recommendations, and the CONTRAST and ELABORATION relations for comparisons. We added another relation to be used during the content-structuring phase, called INFER, which holds for combinations of speech acts for which there is no rhetorical relation expressed in the content plan, as in (Marcu, 1997). By explicitly representing the discourse structure of the information presentation, we can generate information presentations with considerably more internal complexity than those generated in (Walker, Rambow and Rogati, 2002) and eliminate those that violate certain coherence principles, as described in Section 2.

The clause-combining operations are general operations similar to aggregation operations used in other research (Rambow and Korelsky, 1992; Danlos, 2000). The operations and the

[1] Although the probability distribution here is handcrafted based on assumed preferences for operations such as MERGE, RELATIVE-CLAUSE and WITH-REDUCTION, it might also be possible to learn this probability distribution from the data by training in two phases.

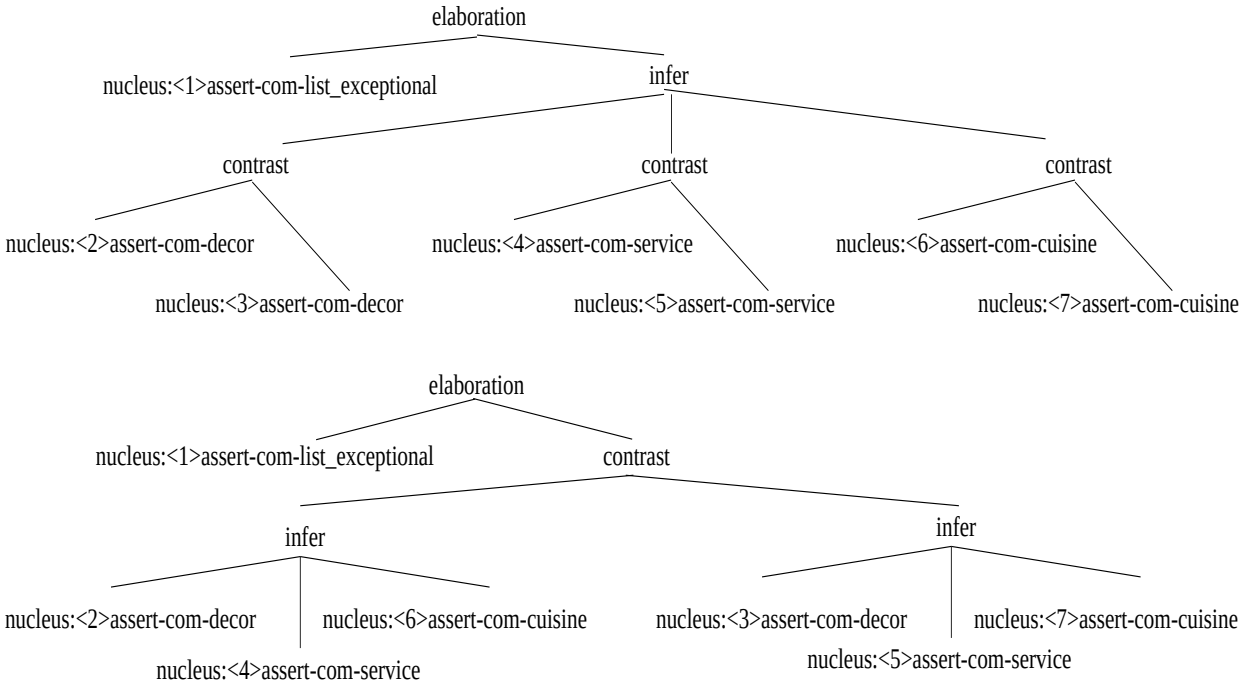


Figure 6: Two tp-trees for alternative 13 in Figure 4.

constraints on their use are described below.

MERGE applies to two clauses with identical matrix verbs and all but one identical arguments. The clauses are combined and the non-identical arguments coordinated. For example, MERGE(*Above has good service*;*Carmine's has good service*) yields *Above and Carmine's have good service.* MERGE applies only for the relations INFER and CONTRAST.

WITH-REDUCTION is treated as a kind of "verbless" participial clause formation in which the participial clause is interpreted with the subject of the unreduced clause. For example, WITH-REDUCTION(*Above is a New American restaurant*;*Above has good decor*) yields *Above is a New American restaurant, with good decor.* WITH-REDUCTION uses two syntactic constraints: (a) the subjects of the clauses must be identical, and (b) the clause that undergoes the participial formation must have a *have*-possession predicate. In the example above, for instance, the *Above is a New American restaurant* clause cannot undergo participial formation since the predicate is not one of *have*-possession. WITH-REDUCTION applies only for the relations INFER and JUSTIFY.

RELATIVE-CLAUSE combines two clauses with identical subjects, using the second clause to relativize the first clause's subject. For example, RELATIVE-CLAUSE(*Chanpen Thai is a Thai restaurant, with decent decor and good service*;*Chanpen Thai has the best overall quality among the selected restaurants*) yields *Chanpen Thai, which is a Thai restaurant, with decent decor and good service, has the best overall quality among the selected restaurants.* RELATIVE-CLAUSE also applies only for the relations INFER and JUSTIFY.

CUE-WORD inserts a discourse connective (one of *since*, *however*, *while*, *and*, *but*, and *on the other hand*), between the two clauses to be combined. CUE-WORD CONJUNCTION combines two distinct clauses into a single sentence with a coordinating or subordinating conjunction (e.g. *Above has decent decor BUT Carmine's has good decor*), while CUE-WORD INSERTION inserts a cue word at the start of the second clause, producing two separate sentences (e.g. *Carmine's is an Italian restaurant. HOWEVER, Above is a New American restaurant*). The choice of cue word is dependent on the rhetorical relation holding between the clauses.

Finally, PERIOD applies to two clauses to be treated as two independent sentences.

Note that a tp-tree can have very different realizations, depending on the operations of the SPG. For example, the second tp-tree in Figure 6 yields both Alt 11 and Alt 13 in Figure 4. However, Alt 13 is more highly rated than Alt 11. The sp-tree and d-tree produced by the SPG for Alt 13 are shown in Figures 7 and 8. The composite labels on the interior nodes of the sp-

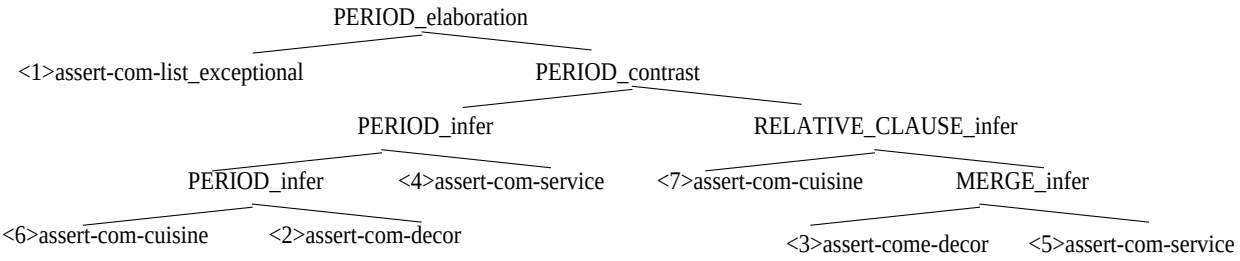


Figure 7: Sentence plan tree (sp-tree) for alternative 13 in Figure 4

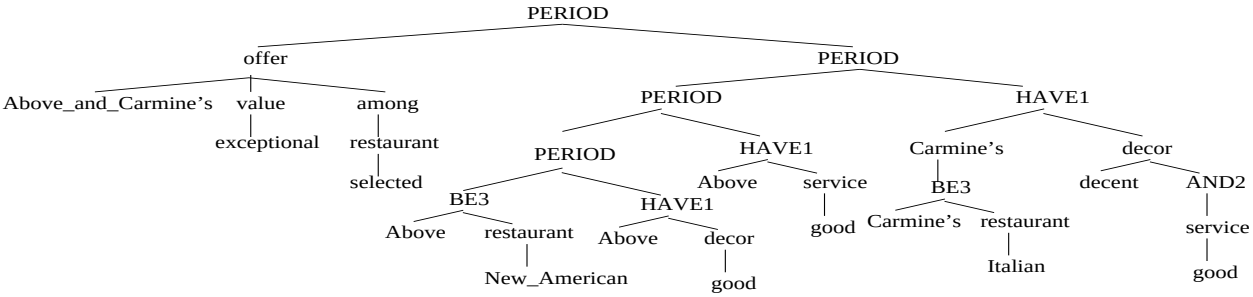


Figure 8: Dependency tree (d-tree) for alternative 13 in Figure 4

tree indicate the clause-combining relation selected to communicate the specified rhetorical relation. The d-tree for Alt 13 in Figure 8 shows that the SPG treats the PERIOD operation as part of the lexico-structural representation for the d-tree. After sentence planning, the d-tree is split into multiple d-trees at PERIOD nodes; these are sent to the RealPro surface realizer.

Separately, the SPG also handles referring expression generation by converting proper names to pronouns when they appear in the previous utterance. The rules are applied locally, across adjacent sequences of utterances (Brennan et al., 1987). Referring expressions are manipulated in the d-trees, either intrasententially during the creation of the sp-tree, or intersententially, if the full sp-tree contains any PERIOD operations. The third and fourth sentences for Alt 13 in Figure 4 show the conversion of a named restaurant (*Carmine's*) to a pronoun.

## 4 Training the Sentence Plan Ranker

The SPR takes as input a set of sp-trees generated by the SPG and ranks them. The SPR's rules for ranking sp-trees are learned from a labeled set of sentence-plan training examples using the RankBoost algorithm (Schapire, 1999).

**Examples and Feedback:** To apply RankBoost, a set of human-rated sp-trees are encoded in terms of a set of features. We started with a set of 30 representative content plans for each strategy. The SPG produced as many as 20 distinct sp-trees for each content plan. The sentences, realized by RealPro from these sp-trees, were then rated by two expert judges on a scale from 1 to 5, and the ratings averaged. Each sp-tree was an example input for RankBoost, with each corresponding rating its feedback.

**Features used by RankBoost:** RankBoost requires each example to be encoded as a set of real-valued features (binary features have values 0 and 1). A strength of RankBoost is that the set of features can be very large. We used 7024 features for training the SPR. These features count the number of occurrences of certain structural configurations in the sp-trees and the d-trees, in order to capture declaratively decisions made by the randomized SPG, as in (Walker, Rambow and Rogati, 2002). The features were automatically generated using feature templates. For this experiment, we use two classes of feature: (1) Rule-features: These features are derived from the sp-trees and represent the ways in which MERGE, INFER and CUE-WORD operations are applied to the tp-trees. These feature names start with "rule". (2) Sent-features: These features are derived from the DSyntSs, and describe the deep-syntactic structure of the utterance, including the chosen lexemes. As a result, some may be domain specific. These feature names are prefixed with "sent".

We now describe the feature templates used in the discovery process. Three templates were

used for both sp-tree and d-tree features; two were used only for sp-tree features. *Local feature templates* record structural configurations local to a particular node (its ancestors, daughters etc.). *Global feature templates*, which are used only for sp-tree features, record properties of the entire sp-tree. We discard features that occur fewer than 10 times to avoid those specific to particular text plans.

| Strategy | System | Min | Max | Mean | S.D. |
|---|---|---|---|---|---|
| Recommend | SPaRKy | 2.0 | 5.0 | 3.6 | .71 |
| | HUMAN | 2.5 | 5.0 | 3.9 | .55 |
| | RANDOM | 1.5 | 5.0 | 2.9 | .88 |
| Compare2 | SPaRKy | 2.5 | 5.0 | 3.9 | .71 |
| | HUMAN | 2.5 | 5.0 | 4.4 | .54 |
| | RANDOM | 1.0 | 5.0 | 2.9 | 1.3 |
| Compare3 | SPaRKy | 1.5 | 4.5 | 3.4 | .63 |
| | HUMAN | 3.0 | 5.0 | 4.0 | .49 |
| | RANDOM | 1.0 | 4.5 | 2.7 | 1.0 |

Table 1: Summary of Recommend, Compare2 and Compare3 results (N = 180)

There are four types of **local feature template**: traversal features, sister features, ancestor features and leaf features. Local feature templates are applied to all nodes in a sp-tree or d-tree (except that the leaf feature is not used for d-trees); the value of the resulting feature is the number of occurrences of the described configuration in the tree. For each node in the tree, traversal features record the preorder traversal of the subtree rooted at that node, for all subtrees of all depths. An example is the feature "rule_traversal_assert-com-list_exceptional" (with value 1) of the tree in Figure 7. Sister features record all consecutive sister nodes. An example is the feature "rule_sisters_PERIOD_infer_RELATIVE _CLAUSE_infer" (with value 1) of the tree in Figure 7. For each node in the tree, ancestor features record all the initial subpaths of the path from that node to the root. An example is the feature "rule_ancestor_PERIOD_contrast*PERIOD _infer" (with value 1) of the tree in Figure 7. Finally, leaf features record all initial substrings of the frontier of the sp-tree. For example, the sp-tree of Figure 7 has value 1 for the feature "leaf_#assert-com-list_exceptional#assert-com-cuisine".

**Global features** apply only to the sp-tree. They record, for each sp-tree and for each clause-combining operation labeling a non-frontier node, (1) the minimal number of leaves dominated by a node labeled with that operation in that tree (MIN); (2) the maximal number of leaves dominated by a node labeled with that operation (MAX); and (3) the average number of leaves dominated by a node labeled with that operation (AVG). For example, the sp-tree in Figure 7 has value 3 for "PERIOD_infer_max", value 2 for "PERIOD_infer_min" and value 2.5 for "PERIOD_infer_avg".

## 5 Experimental Results

We report two sets of experiments. The first experiment tests the ability of the SPR to select a high quality sentence plan from a population of sentence plans randomly generated by the SPG. Because the discriminatory power of the SPR is best tested by the largest possible population of sentence plans, we use 2-fold cross validation for this experiment. The second experiment compares SPaRKy to template-based generation.

**Cross Validation Experiment**: We repeatedly tested SPaRKy on the half of the corpus of 1756 sp-trees held out as test data for each fold. The evaluation metric is the human-assigned score for the variant that was rated highest by SPaRKy for each text plan for each task/user combination. We evaluated SPaRKy on the test sets by comparing three data points for each text plan: HUMAN (the score of the top-ranked sentence plan); SPARKY (the score of the SPR's selected sentence); and RANDOM (the score of a sentence plan randomly selected from the alternate sentence plans).

We report results separately for comparisons between two entities and among three or more entities. These two types of comparison are generated using different strategies in the SPG, and can produce text that is very different both in terms of length and structure.

Table 1 summarizes the difference between SPaRKy, HUMAN and RANDOM for recommendations, comparisons between two entities and comparisons between three or more entities. For all three presentation types, a paired t-test comparing SPaRKy to HUMAN to RANDOM showed that SPaRKy was significantly better than RANDOM ($df = 59$, $p < .001$) and significantly worse than HUMAN ($df = 59$, $p < .001$). This demonstrates that the use of a trainable sentence planner can lead to sentence plans that are significantly better than baseline (RANDOM), with less human effort than programming templates.

**Comparison with template generation:** For each content plan input to SPaRKy, the judges also rated the output of a template-based generator for MATCH. This template-based generator performs text planning and sentence planning (the focus of the current paper), including some discourse cue insertion, clause combining and referring expression generation; the templates themselves are described in (Walker et al., 2002). Because the templates are highly tailored to this domain, this generator can be expected to perform well. Example template-based and SPaRKy outputs for a comparison between three or more items are shown in Figure 9.

| Strategy | System | Min | Max | Mean | S.D. |
|---|---|---|---|---|---|
| Recommend | Template | 2.5 | 5.0 | 4.22 | 0.74 |
| | SPaRKy | 2.5 | 4.5 | 3.57 | 0.59 |
| | HUMAN | 4.0 | 5.0 | 4.37 | 0.37 |
| Compare2 | Template | 2.0 | 5.0 | 3.62 | 0.75 |
| | SPaRKy | 2.5 | 4.75 | 3.87 | 0.52 |
| | HUMAN | 4.0 | 5.0 | 4.62 | 0.39 |
| Compare3 | Template | 1.0 | 5.0 | 4.08 | 1.23 |
| | SPaRKy | 2.5 | 4.25 | 3.375 | 0.38 |
| | HUMAN | 4.0 | 5.0 | 4.63 | 0.35 |

Table 2: Summary of template-based generation results. N = 180

Table 2 shows the mean HUMAN scores for the template-based sentence planning. A paired t-test comparing HUMAN and template-based scores showed that HUMAN was significantly better than template-based sentence planning only for compare2 ($df = 29$, $t = 6.2$, $p < .001$). The judges evidently did not like the template for comparisons between two items. A paired t-test comparing SPaRKy and template-based sentence planning showed that template-based sentence planning was significantly better than SPaRKy only for recommendations ($df = 29$, $t = 3.55$, $p < .01$). These results demonstrate that trainable sentence planning shows promise for producing output comparable to that of a template-based generator, with less programming effort and more flexibility.

The standard deviation for all three template-based strategies was wider than for HUMAN or SPaRKy, indicating that there may be content-specific aspects to the sentence planning done by SPaRKy that contribute to output variation. The data show this to be correct; SPaRKy learned content-specific preferences about clause combining and discourse cue insertion that a template-based generator can-

| System | Realization | H |
|---|---|---|
| Template | Among the selected restaurants, the following offer exceptional overall value. Uguale's price is 33 dollars. It has good decor and very good service. It's a French, Italian restaurant. Da Andrea's price is 28 dollars. It has good decor and very good service. It's an Italian restaurant. John's Pizzeria's price is 20 dollars. It has mediocre decor and decent service. It's an Italian, Pizza restaurant. | 4.5 |
| SPaRKy | Da Andrea, Uguale, and John's Pizzeria offer exceptional value among the selected restaurants. Da Andrea is an Italian restaurant, with very good service, it has good decor, and its price is 28 dollars. John's Pizzeria is an Italian , Pizza restaurant. It has decent service. It has mediocre decor. Its price is 20 dollars. Uguale is a French, Italian restaurant, with very good service. It has good decor, and its price is 33 dollars. | 4 |

Figure 9: Comparisons between 3 or more items, H = Humans' score

not easily model, but that a trainable sentence planner can. For example, Table 3 shows the nine rules generated on the first test fold which have the largest negative impact on the final RankBoost score (above the double line) and the largest positive impact on the final RankBoost score (below the double line), for comparisons between three or more entities. The rule with the largest positive impact shows that SPaRKy learned to prefer that justifications involving price be merged with other information using a conjunction.

These rules are also specific to presentation type. Averaging over both folds of the experiment, the number of unique features appearing in rules is 708, of which 66 appear in the rule sets for two presentation types and 9 appear in the rule sets for all three presentation types. There are on average 214 rule features, 428 sentence features and 26 leaf features. The majority of the features are ancestor features (319) followed by traversal features (264) and sister features (60). The remainder of the features (67) are for specific lexemes.

To sum up, this experiment shows that the ability to model the interactions between domain content, task and presentation type is a strength of the trainable approach to sentence planning.

## 6 Conclusions

This paper shows that the training technique used in SPoT can be easily extended to a new

| N | Condition | $\alpha_s$ |
|---|---|---|
| 1 | sent_anc_PROPERNOUN_RESTAURANT *HAVE1 $\geq$ 16.5 | -0.859 |
| 2 | sent_anc_II_Upper_East_Side*ATTR_IN1* locate $\geq$ 4.5 | -0.852 |
| 3 | sent_anc_PERIOD_infer*PERIOD_infer *PERIOD_elaboration $\geq$ -$\infty$ | -0.542 |
| 4 | rule_anc_assert-com-service*MERGE_infer $\geq$ 1.5 | -0.356 |
| 5 | sent_tvl_depth_0_BE3 $\geq$ 4.5 | -0.346 |
| 6 | rule_anc_PERIOD_infer*PERIOD_infer *PERIOD_elaboration $\geq$ -$\infty$ | -0.345 |
| 7 | rule_anc_assert-com-decor*PERIOD_infer *PERIOD_infer*PERIOD_contrast *PERIOD_elaboration $\geq$ -$\infty$ | -0.342 |
| 8 | rule_anc_assert-com-food_quality*MERGE _infer $\geq$ 1.5 | 0.398 |
| 9 | rule_anc_assert-com-price*CW _CONJUNCTION_infer*PERIOD_justify $\geq$ -$\infty$ | 0.527 |

Table 3: The nine rules generated on the first test fold which have the largest negative impact on the final RankBoost score (above the double line) and the largest positive impact on the final RankBoost score (below the double line), for Compare3. $\alpha_s$ represents the increment or decrement associated with satisfying the condition.

domain and used for information presentation as well as information gathering. Previous work on SPoT also compared trainable sentence planning to a template-based generator that had previously been developed for the same application (Rambow et al., 2001). The evaluation results for SPaRKy (1) support the results for SPoT, by showing that trainable sentence generation can produce output comparable to template-based generation, even for complex information presentations such as extended comparisons; (2) show that trainable sentence generation is sensitive to variations in domain application, presentation type, and even human preferences about the arrangement of particular types of information.

## 7 Acknowledgments

We thank AT&T for supporting this research, and the anonymous reviewers for their helpful comments on this paper.

## References

I. Langkilde. Forest-based statistical sentence generation. In *Proc. NAACL 2000*, 2000.

S. E. Brennan, M. Walker Friedman, and C. J. Pollard. A centering approach to pronouns. In *Proc. 25th Annual Meeting of the ACL, Stanford*, pages 155–162, 1987.

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

M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. MATCH: An architecture for multimodal dialogue systems. In *Annual Meeting of the ACL*, 2002.

A. Knott, J. Oberlander, M. O'Donnell and C. Mellish. Beyond Elaboration: the interaction of relations and focus in coherent text. In *Text Representation: linguistic and psycholinguistic aspects*, pages 181-196, 2001.

B. Lavoie and O. Rambow. A fast and portable realizer for text generation systems. In *Proc. of the 3rd Conference on Applied Natural Language Processing, ANLP97*, pages 265–268, 1997.

W.C. Mann and S.A. Thompson. Rhetorical structure theory: A framework for the analysis of texts. Technical Report RS-87-190, USC/Information Sciences Institute, 1987.

D. Marcu. From local to global coherence: a bottom-up approach to text planning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'97)*, 1997.

C. Mellish, A. Knott, J. Oberlander, and M. O'Donnell. Experiments using stochastic search for text planning. In *Proceedings of INLG-98. 1998*.

I. A. Melčuk. *Dependency Syntax: Theory and Practice*. SUNY, Albany, New York, 1988.

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

O. Rambow, M. Rogati and M. A. Walker. Evaluating a Trainable Sentence Planner for a Spoken Dialogue Travel System In *Meeting of the ACL*, 2001.

R. E. Schapire. A brief introduction to boosting. In *Proc. of the 16th IJCAI*, 1999.

D. R. Scott and C. Sieckenius de Souza. Getting the message across in RST-based text generation. In *Current Research in Natural Language Generation*, pages 47–73, 1990.

A. Stent, M. Walker, S. Whittaker, and P. Maloor. User-tailored generation for spoken dialogue: An experiment. In *Proceedings of ICSLP 2002.*, 2002.

M. A. Walker, S. J. Whittaker, A. Stent, P. Maloor, J. D. Moore, M. Johnston, and G. Vasireddy. Speech-Plans: Generating evaluative responses in spoken dialogue. In *Proceedings of INLG-02.*, 2002.

M. Walker, O. Rambow, and M. Rogati. Training a sentence planner for spoken dialogue using boosting. *Computer Speech and Language: Special Issue on Spoken Language Generation*, 2002.